INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Automated Matching Skills to Improve the Accuracy of Job Applicant Selection Using Indonesian National Work Competency Standards

Abdul Azzam Ajhari [a,*], Dimas Febriyan Priambodo [b], Henny Yulianti [a]

[a] *Department of Informatics, Universitas Siber Asia, Pasar Minggu, Jakarta, 12550, Indonesia*
[b] *Cyber Security Engineering, Politeknik Siber dan Sandi Negara, Ciseeng, Bogor, 16120, Indonesia*
Corresponding author: [*]abdulazzam@lecturer.unsia.ac.id

***Abstract*— The high number of cyberattack anomalies and data leaks in Indonesia increases the need for cybersecurity in various companies. Cybersecurity capabilities and skills in Indonesia are divided into three categories based on the Indonesian National Work Competency Standards (SKKNI), namely Security Operation Center (SOC), Cybersecurity test/Penetration testing (Pentest), and Information Security Audit. Although various approaches have been applied in different companies to select job applicants, a new method with automated matching is explored in this study. This method matches the skills possessed by prospective job applicants with the profile of their job task requirements based on the SKKNI Decree of the Minister of Manpower of the Republic of Indonesia using Machine Learning (ML) models. The empirical comparison of results comes from automated matchmaking processed by Multinomial Naive Bayes (MNB) and Decision Tree algorithm models. Before modeling, the data is trained and evaluated for testing. Then to assess the most optimal algorithm between MNB and Decision Tree, a confusion matrix is proposed and used to find the best model. From the evaluation results, both models performed well and were highly accurate during training and test evaluation. The Decision Tree model performs slightly better than the MNB model, but both still provide satisfactory results in classifying data based on the Indonesian National Work Competency Standards (SKKNI) categories. This study offers a solution to minimize the number of potential applicants who are not competent in the three SKKNI cybersecurity job categories due to the mismatch of their abilities and skills.**

***Keywords*— Indonesian National Work Competency Standards (SKKNI); automated matching; machine learning; Multinomial Naïve Bayes; decision tree.**

Manuscript received 28 Jul. 2023; revised 24 Dec. 2023; accepted 9 Jan. 2024. Date of publication 31 May. 2024.

I. INTRODUCTION

In 2023, cyber-attacks are increasingly common in Indonesia. One of the most significant attacks that paralyzed financial transactions was the LockBit 3.0 ransomware attack at Bank Syariah Indonesia (BSI) from 8 May 2023 to 11 May 2023. Therefore, special knowledge and skills are required for IT staff to be in line with the development of conditions for cybersecurity needs, which are one of the most critical elements of human resources in Indonesia [1]. Human resource issues related to cyber security have been explained and divided into three categories based on the Decree of the Minister of Manpower of the Republic of Indonesia concerning applying the Indonesian national work competency standards (SKKNI).

The three categories of cyber security competencies include the Security Operation Center (SOC) regulated in the Decree of the Minister of Manpower Number 391 of 2020 [2], The Cyber Security Test/Penetration Testing (Pentest) is regulated in the Decree of the Minister of Manpower Number 23 of 2022 [3], and Information Security Audit is regulated in the Decree of the Minister of Manpower Number 24 of 2022 [4].

Some researchers have studied methods for automatic matching skills between applicant profiles and jobs that can alleviate some of these problems. Such research [5] uses a simple approach: the higher the overall transformation cost, the worse the candidate's position in the ranking, and vice versa. The lower the transformation cost, the better the candidate's position in the final ranking. Models based on automatic calculation of transformation costs use background knowledge (in the form of familiar taxonomies) to evaluate a job applicant's suitability for a job offer in a way that a human expert would. In addition, many studies have focused on keyword search systems that suit the wishes of applicants [6], [7].

Research by [8] performed profile matching using MNB to determine the ideal position of soccer players in the football manager game. MNB is also used for profile matching of prospective tenants to predict their rentability [9]. Having similarities with profile-matching skills, research by [10] utilizes MNB to predict the pass rate of Microsoft Office Specialist (MOS) certification participants. However, only one research by [11] has conducted a profile-matching study to comprehensively assess candidates using MNB to determine or predict employee placement based on their characteristics.

Another model, the decision tree, was used in the study by [12] to predict students' grades and assess their math competency based on profile matching. Other research [13] proposes a capability-matching concept that essentially relies on standardized recipes by offering seamless integration into the available industrial infrastructure. However, only one research [14] decision tree model has similarities and is used for profile matching to determine the promotion of contract employees to permanent employees.

Thus, from all the existing research, no method is suitable or similar to the cybersecurity talent needs in companies based on the Indonesian National Work Competency Standards (SKKNI). This research proposes a new method, utilizing the Indonesian National Work Competency Standards (SKKNI), to create an automated matching model to increase the accuracy of job applicant selection and meet the needs of cybersecurity talent in Indonesia. This study aims to produce the best job applicant selection accuracy performance with the proposed model, namely Multinomial Naïve Bayes (MNB) and decision tree models from existing research. Thus, it can be used as an automated skills-matching framework using the Indonesian National Work Competency Standards (SKKNI) that suits the needs of the cybersecurity market in Indonesia.

