Business Description Categorization to the Five-Digit Indonesian Standard Classification of Business Field (KBLI) Using Machine Learning and Transfer Learning

M A Amnur[1], L O M Gazali[1], A M Siregar[1], F A Jalaksana[1], M N R A Suwendra[1], N F Utami[1], A M Ramadhan[1], E K Fabrianne[1], E W R Panjaitan[1], F A Izzati[1], J B Y Manalu[1], M G Hidayat[1], L H Suadaa[1], B Yuniarto[1], S Pramana[1]

[1] STIS Polytechnic of Statistics, Indonesia, Jakarta, Indonesia

*Corresponding author's email: 222212736@stis.ac.id

Abstract. The Indonesian Standard Classification of Business Fields (KBLI) is essential for economic statistics, yet manual classification of business descriptions to five-digit KBLI codes is time-consuming and prone to inconsistencies. This study aims to develop and compare machine learning (Support Vector Machine and Random Forest) and transfer learning (IndoBERT) models for automating KBLI classification, supported by the preparation of synthetic and real-world datasets for model training. The synthetic data were generated using large language models, validated through human majority voting and complemented with real-world data from the National Labor Force Survey (Sakernas) and the Micro and Small Industry Survey (IMK). The findings indicate that Fine-tuned IndoBERT achieved superior performance, achieving an F1-score of 92.99% and an accuracy of 93.40% on synthetic data, alongside top-1, top-5, and top-10 accuracies of 32.93%, 54.71%, and 63.24% on real-world data. The deployment of fine-tuned IndoBERT as a RESTful API demonstrates its scalability and efficiency, presenting a reliable solution for large-scale KBLI classification in official statistics.

Keyword: IndoBERT, KBLI, machine learning, transfer learning, text classification

1. Introduction

The Indonesian Standard Classification of Business Fields (KBLI) is a key component of the national statistical system that standardizes the classification of economic activities in Indonesia. Derived from the International Standard Industrial Classification (ISIC) Rev. 4, KBLI ensures the comparability of economic data across national, regional, and international levels, supporting consistent economic analysis and policy formulation [1]. Beyond its statistical function, KBLI codes are also essential for licensing, taxation, and regulatory compliance, making them indispensable for both public administration and private sector operations [2].

Despite its importance, assigning KBLI codes remains a complex and error-prone process. The manual approach requires enumerators to interpret open-ended business descriptions, which is time-consuming and highly dependent on their understanding of the KBLI structure. This often results in inconsistencies, subjective judgments, and human error [3], reducing the reliability of official statistics

[4]. These challenges are amplified in large-scale data collections such as the Economic Census, which covers millions of enterprises. Misclassification not only causes operational inefficiency but also distorts economic indicators, misguides policy decisions, and undermines the accuracy of national economic statistics. Hence, automating the classification process is crucial to ensure accuracy, efficiency, and the credibility of Indonesia's economic data.

Classifying business activities at the five-digit KBLI level is particularly challenging due to its high granularity and complexity [5]. The first two digits represent broad sectors [6], while the five-digit level provides detailed classifications essential for sector-specific planning and precise analysis [1]. With 1,789 unique classes [1], the task becomes increasingly difficult due to labeling inconsistencies and subjective interpretation, where even minor wording differences can lead to different classifications. These linguistic nuances require models with advanced language understanding [7].

Recent developments in Natural Language Processing (NLP) offer promising solutions. Machine learning (ML) and transfer learning (TL) techniques can learn from textual data to perform classification efficiently. Transfer learning allows models to leverage large pre-trained corpora and adapt them to specific tasks, improving accuracy even with limited labeled data. Prior studies show that models such as Support Vector Machine (SVM) and Random Forest can classify business descriptions with reasonable accuracy [8], though they often struggle with complex linguistic patterns and large label spaces. Transformer-based models such as IndoBERT, trained on extensive Indonesian text corpora, have demonstrated improved performance in similar contexts [9]. However, because business descriptions differ from general pretraining corpora, domain adaptation remains essential. Studies have shown that continual pretraining or fine-tuning on in-domain data can enhance model performance on specialized tasks [10][11]. Moreover, KBLI classification represents a low-resource NLP problem with limited and imbalanced labeled data, which necessitates techniques such as data augmentation, distant supervision, and transfer learning [12].

Despite these advances, systematic comparisons between traditional ML and modern TL models for fine-grained KBLI classification remain limited. Such research is essential to evaluate model performance under real-world conditions characterized by imbalanced datasets and ambiguous text inputs. Previous studies have reported varying results, with Random Forest and SVM achieving 85–86% accuracy, while fine-tuned IndoBERT models reached approximately 87% [8]. Another study found IndoBERT achieving 76% accuracy for five-digit KBLI classification, slightly outperforming SVM's 74% but with higher computational cost [4]. Therefore, further research is needed to identify models that balance accuracy, efficiency, and scalability for national implementation. This study contributes methodologically to NLP-based economic classification and practically to improving statistical operations in Indonesia by:

- Developing and evaluating ML (SVM, Random Forest) and TL (IndoBERT) models for classifying business descriptions into five-digit KBLI codes.
- Comparing model performance using accuracy, F1-score, and computational efficiency to determine the most effective and scalable approach.
- Demonstrating the implementation of the best-performing model as a RESTful API to support official statistical operations in Indonesia.

