Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Development of Smart Study Web Application for Classifying Student Material Understanding Levels Using Naive Bayes Classifier

Purnomo Hadi Susilo*[1], Vita Ihwatin Mujtahidah[2], Nur Qomariyah Nawafilah[3], Azizul Azhar Ramli[4]

[1,2,3]Informatics Engineering, Faculty of Science and Technology, Universitas Islam Lamongan, Indonesia
[4]Software Engineering Department, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), Malaysia

Email: [1]purnomo@unisla.ac.id

Received : Dec 2, 2025; Revised : Dec 23, 2025; Accepted : Jan 19, 2026; Published : Jun 15, 2026

Abstract

The rapid development of information and communication technology requires adaptive digital learning systems that are able to evaluate students' learning outcomes objectively. However, the Smart Study application previously functioned only as a quiz delivery platform and lacked analytical capabilities to assess students' levels of material understanding, particularly in practical courses such as Computer Networks. This study aims to design and develop a web-based Smart Study application integrated with the Naive Bayes classification algorithm to determine students' understanding levels based on quiz performance data. The research methodology includes data collection from Informatics Engineering students at Universitas Islam Lamongan, followed by data preprocessing through cleaning and categorical conversion of features, including final score, average response time, response time variability, and correct incorrect response time ratio. The dataset was divided into 80% training data and 20% testing data. The Naive Bayes model was trained and evaluated using accuracy, precision, recall, F1-score, and a confusion matrix. The results show that the proposed model achieved an accuracy of 75%, correctly classifying 15 out of 20 testing samples. The model demonstrated strong performance in identifying the Comprehended class with an F1-score of 0.83, while performance for the Not Comprehended class was lower with an F1-score of 0.55 due to class imbalance. This study contributes to the fields of learning analytics and educational data mining by demonstrating the integration of a simple machine learning method into an e-learning application to support early detection of learning difficulties and data-driven evaluation of digital learning processes in higher education.

Keywords : *Data Mining, E-Learning, Naive Bayes, Smart Study application, Student Comprehension Assessment*

1. INTRODUCTION

The development of information and communication technology (ICT) has made the learning process increasingly required to be adaptive through the use of digital platforms. One of its implementations is E-learning, a web-based learning system designed to improve the effectiveness, efficiency, transparency, and convenience of teaching and learning activities [1]. Recent studies highlight that predicting student academic performance is essential for improving educational quality and providing early intervention for students at risk [2]. Machine-learning approaches have therefore been widely adopted in higher education because they allow institutions to analyze large scale learning data more effectively [3]. As digital learning activities continue to expand, the need for analytical systems that can objectively measure learning success becomes even more urgent. These developments indicate that data-driven methods are now an integral part of modern educational evaluation and digital learning ecosystems. In Indonesia, data-driven learning analytics also continues to grow as shown in studies that utilize LMS log data to predict student performance through ensemble machine learning techniques [4]. Other studies demonstrate that Naïve Bayes is capable of classifying student

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

performance based on academic and non-academic variables, proving that this algorithm can be applied across various education levels [5]. As an example of such implementation, a study applying Naive Bayes and Random Forest algorithms to classify students' learning styles based on the VAK model demonstrates how machine-learning techniques can support the analysis of learner characteristics and contribute to improved educational evaluation [6]. This aligns with international findings showing that machine-learning algorithms including Naïve Bayes, Random Forest, SVM, and KNN can accurately predict academic performance based on midterm and final examination date [7]. More recent international research also confirms the effectiveness of classification algorithms in predicting student exam performance with high accuracy [8].

However, not all institutions have been able to provide well-integrated online systems, especially in courses that are practical in nature and require active student engagement. Another challenge that significantly affects the quality of learning is the low reading interest of students, which directly impacts their ability to understand course material [9][10]. Similar classification-based approaches have also been used outside the education domain to interpret user behavior and satisfaction levels. For example, a study conducted at the Rumbai POS Office showed that insights generated from classification models can reveal behavior patterns that are not easily visible through manual evaluation. This finding reinforces that analytical techniques such as Naïve Bayes can help institutions understand user responses more objectively and support more accurate decision-making processes [11].

To measure understanding quickly, quizzes are commonly used as evaluation instruments, especially in digital learning environments where direct supervision is limited [12]. However, the quiz system in the previous Smart Study application was passive because it was not able to provide an analysis of understanding levels. Several previous studies have demonstrated that classification methods can be used to analyze levels of understanding and assess learner performance more accurately. The study by used the K-Nearest Neighbor algorithm and found that most students had a "Poor" level of understanding in online learning environments [13]. Another study used the C4.5 algorithm and produced 27 rules with an understanding accuracy of 70% [14]. Study compared Naïve Bayes and C4.5 and showed that Naïve Bayes performed better with an accuracy of 99% compared to 94% [15]. Furthermore, proved that Naïve Bayes was able to predict student graduation with an accuracy of 90% [16]. Another study by showed that applying SMOTE to Gaussian Naïve Bayes increased the classification accuracy of dropout cases from 84% to 86% [17]. Several Indonesian studies also support this finding, such as the use of Naïve Bayes to predict GPA during online learning and the use of Naïve Bayes to predict students' final grades in higher education [18][19]. Comparative research likewise shows that Naïve Bayes remains competitive among other algorithms such as KNN and SVM in predicting student graduation outcomes [20]. Broader LMS-based research further indicates that machine-learning models integrating participation, access frequency, and assignment performance can produce more comprehensive predictions of student success [21]. These results indicate that machine learning algorithms, especially Naïve Bayes, are highly promising for classification tasks related to student behavior and academic performance.

Despite the extensive use of Naïve Bayes and other classification algorithms in previous studies, most existing research focuses on offline analysis or experimental datasets and does not emphasize direct integration into operational e-learning applications [22]. In particular, limited studies have implemented Naïve Bayes as an automated quiz analysis mechanism embedded within a web-based learning system for practical courses that require immediate feedback and adaptive evaluation. This condition indicates a clear research gap related to the lack of intelligent quiz analysis features that can classify students' understanding levels in real learning environments.

Therefore, the novelty of this research lies in the integration of the Naïve Bayes classification algorithm into the Smart Study web application to perform automatic analysis of quiz results and classify

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

students' levels of material understanding [23]. Unlike previous studies that primarily evaluate algorithmic performance, this research emphasizes practical system implementation to support adaptive learning evaluation and early identification of students who require academic intervention [24].

