Jurnal E-Komtek
Vol. 10, No. 1. (2026) pp. 125-142
https://jurnal.politeknik-kebumen.ac.id/index.php/E-KOMTEK
p-ISSN : 2580-3719 e-ISSN : 2622-3066

Implementation of the K-Nearest Neighbors (KNN) Algorithm in the Process of Student Graduation Prediction (Case Study of The Bachelor of Informatics Engineering Program, PLN Institute of Technology, Jakarta)

Andi Abd. Jalil. L[1], Herman Bedi Agtriadi [2], Meilia Nur Indah Susanti[3], Rakhmadi Ifansyah Putra[4]

[1-4] Department of Magister Computer Science, Institut Teknologi PLN, Indonesia, 11750

andijalil00@gmail.com

https://doi.org/10.37339/e-komtek.v10i1.3299

Published by Politeknik Piksi Ganesha Indonesia

Artikel Info
Submitted:
07-05-2026
Revised:
17-05-2026
Accepted:
30-06-2026
Online first :
30-06-2026

Abstract

Student graduation is one of the key indicators in a university's Internal Quality Assurance System (SPMI). Based on data from the Bachelor of Informatics Engineering program, out of 305 students from the 2016 cohort, 227 graduated on time and 78 graduated late. This study aims to predict student graduation using the K-Nearest Neighbors (KNN) algorithm. The research stages include data collection and division for training and testing, parameter determination with K=3, and distance calculation between data points. The results show that the KNN model with parameter K=3 achieved an accuracy rate of 90% in predicting student graduation. This demonstrates that the KNN method is effective in predicting student graduation outcomes.

Keywords: *Student Graduation, Prediction, K-Nearest Neighbors, Accuracy*

Abstrak

Kelulusan mahasiswa merupakan salah satu indikator penting dalam Standar Penjaminan Mutu Internal (SPMI) perguruan tinggi. Berdasarkan data Program Studi S1 Teknik Informatika angkatan 2016, dari 305 mahasiswa terdapat 227 mahasiswa lulus tepat waktu, 78 tidak tepat waktu. Penelitian ini bertujuan untuk memprediksi kelulusan mahasiswa menggunakan algoritma K-Nearest Neighbors (KNN). Tahapan penelitian meliputi pengumpulan dan pembagian data untuk pelatihan serta pengujian model, penentuan parameter K=3, dan perhitungan jarak antar data. Hasil penelitian menunjukkan bahwa model KNN dengan parameter K=3 mampu memprediksi kelulusan mahasiswa dengan tingkat akurasi sebesar 90%. Hal ini membuktikan bahwa metode KNN efektif digunakan dalam memprediksi kelulusan mahasiswa.

***Kata-kata kunci**: kelulusan mahasiswa, prediksi, K-Nearest Neighbors, akurasi*

1. Introduction

Student graduation is one of the areas included in the *standar penjaminan mutu internal* (SPMI) of higher education institutions. One of the university's standards for producing students who graduate on time, particularly in the Informatics Engineering Undergraduate Program, is that students can complete a maximum of eight semesters with a total study load of at least 149 credits **[1], [2]**. Based on interview data obtained from the Informatics Engineering Undergraduate Study Program with a total of 305 students from the 2016 Class with 227 students graduating on time, 78 students graduating late, meaning that there are still many students who have not been able to graduate on time due to various obstacles, for example, not meeting the minimum number of credits that have been set and low cumulative grade point average (GPA). Indications of delays can usually be found in semester 6 or 7. So a system is needed that is able to capture indications of delays in student graduation early on **[2]**.

To prevent this from happening, prediction is necessary. One technology that can predict student graduation rates is the K-Nearest Neighbors (KNN) algorithm **[3]** The K-Nearest Neighbors (KNN) algorithm is an object classification method based on learning data closest to the object. KNN is a supervised learning algorithm in which the results of a new query instance pattern are classified based on the majority of categories in the KNN **[4]**. The most frequently occurring class is the class that emerges from the classification results. The goal of this algorithm is to classify new objects based on their attributes and the training data examples available **[5]**

In determining classification, the K-Nearest Neighbors (KNN) algorithm is used. The reason for choosing the KNN algorithm is based on the data used, namely secondary data. The goal of this algorithm is to classify new objects based on attributes and training **[6]**. This classification uses data from students' Semester Grade Point Average (IPS) and their on-time or late graduation status as training data. The KNN algorithm offers several advantages, including very fast training, simplicity and ease of learning, robustness to noisy training data, and effectiveness for both small and large datasets **[7]**.

The purpose of this study is to implement the K-Nearest Neighbors (KNN) Algorithm in predicting the graduation of Informatics Engineering students and to investigate the level of accuracy and feasibility of the K-Nearest Neighbors (KNN) Algorithm in predicting student graduation.

2. Method

This research was conducted to address the problem of delays in student permits for the Informatics Engineering Undergraduate Program at the PLN Institute of Technology, which until now cannot be predicted systematically based on academic data. Therefore, this research applies a quantitative approach by utilizing the K-Nearest Neighbors (KNN) learning algorithm to predict student admission status based on Semester Achievement Index (IPS) scores. The research method is systematically structured starting from the place and time of the research, research design, data collection and preparation, modeling, to evaluation of the performance of the prediction model.

2.1 Place and Time Research

This research was conducted at the PLN Institute of Technology, Informatics Engineering Undergraduate Program, Faculty of Energy Telematics. The research period was from March 29, 2022, to July 20, 2022.

2.2 Research Design

This study employed a quantitative research design with an experimental approach. This design aimed to test the performance of the K-Nearest Neighbors (KNN) algorithm in predicting student graduation based on available academic data. The experimental approach involved building a classification model using training data and testing its performance using separate test data.

This research falls into the supervised learning category because the data used already had class labels in the form of student graduation status, namely graduating on time and not on time. The research stages included data collection, data preparation, namely data division using the holdout validation method, modeling using the KNN algorithm, and model evaluation using a confusion matrix.