2. Research Method

2.1 Business Understanding

The business understanding phase was initiated by examining the operational workflow of Statistics Indonesia (BPS) in conducting business classification based on the Indonesian Standard Classification

of Business Fields (KBLI). A comprehensive review of the five-digit KBLI was undertaken, encompassing its definitions, associated sectors, and hierarchical taxonomy. Two training sessions on the procedures for classifying businesses into KBLI, facilitated by subject-matter experts from BPS, were attended to enhance domain-specific understanding. Furthermore, the official questionnaires were analyzed to determine the types of questions posed in the collection of business field data. Enumeration records from various surveys were also compiled and examined to identify patterns in respondents' answers, assess how these responses were recorded by field officers, and detect points within the process where misclassifications were likely to occur. This phase additionally involved identifying the relevant stakeholders and evaluating potential integration pathways with the planned web-based system. The knowledge acquired through these activities serves as the foundational basis for designing an automated classification feature that is consistent with BPS's established procedures and operational requirements.

2.2 Data Understanding

The data understanding phase emphasizes the construction of a synthetic dataset for KBLI 2020, based on five-digit business descriptions corresponding to the research that has been done by Kaffah et al [13]. That study involved the creation of a synthetic dataset by constructing company descriptions corresponding to each five-digit KBLI 2020 code, using official internet sources (https://klasifikasi.web.bps.go.id). The data creation process was automated via large language models (LLMs), notably OpenAI's ChatGPT-3.5 and ChatGPT-4o, employing a one-shot prompting methodology that has been demonstrated to improve output relevance relative to zero-shot methods [14]. Subsequent to the accumulation of AI-generated data, the annotation phase was executed utilizing a majority voting methodology. This method involves evaluating each data pair by numerous annotators, with the final label determined by the most frequently selected category among them [15].

In the present study, each generated business description, along with its corresponding KBLI code, was independently assessed by two annotators who were not involved in the creation of the descriptions, thereby minimizing potential bias and subjectivity. Each annotator assigned a flag to indicate whether the business description matched the given KBLI code. If both annotators approved the description, its status was marked as "Match." Conversely, if one or both annotators did not approve, the status was marked as "Mismatch." Further annotation was conducted by modifying the descriptions through word substitutions, adding examples, and content improvements. In a subsequent step, annotators also revised the descriptions labelled as "Mismatch" during the Majority Voting phase. The statistics of the valid synthetic dataset are shown in table 1.

Table 1. A Synthetic Dataset Statistics [13]

Category	Number of Business Descriptions *Generate*	Average Business Description *Generate*	Category	Number of Business Descriptions *Generate*	Average Business Description *Generate*
A	6,076	30.10	L	143	28.60
B	1,426	33.69	M	1,895	29.50
C	13,890	29.31	N	2,355	28.81
D	475	29.68	O	983	29.78
E	471	31.00	P	1,629	29.08
F	2,094	31.63	Q	920	30.47

G	8,655	27.74	R	2,265	29,41
H	3,425	32.53	S	825	35.84
I	937	41.52	T	97	30.25
J	2,047	37.23	U	33	33.00
K	3,603	30.51			

In addition to the synthetic dataset, this study incorporates a case study dataset derived from the survey and enumeration data of the National Labor Force Survey (Sakernas) and the Micro and Small Industry Survey (IMK). The next phase involved the integration and formatting of datasets into standardized structures, preserved in either Microsoft Excel (.xlsx) or Comma-Separated Values (.csv) formats. Following integration, a dataset quality assessment was performed to guarantee the creation of accurate and unbiased classification models. The assessment was established on two main criteria: data coverage and description accuracy. Data coverage was assessed by verifying that each five-digit KBLI code contained a minimum of 20 valid business descriptions, whereas description accuracy was determined through validation procedures involving data cleaning, majority voting, and annotation.

2.3 Data Preparation

During the data preparation phase, several sequential steps were undertaken prior to model training. First, the dataset was partitioned into training (70%), validation (15%), and testing (15%) subsets. A stratified sampling strategy was employed to ensure proportional representation of all KBLI codes across the splits, thereby minimizing potential bias in both training and evaluation. The input feature comprised the Business Description, while the target variable was the corresponding five-digit KBLI code. The second step consisted of a comprehensive preprocessing procedure. This included the removal of irrelevant or non-linguistic characters (text cleaning), segmentation of text into smaller linguistic units (tokenization), and conversion of all characters to lowercase (case folding) to ensure consistency. Additionally, stopword removal was applied to eliminate semantically uninformative words, and stemming was performed to reduce words to their base forms, thereby mitigating morphological variations and enhancing the model's generalization capability. For the IndoBERT model specifically, preprocessing was limited to text cleaning and case folding in accordance with the model's pretrained requirements. The third step involved feature extraction. For conventional machine learning models such as Support Vector Machine (SVM) and Random Forest, textual data were transformed using the Term Frequency–Inverse Document Frequency (TF-IDF) representation. In contrast, IndoBERT utilized a word embedding approach to generate dense vector representations capturing rich semantic information. Furthermore, IndoBERT incorporated positional embeddings to encode token order and segment embeddings to differentiate between sentence pairs within a single input sequence. Finally, label encoding was applied to the target variable, as it was categorical in nature. Each KBLI code was assigned a unique integer identifier, ranging from 0 to 1,788, corresponding to the 1,789 distinct five-digit KBLI codes in the dataset. This encoding scheme was selected for its storage efficiency, ease of implementation, and the ability to facilitate reverse mapping to the original codes.

2.4 Modelling

2.4.1 Support Vector Machine

The Support Vector Machine (SVM) algorithm was selected as the primary classification model for automating KBLI code assignment due to its proven effectiveness in handling high-dimensional textual data, particularly when represented using Term Frequency–Inverse Document Frequency (TF-IDF)

features [16]. Compared to deep learning models such as IndoBERT, SVM offers significant advantages in terms of computational efficiency and lower resource requirements. Additionally, SVM supports multi-class classification through One-vs-Rest and One-vs-One strategies [17], making it suitable for handling over 1,700 distinct five-digit KBLI codes.