II. MATERIAL AND METHOD

This stage discusses implementing MNB and decision tree methods in classifying abilities as an automated matching skills framework. Figure 1 shows the automated matching skills method we can use in this study.

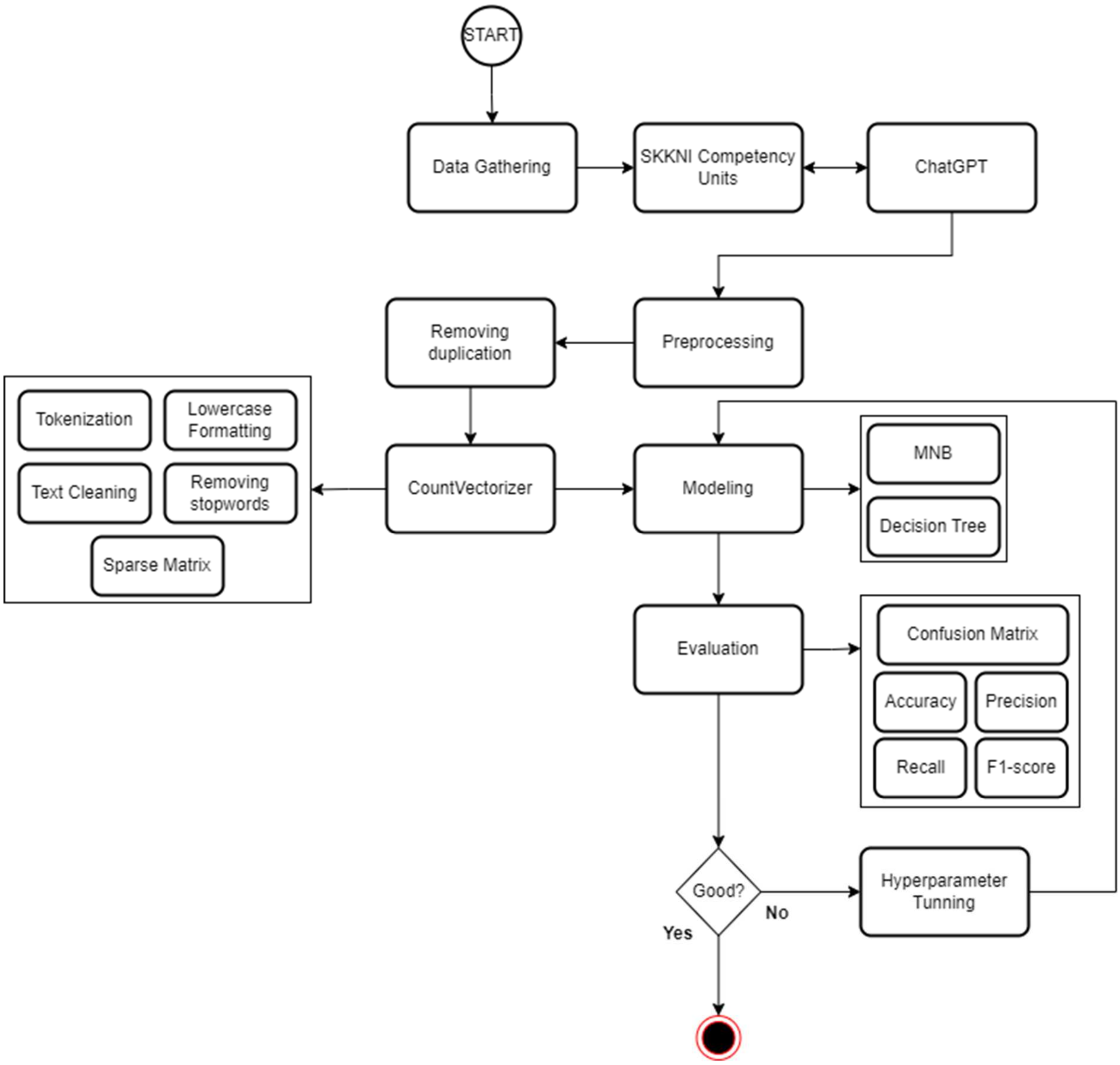

Fig. 1 Automated Matching Skills Method Process

Fig 1 shows four processes: data gathering, pre-processing, modeling, and evaluation, which are explained below.

A. Data Gathering

In this step, ChatGPT is used as a data collection tool by utilizing initial data related to the Indonesian National Work Competency Standards (SKKNI) list of competency units. Using ChatGPT, this research hopes to collect relevant and in-depth information about the unit of competency because ChatGPT performs well in many tasks that prioritize reasoning ability [15] to handle these data annotation tasks [16].

