Syiva Awaliyah Maqdis

by dts digital

Submission date: 04-Aug-2024 03:45PM (UTC+0700)
Submission ID: 2426944782
File name: 8539-23259-1-RV.docx (1.62M)
Word count: 3840
Character count: 20694

JIKO (Jurnal Informatika dan Komputer) Accredited KEMENDIKBUD RISTEK, No.105/E/KPT/2022
Vol. x, No. x, April 2024, pp. x-x p-ISSN: 2614-8897
DOI: 10.33387/jiko e-ISSN: 2656-1948

DESIGN OF MICROSLEEP DETECTION SYSTEM IN 32 BIT MICROCONTROLLER-BASED MOTORISTS WITH RANDOM FOREST METHOD

Syiva Awaliyah Maqdis [1] , Anugrah Adiwilaga [2] , Munawir Munawir[3]

[1] Universitas Pendidikan Indonesia
[2] Universitas Pendidikan Indonesia
[3] Universitas Pendidikan Indonesia
[*] Email: [1] maqdissyiva@upi.edu, [2] anugrah.adiwilaga@upi.edu, [3] munawir@upi.edu

(Received: dd mmm yyyy, Revised: dd mmm yyyy, Accepted: dd mmm yyyy)

Abstract

The number of motorcycle accidents has increased rapidly every year, many of which occur due to drowsiness or fatigue, because motorists often force themselves to keep driving. The state of fatigue while driving is also known as microsleep. Therefore, a prototype system can be installed on the helmet of a motorized user so that the driver is more alert when driving a vehicle. This system utilizes machine learning technology with the Random Forest algorithm with 2 prediction results, namely, prediction 1 which means the rider is tired and prediction 0 the rider is in a normal state embedded in the ESP32 microcontroller, and a tilt sensor that can detect signs of drowsiness in motorists. This system design will use the MPU6050 sensor to measure changes in the angle of the rider's head. The data obtained will be processed by the microcontroller to identify head changes that indicate the possibility of drowsiness, which will then appear as a warning through the sound of a buzzer to warn the driver to take a short break. The test results in sleepy conditions with an angle of 10°-30° resulted in 100% accuracy and normal conditions only at an angle of 0°-6° which resulted in 100% accuracy. The detection device can be a system design that can reduce the number of accidents caused by drowsiness.

Keywords: *Microsleep, Prototype, Microcontroller, MPU6050, Random Forest*

**Corresponding Author*: *Syiva Awaliyah Maqdis*

1. INTRODUCTION (UPPERCASE, 10pt, bold)

Traffic accidents continue to be a serious problem that threatens lives. According to data from the Central Statistics Agency (BPS), the number of accidents in Indonesia from 2019-2021 is high. Safety in mobilization is a crucial aspect that receives great attention. One of the factors that cause accidents when driving is due to drowsiness, which is often ignored by society [1]. Motorists who force themselves to drive when they are excessively sleepy can reduce the driver's alertness and can cause traffic accidents that endanger other people and the driver himself. The occurrence of fatigue when driving to the point of feeling sleepy is also known as microsleep. Microsleep is an event where a person experiences a loss of consciousness for just a fraction of a second, and it can happen at any time. Microsleep events do not last long, only seconds to 2 minutes. If a person experiences closed eyes for 0.5 seconds or more periodically, every 10 seconds or so, this indicates that the person is experiencing microsleep. Microsleep makes a person fail to focus, ignoring the surroundings alike as sensory input that later enters a person's subconscious [2].

When humans reach adulthood or are calculated starting from the age of 17 years and above, they need sleep time for approximately 8 hours every night. This microsleep factor is caused by a lack of sleep, making the body tired [3]. Drowsiness is a condition of a person between conscious and unconscious that causes a decrease in function in all five senses so that the level of a person's head tilt when sleepy is 15-45 degrees within 3 seconds [4][5], drowsiness can be categorized when the condition of the eyelids starts to be heavy, the view begins to blur, and the head starts to be unbalanced [6]. Several methods have been found to carry out early detection when someone is sleepy, one of which is the head position change method [7].