In this study, the classification pipeline consisted of a TF-IDF vectorizer to convert business descriptions into numerical feature vectors, followed by an SVM classifier to perform the classification. To optimize the model's performance, hyperparameter tuning was conducted using grid search on a combined training and validation set. This approach enabled systematic exploration of parameter combinations and ensured robust evaluation through cross-validation. The hyperparameters optimized in this study included:

1. Kernel: Options include linear, radial basis function (RBF), sigmoid, and polynomial kernels, which determine the method of mapping input data into a higher-dimensional feature. The linear kernel computes the inner product between two feature vectors x and x', expressed as:

$$K(x, x_i) = x \cdot x^T \tag{1}$$

The output of the polynomial kernel function depends on the direction of the two vectors in low dimensional space. This is due to the dot product in the kernel. This kernel is commonly expressed as:

$$K(x, x_i) = \left(1 + x \cdot x_i^T\right)^d \tag{2}$$

Radial Basis Function is one of the most popular kernel functions. It measures similarity based on the Euclidean distance between vectors, controlled by the parameter γ, and is defined as:

$$K(x, x_i) = exp(-\gamma||x - x_i||)^2 \tag{3}$$

The sigmoid kernel applies a hyperbolic tangent transformation to the inner product, parameterized by slope α and intercept r:

$$K(x_i, x_j) = tanh\ tanh\left(\gamma x_i^T x_j + r\right) \tag{4}$$

2. Regularization Parameter (C): Values explored were 0.1, 1, 10, 100, and 1000. This parameter controls the trade-off between minimizing classification error and maximizing the decision margin.
3. Gamma: Values tested include 1, 0.1, 0.001, 0.0001, and scale. This parameter, particularly relevant for non-linear kernels like RBF, determines the influence of a single training example.
4. Probability: Boolean values TRUE and FALSE were evaluated to determine whether the model should estimate class probabilities using Platt Scaling.

2.4.2 Random Forest

Random Forest is one of the most effective classifiers, widely used for both regression and classification tasks [18]. Its diversity, simplicity, and strong performance make it particularly popular for classification [19]. It is an ensemble learning method that combines multiple decision trees to enhance prediction accuracy and reduce overfitting. Each tree is trained on a randomly selected subset of the training data using bootstrapping, and at each split, only a random subset of features is considered to promote model diversity. During prediction, the outputs of all trees are aggregated through voting (for classification) or averaging (for regression), producing more stable, accurate, and robust predictions compared to a single decision tree [20]. In this study, optimal hyperparameters were determined using Grid Search on the training and validation datasets, and the final model was trained on the full dataset using TF-IDF

Vectorizer for feature extraction and Random Forest with bootstrapped decision trees built on random subsets of features and samples. The hyperparameters optimized in this study include:

1. n_estimators: number of decision trees in the ensemble (10, 100, 1000), where more trees improve stability but increase computation time.
2. max_depth: maximum depth of each tree (10, 50, 100, none) to control model complexity.
3. max_features: maximum number of features considered per split (*auto*, *sqrt*, *log2*), where *sqrt* uses the square root of total features, *log2* uses the base-2 logarithm, and *auto* defaults to *sqrt* for classification and all features for regression.

2.4.3 IndoBERT

IndoBERT is a BERT-based model pretrained on a large Indonesian corpus (Indo4B) using Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks to capture bidirectional text representations [21]. After pretraining, IndoBERT can be fine-tuned for specific tasks by adding a task-specific output layer and updating the pretrained weights using the target dataset [22]. In this study, IndoBERT was fine-tuned on the Synthetic KBLI 2020 dataset to enhance its ability to classify KBLI codes. Hyperparameter tuning was performed on the training and validation datasets to identify optimal configurations that yield the best performance and stable training. The final model implementation used the Hugging Face Transformers library, which provides the IndoBERT model, tokenizer, and APIs for training, evaluation, and prediction. The hyperparameters optimized in this study include:

1. Learning rate: values tested in the range 5×10^{-6} to 5×10^{-5} to control weight updates and convergence speed.
2. Optimizer: Adam from torch.optim, chosen for its adaptability to learning rate changes and efficiency for deep learning models.
3. Batch size: dynamically adjusted via a _collate_fn function according to the maximum sequence length (max_seq_len), ensuring uniform input sizes without exceeding the defined limit.

The transformer architecture of the Fine-tuned IndoBERT model used in this study consists of multiple hierarchical processing layers. The following image illustrates the architecture of the Fine-tuned IndoBERT model employed in this research.

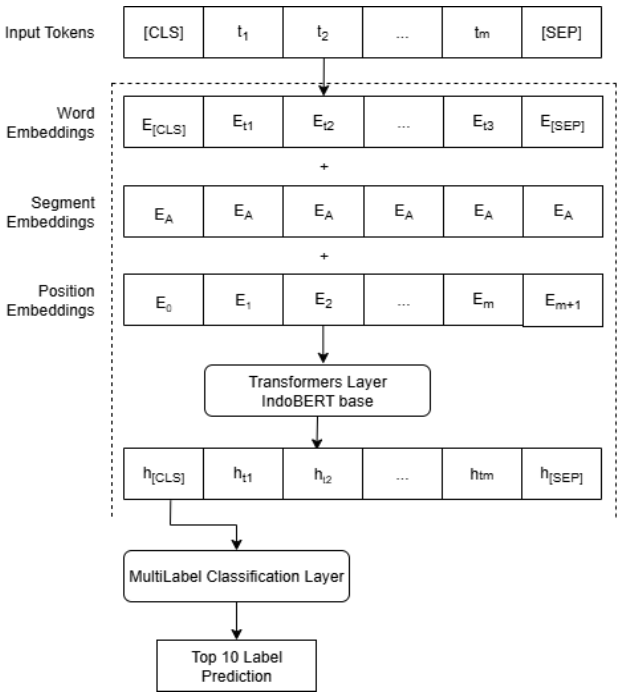

Figure 1. Fine-tuned IndoBERT architecture.