TABLE I
LIST OF SOC COMPETENCY UNITS

No	Unit Code	Competency Unit Name
1	J.62SOC00.001.1	Creating the desired Security Operations Center (SOC) Operating Model and Strategy
2	J.62SOC00.002.1	Designing Security Operations Center (SOC) Capability
3	J.62SOC00.003.1	Develop Cybersecurity Incident Handling Procedures
4	J.62SOC00.004.1	Managing the Cybersecurity Incident Response Team
5	J.62SOC00.005.1	Conduct a Cybersecurity Analysis of a Cybersecurity Incident to Determine Controls
6	J.62SOC00.006.1	Conducting Information Technology (IT) Asset Vulnerability Detection
7	J.62SOC00.007.1	Analyzing Cybersecurity Threats/Anomalies (Threat Intelligence) on the Security Perimeter
8	J.62SOC00.008.1	Monitoring Information Technology (IT) Assets for Cyber Threat Activity
9	J.62SOC00.009.1	Categorizing Cybersecurity Incidents that Occur According to Level of Severity
10	J.62SOC00.010.1	Ticketing Cyber Security Incidents
11	J.62SOC00.011.1	Analyzing Logs on Security Operations Center (SOC)
12	J.62SOC00.012.1	Perform Security Operations Center (SOC) Data Backup
13	J.62SOC00.013.1	Communicating Cybersecurity Incident Handling and Crisis Management
14	J.62SOC00.014.1	Investigate how individuals or groups of criminals operate in carrying out their crime plans (Modus Operandi) Cyber Security Incidents
15	J.62SOC00.015.1	Identifying Technical Solutions to Cyber Security Incidents that Occur
16	J.62SOC00.016.1	Isolating Affected Information Technology (IT) Assets to Stop Cyber Security Incidents
17	J.62SOC00.017.1	Perform Incident-Affected Information Technology (IT) Asset Service Termination for Remediation
18	J.62SOC00.018.1	Analyzing the Impact of a Cybersecurity Incident
19	J.62SOC00.019.1	Ending the Response Process to a Cybersecurity Incident
20	J.62SOC00.020.1	Making Recommendations for Improvement after a Cybersecurity Incident

This method is expected to contribute to gaining a broader understanding of the Indonesian National Work Competency Standards (SKKNI) and improving the quality of work competency development in Indonesia. In Table I, it is known that there are 20 names of competency units from the Indonesian National Work Competency Standards (SKKNI). In obtaining further data, this research uses the query "create 260 lists of technical ability data needed for the SOC field of work from Table I. So, in the data collection process, ChatGPT will provide 260 data regarding technical abilities in the SOC field.

In Table II, it is known that there are 10 names of competency units from the Indonesian National Work Competency Standards (SKKNI). This research uses the query "create 260 lists of technical ability data needed for the cybersecurity test/pentest work field from Table II to obtain advanced data. In the data collection process, ChatGPT will provide a total of 260 data regarding technical capabilities in cybersecurity test/test work.

TABLE II
LIST OF CYBERSECURITY TEST/PENTEST COMPETENCY UNITS

No	Unit Code	Competency Unit Title
1	J.62UKS00.001.1	Planning a Cybersecurity Test Procedure
2	J.62UKS00.002.1	Determine the Vulnerability Assessment Method
3	J.62UKS00.003.1	Determining the Scope of Cybersecurity Testing
4	J.62UKS00.004.1	Managing the Cybersecurity Incident Response Team
5	J.62UKS00.005.1	Collecting Information Required for Cybersecurity Testing
6	J.62UKS00.006.1	Finding Vulnerabilities Within the Scope of Cybersecurity Testing
7	J.62UKS00.007.1	Testing the Vulnerability of the Test Object
8	J.62UKS00.008.1	Conduct Post-Exploitation Activities Based on the Scope of Cybersecurity Testing
9	J.62UKS00.009.1	Compile Cybersecurity Testing Findings
10	J.62UKS00.010.1	Compile Cybersecurity Test Results Report

The Indonesian National Work Competency Standards (SKKNI) in the field of information security audit in Table III, is known to have 21 names of competency units. In obtaining advanced data, this research uses the query "create 260 lists of technical ability data needed for the information security audit work field from Table III. So, in the data collection process, ChatGPT will provide a total of 260 data regarding technical capabilities in the field of information security audit work.

TABLE III
LIST OF INFORMATION SECURITY AUDIT COMPETENCY UNITS

No	Unit Code	Competency Unit Title
1	J.62AKI00.001.1	Defining the Objectives and Scope of an Information Security Audit
2	J.62AKI00.002.1	Conduct Information Security Audit Risk Analysis
3	J.62AKI00.003.1	Create an Information Security Audit Procedure