People should be more aware of the incidence of microsleep. Uniquely, microsleep does not only occur due to lack of sleep; there are several cases in which microsleep occurs in individuals who perform monotonous activities. What needs to be understood about microsleep is its impact or risk, an example of the impact that must be considered, namely, fatigue which is a decrease in a person's productivity performance when doing activities. Loss of alertness and awareness of the surrounding environment so as not to be able to detect when there is danger can also make a person have a social relationship disorder that makes a person not interested in doing social activities [8][9][10].

In an effort to keep drivers safe from the dangers and impacts of microsleep, technology has played a crucial role in developing effective solutions. The primary goal is to improve traffic safety for riders, thereby reducing the risk of accidents caused by microsleep while driving. Another objective of this research is to create a drowsiness warning system integrated into the rider's helmet. This system detects signs of drowsiness in the rider and provides warning signals to prevent them from entering a state of microsleep.

Several studies have explored the dangers of microsleep, leading to an innovation in creating a system to detect drowsiness in drivers. The research aims to detect drivers in a state of microsleep through eye blinks and eye aspect ratio combined with machine learning algorithms [11]. Another study used MPU6050 sensors to detect the tilt of an accelerometer and a gyroscope to enter the reference points of roll tilt, yaw, and pitch stored on the seat or trunk of a motorcycle [12]. In addition, some studies use the classification of Random Forest to detect human movement during activities such as climbing stairs, walking, talking, standing, and working [13]. Therefore, a new innovation was formed to prevent accidents caused by microsleep. This innovation utilizes the Lolin32 Liteas a microcontroller or component controller that connects to other components. The system is enhanced by integrating the MPU6050 sensor and a Buzzer. Lolin32 Lite has two cores that allow the microcontroller to work faster [14]. The system is a combination of machine learning, microcontroller, and sensor technology that aims to detect early signs of fatigue or drowsiness in drivers and provide a quick warning to reduce the risk of accidents. By implementing machine learning with the Random Forest algorithm, it is expected to get accurate results to distinguish between drowsy and non-drowsy positions. Random Forest is an ensemble algorithm based on bootstrap aggregating, known for its superior performance compared to other Decision-Tree algorithms [15][16].

The MPU6050 sensor is widely used due to its ability to detect changes in helmet movement and orientation, which can indicate drowsiness in the rider. Information from the MPU6050 gyro sensor is obtained by reading the angular tilt based on input data, which converts acceleration data and rotational motion into digital data [17]. The gyro sensor then integrates the angle value into machine learning, sending a signal to the Lolin32 Lite to obtain the appropriate sensor output value and distinguish between head nods and head shakes. To identify and limit the conditions so that this design functions optimally, it is by considering the condition of the road. However, this system is specifically designed for paved roads with flat and smooth surfaces. In addition, when the system detects a drowsy head position, the buzzer sounds within 2-3 seconds to provide timely warnings. Initial testing focuses on smooth road conditions to ensure the tool's effectiveness. By addressing this limitation, the design is expected to perform well in various real-world road conditions and contribute to reducing motor vehicle accidents caused by drowsiness.

2. RESEARCH METHOD

This research uses the waterfall method, namely by taking an approach in software development that describes the process in a linear sequential manner. The waterfall method will carry out system design sequentially, starting with identification and analysis according to needs to testing and maintenance.

2.1 Identification and Analysis

In this section, identification of needs is carried out, namely by making a prototype system design that is tested by pairing the prototype with a helmet, then after testing, the existing data will be integrated from the Random Forest algorithm machine learning, then when the data has been read it will be sent to Lolin32 Lite which is then sent to the buzzer to signal whether the rider is sleepy or not. If detected sleepy or in a state of head leaning forward, the buzzer will sound to warn the driver to avoid accidents.

Random Forest is a machine learning algorithm that combines the results of many decision trees to achieve one result. Each tree is built using a random subset of the dataset already obtained by retrieving data to measure a random subset of features from each partition. After that Random Forest will make predictions to combine the results of all trees [18]. The advantages of Random Forest are that it can combine many models, can cope with complex data, is more robust, and can reduce the risk of overfitting. Because it uses ensemble averaging, Random Forest is more resistant to overfitting. It is more stable because it takes the average of many models from the introduction of variability between individual trees, and provides improved overall prediction performance. The Random Forest work stages used in this research include sampling, building decision trees, prediction, and final result output.