As illustrated in figure 1, the process begins with a sequence of input tokens generated from the tokenization of business description texts, including the addition of special tokens such as [CLS] at the beginning and [SEP] at the end. The model constructs input representations by summing three types of embeddings: Word Embeddings, which capture the semantic meaning of words; Segment Embeddings, which distinguish between different segments of the input; and Position Embeddings, which encode the relative positions of words within the sequence [23]. This combination results in a context-rich input representation.

Consequently, the combined representation is passed through the transformer layers of the IndoBERT-base, which comprise multiple encoder layers. Each layer applies a self-attention mechanism to capture bidirectional relationships between words, allowing the model to understand context from both directions in the sentence [24]. The output hidden states from these layers are then fed into a multilabel classification layer specifically designed for multi-class KBLI classification. This final layer generates predictions of the most relevant five-digit KBLI codes along with their associated probabilities, based on the given business description.

2.5 Model Evaluation

In the evaluation phase, the model's ability to recommend five-digit KBLI codes based on business descriptions was assessed using a separate test dataset. Four standard classification metrics were used: precision, recall, F1-score, and accuracy. Precision is the proportion of correctly classified positive samples to all samples predicted as positive, indicating how accurate the model's positive predictions are. Recall, or sensitivity, is the proportion of true positive samples to all actual positive samples, measuring the model's ability to identify all relevant positive cases. The F1-score, which ranges from 0 to 1, is the harmonic mean of precision and recall, offering a balanced evaluation when both false positives and false negatives are important. Accuracy is the proportion of all correctly classified samples to the total number of samples, reflecting the overall correctness of the model's predictions [25].

However, due to potential class imbalance, accuracy alone was considered insufficient, and greater emphasis was placed on precision, recall, and F1-score [26].

The optimal model was selected based on both quantitative metrics and practical considerations such as algorithm behavior, model stability, and computational efficiency in real-time use. Random Forest demonstrated consistent performance under balanced class distributions, whereas Support Vector Machine and IndoBERT exhibited greater sensitivity to class imbalance. Thus, the F1-score is the most suitable metric for evaluating the performance of a classification model on unbalanced data [27]. External validation using the National Labor Force Survey (Sakernas) and the Micro and Small Industry (IMK) Survey of South Sumatra for the years 2022 to 2024 was conducted to assess model robustness and generalizability.

2.6 Model Deployment

In developing the KBLI classification system, a fine-tuned IndoBERT model is deployed as an API service to process user-submitted business descriptions and generate the most relevant KBLI code predictions. This architecture separates the inference process from the user interface, improving the system's modularity, scalability, and ease of maintenance. The API is implemented using FastAPI, a high-performance framework that simplifies backend development and ensures efficient request handling [28]. One of FastAPI's notable features is its automatic generation of interactive API documentation using the OpenAPI standard, which greatly facilitates understanding and direct testing by developers [29]. The service is accessible through a public endpoint, allowing for smooth integration with external systems. To support consistent performance under high demand, the API is deployed with a stable server configuration and detailed online documentation. This approach ensures that the system is not only accurate but also robust and ready for long-term, large-scale use.

3. Result and Discussion

3.1. Evaluation of Machine Learning and Transfer Learning Models

The performance of each model was systematically evaluated using a synthetic dataset. The evaluation metrics include accuracy, precision, recall, F1-score, and computational time

3.1.1. Support Vector Machine (SVM)

The Support Vector Machine (SVM) model's hyperparameters were optimized using a grid search with 5-fold cross-validation. This procedure evaluated various combinations of the C, kernel, and gamma parameters to identify the set that yielded the highest validation performance. The optimal hyperparameter combination is presented in table 2**.**

Table 2. Optimal hyperparameter selection for the SVM model using Grid Search

Parameter	Selected Value
C	10
Kernel	RBF
Gamma	Scale

The grid search identified an optimal C value of 10, indicating that the model prioritizes minimizing training errors, even at the cost of a narrower decision margin. The Radial Basis Function (RBF) kernel was selected, as it consistently outperformed the linear kernel during cross-validation. The gamma parameter was set to 'scale', which automatically adjusts the value based on the number of input features. Following training with these optimal parameters, the SVM model was evaluated on the business description classification task.

The SVM model achieved strong classification performance, with an accuracy of 89%, precision of 90%, recall of 89%, and an F1-Score of 89%. The high F1-Score indicates a robust balance between precision and recall, showing that SVM can effectively separate relevant features even in a high-dimensional KBLI dataset. However, compared to other models, its training time was considerably longer (approximately 4,689 seconds or 78 minutes), while inference time remained efficient at 0.456 seconds, making it suitable for real-time use.

To simulate a real-world scenario, the model was tested on the raw, unprocessed input "INDUSTRI KUE BASAH" (WET CAKE INDUSTRY). The resulting prediction is shown in figure 1.