No	Unit Code	Competency Unit Title
4	J.62AKI00.004.1	Determining Information Security Audit Resource Requirements
5	J.62AKI00.005.1	Implement Information Security Audit Procedures against Organizational Controls
6	J.62AKI00.006.1	Implementing Information Security Audit Procedures for Technology Controls
7	J.62AKI00.007.1	Implement Information Security Audit Procedures on Physical Controls
8	J.62AKI00.008.1	Implement Information Security Audit Procedures on Personnel Controls
9	J.62AKI00.009.1	Create an Information Security Audit Working Paper
10	J.62AKI00.010.1	Creating Information Security Audit Evidence Documentation
11	J.62AKI00.011.1	Overseeing the Adequacy of Audit Implementation by Information Security Audit Procedures
12	J.62AKI00.012.1	Oversee the Technical Feasibility of Implementing Information Security Audit Procedures
13	J.62AKI00.013.1	Overseeing the Appropriateness of Information Security Audit Workpaper Documentation
14	J.62AKI00.014.1	Oversee the Appropriateness of Information Security Audit Evidence Documentation
15	J.62AKI00.015.1	Present the Information Security Audit Procedures Implemented in the Information Security Audit Report
16	J.62AKI00.016.1	Delivering the Information Security Audit Resources Used in the Information Security Audit Report
17	J.62AKI00.017.1	Submitting Information Security Audit Findings in the Information Security Audit Report
18	J.62AKI00.018.1	Submitting Information Security Audit Recommendations in the Information Security Audit Report
19	J.62AKI00.019.1	Delivering Information Security Audit Conclusions
20	J.62AKI00.020.1	Collecting Evidence of Information Security Audit Follow-up Implementation
21	J.62AKI00.021.1	Evaluate Evidence of Follow-Up Implementation of Information Security Audit Recommendations

B. Preprocessing

The total data amounted to 780 from each field of work contained in the Indonesian National Work Competency Standards (SKKNI). This stage uses word and text processing techniques commonly called natural language processing (NLP). The preprocessing carried out in this study is described below.

1) Data duplication removal: it identifies and removes duplicate entries or rows from the dataset so that only one unique entry of the same data remains. In this process, data duplication removal is performed on data with the same attribute value for all or most columns in the dataset. The quality of the data is improved by removing redundancies [17].

2) CountVectorizer: one of the methods used for text processing and text analysis. In this process, a text collection is converted into a numerical vector representation that the model can understand. This method works by calculating the frequency of occurrence of each word (or term) in each document and compiling it in vector form. The CountVectorizer process is divided into 6 processes as follows.

- Tokenization: this process involves dividing the text into smaller units, usually words to recover the elements of interest in a sequence of data [18]. Tokenization can be done in various ways, including by using spaces as separators or more complex ones such as grammar-based separations.
- For text cleaning, data often contain special characters, numbers, punctuation marks, or words that are not relevant to text analysis. Before being used with CountVectorizer, these characters' text is frequently cleaned. Text cleaning has great value in real usage and is an important component in modern writing assistance systems [19].
- Conversion to lowercase, this process is done so that the difference in the size of the letters does not affect the representation of the same word. The given text depends on the chosen way of understanding the boundaries of proper names for various structures [20].
- Stop word removal, this process is necessary to remove common words that appear frequently in the text, most preprocessing steps in NLP [21], and do not provide important information for text analysis.

TABLE IV
VECTORIZATION PROCESS

Class Code	Skill Based on ChatGPT	wireless	network	…
1	Wireless **network** intrusion monitoring analysis	1	**1**	…
2	Mobile app **network** communication vulnerabilities	0	**1**	…
3	Knowledge of secure remote **network** access practices	0	**1**	…

- Vectorization, after the text cleaning of the previous 4 processes is done, Table IV describes the CountVectorizer that counts the frequency of occurrence of each word in each data. It will generate a meaningful numerical vector representation [22] for each document in the data set. Each dimension in the vector will represent a particular word, and the value in that dimension will indicate the number of times that word appears in that document.
- Sparse matrix, which is the result of CountVectorizer, is a form of data representation where most of the elements have a value of one or zero that can better represent a model [23]. This is because many words in the data set rarely appear in every data or document. To save memory space, this representation is usually stored in the form of a sparse matrix.

C. Modeling

1) Multinomial Naïve Bayes (MNB): it is a variant of the Naive Bayes classification method [24] that is specifically used for data with features calculated based on a multinomial distribution [25]. This method is prevalent in text classification, especially in natural language processing. The model assumes that a multinomial distribution represents features (words in the text), and the Naive Bayes assumption is that each feature is independent.

In this research text classification, the MNB model is implemented to classify data (job text) into appropriate categories or labels (e.g., job category or job type). This model performs probability calculations by calculating the probability of each word or feature in the document appearing in a particular category. In this case, the job category or class will be the target class, and the words or features from the job text will be the features for which the probability is calculated.

First, the model will calculate the initial probability for each job category based on the training data. This initial probability is the proportion of the number of jobs in each category to the total number of jobs in the training set. Then each job text is converted into a numerical representation through the feature extraction process, as done using CountVectorizer in the code above. CountVectorizer will calculate the frequency of occurrence of each word in the text. Next, the model will calculate the probability of each word or feature in each job category.

This probability is calculated based on the number of occurrences of words in each category divided by the total number of words in that category. Once the model has been trained with the training data, it can predict new data (e.g. testing data). For each new job text, the model will calculate the probability of each category based on the features present in the text. The category with the highest probability will be the model's prediction for that text.

2) Decision Tree: it is one of the ML algorithms [26] and significant model [27] used for classification [28] and regression tasks. A Decision Tree model can be described as a tree structure consisting of a series of decisions (nodes) that generate predictions [29] on data based on input features. Each node in the tree represents a feature of the data, and each branch of the node represents a different value of that feature. Each branch's end is a leaf with a prediction for a particular class.