Based on these findings, this research integrated the Naïve Bayes algorithm to determine students' understanding levels based on quiz data such as the number of correct answers, incorrect answers, completion time, and final score. The model was designed to categorize learners automatically, making the assessment process more adaptive and responsive to individual performance. Accordingly, the objectives of this study are (1) to design and develop a web-based Smart Study application integrated with the Naïve Bayes classification method for automatic quiz analysis, and (2) to evaluate the performance of the proposed model in classifying students' understanding levels using accuracy, precision, recall, F1-score, and confusion matrix metrics [25]. Thus, this research not only presented a web-based learning system but also applied artificial intelligence to support a more adaptive and personalized learning process, particularly in the Computer Networks course at the Islamic University of Lamongan.

2. METHOD

2.1. Data Collection

The data presented in the following table were the results of students' answers after completing a quiz designed to produce classification data [9]. The experimental data collection was conducted on students of Universitas Islam Lamongan, Informatics Engineering Department, class of 2021-D, so the obtained data reflected the real condition of the learning process in that class [26].

Table 1. Initial Data

RT_List	Truth / Correctness	Score	Mean _RT	Std_Dev _RT	RT_ Correct	RT_ Wrong	RT_Correct_ vs_Wrong
[3.76, 13.54, 14.12, 7.79, 7.05]	[0, 1, 1, 1, 1]	80	9.25	4.45	10.62	3.76	2.82
[7.92, 12.41, 2.54, 12.0, 2.04]	[0, 1, 1, 1, 0]	60	7.38	4.97	8.98	4.98	1.8
[10.66, 14.84, 8.09, 3.6, 13.91]	[1, 0, 1, 0, 0]	40	10.22	4.57	9.38	10.78	0.87
[6.22, 14.88, 8.67, 13.39, 2.88]	[0, 1, 1, 1, 1]	80	9.21	4.97	9.96	6.22	1.6
[14.08, 8.49, 12.43, 13.66, 8.27]	[1, 0, 1, 0, 1]	60	11.39	2.81	11.59	11.08	1.05
[10.43, 13.69, 3.35, 6.05, 3.02]	[1, 1, 0, 1, 1]	80	7.31	4.64	8.3	3.35	2.48
[5.42, 10.71, 6.89, 14.81, 10.67]	[1, 1, 0, 1, 1]	80	9.7	3.68	10.4	6.89	1.51
…	…	…	…	…	…	…	…
[9.51, 14.84, 4.38, 3.95, 11.67]	[0, 1, 1, 1, 0]	60	8.87	4.7	7.72	10.59	0.73
[2.81, 6.35, 6.64, 14.31, 4.32]	[1, 1, 1, 0, 1]	80	6.89	4.43	5.03	14.31	0.35
[3.16, 9.24, 9.07, 9.34, 10.06]	[1, 1, 1, 1, 1]	100	8.17	2.83	8.17	1.0	8.17

2.2. Data Cleaning

Data cleaning was performed to remove missing, inconsistent, and irrelevant entries, as such issues may reduce the accuracy of the data mining process [27]. The cleaning involved integration, transformation, and reduction stages. Since Naïve Bayes supports nominal, ordinal, and continuous attributes, no additional transformation was required [28]. Four primary attributes were selected: score, mean response time, standard deviation of response time, and correct vs. incorrect response time. Table 2 shows the cleaned dataset used for further processing [3].

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Table 2. Data Cleaning

Score	Mean_RT	Std_Dev_RT	RT_Correct_vs_Wrong
80	9.25	4.45	2.82
60	7.38	4.97	1.8
40	10.22	4.57	0.87
80	9.21	4.97	1.6
60	11.39	2.81	1.05
80	7.31	4.64	2.48
80	9.7	3.68	1.51
…	…	…	…
60	8.87	4.7	0.73
80	6.89	4.43	0.35
100	8.17	2.83	8.17

2.3. Data Conversion

Data conversion transformed numerical values into categorical labels to meet the Naïve Bayes classification requirements [29]. The final score was converted into comprehension levels as shown in Table 3. Additional categorical features mean response time, response time variability, and response time ratio were also generated (Tables 4-6) [11]. These converted attributes served as inputs for the classification model used to predict students' understanding levels [16].

Table 3. Student Comprehension Categories

Category	Final Score Threshold
Low	≤ 40
Medium	41 – 74
High	≥ 75

The score value was calculated based on the comparison between the number of correct answers and the total number of questions, using the formula:

Equation 1. Correct Count

$$\textit{Correct Count} = \frac{\textit{Correct Answers Count}}{\textit{Total Questions}} \times 100\% \quad (1)$$

Table 4. Categories of Mean Response Time (Mean_RT)

Mean RT Score (Seconds)	Speed Category
< 5	Fast
5 – 10	Medium
> 10	Slow

Table 5. Categories of Response Time Variability (Std_Dev_RT)

Std_Dev_RT Score (Seconds)	Consistency Category
< 2	Consistent
2 – 4	Slightly Variable
> 4	Variable

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Table 6. Categories of Response Time Ratio (Correct vs Incorrect)

RT Ratio (Correct/Incorrect)	Correct RT vs Incorrect Category
< 0.9	Fast on Incorrect
0.9 – 1.1	Balanced
> 1.1	Fast on Correct

2.4. Dataset

After preprocessing, a dataset of 100 entries was prepared. Each entry contained four predictor attributes and one target class (comprehension level) [21]. Table 7 illustrates several samples of the processed dataset.

Table 7. Dataset

Score	Mean_RT	Std_Dev_RT	RT Correct vs Incorrect	Comprehension
High	Medium	Variable	Fast on Correct	Comprehended
Medium	Medium	Variable	Fast on Correct	Not Comprehended
Low	Slow	Variable	Fast on Incorrect	Not Comprehended
High	Medium	Variable	Fast on Correct	Comprehended
Medium	Slow	Slightly Variable	Balanced	Not Comprehended
High	Medium	Variable	Fast on Correct	Comprehended
…	…	…	…	…
Medium	Medium	Variable	Fast on Incorrect	Not Comprehended
High	Medium	Variable	Fast on Incorrect	Not Comprehended
High	Medium	Slightly Variable	Fast on Correct	Comprehended

2.5. Data Spliting

Following previous recommendations, the dataset was divided into 80% training data and 20% testing data, a ratio commonly used for Naïve Bayes classification tasks [4]. The training data were used to calculate prior and conditional probabilities, while the testing data were employed to evaluate model performance [20]. Samples of training and testing data are shown in Tables 8 and 9.