2.3 Data Collection

The data used in this study is secondary data obtained from the Informatics Engineering Undergraduate Program at the PLN Institute of Technology. Data collection was conducted through observation of student academic data and interviews with the study program to obtain student graduation data.

The dataset used consisted of 305 student data from the 2016 intake. The attributes used in this study included Semester Achievement Index (IPS) from semester 1 to semester 4, as well as

one label attribute in the form of student graduation status which was classified into graduating on time and not on time.

2.4 Data Preparation

The data preparation stage is carried out to ensure the data used is ready for processing in the modeling. The collected data is organized into a dataset and focused on the attributes of social studies from semesters 1 to 4 as input variables, and graduation status as output variables.

Next, the dataset is divided using the holdout validation method, with a proportion of 80% of the data as training data and 20% of the data as testing data **[8]**. The training data is used to train the KNN model, while the testing data is used to test the model's ability to predict data that has never been studied before. This data division aims to objectively measure the model's generalization ability **[9]**.

2.5 Modeling Using the K-Nearest Neighbors (KNN) Algorithm

Data modeling in this study was performed using the K-Nearest Neighbors (KNN) algorithm. KNN is a supervised learning algorithm that classifies new data based on its proximity to existing training data. The proximity between data is calculated using the Euclidean Distance formula **[10]**.

The modeling process begins by determining the value of the parameter K, which is the number of nearest neighbors used in the classification process. Next, the distance between each test data point and the entire training data point is calculated, then sorted by the closest distance. The K nearest neighbors are selected, and the class that appears most frequently among these neighbors is determined as the predicted student graduation status.

In this study, several K values were tested to obtain the best prediction results. Based on the test results, the value of K=3 produced the highest level of accuracy compared to other K values. Therefore, the parameter K=3 was used in the graduation prediction modeling process

2.6 Model Evaluation

The model evaluation stage is conducted to determine the performance and reliability of the prediction model that has been developed. The evaluation method used is a confusion matrix, which produces True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values.

Based on these values, several evaluation metrics were calculated, namely accuracy, precision, recall, and F1-score. The evaluation results showed that the KNN model with

parameter K=3 was able to predict student graduation with an accuracy rate of 90%. These results indicate that the K-Nearest Neighbors (KNN) algorithm is effective and performs well in predicting student graduation based on data.

3. Result and Discussion

3.1 Result

This section will detail the results obtained, including data collection, data preparation, data training (modeling), and model testing using a confusion matrix. These are explained as follows:

3.1.1 Data Collection Results

The data consists of 305 datasets with attributes such as no, noreg, Semester Grade Index (IPS) 1 to 4 and Label (Graduation Status) obtained directly from the Informatics Engineering Undergraduate Study Program on May 31, 2022. The data obtained can be seen in the table below.

Table 1. Data Collection Results

NO	NOREG	IPS-1	IPS-2	IPS-3	IPS-4	STATUS
1	20163328	3.68	3.5	3.94	3.61	ON TIME
2	20165872	3.64	3.38	3.02	3.04	ON TIME
3	20162698	3.92	3.64	3.05	3.7	ON TIME
4	20163030	3.37	3.03	3.35	3.48	ON TIME
5	20160277	3.04	3.29	3.34	3.14	ON TIME
6	20163261	2.74	2.83	0	0	LATE
7	20162028	2.83	2.7	2.97	2.91	LATE
8	20163051	3.39	2.89	2.91	3.17	ON TIME
9	20162012	3.82	3.93	3.9	3.78	ON TIME
10	20163461	3.33	3.29	3.66	3.39	ON TIME
…						
305	20166880	2.89	2.53	2.93	3.07	ON TIME

The table above is the data on 2020 graduates of the Informatics Engineering Undergraduate Study Program with a total of 305 data, with 227 data classified as graduating on time and 78 data as graduating not on time.

3.1.2 Data Preparation Results

The results obtained from data preparation are focused on the Semester Predicate Index (IPS) 1 to 4 and Label (Graduation Status) attributes, namely the results of dividing the training data and testing data using holdout validation, namely from 305 data divided by 80% of the

training data with 244 data and 20% of the testing data with 61 data. The results of dividing the training data and testing data can be seen in the image below.