The decision tree algorithm searches for the most informative features to separate the data best. Feature selection is based on metrics measuring data diversity and purity (for example, Gini Impurity or Entropy). Once the best feature is selected, the data is split based on the value of the feature. This process is done recursively, and the decision tree continues to be formed until it reaches a stopping condition, such as reaching maximum depth or perfect splitting at a node (e.g., all data at that node has the same label). Once the decision tree is formed, testing or prediction is performed by passing the test data through the decision tree, starting from the root until it reaches a leaf node (childless node). The label at the leaf node will be the prediction for that particular data.

D. Evaluation

After all stages have been carried out, the model can learn the data, and this stage can be carried out. The evaluation process is done using the confusion matrix technique. A confusion matrix is a performance evaluation method commonly used in ML and statistics by [30] to measure the predictive quality of a model. The confusion matrix allows researchers to evaluate model performance in more detail than simple evaluation methods such as accuracy. A confusion matrix is mainly used for classification problems [31], where the model attempts to predict a certain category or label for each data example.

The evaluation process is done by testing the model using independent test data (data not used to train the model). The model's prediction results will be compared with the actual labels of the test data. Each data instance is tested, and its prediction results are compared with the actual labels. Based on the comparison, we calculate True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Once we have the TP, TN, FP, and FN numbers, we can calculate various evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics provide deeper insights into the performance of the model. By looking at the four main elements in the confusion matrix, we can better understand the types of errors made by the model and take appropriate actions to improve its performance.

In our detailed methodology, we employ a unique approach utilizing the Indonesian National Work Competency Standards (SKKNI) to refine the accuracy of job applicant selection. This method integrates advanced Machine Learning algorithms - notably Multinomial Naïve Bayes and Decision Tree - to adeptly match job applicants with the right cybersecurity roles. Our approach stands out for its nuanced application of SKKNI, ensuring a tailored and culturally relevant selection process.

III. Results and Discussion

A. Data Sample Collection

The initial stage of data sampling is done by dividing the data and each class into two parts: training data and test data. The training data is used to train the model and then the test data is used in the evaluation process to determine the model's performance.

TABLE V
SAMPLE DATA SHARING

SKKNI	Class Code	Data
Security Operation Center (SOC)	1	260
Penetration testing (Pentest)	2	260
Information Security Audit	3	260
Total		780

In Table V, the three categories or classes for SOC, cybersecurity test/pentest, and Information Security Audit in the Indonesian National Work Competency Standards (SKKNI) are coded and data preprocessing is performed. Then the required data is divided for model training into 70% and 80% for training data and 20% and 30% for test data after the data separation process. The data is trained using several hyperparameters including the proportion data.

B. Evaluation Result

Six experiments were conducted for each model with various parameters (alpha for MNB and max_depth for decision tree) set and the proportion of data included. The process involved evaluating the training results from the given data and evaluating the test results using the confusion matrix technique. The results in Table VI show the various combinations of hyperparameters and test sizes used in the performance evaluation of the two models. By looking at this table, we can select the most optimal combination of hyperparameters and test sizes for each model based on the evaluation criteria corresponding to the desired use case. The best performance resulting from the six training experiments conducted for the MNB model, we can see that the model with alpha=0.1 and test size 0.2 has the highest training accuracy (94.71%), while for the Decision Tree, the model with max_depth=15 and test size 0.2 has the highest training accuracy (95.99%).

TABLE VI
TRAIN EVALUATION

Model	Hyperparameter		Train Accuracy
Multinomial Naïve Bayes (MNB)	**test_size=0.2**	**alpha=0.1**	**94.71%**
		alpha=0.5	93.10%
		alpha=1.0	92.14%
	test_size=0.3	alpha=0.1	93.95%
		alpha=0.5	93.04%
		alpha=1.0	91.39%
Decision Tree	**test_size=0.2**	max_depth=5	90.54%
		max_depth=10	93.10%
		max_depth=15	**95.99%**
	test_size=0.3	max_depth=5	90.65%
		max_depth=10	93.04%
		max_depth=15	95.60%

TABLE VII
TEST EVALUATION

Model	Hyperparameter		Testing Evaluation (Confusion Matrix)			
			Overall Accuracy	Precision	Recall	F1-score
Multinomial Naïve Bayes (MNB)	**test_size=0.2**	alpha=0.1	83.97%	83.73%	83.97%	83.76%
		alpha=0.5	**85.25%**	**85.54%**	**85.25%**	**85.06%**
		alpha=1.0	84.61%	85.13%	84.61%	84.26%
	test_size=0.3	alpha=0.1	82.05%	82.19%	82.05%	81.67%
		alpha=0.5	84.18%	84.37%	84.18%	83.79%
		alpha=1.0	81.62%	82.29%	81.62%	80.82%
Decision Tree	**test_size=0.2**	max_depth=5	89.74%	90.29%	89.74%	89.60%
		max_depth=10	89.74%	90.34%	89.74%	89.69%
		max_depth=15	**90.38%**	**90.62%**	**90.38%**	**90.36%**
	test_size=0.3	max_depth=5	89.74%	90.01%	89.74%	89.73%
		max_depth=10	90.17%	90.42%	90.17%	90.19%
		max_depth=15	89.74%	89.76%	89.74%	89.74%