Table 8. Data Training

X1	X2	X3	X4	Y
High	Medium	Variable	Fast on Correct	Comprehended
Medium	Medium	Variable	Fast on Correct	Not Comprehended
Rendah	Slow	Variable	Fast on Incorrect	Not Comprehended
High	Medium	Variable	Fast on Correct	Comprehended
Medium	Slow	Slightly Variable	Balanced	Not Comprehended
High	Medium	Variable	Fast on Correct	Comprehended
High	Medium	Slightly Variable	Fast on Correct	Comprehended
…	…	…	…	…
High	Medium	Slightly Variable	Fast on Correct	Comprehended
Medium	Medium	Slightly Variable	Fast on Incorrect	Not Comprehended
High	Slow	Slightly Variable	Fast on Correct	Comprehended

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Table 9. Data Testing

X1	X2	X3	X4	Y
High	Medium	Variable	Fast on Incorrect	Not Comprehended
High	Medium	Slightly Variable	Fast on Correct	Comprehended
High	Medium	Slightly Variable	Fast on Correct	Comprehended
High	Medium	Slightly Variable	Fast on Correct	Comprehended
Low	Medium	Slightly Variable	Fast on Correct	Not Comprehended
Low	Low	Consistent	Fast on Incorrect	Not Comprehended
----	----	----	----	----
Medium	Medium	Variable	Fast on Incorrect	Not Comprehended
Medium	Medium	Variable	Fast on Incorrect	Not Comprehended
High	Medium	Variable	Fast on Incorrect	Not Comprehended
High	Medium	Slightly Variable	Fast on Correct	Comprehended

2.6. Naive Bayes Method

Naïve Bayes is a probabilistic classification technique that assumes independence among predictor attributes [6]. In this study, the Naive Bayes algorithm was applied to classify students' levels of material understanding into two classes, namely Comprehended and Not Comprehended, and these terms were used consistently throughout the system and analysis sections [30]. It calculates the probability of each class based on the features contained in the data [31]. The classification result is determined by selecting the class with the highest posterior probability [32]. The mathematical formulation of the Naive Bayes theorem is presented in Equation (2), while the step-by-step classification procedure is illustrated in Figure 1.:

Equation 2. Naïve Bayes Theorem

$$\boldsymbol{P(H|C) = \frac{P(C|H).P(H)}{P(C)}} \tag{2}$$

where:

- C = Data whose class was not yet known.
- H = The hypothesis that the data belonged to a specific class.
- P(H|C) = The probability of hypothesis H based on condition C (Posterior Probability).
- P(H) = The probability of hypothesis H (Prior Probability).
- P(C|H) = The probability of C based on H.
- P(C) = The probability of C.

Figure 1 specifically represents the Naive Bayes classification workflow, including prior probability calculation, conditional probability estimation, posterior probability computation, and class selection based on maximum probability [17]. The flowchart illustrates the Naïve Bayes classification process, beginning with the input of training and testing data. The system first computes the prior probability for each class (P(H)) and then calculates the conditional probabilities of each attribute given the class (P(C|H)) [2]. Using these values, the posterior probability (P(H|C)) is obtained to determine how likely the test data belong to each class. Because P(C) is constant for all classes, it is omitted from the calculation [14]. The class with the highest posterior probability is finally selected as the prediction result.

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

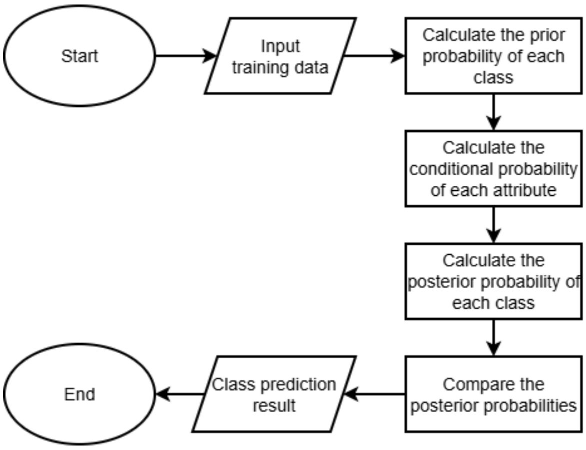

Figure 1. Naïve Bayes Procedure

2.7. Model Evaluation

Model evaluation used standard metrics: accuracy, precision, recall, and F1-score [33]. The confusion matrix summarizes the model's prediction results in terms of true and false classifications These metrics provide a comprehensive assessment of the classifier's performance, especially in identifying both correctly and incorrectly predicted classes [34]. Table 10 summarizes the formulas used to compute each metric [35].

Table 10. Performance Measurement Formula

Matrix	Formula
Accuracy	(TP+TP+TP) / (Total Data)
Precision	TP / (TP+FP/FN+FP/FN)
Recall	TP / (TP+FP/FN+FP/FN)
F1-Score	$1 \times \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})}$

2.8. Research Procedure

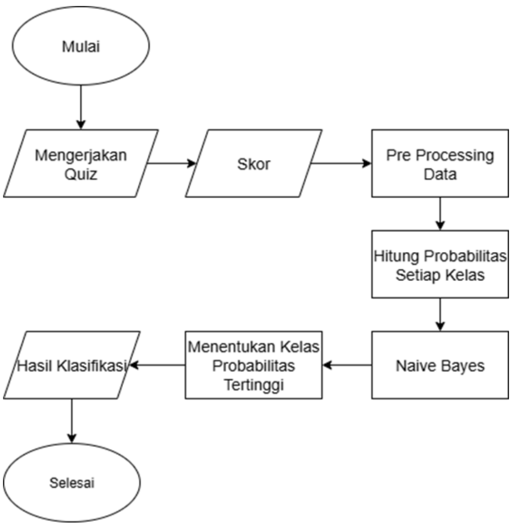

Figure 2. Overall Research Methodology Flowchart

P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

To ensure clarity and reproducibility, this study followed a systematic research procedure consisting of several main stages: data collection, data preprocessing, feature categorization, dataset splitting, model training using the Naive Bayes algorithm, performance evaluation, and system implementation within the Smart Study application [36]. The overall research workflow is illustrated in Figure 2, which presents the complete methodological stages from raw quiz data acquisition to classification result output.

Figure 2 illustrates the complete research process, beginning with quiz data collection from students, followed by preprocessing and feature categorization, model training and testing, performance evaluation using standard metrics, and final deployment of the classification model in the Smart Study web application [37]. This flowchart complements Figure 1, which specifically describes the internal procedure of the Naive Bayes classification algorithm [38].

2.9. System Implementation and Tools

To support the practical implementation of the proposed method, the Smart Study application was developed as a web-based system [12]. The application backend was implemented using PHP for server-side processing and MySQL for data storage, while the frontend utilized HTML, CSS, and JavaScript to provide interactive user interfaces [39]. The Naive Bayes classification process was implemented programmatically within the application to automatically compute prior probabilities, conditional probabilities, and posterior probabilities based on quiz input data [40]. All calculations for training, testing, and evaluation metrics including accuracy, precision, recall, F1-score, and confusion matrix were performed using algorithmic logic embedded in the application [41]. This implementation enabled real-time classification of students' understanding levels immediately after quiz completion. The system design ensured that classification results could be updated dynamically as new quiz data were added to the dataset, supporting adaptive learning evaluation within the Smart Study platform [42].