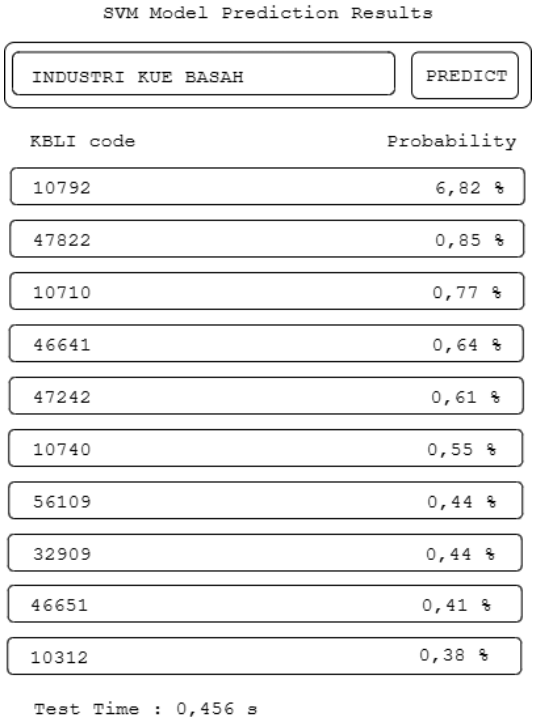

Figure 2. Example of SVM model prediction results.

As shown in figure 2, the model identified KBLI code 10792 as the top prediction with a probability of 6.82%. This code, which corresponds to the 'Wet Cake Industry,' is highly relevant to the input query. However, the low probability scores across all predicted classes suggest the model had low confidence in its top choice. Despite this uncertainty, the presence of other relevant codes among the alternatives indicates that the model successfully captured key patterns from the input, even if it could not assign a high probability to a single, specific class. The inference time of 0.456 seconds for this prediction is considered highly efficient.

3.1.2. Random Forest

The hyperparameter optimization for the Random Forest model began with pre-configuring the feature extraction step. A TfidfVectorizer was initialized with a vocabulary limit of 5,000 features (max_features) and a document frequency threshold of 90% (max_df). Subsequently, a grid search was performed to identify the optimal set of hyperparameters for the Random Forest Classifier. The resulting configuration is presented in table 3.

Table 3. Optimal hyperparameter selection for the Random Forest model using Grid Search

Parameter	Selected Value
n_estimators	100
max_depth	None
random_state	42

The grid search identified n_estimators=100 as the optimal value, striking a balance between predictive power and training efficiency. By setting max_depth to None, each tree in the forest was permitted to grow until all its leaves were pure, maximizing the model's capacity to learn intricate data relationships. Reproducibility was ensured by setting random_state to 42. The final model was trained using a pipeline incorporating these settings

The Random Forest model yielded an accuracy of 83%, precision of 84%, and an F1-Score of 82% (table 3). This demonstrates solid predictive capability and high computational efficiency, requiring only 0.011 seconds for inference. Compared with SVM, Random Forest achieved faster prediction time but slightly lower accuracy. figure 3 displays a sample output for the query "INDUSTRI KUE BASAH" (WET CAKE INDUSTRY).

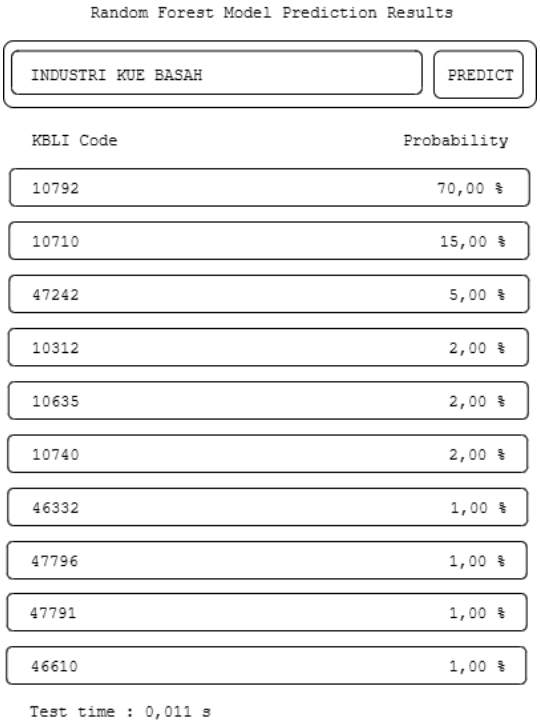

Figure 3. Example of Random Forest model prediction results.

The Random Forest model demonstrated a significant improvement in prediction confidence, assigning a 70% probability to the correct class, KBLI 10792. This high degree of certainty, especially when compared to the SVM's output, reflects a more decisive and accurate classification. The model's secondary predictions, such as 10710 (15%) and 47242 (5%), were also thematically relevant, indicating a well-generalized understanding of the input.

3.1.3. IndoBERT

The implementation of IndoBERT began by loading the pretrained indobenchmark/indobert-base-p1 model and its associated BertTokenizer from Hugging Face. This tokenizer, which is based on the WordPiece algorithm and trained on an Indonesian corpus, was used to convert the input text into a format suitable for the model. Subsequently, the model underwent a fine-tuning phase on the KBLI 2020 dataset. This training was performed using the Adam optimizer with a variable learning rate, configured according to the parameters detailed in table 4.

Table 4. Optimal hyperparameter selection for Fine-tuning IndoBERT.

Parameter	Selected Value
Max Input Length	128
Batch Size	32
DataLoader Workers	2
Learning Rate	5×10^{-6} to 5×10^{-5}
Epochs	10

The fine-tuning process involved a series of experiments to determine the optimal learning rate. Several rates between 5×10^{-6} to 5×10^{-5} were tested to identify the configuration that yielded the best validation performance. The key hyperparameters used during this fine-tuning stage, including batch size and the number of epochs, are detailed in table 5.

Table 5. Evaluation results for the Fine-tuned IndoBERT model with different learning rates.