It can be seen that the best test evaluation results from Table VII obtained by the MNB model are with hyperparameter alpha = 0.5 and test_size = 0.2 having overall accuracy 85.25%, precision 85.54%, recall 85.25%, and f1-score 85.06%. The best test evaluation results of the decision tree model are with hyperparameter max_depth=15 and test_size=0.2 having an overall accuracy of 90.38%, precision of 90.62%, recall of 90.38%, and f1-score 90.36%. Several misclassifications occur in the best MNB model which can be seen in Fig. 2. Misclassification occurs in category one (SKKNI regarding SOC) with a total of 14 data. Then, category two (SKKNI regarding cybersecurity test/pentest) has a total of 7 data misclassifications. Meanwhile, the third category (SKKNI regarding Information Security Audit) only has 2 data misclassifications.

Several misclassifications occur in the best decision tree model which can be seen in Fig. 2. Misclassification occurs in category one (SKKNI regarding SOC) with a total of 4 data. Then, category two (SKKNI regarding cybersecurity test/pentest) has a total of 10 data misclassifications. Meanwhile, the third category (SKKNI regarding Information Security Audit) only has 1 data misclassification.

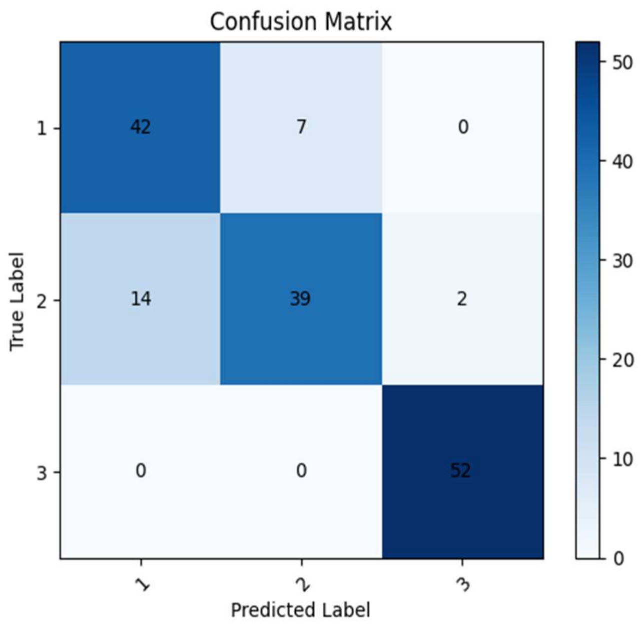

Fig. 2 Best MNB Model Test Evaluation

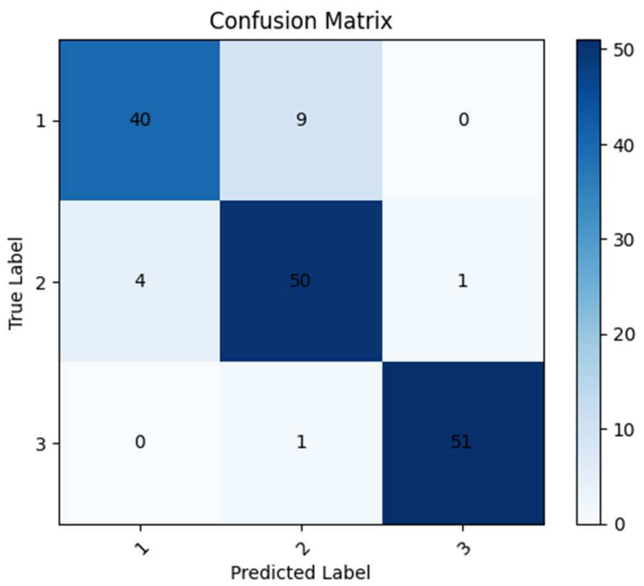

Fig. 3 Best Decision Tree Model Test Evaluation

Furthermore, this study compares the performance of similar previous studies using the MNB model [11] with 93% accuracy and the decision tree [14] with 91.54% accuracy. It was found that the training accuracy of the MNB model used in this study was superior reaching 94.71% but had a low overall accuracy (confusion matrix) of around 85.06%. Meanwhile, the training accuracy of the decision tree model used in this study is superior reaching 95.99% with a lower overall accuracy (confusion matrix) of around 90.36%.

Previous research in automated job matching has explored various methods, yet often lacked the integration of specific national competency standards like SKKNI. Our review of these studies reveals a gap in culturally adaptive methods. By incorporating SKKNI, our approach not only aligns with national standards but also bridges this gap, offering a more effective and contextually relevant solution. Unlike previous methods that apply generic algorithms, our research tailors these algorithms to the Indonesian context. We leverage local competency standards, a step not taken in prior studies. This divergence allows for a more accurate and culturally sensitive matching process, specifically honed for Indonesia's cybersecurity job market.