3. RESULT

In this section, the researcher applied the Naive Bayes method that had been explained in the previous section to perform the training, testing, probability calculation processes, as well as the system implementation in the Smart Study application.

3.1. Manual Calculation of Naïve Bayes

This subsection provides an example of the manual computation process to illustrate how Naïve Bayes performs classification and generates predictions for the selected test data. The example uses all 100 available entries as training data.

3.1.1. Calculating Prior Probability

The prior probability was calculated using 100 data entries, producing two classes: "Understand" and "Not Understand." Table 11 summarizes the distribution of each attribute value across both classes and the resulting prior probabilities used in the next calculation stage.

3.1.2. Calculating Posterior Probability

After the calculation of the Prior Probability was completed, the next step was to calculate the Posterior Probability by utilizing the Prior Probability values for the new case. The case that was used was shown in Table 12.

The table 12 represented an example of test data as a new case that was used by the researcher to calculate and determine the output of class Y. After determining the test data, the next step was to identify the number of occurrences for each test data attribute, namely X1, X2, X3, and X4. The input data were then transformed based on the categorization of each variable.

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Table 11. Posterior Probability

Variable Attribute		Total Data (J))	Comprehended	Not Comprehended	P(H)C Comprehended	P(H)C Not Comprehended
Total		100	40	60	0,4	0,6
Score	High	60	40	20	1,5	1
	Medium	23	0	23	0,575	0,383333
	Low	17	0	17	0,425	0,283333
Mean_RT	Fast	1	1	0	0,025	0,016667
	Medium	81	33	48	2,025	1,35
	Low	18	6	12	0,45	0,3
Std_dev_RT	Consistent	4	0	4	0,1	0,066667
	Slightly Variable	56	25	31	1,4	0,933333
	Variable	40	15	25	1	0,666667
RT_Benar_salah	Fast on Incorrect	40	0	40	1	0,666667
	Balanced	9	5	4	0,225	0,15
	Fast on Correct	51	35	16	1,275	0,85

Table 12. Test Data

X1	X2	X3	X4	Y
60	8	3	1	?

Table 13. Input Data Categorization

Variable	Input	Categorization
X1	60	Medium
X2	8	Medium
X3	3	Slightly Variable
X4	1	Balanced

In the Table 13 were the categorization results based on the boundary rules (categorization) that had been determined previously.

Table 14. Posterior Probability

Test Data Attribute	Test Data Score	P(H)C Comprehended	P(H)C Not Comprehended
Score	Medium	0,575	0,38333
Mean_RT	Medium	2,025	1,35
Std_Dev_RT	Slightly Variable	1,4	0,933333
RT_Benar_Salah	Balanced	0,225	0,15

After the attributes were calculated, the next step was to multiply all the values. The result corresponded to the class of the X data being evaluated. The calculation was as follows:
Formula:

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Equation 3. Naïve Bayes Classification Formula for Multiple Attributes

$$\mathbf{P(C).P(H|C) = P(C).\prod_{i=1}^{n} P(H_i|C)} \quad (3)$$

Calculations:

P(H|Comprehension=Comprehended)=P(Score=Medium|Comprehension=Comprehended)×P(Mean_RT=Medium|Comprehension=Comprehended)×P(Std_dev_RT=Slightly_Varied|Comprehension=Comprehended)×P(RT_Correct_Wrong=Balanced|Comprehension=Comprehended)P(Comprehended)·P(H|Comprehended)=0.575×2.025×1.4×0.225=0.36678.

P(H|Comprehension = Not Comprehended) = P(Score = Medium | Comprehension = Not Comprehended) ×P (Mean _ RT = Medium | Comprehension = Not Comprehended) × P (Std_dev_RT=Slightly_Varied | Comprehension = Not Comprehended) × P (RT_Correct_Wrong = Balanced | Comprehension = Not Comprehended) P (Not Comprehended) ·P (H|Not Comprehended) =0.38333 × 1.35 × 0.9333 × 0.15 = 0.07245.

From these results, we could still perform a maximization calculation to obtain the maximal outcome by multiplying (H1) * (H2). The following table showed the computations:

Table 15. Maximization Results

P(H) (Understanding = Understand)		
H1	H2	Results
0,36678	0,4	0,14671
P(H) (Understanding =Not Comprehended)		
H1	H2	Results
0,07245	0,6	0,04347

The calculations in the data were the results of the posterior probability values (H1) that were multiplied by the probabilities of each class, namely Comprehended and Not Comprehended (H2). Based on the highest value shown in Table 15, it was concluded that the data were classified into the Comprehended class in determining the students' level of understanding.

3.1.3. Calculating System Accuracy

To measure the accuracy of the prediction system, testing was carried out using the testing data that had been explained in the previous chapter, which consisted of 20 data entries in Table 16. The data were tested by comparing the actual class and the predicted class, resulting in either matching or non-matching outputs. Based on these results, the accuracy value was then calculated, and an evaluation was performed using a confusion matrix to obtain the precision, recall, and F1-score values.

In the table 16, it could be seen that there were 20 data entries used in the accuracy testing. From all of these entries, 15 were classified as correct and 5 were classified as incorrect. Based on these results, the accuracy calculation could then be performed using the following formula:

Equation 4. Accuracy

$$Accuracy = \frac{Total\ TP}{Total\ Data} \times 100 \quad (4)$$

$$Accuracy = \frac{15}{20} \times 100$$

$$Accuracy = 75\%$$

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Table 16. Accuracy Test Results

ID	Actual Class	Predicted Class	Results
6	Understand	Not Comprehended	Not Matched
3	Understand	Not Comprehended	Not Matched
20	Understand	Understand	In accordance
43	Understand	Understand	In accordance
45	Not Comprehended	Not Comprehended	In accordance
61	Not Comprehended	Not Comprehended	In accordance
29	Not Comprehended	Not Comprehended	In accordance
15	Not Comprehended	Not Comprehended	In accordance
34	Understand	Not Comprehended	Not Matched
80	Not Comprehended	Not Comprehended	In accordance
98	Not Comprehended	Not Comprehended	In accordance
86	Understand	Not Comprehended	Not Matched
85	Not Comprehended	Not Comprehended	In accordance
68	Not Comprehended	Not Comprehended	In accordance
31	Not Comprehended	Not Comprehended	In accordance
11	Not Comprehended	Not Comprehended	In accordance
38	Not Comprehended	Not Comprehended	In accordance
24	Not Comprehended	Not Comprehended	In accordance
17	Understand	Understand	In accordance
19	Understand	Not Comprehended	Not Matched

3.1.4. Confusion Matrix Calculation

To obtain a deeper understanding of the classification model's performance, calculations using the confusion matrix were also carried out. This method allowed a comprehensive evaluation of the metrics for each class. Through the confusion matrix, the precision, recall, and F1-score values were also calculated, providing an overview of the model's ability to classify the data correctly or incorrectly. The following confusion matrix was presented based on the calculation results in Table 17.