Epoch	Learning Rate	Batch Size	F1-Score (percent)	Accuracy (percent)	Recall (percent)	Precision (percent)	Computational Time (ms)
20	5×10^{-6}	32	88.77	90.03	89.27	90.55	14
15	7×10^{-6}	32	89.00	90.25	89.49	91.04	14

20	9×10^{-6}	32	92.99	93.40	93.12	94.22	13
15	1×10^{-5}	32	92.23	92.81	92.47	93.50	15
10	3×10^{-5}	32	91.75	92.30	92.02	92.98	13
10	5×10^{-5}	32	91.83	92.32	92.09	93.46	15

The fine-tuned model demonstrated exceptional predictive power. The optimal learning rate of 9×10^{-6} led to a precision of 94.22% and an F1-Score of 92.99%, underscoring its ability to make accurate classifications with a strong balance between precision and recall. The model's robustness is further evidenced by its training and validation loss curves (figure 4).

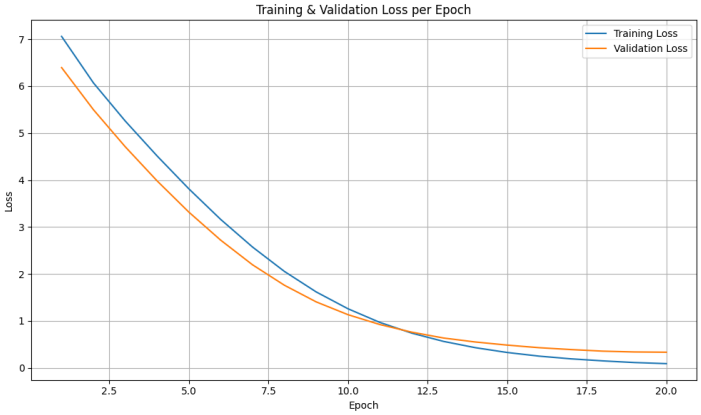

Figure 4. Training & Validation Loss per Epoch at a learning rate of 5×10^{-5}.

These curves display an ideal convergence pattern where the validation loss flattens after epoch 12 instead of increasing. This behavior confirms that the model effectively learned to generalize from the training data while successfully avoiding overfitting. When tested with the input "INDUSTRI KUE BASAH" (WET CAKE INDUSTRY), the Fine-tuned IndoBERT model provided a highly confident prediction, as seen in figure 5.

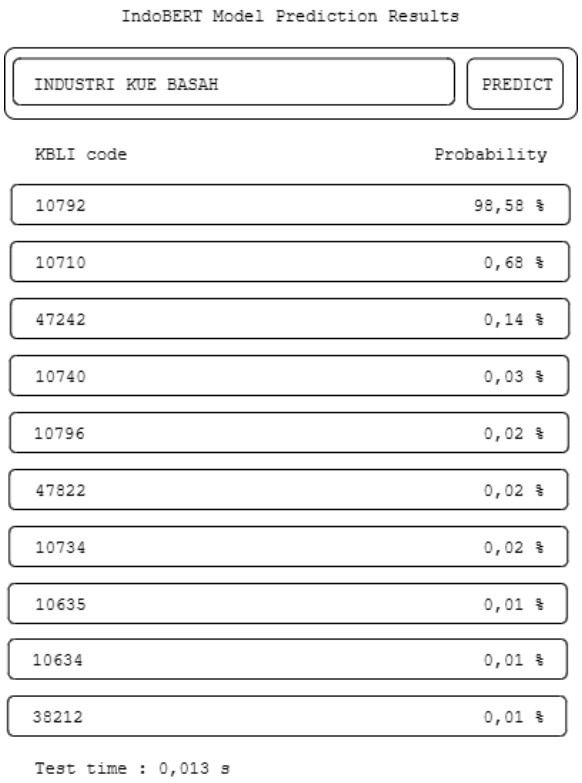

Figure 5. Example of Fine-tuned IndoBERT model prediction results.

The model predicted KBLI code 10792 with an overwhelming probability of 98.58%, demonstrating its superior ability to understand the context of the business description and assign the most relevant code with a high degree of certainty.

3.2. Comparative Analysis and Best Model Selection

To determine the most effective model, a comparative analysis was conducted based on the evaluation metrics and computational load. table 6 provides a summary of the performance of the three models.

Table 6. Performance Comparison of SVM, Random Forest, and Fine-tuned IndoBERT Models.

Model	Accuracy (percent)	Recall (percent)	Precision (percent)	F1-Score (percent)	Computational Time (second)
SVM*	89.00	89.00	90.00	89.00	0.456
Random Forest*	83.00	83.00	84.00	82.00	0.011
Fine-tuned IndoBERT**	93.40	93.12	94.22	92.99	0.013

* trained on v2-8 TPU runtime

** trained on T4 GPU runtime

Based on the evaluation metrics, the Fine-tuned IndoBERT model demonstrated the best performance across the board, achieving the highest accuracy (93.40%), recall (93.12%), precision (94.22%), and F1-Score (92.99%). Regarding computational cost, Fine-tuned IndoBERT's testing time of 0.013 seconds is highly efficient, only marginally slower than Random Forest (0.011s) and significantly faster than SVM (0.456s). This high-speed inference is a major advantage for practical implementation. Although Fine-tuned IndoBERT's training time (2,043 seconds) was longer than Random Forest's (486 seconds), it was chosen as the best model. This decision was based on the favorable trade-off of a substantial 10.4% increase in accuracy over Random Forest for a minimal 2-millisecond increase in testing time. Since model quality was the priority and training is a one-time cost, fine-tuned IndoBERT provides the optimal solution with high accuracy and rapid, practical deployment capabilities.

These findings are consistent with prior studies demonstrating the effectiveness of transformer-based models for Indonesian text classification. A Fine-tuned IndoBERT model achieved the highest F1-score of 87% for KBLI classification, outperforming SVM and Random Forest [8]. Similarly, Fine-tuned IndoBERT attained 97% across all evaluation metrics in a study on exam question classification, further highlighting its strong performance in handling Indonesian-language classification tasks [9].