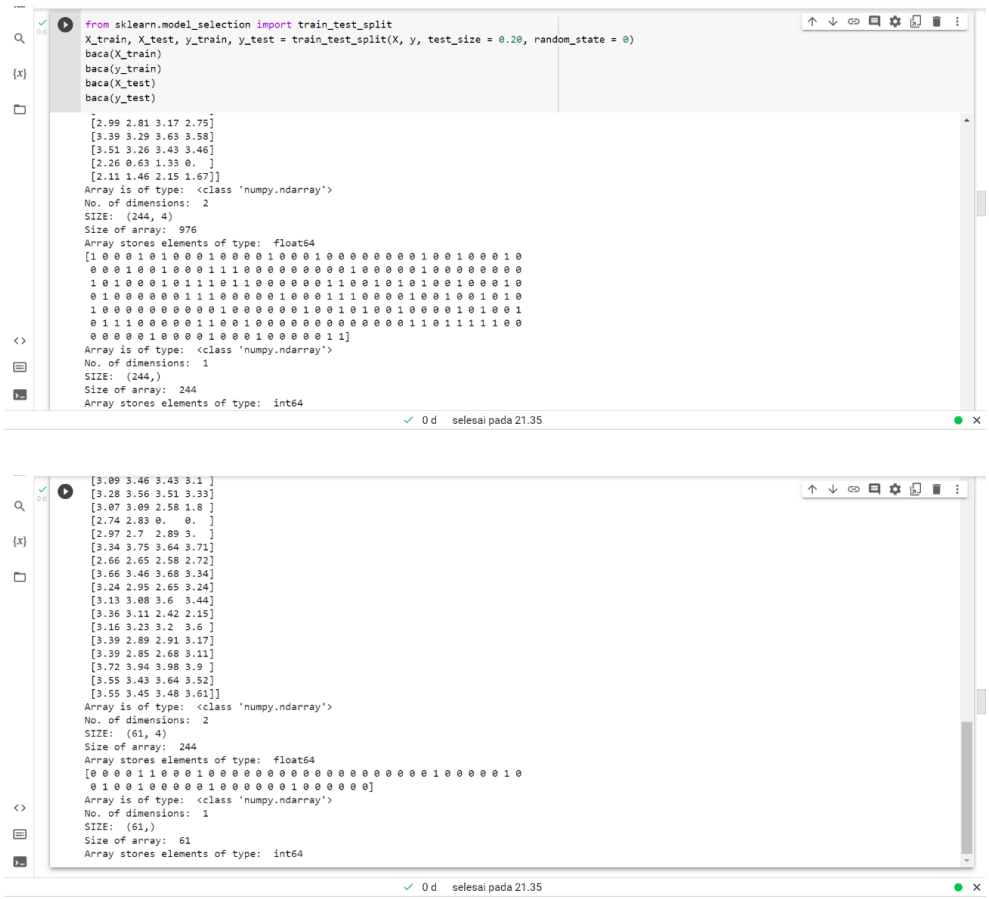


Figure 1. Data Preparation Results

The image above shows the results of randomly dividing the dataset using the Python programming language on the Google Collab platform, which is divided into two: training data (80% of the dataset with 244 data points) and testing data (20% of the dataset with 61 data points). The results of this division of training and testing data will be used for data modeling, where the training data is used to train the testing data, and the testing data will be used to test and evaluate the model.

3.1.3 Data Training Results (Modeling)

At this stage, the data that has been divided into training data and testing data will be processed with the K-Nearest Neighbors algorithm using the K-3 parameter. The use of the K-3 parameter is used because the results of the experiment using the K-1 to K-9 parameters with the accuracy results using K-1 are 81%, K-3 is 90%, K-5 = 88%, K-7 = 88%, K-9 = 88%, therefore the

author sets the K-3 parameter in the process of predicting student graduation because the highest accuracy is in the K-3 parameter.

Table 2. Data Training

DATA	IPS-1	IPS-2	IPS-3	IPS-4	STATUS
1	2.74	2.64	1.85	0	LATE
2	3.25	3.25	3.63	3.57	ON TIME
3	3.72	3.59	3.66	3.64	ON TIME
4	3.66	3.38	3.26	3.31	ON TIME
5	3.05	3.15	2.57	2.44	LATE
6	3.57	3.31	3.23	3.46	ON TIME
7	2.16	2.76	2.33	2.1	LATE
8	3.76	3.85	3.56	3.45	ON TIME
9	2.84	3.21	3.18	2.81	ON TIME
10	3.43	3.18	3.38	3.69	ON TIME
…					
244	2.11	1.46	2.15	1.67	LATE

The table above is training data from a dataset that has been divided into training data and testing data, where this training data will be used to train the testing data.

Tabla 3. Data testing

DATA	IPS-1	IPS-2	IPS-3	IPS-4	STATUS
1	3.54	3.61	3.63	3.81	?
2	3.11	2.44	2.68	2.72	?
3	2.8	2.69	3.39	3.03	?
4	3.53	3.41	3.03	3.35	?
5	2.42	2.56	2.09	2.4	?
6	3.09	3.05	3.16	3.25	?
7	3.21	3.05	3.45	3.48	?
8	2.97	3.31	2.82	2.24	?
9	3.66	3.41	3.68	3.73	?
10	3.11	2.63	0.43	1.34	?
…					
61	3.55	3.45	3.48	3.61	?

.

The table above shows 61 testing data that were tested against the training data using Python programming language-based modeling on the Google Colab platform, by applying the K-Nearest Neighbors (KNN) algorithm and parameter K = 3.

1. Data Training Results 1

The experimental results of training data 1 with IPS values of 3.54, 3.61, 3.63, 3.81 using the KNN algorithm with K-3 parameters, the results obtained are as follows:

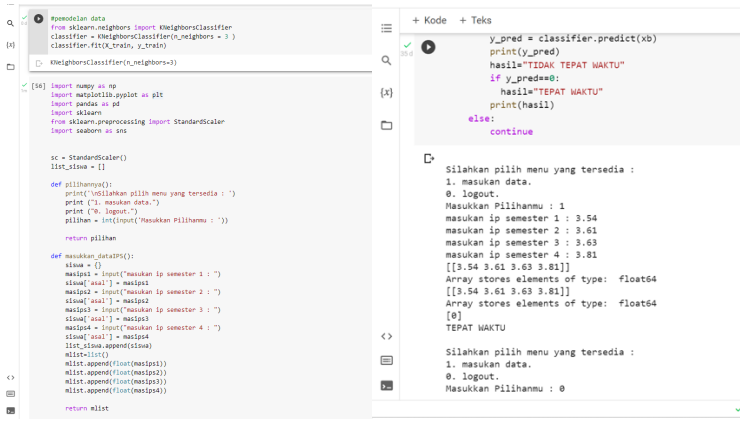


Figure 2. Data Training Results 1

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and K-3 parameters. Based on the test results on data with IPS values of 3.54, 3.61, 3.63, and 3.81, the model predicts that the data falls into the ON-TIME category.

2. Data Training Results 2

The experimental results of training data 2 with IPS values of 3.11, 2.44, 2.68, 2.72 using the KNN algorithm with K-3 parameters are as follows:

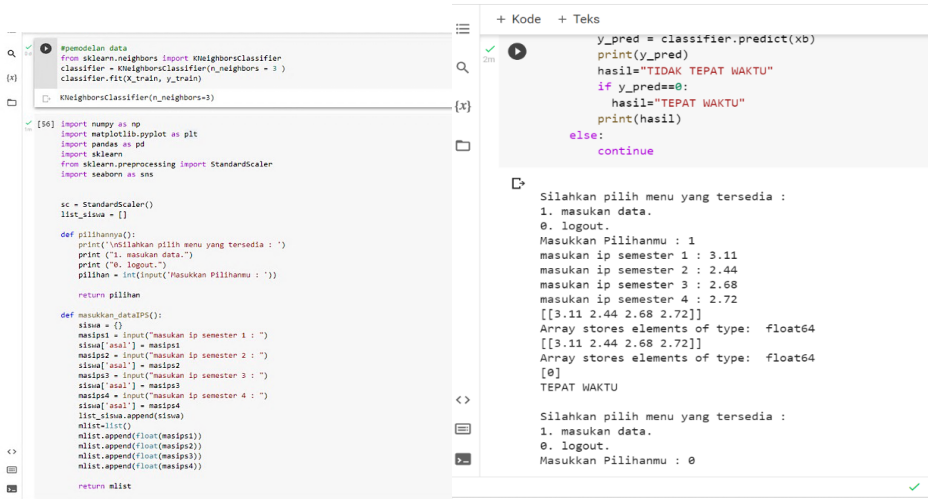


Figure 3. Data Training Results 2

.