IV. Conclusion

The MNB model performed well in terms of accuracy and other evaluation metrics, especially during the training process. However, in the test evaluation, the performance dropped slightly but produced quite good results. The model also had a few cases of misclassification, but overall it still produced satisfactory results. The Decision Tree model also performed well, especially during the training process. However, the test evaluation results also showed high performance, although slightly lower than the training results. The model had some cases of misclassification, but overall it still gave satisfactory results.

From the evaluation results, both models performed well with high accuracy during training and test evaluation. The Decision Tree model performs slightly better than the MNB model, but both still provide satisfactory results in classifying data based on the Indonesian National Work Competency Standards (SKKNI) categories.

This research is the first to conduct automated matching skills to improve the accuracy of job applicant selection using the Indonesian National Work Competency Standards (SKKNI). This research can be the basis for conducting a profile-matching process based on the Indonesian National Work Competency Standards (SKKNI) with deeper features as the next improvement for other researchers.

Reflecting on our findings, it is evident that integrating SKKNI into automated job matching presents a significant advancement in the field. This approach not only enhances accuracy but also resonates with the Indonesian job market's unique needs. Future research could further explore this integration in other sectors, potentially transforming job applicant selection across diverse industries.

References

[1] E. Budi, D. Wira, and A. Infantono, “Strategi Penguatan Cyber Security Guna Mewujudkan Keamanan Nasional di Era Society 5.0,” Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia (SENASTINDO), vol. 3, pp. 223–234, Dec. 2021, doi:10.54706/senastindo.v3.2021.141.

[2] K. K. R. Indonesia, *Keputusan Menaker Nomor 391 Tahun 2020 tentang Penetapan Standar Kompetensi Kerja Nasional Indonesia Kategori Informasi dan Komunikasi Golongan Pokok Aktivitas Pemrograman, Konsultasi Komputer, dan Kegiatan yang Berhubungan Dengan Itu (YBDI) Bidang Securi*. Indonesia, 2020.

[3] Kementerian Ketenagakerjaan Republik Indonesia, *Keputusan Menaker Nomor 23 Tahun 2022 tentang Standar Kompetensi Kerja Nasional Indonesia Kategori Informasi Dan Komunikasi Golongan Pokok Aktivitas Pemrograman, Konsultasi Komputer Dan Kegiatan yang Berhubungan Dengan Itu (YBDI) Bidang Uji Keamanan Siber*. Indonesia, 2022.

[4] Kementerian Ketenagakerjaan Republik Indonesia, *Keputusan Menaker Nomor 24 Tahun 2022 tentang Standar Kompetensi Kerja Nasional Indonesia Kategori Informasi dan Komunikasi Golongan Aktivitas Pemrograman, Konsultasi Komputer dan Kegiatan yang Berhubungan Dengan Itu (YBDI) Bidang Audit Keamanan Infor*. 2022.

[5] J. Martinez-Gil, A. L. Paoletti, and M. Pichler, “A Novel Approach for Learning How to Automatically Match Job Offers and Candidate Profiles,” Information Systems Frontiers, vol. 22, no. 6, pp. 1265–1274, Jun. 2019, doi: 10.1007/s10796-019-09929-7.

[6] N. Kamaruddin, A. W. A. Rahman, and R. A. Mohd Lawi, “Jobseeker-industry matching system using automated keyword selection and visualization approach,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 13, no. 3, p. 1124, Mar. 2019, doi:10.11591/ijeecs.v13.i3.pp1124-1129.

[7] A. Mital, N. Siltala, E. Järvenpää, and M. Lanz, “Web-based solution to automate capability matchmaking for rapid system design and reconfiguration,” Procedia CIRP, vol. 81, pp. 288–293, 2019, doi:10.1016/j.procir.2019.03.050.

[8] F. F. Pratama and Y. I. Nurhasanah, “Penggunaan Metode Profile Matching dan Naïve Bayes Untuk Menentukan Starting Eleven pada Sepak Bola,” Jurnal Tekno Insentif, vol. 14, no. 2, pp. 59–68, Aug. 2020, doi: 10.36787/jti.v14i2.268.

[9] R. Rachman and R. N. Handayani, “Klasifikasi Algoritma Naive Bayes Dalam Memprediksi Tingkat Kelancaran Pembayaran Sewa Teras UMKM,” Jurnal Informatika, vol. 8, no. 2, pp. 111–122, Aug. 2021, doi: 10.31294/ji.v8i2.10494.

[10] M. F. Rifai, H. Jatnika, and B. Valentino, “Penerapan Algoritma Naïve Bayes Pada Sistem Prediksi Tingkat Kelulusan Peserta Sertifikasi Microsoft Office Specialist (MOS),” PETIR, vol. 12, no. 2, pp. 131–144, Sep. 2019, doi: 10.33322/petir.v12i2.471.

[11] F. A. Bachtiar, F. Pradana, and R. D. Yudiari, “Employee Recruitment Recommendation Using Profile Matching and Naïve Bayes,” 2019 International Conference on Sustainable Information Engineering and Technology (SIET), Sep. 2019, doi: 10.1109/siet48054.2019.8985988.