A detailed evaluation was conducted to assess the Fine-tuned IndoBERT model's predictive performance across all 1,789 KBLI five-digit codes. Since the number of codes is very large, the complete results cannot be fully displayed. Therefore, only the top ten and bottom ten KBLI five-digit codes are presented in table 7, representing the highest and lowest F1-scores based on precision and recall values.

Table 7. Evaluation of Fine-tuned IndoBERT Model on the Top and Bottom Ten KBLI five-Digit.

Five-digit KBLI Codes	Recall (percent)	Precision (percent)	F1-Score (percent)
01115	1.0	1.0	1.0
01191	1.0	1.0	1.0
01160	1.0	1.0	1.0
01150	1.0	1.0	1.0
01140	1.0	1.0	1.0
01139	1.0	1.0	1.0
01136	1.0	1.0	1.0
58120	1.0	1.0	1.0
56305	1.0	1.0	1.0

56304	1.0	1.0	1.0
:	:	:	:
43905	1.00	0.20	0.33
49426	0.50	0.25	0.33
45302	0.50	0.25	0.33
35115	0.29	0.33	0.31
28130	0.33	0.25	0.29
87202	0.25	0.25	0.25
62029	0.50	0.14	0.22
90090	0.00	0.00	0.00
47736	0.00	0.00	0.00
35117	0.00	0.00	0.00

The evaluation shows that the Fine-tuned IndoBERT model achieved strong classification accuracy overall, with approximately 1,000 KBLI five-digit codes reaching perfect precision and recall (100%). Only three codes out of 1,789 recorded both precision and recall values of 0.00, indicating that none of their samples were correctly identified. Several other codes achieved moderate recall but lower precision, showing that while the model could detect some correct samples, some predictions for these codes were incorrect. The results indicate that the model performs accurately for the majority of KBLI five-digit codes, and that the few low-performing codes represent only a small portion of the total number of categories.

3.3. Model Performance on Real-World Case Study Data

A case study was designed to evaluate the models' generalization capabilities on more challenging real-world data. A composite dataset was created, drawing from the National Labor Force Survey (Sakernas) (2022-2024) and the Micro and Small Industry (IMK) survey (2024), comprising 6,231 business descriptions. The models were then tasked with classifying the varied business descriptions from this dataset, with their performance quantified by top-1, top-5, and top-10 prediction accuracy.

Table 8. Model test results on the case study dataset (Sakernas/IMK).

Model	Top-1		Top-5		Top-10	
	Correct Predictions	Accuracy (percent)	Correct Predictions	Accuracy (percent)	Correct Predictions	Accuracy (percent)

SVM	1,487	23.86	2,510	40.28	2,895	46.46
Random Forest	1,547	24.83	2,864	45.96	3,349	53.75
Fine-tuned IndoBERT	2,052	32.93	3,409	54.71	3,941	63.24

From table 8, we find the robustness of the Fine-tuned IndoBERT model, which significantly outperformed its counterparts. With a top-1 accuracy of 32.93% and a top-10 accuracy of 63.24%, it demonstrated a much stronger capability to handle complex, real-world data than either SVM or Random Forest.

However, the general decline in performance across all models when moving from synthetic to real-world data is a critical finding. This suggests that the less-structured and more diverse language found in practical scenarios poses a significant challenge. The result strongly supports the initial hypothesis that training on clean, structured, and synthetically generated data is a more effective strategy for maximizing a model's predictive power.

4. Conclusion

This study successfully automated five-digit KBLI classification utilizing machine learning and transfer learning techniques on Indonesian business descriptions. Fine-tuned IndoBERT as a result of the transfer learning approach, consistently outperformed conventional machine learning models such as Support Vector Machine and Random Forest, achieving better precision and balance across evaluation metrics while ensuring computing efficiency. This work emphasizes the effectiveness of transfer learning for fine-grained, multi-class text classification problems within the Indonesian context, in addition to technical performance. The implementation of Fine-tuned IndoBERT as a RESTful API validates its practical utility, facilitating intuitive integration into statistical frameworks. The proposed method enhances the accuracy and consistency of official statistics processes while promoting the application of artificial intelligence in large-scale economic data management in Indonesia.

References

[1] Badan Pusat Statistik, *Klasifikasi Baku Lapangan Usaha Indonesia (KBLI) 2020*. Jakarta: BPS, 2020.

[2] F. Aiman, N. M. Septiansyah, I. F. Ardianto, E. Sutrisno, and G. Y. Andri, "The excessive implementation of ISIC on the obstruction in the implementation of KBLI 2020," Journal of World Science, vol. 3, no. 6, pp. 582–589 Jun. 2024, doi: https://doi.org/10.58344/jws.v3i6.613

[3] M. R. Syazali and E. Yulianti, "Classification of Economic Activities in Indonesia Using IndoBERT Language Model," *Jurnal Ilmu Komputer dan Informasi*, vol. 18, no. 2, pp. 155–165, Jun. 2025, doi: 10.21609/jiki.v18i2.1446.

[4] A. S. Dwicahayaniawan, T. Devara, and T. Saadi, "Potensi Pemanfaatan Machine Learning dan Transfer Learning untuk Klasifikasi Baku Pekerjaan (Leveraging Machine Learning and Transfer Learning for Standardized Job Classification: A Case Study in West Nusa Tenggara)," in *Seminar Nasional Official Statistics 2024*, 2024.