Table 17. Confusion Matrix

Actual / Predicted	Understand	Not Comprehended	Total Actual
Understand	12	0	12
Not Comprehended	5	3	8
Total Predicted	17	3	20

3.1.5. Precision, Recall and F1-Score Calculation

Based on the results in Table 17 of the confusion matrix, calculations of precision, recall, and F1-score for each class, namely Comprehended and Not Comprehended, could be carried out.

Equation 5. Precision

$$\text{Precision} = TP : (TP + FP) \tag{5}$$

Equation 6. Recall

P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Recall = TP : (TP + FN) (6)

Equation 7. F1-Score

F1-Score = 2 × (Precision × Recall) / (Precision + Recall) (7)

Precision calculations for each class:
Comprehended:
Precision = 12 / (12 + 5) = 0.706 (70.6%)

Not Comprehended:
Precision = 3 / (3 + 0) = 1.0 (100%)

Recall calculations for each class:
Comprehended:
Recall = 12 / (12 + 0) = 1.0 (100%)

Not Comprehended:
Recall = 3 / (3 + 5) = 0.375 (37.5%)
F1-score calculations for each class:

Comprehended:
F1 Score = 2 × (0.706 × 1.0) / (0.706 + 1.0)
= 1.412 / 1.706
= 0.828 (82.8%)

Not Comprehended:
F1 Score = 2 × (1.0 × 0.375) / (1.0 + 0.375)
= 0.75 / 1.375
= 0.545 (54.5%)

The model had an accuracy of 75%, with 15 out of 20 test data correctly classified. In the "Comprehended" class, the model showed good performance with an F1-score of 0.83. However, in the "Not Comprehended" class, its performance was still low with an F1-score of 0.55 due to the low recall value, even though the precision reached 100%. Thus, it could be concluded that the model was quite effective in classifying data in the "Comprehended" class but was less optimal in detecting the "Not Comprehended" class. Therefore, improvements were needed, such as balancing the training data or using a more suitable algorithm to handle class imbalance.

3.2. System Implementation

3.2.1. Login Page

The figure 3 displayed the result of the login page implementation, which was the initial page accessed by users before entering the e-learning system. This page functioned to restrict access so that only users who had access rights could enter the system. On the login page, a form was provided containing Email and Password inputs according to the user's data, as well as a Login button to perform the access validation process.

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

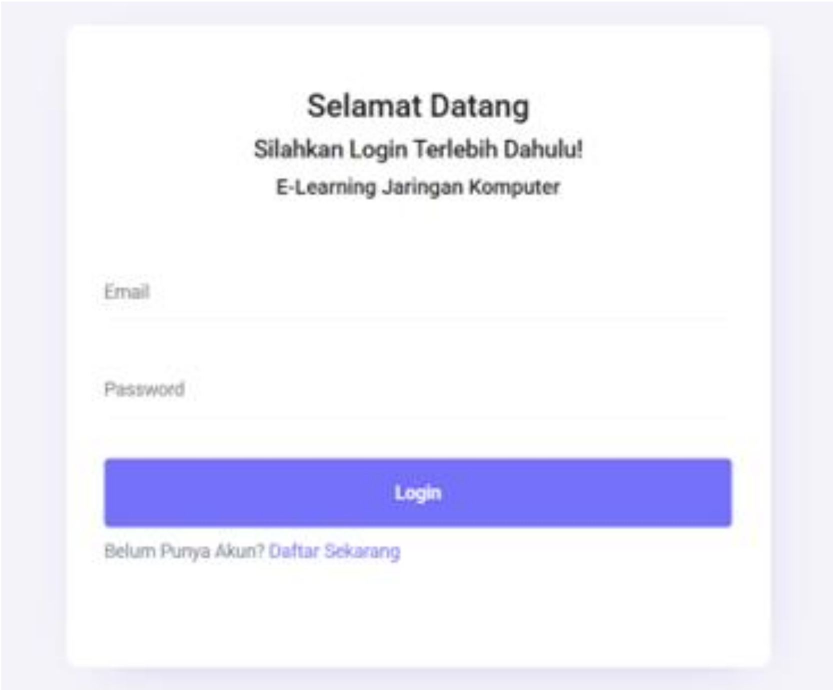

Figure 3. Login Page

3.2.2. Student Classification Data Page

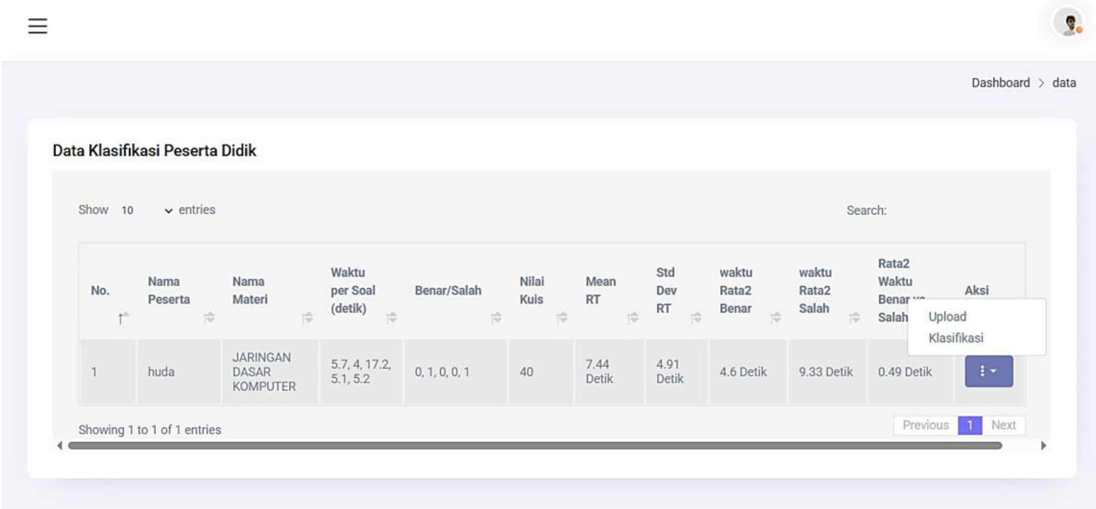

Figure 4. Student Classification Data Page

The figure 4 displayed the student data that had completed the quiz along with their classification results. Through this page, the admin could upload the data into the dataset for updating the classification data so that the model's accuracy became more valid and relevant.