A. Sampling
1. Start by taking several random samples to form the training set.
2. The samples that have been taken will be used to build a Decision Tree.

B. Building a Decision Tree
1. Split the dataset into a sleepy and normal subset.
2. Each internal node in the Decision Tree represents a feature or attribute, the branches represent decision rules, and the leaves represent results or labels.

C. Prediction by Voting
1. After the Decision Tree is built, Random Forest will make predictions on the test data.
2. Each Decision Tree will provide its prediction, where prediction 1 is identified as sleepy and prediction 0 is identified as normal.
3. A voting process is carried out for each prediction result that comes out of all Decision Trees.

D. Final Result
1. Random Forest will choose the most voted prediction result as the final prediction.

2.2 Design

A. Block Diagram

In Figure 1. the block diagram can be explained that the component will turn on when there is a power supply connected to Lolin32 Lite. Then Lolin32 Lite will receive data from MPU6050 to give commands to the buzzer according to the prediction of head tilt conditions to give a warning if drowsiness is detected.

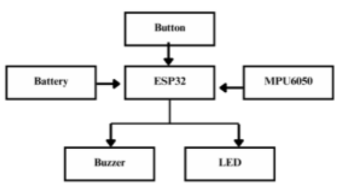

Figure 1. Block Diagram

B. Flowchart

Figure 2 will explain the flow, namely when the device has started, the MPU6050 will read the accelerometer and gyro values which will then be predicted by whether the result is 0 (in normal condition) or 1 (in a sleepy condition). If the prediction result is 1, the buzzer will turn on and if the prediction result is 0, the system will re-read the accelerometer and gyro values.

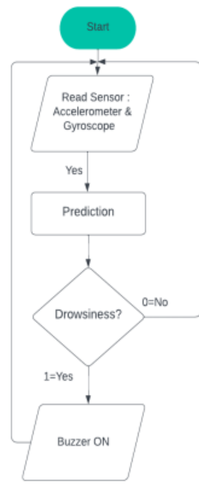

Figure 2. Flowchart of drowsiness detection device microcontroller program

C. Wiring Diagram

Figure 3, the wiring diagram illustrates the components that are connected in one circuit.

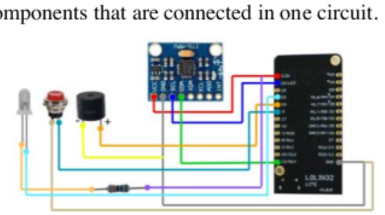

Figure 3. Wiring Diagram

D. Component Prototype

Figure 4, Figure 5, and Figure 6 Pins are connected between Lolin32 Lite, MPU6050, buzzer, LED, and button.

a. Pin GND Lolin32 Lite to GND MPU6050
b. Pin 3V Lolin32 Lite to VCC MPU6050
c. Pin 22 Lolin32 Lite to SCL MPU6050
d. Pin 13 Lolin32 Lite to SDA MPU6050
e. Pin 18 Lolin32 Lite to + Buzzer
f. Pin GND MPU6050 to - Buzzer
g. Pin 5 Lolin32 Lite to Push Button
h. Pin GND Lolin32 to Push Button
i. Pin 23 Lolin32 Lite to LED Cathode
j. Pin 3V Lolin32 Lite to Resistor
k. Anode LED to Resistor

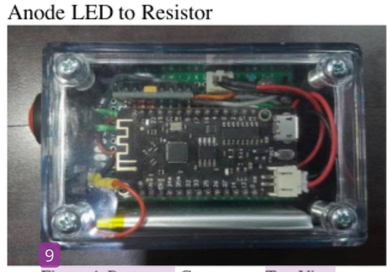

Figure 4. Prototype Component Top View

Figure 5. Prototype Component Side View

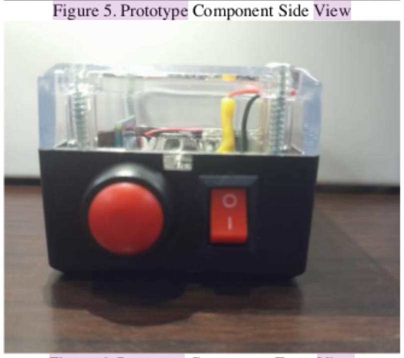

Figure 6. Prototype Component Front View

E. Illustration of Head Tilt Posture

Figure 7 explains that if someone is in a sleepy state the initial position of the head tilt is leaning towards the front, because of the initial condition when sitting upright. [19].