[5] M. Dwicahyo and B. Yuniarto, "Deep Learning for Indonesia Standard Industrial Classification," in *Proceedings of the International Conference on Electrical Engineering and Informatics*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020. doi: 10.1109/ICELTICs50595.2020.9315361.

[6] United Nations, *International Standard industrial classification of all economic activities (ISIC)*. United Nations, 2008.

[7] H. Bechara, R. Zhang, S. Yuan, and S. Jankin, "Applying NLP Techniques to Classify Businesses by their International Standard Industrial Classification (ISIC) Code," in *Proceedings - 2022 IEEE International Conference on Big Data, Big Data 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 3472–3477. doi: 10.1109/BigData55660.2022.10020787.

[8] L. H. Suadaa, F. Ridho, A. K. Monika, and N. W. K. Projo, "Automatic Text Categorization to Standard Classification of Indonesian Business Fields (KBLI) 2020," in *2023 International Conference on Electrical Engineering and Informatics (ICEEI)*, IEEE, 2023. doi: https://doi.org/10.1109/ICEEI59426.2023.10346866.

[9] F. Baharuddin and M. F. Naufal, "Fine-Tuning IndoBERT for Indonesian Exam Question Classification Based on Bloom's Taxonomy," *Journal of Information Systems Engineering and Business Intelligence*, vol. 9, no. 2, pp. 253–263, Oct. 2023, doi: 10.20473/jisebi.9.2.253-263.

[10] S. Gururangan *et al.*, "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks," May 2020, [Online]. Available: http://arxiv.org/abs/2004.10964

[11] R. Pasunuru, V. Stoyanov, and M. Bansal, "Continual Few-Shot Learning for Text Classification," in *Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2021, pp. 5688–5702. doi: https://doi.org/10.18653/v1/2021.emnlp-main.460.

[12] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow, "A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios," in *Conference ofthe North American Chapter ofthe Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2021, pp. 2545–2568. doi: https://doi.org/10.18653/v1/2021.naacl-main.201.

[13] M. Ihsan Silmi Kaffah *et al.*, "Development of a Synthetic Dataset for the Indonesia Standard Industrial Classification 2020 using Generative Artificial Intelligence," in *Seminar Nasional Official Statistics 2025*, 2025.

[14] H. Dang, S. Goller, L. Mecke, D. Buscheck, and F. Lehmann, "How to Prompt? Opportunities and Challenges of Zero- and Few-Shot Learning for Human-AI Interaction in Creative Applications of Generative Models," in *CEUR Workshop Proceedings*, CEUR-WS, Sep. 2020, pp. 1–9. doi: https://doi.org/10.48550/arXiv.2209.01390.

[15] J.-C. Klie and R. Eckart De Castilho, "Analyzing Dataset Annotation Quality Management in the Wild Iryna Gurevych Ubiquitous Knowledge Processing Lab," 2024, doi: 10.1162/coli.

[16] D. Pakpahan, V. Siallagan, and S. Siregar, "Classification of E-Commerce Product Descriptions with The Tf-Idf and Svm Methods," *sinkron*, vol. 8, no. 4, pp. 2130–2137, Oct. 2023, doi: 10.33395/sinkron.v8i4.12779.

[17] N. Arifin, U. Enri, and N. Sulistiyowati, "PENERAPAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) DENGAN TF-IDF N-GRAM UNTUK TEXT CLASSIFICATION," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. Vol. 6 No. 2, Dec. 2021.

[18] N. Jalal, A. Mehmood, G. S. Choi, and I. Ashraf, "A novel improved random forest for text classification using feature ranking and optimal number of trees," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2733–2742, Jun. 2022, doi: 10.1016/j.jksuci.2022.03.012.

[19] S. U. Hassan, J. Ahamed, and K. Ahmad, "Analytics of machine learning-based algorithms for text classification," *Sustainable Operations and Computers*, vol. 3, pp. 238–248, Jan. 2022, doi: 10.1016/j.susoc.2022.03.001.

[20] C. Che, H. Hu, X. Zhao, S. Li, and Q. Lin, "Advancing Cancer Document Classification with Random Forest," *Academic Journal of Science and Technology*, vol. Vol. 8, No. 1, 2023.

[21] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in *Proceedings ofthe 1st Conference ofthe Asia-Pacific Chapter ofthe Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, Dec. 2020, pp. 843–857. [Online]. Available: https://github.com/annisanurulazhar/absa-playground

[22] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.00677

[23] A. K. Nandanwar and J. Choudhary, "Contextual Embeddings-Based Web Page Categorization Using the Fine-Tune BERT Model," *Symmetry (Basel)*, vol. 15, no. 2, Feb. 2023, doi: 10.3390/sym15020395.

[24] D. Suhartono, M. R. N. Majiid, and R. Fredyan, "Towards automatic question generation using pre-trained model in academic field for Bahasa Indonesia," *Educ Inf Technol (Dordr)*, vol. 29, no. 16, pp. 21295–21330, Nov. 2024, doi: 10.1007/s10639-024-12717-9.

[25] A. S. Rizky and E. Y. Hidayat, "Emotion Classification in Indonesian Text Using IndoBERT," *Computer Engineering and Applications*, vol. 14, no. 1, pp. 2252–4274, 2025.

[26] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0192-5.

[27] S. Riyanto, I. S. Sitanggang, T. Djatna, and T. D. Atikah, "Comparative Analysis using Various Performance Metrics in Imbalanced Data for Multi-class Text Classification," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, p. 2023, 2023, [Online]. Available: http://gcancer.org/pdr

[28] C. D. Bandaranayake, "Development of Scalable Tool for Big Data Analysis and Visualization," 2025.

[29] J. Chen, "Model Algorithm Research based on Python Fast API," *Frontiers in Science and Engineering*, vol. Volume 3, 2023.