3.2.3. Classification Results Page

This page displayed the calculation process and the classification results of the learners who had completed the quiz. The system automatically calculated the probability of each class (Understood and Not Understood) based on the learners' input data, such as answering time, quiz score, and response time between questions.

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Figure 5. Classification Results Page

The calculation section showed the probability values for each attribute in every class, while the result section showed the class with the highest probability. In the example shown, the system classified the learner into the Not Understood category.

3.2.4. Accuracy Page

Figure 6. Accuracy Page

The Accuracy page in the figure 6 displayed the values of Accuracy, Precision, Recall, and F1-score that had been calculated from 20% of the dataset, where it showed the categorization of the

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

predetermined time-based results. Through this page, users could understand how effective the system was.

4. DISCUSSIONS

Beyond the evaluation of classification performance, this study has important implications for the field of informatics, particularly in the areas of machine learning and digital learning analytics. The integration of the Naive Bayes algorithm into the Smart Study e-learning application demonstrates how a simple and interpretable machine learning method can be effectively embedded within a real educational system to support automated learning evaluation [15] [16]. This implementation highlights the practical role of artificial intelligence in transforming conventional e-learning platforms into adaptive systems capable of analyzing student learning behavior in near real time [34].

From a computer science perspective, this research contributes empirical evidence regarding the behavior of Naive Bayes when applied to educational datasets characterized by categorical features and class imbalance, a common condition in learning analytics [17] [35]. The findings reinforce previous international studies indicating that Naive Bayes tends to favor majority classes, resulting in reduced recall for minority categories, as observed in the Not Understood class [15] [36]. This insight is valuable for researchers and developers who design machine learning-based educational systems, as it emphasizes the importance of data balancing and feature representation in model performance.

In the context of Indonesian higher education, this study addresses the growing urgency for AI-supported learning evaluation tools by providing a concrete example of a locally developed e-learning application integrated with machine learning [18][19]. The Smart Study application illustrates how Naive Bayes can be utilized to support early identification of students who experience learning difficulties, thereby enabling data-driven academic interventions and personalized learning strategies. This contribution is particularly relevant for practical courses, where timely feedback and adaptive assessment play a crucial role in improving learning outcomes [16] [34].

Furthermore, this research opens opportunities for future improvements and extensions. Techniques such as Synthetic Minority Over-sampling Technique (SMOTE) can be applied to mitigate class imbalance and potentially improve recall for minority classes [17] [36]. Additionally, comparative evaluation using alternative algorithms such as K-Nearest Neighbor, Random Forest, or Support Vector Machine has been suggested in previous studies to achieve more balanced and robust results in academic performance prediction tasks [15] [35]. These directions confirm the relevance of this study as a foundation for continued research in educational artificial intelligence and adaptive e-learning systems.

5. CONCLUSION

This study concludes that the integration of the Naive Bayes classification algorithm into the Smart Study e-learning application is effective in automatically determining students' levels of understanding based on quiz performance data. The evaluation results show that the proposed model achieved an accuracy of 75%, with strong performance in classifying the Understood category, while performance for the Not Understood category was limited due to class imbalance and the categorical representation of features. These findings indicate that Naive Bayes is well suited for identifying dominant learning patterns but requires additional handling to improve minority class detection.

From the perspective of informatics and computer science, this research provides a concrete contribution by demonstrating the practical implementation of a simple and interpretable machine learning algorithm for automated learning evaluation in a real e-learning environment. The proposed approach illustrates how lightweight classification models such as Naive Bayes can be effectively applied to support adaptive learning systems, learning analytics, and early identification of students who experience learning difficulties in higher education.

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Furthermore, this study opens several directions for future research. The classification performance may be improved by applying data balancing techniques such as Synthetic Minority Over-sampling Technique (SMOTE) or by exploring alternative machine learning algorithms, including K-Nearest Neighbor, Random Forest, and deep learning-based approaches. Future studies may also consider larger and more diverse datasets, as well as testing the proposed model on different courses beyond Computer Networks, to enhance the generalizability and robustness of the system. Overall, this research supports the advancement of educational technology by strengthening the role of machine learning in developing more adaptive, data-driven e-learning systems within the field of computer science.

REFERENCES

[1] B. Murtiyasa, A. N. Aulida, and M. A. A. bin Abdullah, "PYTHAGORAS : Jurnal Matematika dan Pendidikan Matematika Analisis efektivitas Brainly sebagai platform e-learning untuk meningkatkan minat belajar siswa SMA," *PYTHAGORAS J. Pendidik. Mat.*, vol. 16, no. 2, pp. 141–150, 2021, doi: https://doi.org/10.21831/pythagoras.v16i2.44875.

[2] D. Kurniasari, R. N. Hidayah, and R. K. Nisa, "CLASSIFICATION MODELS FOR ACADEMIC PERFORMANCE: A COMPARATIVE STUDY OF NAÏVE BAYES AND RANDOM FOREST ALGORITHMS IN ANALYZING UNIVERSITY OF LAMPUNG STUDENT GRADES," *J. Tek. Inform.*, vol. 5, no. 5, pp. 1267–1276, 2024, doi: https://doi.org/10.52436/1.jutif.2024.5.5.2066.

[3] D. A. Fauziah, A. Muliawan, and M. Dimyati, "IMPLEMENTATION OF MACHINE LEARNING ON EMPLOYEE ATTRITION BASED ON PERFORMANCE PARAMETERS USING PARTICLE SWARM IMPLEMENTASI MACHINE LEARNING PADA ATRISI KARYAWAN BERDASARKAN PARAMETER KINERJA DENGAN MENGGUNAKAN METODE," *J. Tek. Inform.*, vol. 5, no. 6, pp. 1823–1831, 2024, doi: https://doi.org/10.52436/1.jutif.2024.5.6.3442.

[4] M. Ardianti, O. D. Nurhayati, and B. Warsito, "Model Prediksi Kinerja Siswa Berdasarkan Data Log LMS Menggunakan Ensemble Machine Learning," *JST (Jurnal Sains Dan Teknol.*, vol. 12, no. 3, pp. 562–571, 2023, doi: https://doi.org/10.23887/jstundiksha.v12i3.59816.

[5] A. A. Pekuwali, V. O. Bano, A. D. D. Panja, and F. I. Prasetyo, "Exploration of Influential Variables and Accuracy of the Naive Bayes Classifier Algorithm for Classifying the Performance of Elementary School Students Eksplorasi Variabel Berpengaruh dan Akurasi Algoritma Naive Bayes Classifier untuk Mengklasifikasikan P," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 5, no. 3, pp. 819–829, 2025, doi: https://doi.org/10.57152/malcom.v5i3.1813.