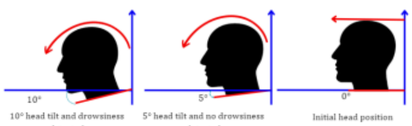

Figure 7. Illustration of head tilt

F. Illustration of the device in the helmet there is AXIS XYZ

The direction of the axis on the MPU6050 after being attached to the helmet is in the roll direction (the x-axis points to the front and back), because the initial position of drowsiness is leaning towards the front. As shown in Figure 8, the detection device is stored on the left side of the helmet and the x-axis will detect when the rider is in a state of drowsiness to give a warning with a buzzer sound.

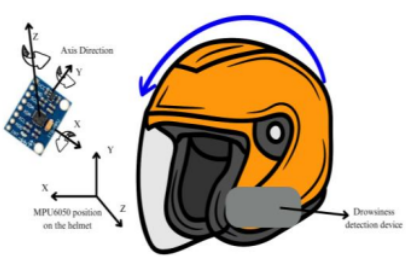

Figure 8. Illustration of Axis MPU6050 on helmet

G. Research structure

Figure 9 explains the flow of this research which is divided into two parts, namely journal writing, and prototype. In writing, carry out a literature review to find journal references to be analyzed and tested, then compare the results of the literature review with the results of the trials that have been carried out to reach conclusions.

Figure 9. Research Structure

2.3 Implementation

The prototype device is mounted on the left helmet with a switch button and push button as shown in Figure 10. The switch button is used to turn ON and OFF the detection device, when it is in use and not in use. The push button is used to pause even though the switch button is ON, when the detection device is on, the LED will light up, and when the device is paused, the LED will turn off.

Figure 10. prototype mounted on the left helmet

2.4 Testing

Testing is carried out with 4 stages, namely, taking a drowsy condition dataset, taking a normal condition dataset, testing in drowsy conditions with a slope of 10°, 15°, 20°, 25°, and 30° with 10 tests each, and testing in normal conditions with a slope of 0°, 2°, 4°, 6°, and 8° with 10 tests each.

The datasets used for training in Random Forest are taken as many as two kinds, namely sleepy datasets and normal datasets. So, there are 30 sets of sleepy data and 30 sets of normal data. Each 1 dataset has 6 values (accelerometer X, accelerometer Y, accelerometer Z, gyro X, gyro Y, and gyro Z), so for a total of 180 datasets.

Next, machine learning will study the results of the datasets that have been taken, then training is carried out using the Random Forest algorithm, after which the model will be transferred to the ESP32 for the interference process (prediction).

2.5 Maintenance

For future maintenance, it is necessary to periodically check the battery and be careful when it is raining heavily. In the detection device, a push button is also prepared for the driver to use if the driver is doing activities other than driving. This is done so that the battery used can last a long time and is not continuously charged. The battery can be charged when it has been used for 2-3 days.

3. RESULT AND DISCUSSION

At this stage, the results of the test are between sleepy conditions and normal conditions. The drowsy condition here is identified by the direction of the head leaning forward or the direction of the head tilt, which is downward. It can be noted that if the motorcyclist is in a sleepy state, a prediction with a value of 1 will appear and if the motorcyclist is in a normal state, a prediction with a value of 0 will appear. Therefore, the Random Forest algorithm was chosen to process the data that had been taken during the trial, with the result that the head tilt when sleepy is leaning forward with a slope of 10°, this was obtained from a trial conducted in a sleepy state. Random Forest can read the sleepy value and make a 10° tilt at the start of the sleepy position, then if the head is at that tilt the buzzer will sound.