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and K-3 parameters. Based on the test results on data with IPS values of 3.11, 2.44, 2.68, and 2.72, the model predicts that the data falls into the ON-TIME category.

3. Data Training Result 3

The experimental results of training data 3 with IPS values of 2.8, 2.69, 3.39, 3.03 using the KNN algorithm with K-3 parameters are as **Figure 4.**

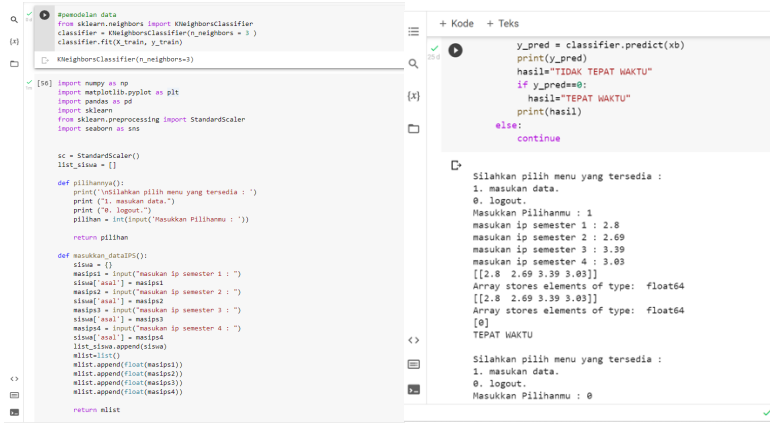


Figure 4. Data Training Results 3

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and the K-3 parameter. Based on the test results on data with IPS values of 2.8, 2.69, 3.39, and 3.03, the model predicts that the data falls into the ON-TIME category.

4. Data Traning Results 4

The experimental results of training data 4 with IPS values of 3.53, 3.41, 3.03, 3.35 using the KNN algorithm with K-3 parameters are as follows:

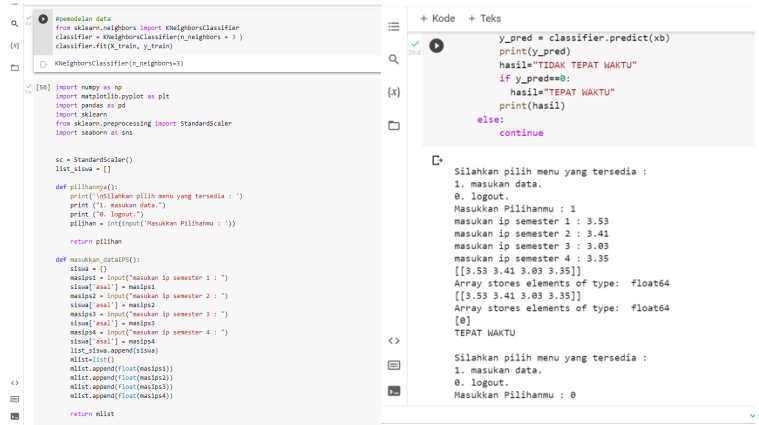


Figure 5. Data Training Results 4

.

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and the K-3 parameter. Based on the test results on data with IPS values of 3.53, 3.41, 3.03, and 3.35, the model predicts that the data falls into the ON-TIME category.

5. Data Training Results 5

The experimental results of training data 5 with IPS values of 2.42, 2.56, 2.09, 2.4 using the KNN algorithm with K-3 parameters are as **Figure 6.**

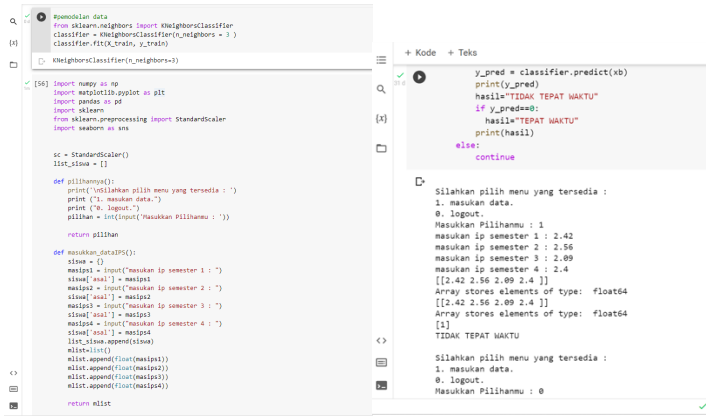


Figure 6. Data Training Results 5

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and K-3 parameters. Based on the test results on data with IPS values of 2.42, 2.56, 2.09, and 2.4, the model predicts that the data falls into the NOT ON-TIME category.