[6] M. Garonga and R. Tanduk, "COMPARISON OF NAIVE BAYES, DECISION TREE, AND RANDOM FOREST ALGORITHMS IN CLASSIFYING LEARNING STYLES OF UNIVERSITAS KRISTEN INDONESIA TORAJA STUDENTS," *J. Tek. Inform.*, vol. 4, no. 6, pp. 1507–1514, 2023, doi: https://doi.org/10.52436/1.jutif.2023.4.6.1020.

[7] M. Yağcı, "Educational data mining : prediction of students ' academic performance using machine learning algorithms," *Yağcı Smart Learn. Environ.*, vol. 9, no. 11, p. 2022, 2022, doi: 10.1186/s40561-022-00192-z.

[8] D. Khairy, N. Alharbi, M. A. Amasha, and M. F. Areed, "Prediction of student exam performance using data mining classification algorithms," *Educ. Inf. Technol.*, vol. 29, no. 16, pp. 21621–21645, 2024, doi: 10.1007/s10639-024-12619-w.

[9] Y. Bardi, R. Nivani, D. Bala, and M. Ermelinda, "Kurangnya Minat Baca di Kalangan Mahasiswa : Studi Kasus di Universitas Muhammadiyah Maumere," *Morfol. J. Ilmu Pendidikan, Bahasa, Sastra dan Budaya*, vol. 3, no. 2, pp. 106–119, 2025, doi: 10.61132/morfologi.v3i2.1484.

[10] S. Krashen, *The Power of Reading: Insights from the Research*. Portsmouth, NH: Heinemann, 2004.

[11] A. Halifa and R. Novita, "APPLICATION OF NAÏVE BAYES CLASSIFIER ALGORITHM IN DETERMINING THE LEVEL OF CUSTOMER SATISFACTION WITH RUMBAI POST

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

OFFICE SERVICES," *J. Tek. Inform.*, vol. 4, no. 6, pp. 1295–1304, 2023, doi: https://doi.org/10.52436/1.jutif.2023.4.6.1054.
[12] Subana, *Statistik Pendidikan*. Bandung: Pustaka Setia, 2005.
[13] M. Raschintasofi, N. Khumairo, E. Rasywir, and A. Feranika, "Analisis Tingkat Pemahaman Mahasiswa Universitas Dinamika Bangsa Dalam Pembelajaran Daring Menggunakan Algoritma K-Nearst Neighbor," *J. Manaj. Teknol. Dan Sist. Inf.*, vol. 2, no. 1, pp. 69–77, 2022, doi: 10.33998/jms.2022.2.1.29.
[14] A. A. Tarigan and M. A. I. Pakereng, "Penentuan Tingkat Pemahaman Mahasiswa dalam Matakuliah Kelas Daring dengan Algoritma C4.5 (Studi Kasus: Mahasiswa/i FTI Angkatan 2019)," *J. Sains Komput. Inform. (J-SAKTI*, vol. 7, no. 1, pp. 339–348, 2023.
[15] N. Trivetisia, R. D. Ramadhani, and M. Wibowo, "Perbandingan Algoritme Naïve Bayes dan C4.5 Pada Pengklasifikasian Tingkat Pemahaman Belajar Mahasiswa Dalam Pembelajaran Daring," *Progresif J. Ilm. Komput.*, vol. 19, no. 1, p. 323, 2023, doi: 10.35889/progresif.v19i1.1081.
[16] Jefri and Z. Fatah, "KLASIFIKASI DATA MINING UNTUK MEMPREDIKSI KELULUSAN MAHASISWA," *J. Ilm. MULTIDISIPLIN ILMU*, vol. 2, no. 1, pp. 29–37, 2025, doi: https://doi.org/10.69714/mhjq1v85.
[17] A. Kurniasih and K. Isyara, "Penggunaan Metode SMOTE pada Naïve Bayes Gaussian untuk Klasifikasi Mahasiswa Drop Out," *Semin. Nas. Mhs. Ilmu Komput. dan Apl.*, vol. 3, no. 2, pp. 616–623, 2023, doi: https://doi.org/10.47709/digitech.v3i2.3513.
[18] M. Jannah and H. Ajie, "Prediction of Grade Point Average (GPA) for Students at Informatics and Computer Engineering Education – Universitas Negeri Jakarta during Online Learning Using Naive Bayes Algorithm," *J. SISFOKOM(Sistem Inf. dan Komputer)*, vol. 13, no. 1, pp. 65–71, 2024, doi: https://doi.org/10.32736/sisfokom.v13i1.1958.
[19] B. R. S. Dwinanda, A. N. Ilhami, H. Choerunisya, and I. J. Thira, "Jurnal Sistem dan Teknologi Informasi Indonesia Prediksi Nilai Akhir Mahasiswa Dengan Metode Naïve Bayes Prediction of Final Student Grades Using Naive Bayes Method," *JUSTINDO (Jurnal Sist. Dan Teknol. Inf. Indones.*, vol. 9, no. 1, pp. 27–35, 2024, doi: https://doi.org/10.32528/justindo.v9i1.1082.
[20] A. Putri *et al.*, "Comparison of K-NN , Naive Bayes and SVM Algorithms for Final-Year Student Graduation Prediction Komparasi Algoritma K-NN , Naive Bayes dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, pp. 20–26, 2023, doi: https://doi.org/10.57152/malcom.v3i1.610.
[21] I. G. S. D. Putra and I. N. T. A. Putra, "IMPLEMENTASI METODE NAÏVE BAYES PADA ANALISIS SENTIMEN PENGGUNA APLIKASI MOBILE," *JITET (Jurnal Inform. dan Tek. Elektro Ter.*, vol. 13, no. 2, pp. 1202–1211, 2025, doi: http://dx.doi.org/10.23960/jitet.v13i2.6423.
[22] A. Qurotul, W. Elvira, N. Nazira, I. Ambarani, S. F. Intan, and D. Ramadhani, "Naïve Bayes Classifier (NBC) Algorithm Analysis for Prediction Medical Device Sales," *IJIRSE Indones. J. Inform. Res. Softw. Eng.*, vol. 3, no. 2, pp. 119–126, 2023, doi: https://doi.org/10.57152/ijirse.v3i2.941.
[23] F. A. Firmansyah, U. Enri, and I. Maulana, "Penerapan Algoritma Naive Bayes Dengan Chi-Square Untuk Klasifikasi Spam Email Berbasis Kata Dan Frekuensi," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 1, pp. 78–87, 2025, doi: https://doi.org/10.23960/jitet.v13i1.5506.
[24] S. Mulyani and R. Novita, "Implementation Of The Naive Bayes Classifier Algorithm For Classification Of Community Sentiment About Depression On Youtube," *J. Tek. Inform.*, vol. 3, no. 5, pp. 1355–1361, 2023, doi: https://doi.org/10.20884/1.jutif.2022.3.5.374.
[25] S. Nilam, S. Muslim, F. Nurdiansyah, and A. Y. Rahman, "PERBANDINGAN ALGORITMA NAIVE BAYESDAN KNNDALAM ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI CAPCUT," *JITET (Jurnal Inform. dan Tek. Elektro Ter.*, vol. 12, no. 3, pp. 3588–3596, 2024, doi: http://dx.doi.org/10.23960/jitet.v12i3S1.5156.
[26] A. Pebdika, R. Herdiana, and D. Solihudin, "KLASIFIKASI MENGGUNAKAN METODE NAIVE BAYES UNTUK MENENTUKAN CALON PENERIMA PIP," *JATI(Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 452–458, 2023, doi: https://doi.org/10.36040/jati.v7i1.6303.
[27] P. H. Susilo, M. G. Rohman, A. B. Laksono, and A. Bachri, "Sistem Pakar Penentuan Kualitas