[12] A. Ivanov, “Decision Trees for Evaluation of Mathematical Competencies in the Higher Education: A Case Study,” Mathematics, vol. 8, no. 5, p. 748, May 2020, doi: 10.3390/math8050748.

[13] H. Elfaham, J. Grothoff, T. Deppe, M. Azarmipour, and U. Epple, "Recipe Based Skill Matching," 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Jul. 2019, doi:10.1109/indin41052.2019.8972204.

[14] A. Alibasyah, A. Ajiz, G. Dwilestari, K. Kaslani, and E. Wahyudin, "Penerapan Algoritma Decision Tree dalam Penentuan Karyawan Kontrak," MEANS (Media Informasi Analisa dan Sistem), pp. 124–129, Jun. 2022, doi: 10.54367/means.v7i1.1844.

[15] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, "Is ChatGPT a General-Purpose Natural Language Processing Task Solver?," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 1339–1384. [Online]. Available: http://arxiv.org/abs/2302.06476

[16] Y. Zhu, P. Zhang, and E. Haq, *Can ChatGPT Reproduce Human-Generated Labels ? A Study of Social Computing Tasks*, vol. 1, no. 1. Association for Computing Machinery, 2016. doi:10.48550/arXiv.2304.10145.

[17] O. Azeroual, M. Jha, A. Nikiforova, K. Sha, and M. Alsmirat, "A O. Azeroual, M. Jha, A. Nikiforova, K. Sha, M. Alsmirat, and S. Jha, "A Record Linkage-Based Data Deduplication Framework with DataCleaner Extension," Multimodal Technologies and Interaction, vol. 6, no. 4, p. 27, Apr. 2022, doi: 10.3390/mti6040027.

[18] R. Friedman, "Tokenization in the Theory of Knowledge," Encyclopedia, vol. 3, no. 1, pp. 380–386, Mar. 2023, doi:10.3390/encyclopedia3010024.

[19] J. Liao, S. Cheng, and M. Tan, "Text Polishing with Chinese Idiom: Task, Datasets and Pre-trained Baselines," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 22, no. 6, pp. 1–24, Jun. 2023, doi: 10.1145/3593806.

[20] E. V. Beshenkova, "Lowercase or uppercase letter in the names of fairytale heroes and literary characters (from the materials of the academic description of Russian spelling)," Russian language at school, vol. 82, no. 6, pp. 79–86, Nov. 2021, doi: 10.30515/0131-6141-2021-82-6-79-86.

[21] D. J. Ladani and N. P. Desai, "Stopword Identification and Removal Techniques on TC and IR applications: A Survey," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Mar. 2020, doi:10.1109/icaccs48705.2020.9074166.

[22] A. K. Singh and M. Shashi, "Vectorization of Text Documents for Identifying Unifiable News Articles," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 7, pp. 305–310, 2019.

[23] "Logistics networks: a sparse matrix application for solving the transshipment problem," Journal of Mathematical and Computational Science, 2021, doi: 10.28919/jmcs/6654.

[24] J. Chen, Z. Dai, J. Duan, H. Matzinger, and I. Popescu, "Naive Bayes with Correlation Factor for Text Classification Problem," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Dec. 2019, doi: 10.1109/icmla.2019.00177.

[25] S. Bagui, K. Devulapalli, and S. John, "MapReduce Implementation of a Multinomial and Mixed Naive Bayes Classifier," International Journal of Intelligent Information Technologies, vol. 16, no. 2, pp. 1–23, Apr. 2020, doi: 10.4018/ijiit.2020040101.

[26] T. Yu, "Decision tree: basic theory, algorithm formulation and implementation," International Conference on Statistics, Applied Mathematics, and Computing Science (CSAMCS 2021), Apr. 2022, doi: 10.1117/12.2628025.

[27] Y. Meng, "Decision tree model in supervised learning," 2nd International Conference on Applied Mathematics, Modelling, and Intelligent Computing (CAMMIC 2022), May 2022, doi:10.1117/12.2639432.

[28] Y. Lu, T. Ye, and J. Zheng, "Decision Tree Algorithm in Machine Learning," 2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Aug. 2022, doi: 10.1109/aeeca55500.2022.9918857.

[29] F. Cheng, "Application of decision tree in student information management system," 2022 International Conference on Artificial Intelligence in Everything (AIE), Aug. 2022, doi:10.1109/aie57029.2022.00068.

[30] H. Turki, M. Ali, H. Taieb, and M. Ben Aouicha, "Knowledge-Based Construction of Confusion Matrices for Multi-Label Classification Algorithms using Semantic Similarity Measures," in *Proceedings of the Workshop on Data meets Applied Ontologies in Explainable AI*, 2021, vol. 2014, pp. 0–3. doi: 10.48550/arXiv.2011.00109.

[31] I. Markoulidakis, G. Kopsiaftis, I. Rallis, I. Georgoulas, A. Doulamis, and N. Doulamis, "Confusion Matrix Analysis for NPS," 24th Pan-Hellenic Conference on Informatics, Nov. 2020, doi:10.1145/3437120.3437305.