6. Data Training Results 6

The experimental results of training data 6 with IPS values of 3.09, 3.05, 3.16, 3.25 using the KNN algorithm with K-3 parameters are as follows:

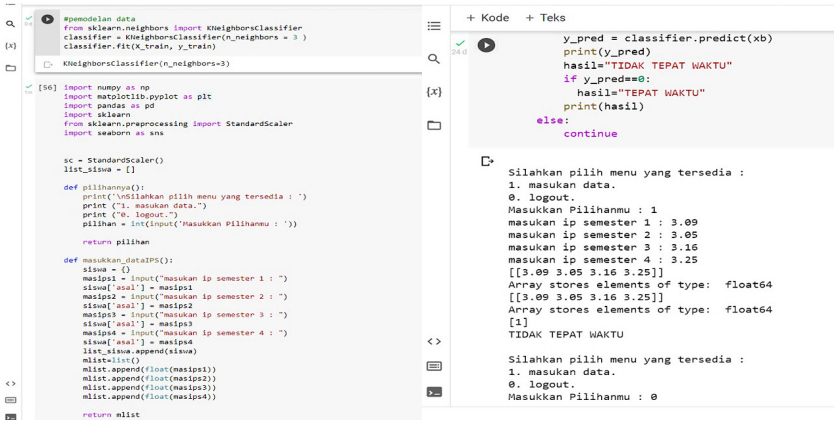


Figure 7. Result Training Data 6

.

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and the K-3 parameter. Based on the test results on data with IPS values of 3.09, 3.05, 3.16, and 3.25, the model predicts that the data falls into the NOT ON-TIME category.

7. Data Training Results 7

The experimental results of training data 7 with IPS values of 3.21, 3.05, 3.45, 3.48 using the KNN algorithm with K-3 parameters are as follows:

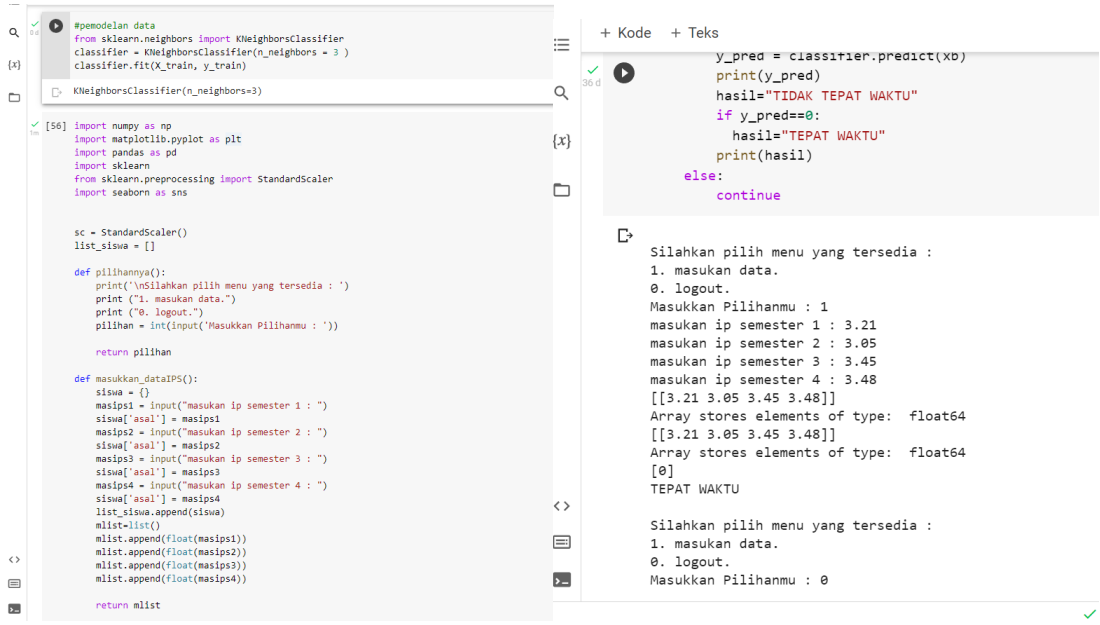


Figure 8. Data Training Results 7

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and the K-3 parameter. Based on the test results on data with IPS values of 3.21, 3.05, 3.45, and 3.48, the model predicts that the data falls into the ON-TIME category.

8. Data Training Results 8

The experimental results of training data 8 with IPS values of 2.97, 3.31, 2.82, 2.24 using the KNN algorithm with K-3 parameters are as follows:

.

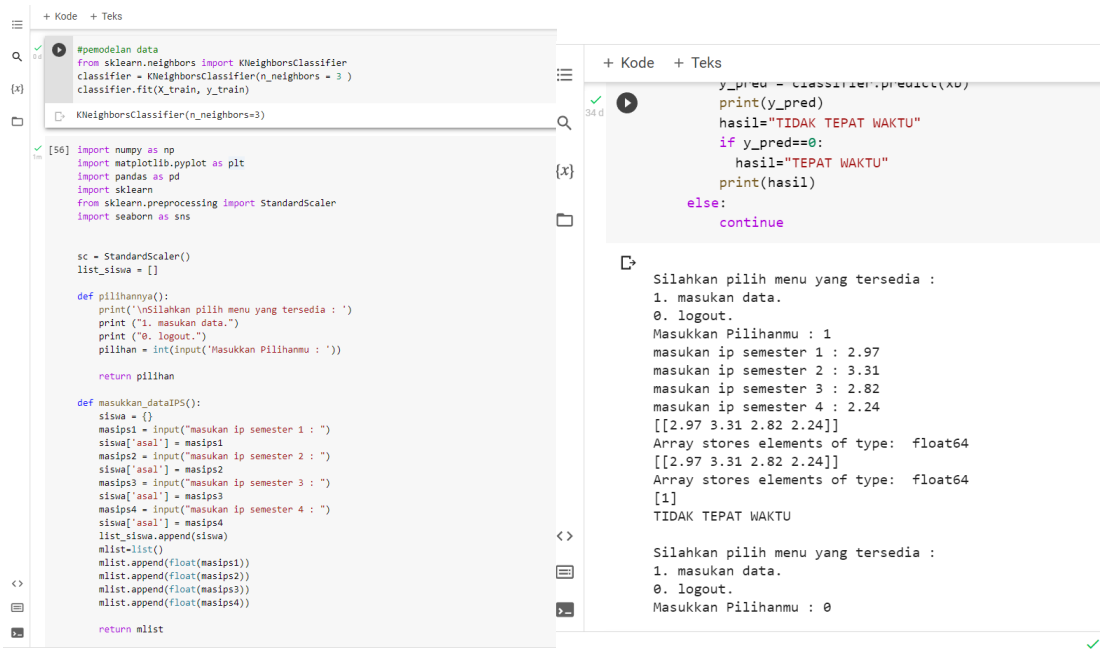


Figure 9. Data Training Results 8

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and K-3 parameters. Based on the test results on data with IPS values of 2.97, 3.31, 2.82, and 2.24, the model predicts that the data falls into the NOT ON-TIME category.