Vol. 7, No. 3, June 2026, Page. 2259-2276
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5507

Jagung Menggunakan Metode Naive Bayes," *Insearch*, vol. 4, no. 2, pp. 47–54, 2024, doi: https://doi.org/10.15548/isrj.v4i02.9280.

[28] P. S. I. Pratiwi, M. G. Rohman, and M. Sholihin, "Sistem Pakar Penyakit Telinga Menggunakan Metode Naïve Bayes," *Gener. J.*, vol. 7, no. 2, pp. 70–82, 2023, doi: 10.29407/gj.v7i2.19991.

[29] R. Y. Hayuningtyas, "Penerapan Algoritma Naïve Bayes untuk Rekomendasi Pakaian Wanita," *J. Inform.*, vol. 6, no. 1, pp. 18–22, 2019, doi: https://doi.org/10.31294/ji.v6i1.4685.

[30] B. P. Salsabila, P. Belva, C. Trana, N. Ramadhani, and A. P. Sari, "PENERAPAN ALGORITMA NAIVE BAYES TERHADAP KUALITAS UDARA DI JAKARTA DAN REKOMENDASI AKTIVITAS MASYARAKAT," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 6, pp. 11732–11738, 2024, doi: https://doi.org/10.36040/jati.v8i6.11592.

[31] M. R. Dwimanhendra *et al.*, "KLASIFIKASI JENIS KAYU BERDASARKAN CITRA SERAT KAYU MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 10, no. 1, pp. 72–80, 2024, doi: https://doi.org/10.29100/jipi.v10i1.5726.

[32] J. H. Saviola and N. D. Hendrawan, "IMPLEMENTASI KLASIFIKASI KUALITAS SUSU MENGGUNAKAN ALGORITMA DECISION TREE , K-NEAREST NEIGHBORS DAN NAIVE BAYES," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 5, pp. 8953–8960, 2025, doi: https://doi.org/10.36040/jati.v9i5.15260.

[33] R. Adrian and M. Ikhsan, "Detection of Hoax News Using TF-IDF Vectorizer and Multinomial Naïve Bayes and Passive Aggressive," vol. 1, no. 2, pp. 54–61, 2024, doi: 10.62205/mjgcs.v1i2.24.

[34] M. H. Ariansyah, E. N. Fitri, and S. Winarno, "IMPROVING PERFORMANCE OF STUDENTS' GRADE CLASSIFICATION MODEL USES NAÏVE BAYES GAUSSIAN TUNING MODEL AND FEATURE SELECTION," *J. Tek. Inform.*, vol. 4, no. 3, pp. 493–501, 2023, doi: https://doi.org/10.52436/1.jutif.2023.4.3.737.

[35] Ratih Rusdiana Ekawati, V. Atina, and J. Maulidar, "Prediksi Ketuntasan Belajar Siswa Menggunakan Naive Bayes dengan Integrasi Data Akademik, Absensi, dan Partisipasi Ekstrakurikuler," *J. Pendidik. dan Teknol. Indones.*, vol. 5, no. 4, pp. 1161–1173, 2025, doi: https://doi.org/10.52436/1.jpti.752.

[36] H. Kusmanto, N. S. Munawaroh, and H. N. Sofhya, "APPLICATION OF NAÏVE BAYES ALGORITHM IN DETERMINING STUDENT CONCENTRATION IN MATHEMATICS LEARNING PROCESS," *Educ. INSIGHTS*, vol. 2, no. 2, pp. 180–194, 2024, doi: https://doi.org/10.58557/eduinsights.v2i2.100.

[37] V. Jackins, S. Vimal, M. Kaliappan, and M. Y. Lee, "AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes," *J Supercomput*, vol. 77, no. 5, pp. 5198–5219, May 2021, doi: 10.1007/s11227-020-03481-x.

[38] W. Chen *et al.*, "Modeling flood susceptibility using data-driven approaches of naïve Bayes tree, alternating decision tree, and random forest methods," *Science of The Total Environment*, vol. 701, p. 134979, Jan. 2020, doi: 10.1016/j.scitotenv.2019.134979.

[39] S. Ruan, H. Li, C. Li, and K. Song, "Class-Specific Deep Feature Weighting for Naïve Bayes Text Classifiers," *IEEE Access*, vol. 8, pp. 20151–20159, 2020, doi: 10.1109/ACCESS.2020.2968984.

[40] A. S. Talita, O. S. Nataza, and Z. Rustam, "Naïve Bayes Classifier and Particle Swarm Optimization Feature Selection Method for Classifying Intrusion Detection System Dataset," *J Phys Conf Ser*, vol. 1752, no. 1, p. 012021, Feb. 2021, doi: 10.1088/1742-6596/1752/1/012021.

[41] V. Sai Krishna Reddy, P. Meghana, N. V Subba Reddy, and B. Ashwath Rao, "Prediction on Cardiovascular disease using Decision tree and Naïve Bayes classifiers," *J Phys Conf Ser*, vol. 2161, no. 1, p. 012015, Jan. 2022, doi: 10.1088/1742-6596/2161/1/012015.

[42] C. Bemando, E. Miranda, and M. Aryuni, "Machine-Learning-Based Prediction Models of Coronary Heart Disease Using Naïve Bayes and Random Forest Algorithms," in *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*, IEEE, Aug. 2021, pp. 232–237. doi: 10.1109/ICSECS52883.2021.00049.