9. Data Training Results 9

Experiments from training data 9 with IPS values of 3.66, 3.41, 3.68, 3.73 using the KNN algorithm with K-3 parameters, the results obtained are as follows:

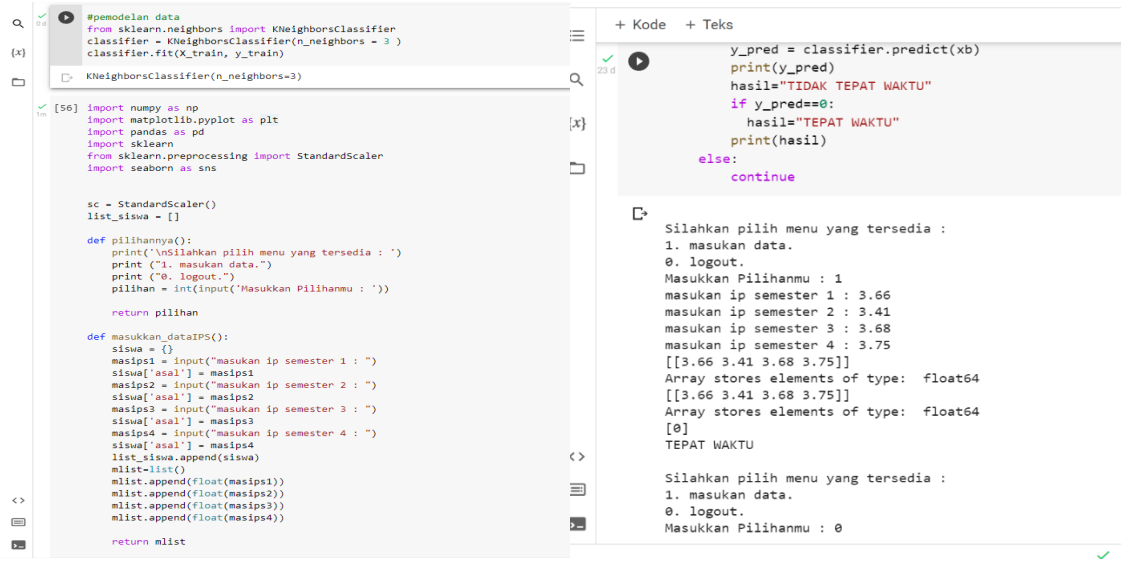


Figure 10. Data Training Results 9

.

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and K-3 parameters. Based on the test results on data with IPS values of 3.66, 3.41, 3.68, and 3.73, the model predicts that the data falls into the ON-TIME category.

10. Data Training Results 10

The experimental results of training data 10 with IPS values of 3.11, 2.63, 0.43, 1.34 using the KNN algorithm with K-3 parameters are as follows:

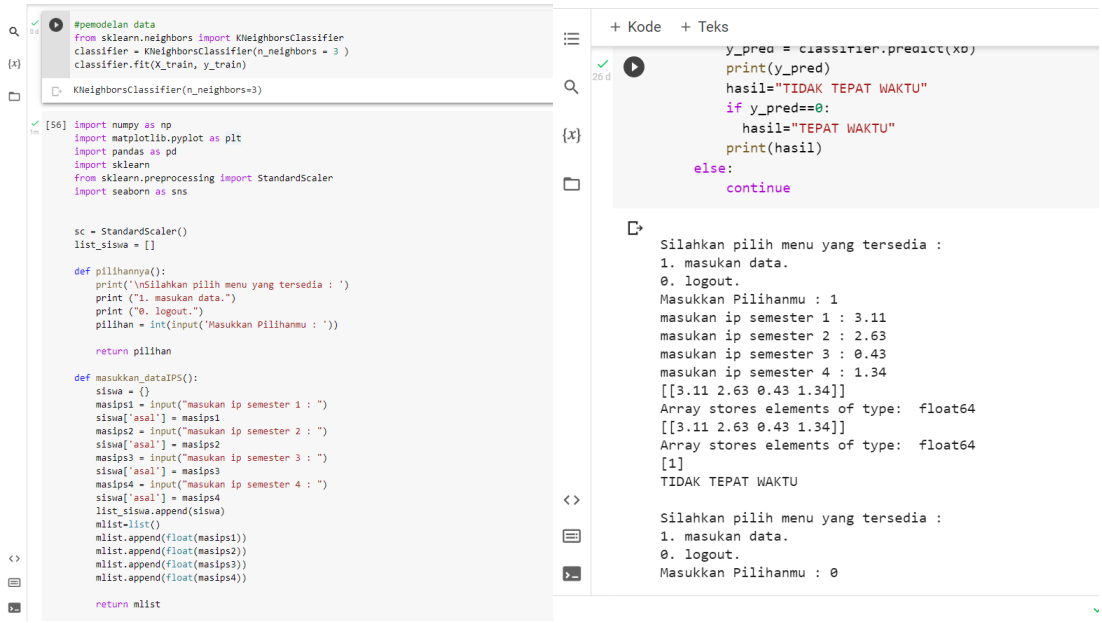


Figure 11. Data Training Results 10

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and the K-3 parameter. Based on the test results on data with IPS values of 3.11, 2.63, 0.43, and 1.34, the model predicts that the data falls into the NOT ON TIME category.

11. Data Training Results 61

The experimental results of training data 61 with IPS values of 3.55, 3.45, 3.48, 3.61 using the KNN algorithm with K-3 parameters are as follows:

.

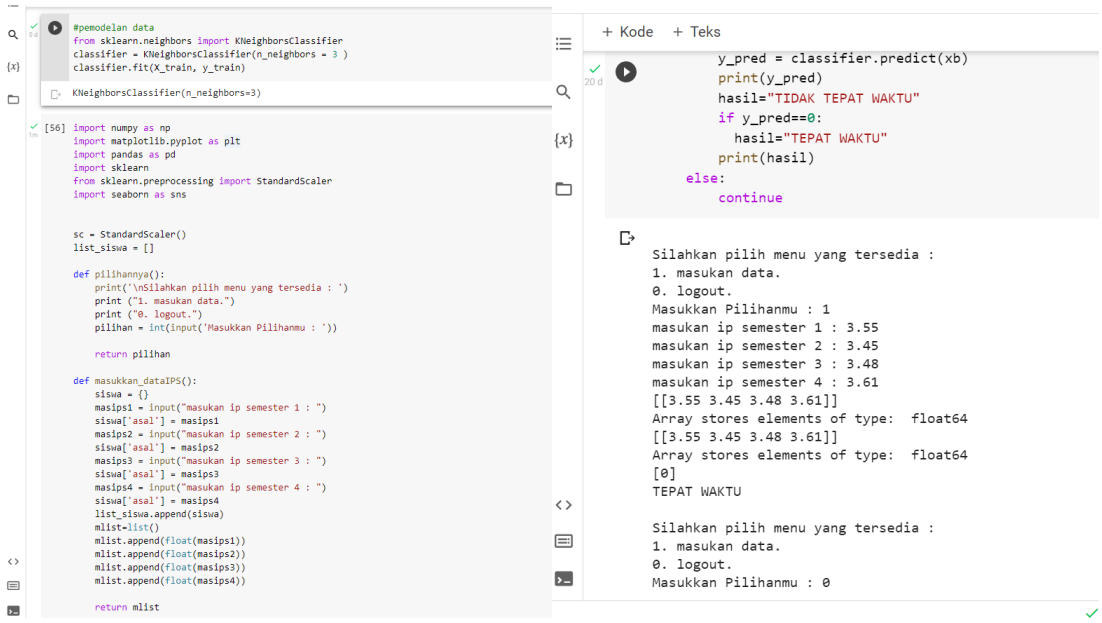


Figure 12. Data Training Results 61

The image above shows the output of a program run using the Python programming language on the Google Colab platform with the K-Nearest Neighbors (KNN) algorithm and K-3 parameters. Based on the test results on data with IPS values of 3.55, 3.45, 3.48, and 3.61, the model predicts that the data falls into the ON-TIME category.

Table 4. Data Training Results

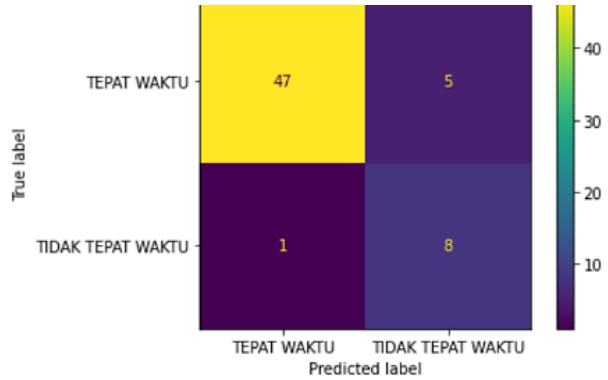
DATA	IPS-1	IPS-2	IPS-3	IPS-4	STATUS
1	3.54	3.61	3.63	3.81	ON TIME
2	3.11	2.44	2.68	2.72	ON TIME
3	2.8	2.69	3.39	3.03	ON TIME
4	3.53	3.41	3.03	3.35	ON TIME
5	2.42	2.56	2.09	2.4	LATE
6	3.09	3.05	3.16	3.25	LATE
7	3.21	3.05	3.45	3.48	ON TIME
8	2.97	3.31	2.82	2.24	LATE
9	3.66	3.41	3.68	3.73	ON TIME
10	3.11	2.63	0.43	1.34	LATE
…					
61	3.55	3.45	3.48	3.61	ON TIME

.

Based on the results of data training, namely by testing the testing data to the training data using the KNN Algorithm, the number of testing data is 61 data which is tested against the training data of 244 data, the classification results obtained are on time for 48 data and not on time for 13 data.

3.1.4 Evaluation (Confusion Matrix)

In the experiment with the results of training data using the K-3 parameters with 61 testing data from the results of dividing the training and test data where the testing data was used for model testing using confusion matrix testing with the following test results:




```
confusion matrix:
[[47  5]
 [ 1  8]]
classification Report:
                  precision    recall  f1-score   support

      TEPAT WAKTU       0.98      0.90      0.94        52
TIDAK TEPAT WAKTU       0.62      0.89      0.73         9

         accuracy                           0.90        61
        macro avg       0.80      0.90      0.83        61
     weighted avg       0.93      0.90      0.91        61

Accuracy: 0.9016393442622951
```

Gambar 13. Confusion Matrix Results

In the Confusion Matrix testing program, as shown in the image above, we obtained True Positive (TP) results with 47 data points, True Negative (TN) results with 8 data points, False Positive (FP) results with 5 data points, and False Negative (FN) results with 1 data point. The final accuracy value was 90%. The resulting accuracy, precision, recall, and F1-score values are as follows:

$$Accuracy = \frac{47 + 8}{47 + 8 + 5 + 1} x\ 100\% = 90\%$$

This means that in the actual data, there were 52 positive (on time) data, 9 negative (not on time), but when predicted, there were 48 positive (on time) data and 13 negative (not on time), meaning that the results of the on time prediction were wrong 4 times and the results of the not on time prediction were wrong 4 times too, where when calculated using the Confusion Matrix accuracy formula, the accuracy was 90%.

$$Precision = \frac{47}{47+5} = 0.90 \text{ x } 100\% = 90\%$$

This means that from the prediction of students graduating on time, all classes that have been predicted correctly are 48 data, but in the actual data, the data that is on time is 52. In this case, the precision of the student graduation prediction is 90%.

$$Recall = \frac{47}{47+1} = 0.97 \text{ x } 100\% = 97\%$$

The ratio of correct predictions passed on time to the total correct data on time was 97%

$$F1 - score = 2\ x\ \frac{0.90\ x\ 0.97}{0.90+0.97} = 0.93 \text{ x } 100\% = 93\%$$

Meanwhile, the f1-score is a weighted comparison of the average precision and recall. In this case, the f1-score is 93%.

Based on the testing data, which was divided into 20% of the total dataset, 61 out of 305, the confusion matrix classification test achieved a 90.00% accuracy in identifying data.

3.2 Discussion

This research was conducted based on observations and interviews conducted by the author with the head of the Informatics Engineering undergraduate study program. The interview results indicated that many graduates are still unable to graduate on time. This is because the students themselves and the study program are also unable to predict whether all prospective graduates will graduate on time or whether many will graduate late each year. Due to these obstacles, a system is needed that can predict student graduation, with the hope that in the future it can help study programs anticipate and prevent potential graduates from the Informatics Engineering undergraduate study program.

The stages are carried out starting from understanding the problem as explained above, data collection where the data is obtained from the results of interviews and observations on May 31, 2022. The data that has been obtained is then processed through a data preparation process, namely dividing the data using the holdout validation technique, namely dividing the training data and testing data, then continued with data training, the activities carried out are testing the testing data to the training data using the K-Nearest Neighbors algorithm with K-3 parameters, and finally the model testing stage is testing the model that has been created previously using the Confusion Matrix with an accuracy result of 90%.

4. Conclusion

Based on the research results of Student Graduation Prediction using Machine learning with the K-Nearest Neighbors Algorithm that has been carried out starting from the initial stage to the evaluation, it can be concluded that the process of making student graduation predictions using the K-Nearest Neighbors algorithm consists of several stages. Starting from data collection and understanding, data preparation with data division using holdout validation, data training by determining the K-3 parameters, the Eucludien Distance process to see the closest distance from the dataset obtained to the model testing stage using the confusion matrix. And from the training process, results were obtained using the K-3 parameters with an accuracy of 90%.

5. Acknowledgement

I would like to express my deepest gratitude to Dr. Herman Bedi Agtriadi, S.Si., M.Kom., as my supervisor in the Master of Computer Science Program at the PLN Institute of Technology. Thanks to his guidance, direction, and support, I was able to successfully complete this journal. I realize that this work still has various shortcomings, so I greatly appreciate your input and suggestions for future improvements.

References

[1] N. Ketut Sriwinarti and N. luh Putu Juniarti, "Analisis Metode K-Nearest Neighbors (K-NN) Dan Naive Bayes Dalam Memprediksi Kelulusan Mahasiswa (Analysis of K-Nearest Neighbors (K-NN) and Naive Bayes Methods in Predicting Student Graduation)," vol. 3, no. 2, pp. 106–112, 2021.

[2] A. Wahyudi and F. Wahyu Wibowo, "PREDIKSI KELULUSAN MAHASISWA TEPAT WAKTU MENGGUNAKAN METODE DECISION TREE DAN NAÏVE BAYES

PREDICTING ON-TIME GRADUATION OF STUDENTS USING DECISION TREE AND NAÏVE BAYES METHODS," 2023.

[3] A. Sabathos Mananta and G. Arther Sandag, "Prediksi Kelulusan Mahasiswa Dalam Memilih Program Magister Menggunakan Algoritma K-NN," vol. 10, no. 2, 2021.

[4] I. Riadi, R. Umar, and R. Anggara, "Prediksi Kelulusan Tepat Waktu Berdasarkan Riwayat Akademik Menggunakan Metode K-Nearest Neighbor," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 2, pp. 249–256, Apr. 2024, doi: 10.25126/jtiik.20241127330.

[5] A. Fakih, M. A. Hamzami, M. R. Hadianto, and N. I. S. Alifah, "Perbandingan Akurasi Algoritma C4.5 dan K-NN Untuk Prediksi Kelulusan Mahasiswa Penerima Beasiswa," *Jurnal Komputer Antartika*, vol. 3, no. 1, pp. 18–25, Jan. 2025, doi: 10.70052/jka.v3i1.623.

[6] I. Ramdhani, "PERBANDINGAN METODE DATA MINING MODEL KLASIFIKASI NAIVE BAYES, DECISSION TREE DAN K-NEAREST NEIGHBOUR DALAM MEMPREDIKSI KETEPATAN KELULUSAN MAHASISWA PRODI TEKNIK INFORMATIKA DI UNIVERSITAS PAMULANG," |*Jurnal Ilmu Komputer JIK*, vol. VI, no. 01, 2023.

[7] E. Etriyanti Sistem Informasi, S. Bina Nusantara Jaya Lubuklinggau Jl Yos Sudarso No, K. Lubuklinggau, and S. Selatan, "Perbandingan Tingkat Akurasi Metode KNN Dan Decision Tree Dalam Memprediksi Lama Studi Mahasiswa," *Jurnal Ilmiah Binary STMIK Bina Nusantara Jaya*, vol. 03, 2021.

[8] A. Putri, C. Syaficha Hardiana, E. Novfuja, F. Try Puspa Siregar, Y. Fatma, and R. Wahyuni, "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," *Institut Riset dan Publikasi Indonesia (IRPI) MALCOM: Indonesian Journal of Machine Learning and Computer Science Journal Homepage*, vol. 3, no. 1, pp. 20–26, 2023.

[9] E. Novianto, A. Hermawan, and D. Avianto, "KLASIFIKASI ALGORITMA K-NEAREST NEIGHBOR, NAIVE BAYES, DECISION TREE UNTUK PREDIKSI STATUS KELULUSAN MAHASISWA S1," *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, vol. 8, no. 2, pp. 146–154, Jul. 2023, doi: 10.36341/rabit.v8i2.3434.

[10] R. H. Hidayatullah, R. Sanjaya, M. J. Alazami, M. Affifudin, R. Samsinar, and T. Elektro, "Perbandingan Algoritma K-Nearest Neighbor dan Random Forest dalam Prediksi Kelulusan Mahasiswa Mahasiswa Universitas Muhammadiyah Jakarta."

.