The test results display the results of normal conditions (prediction 0), what percentage of success from each angle tested, and what percentage of buzzer activation remains in the OFF state. There are 5 angles tested, namely 0°, 2°, 4°, 6°, and 8° with each doing 10 tests. The test results are listed in Table 1.

Table 1 . Table 1. Testing results of 0 device prediction against normal conditions

Slope	Normal Condition Prediction (%)	Number of successful tests	Buzzer OFF (%)	Number of successful tests
0°	100%	10	100%	10
2°	100%	10	100%	10
4°	100%	10	100%	10
6°	100%	10	100%	10
8°	90%	9	90%	9

The test results display the results of the drowsiness condition (prediction 1), what percentage of success from each angle tested, and what percentage of buzzer activation remains in the OFF state. There are 5 angles tested, namely 10°, 15°, 20°, 25°, and 30° with each doing 10 tests. The test results are listed in Table 2.

Table 2. Test results of 1 device's prediction of drowsiness condition

Slope	Drowsiness Condition Prediction (%)	Number of successful tests	Buzzer ON (%)	Number of successful tests
10°	100%	10	100%	10
15°	100%	10	100%	10
20°	100%	10	100%	10
25°	100%	10	100%	10
30°	100%	10	100%	10

Furthermore, testing using the Random Forest algorithm will produce an output of 1.0 which means that the test carried out produces perfect results. As in Figure 11, the cross-validation score is 1.0

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100)

# Membuat KFold dengan 5 splits
kf = KFold(n_splits=5, shuffle=True, random_state=1)

# Melakukan K-fold cross-validation
scores = cross_val_score(model, X, y, cv=kf)

print(f'Skor untuk setiap fold adalah: {scores}')
print(f'Rata-rata skor cross-validation: {scores.mean()}')

Skor untuk setiap fold adalah: [1. 1. 1. 1. 1.]
Rata-rata skor cross-validation: 1.0
```

Figure 11. Results on jupyter

In testing the results read by the program, namely normal conditions and drowsiness conditions. The value that will be read by the program when in normal condition is Prediction 0 and the drowsiness condition is Prediction 1. Figure 12 is an example when the program reads that the driver is in normal condition.

```
from sklearn.metrics import accuracy_score
skor_akurasi = accuracy_score(y_test, y_pred)

print(f'Skor Akurasi: {skor_akurasi}')

Skor Akurasi: 1.0

# Prediksi label untuk data baru, misal prediksi data input baris 45
X_new = df.iloc[45,:-1].to_numpy()
y_pred_new = knn.predict([X_new])

print(f'Hasil Prediksi: {y_pred_new}')

Hasil Prediksi: [0]
```

Figure 12. Figure 12. Prediction Results on jupyter

Figure 13, shows that testing with 18 test data resulted in a precision value of 1.00, recall of 1.00, F1-Score of 1.00, and accuracy of 1.00 which means success without error. In the context of machine learning data analysis, 1.00 refers to the highest value of an evaluation matrix.

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Confusion Matrix:
[[ 8  0]
 [ 0 10]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       1.00      1.00      1.00        10

    accuracy                           1.00        18
   macro avg       1.00      1.00      1.00        18
weighted avg       1.00      1.00      1.00        18
```

Figure 13. Precision, Recall, F1-Score, Accuracy

The test shows prediction 1 (drowsiness condition) and prediction 0 (normal condition), by testing from various degree angles, each carried out 10 times. The test results in drowsy conditions with an angle of 10°-30° resulted in 100% accuracy and in normal conditions only the angle of 0°-6° resulted in 100% accuracy.

4. CONCLUSION

Based on the results of the research that has been done, it can be concluded that the drowsiness detection device can detect head tilt when drowsiness occurs, this can be proven when the head tilt at the roll angle reaches 10 ° then the buzzer will sound. So, when the rider is in a state of drowsiness, the device will sound the buzzer attached to the left helmet, so that the rider can immediately wake up when in a state of microsleep. The MPU6050 can detect when the rider experiences one of the drowsiness characteristics in the form of a forward head nod with an angle of at least 10° and within 2-3 seconds.

5. REFERENCE

[1] S. A. Khan, H. Mukhtar, and B. A. Pramudita, "Perancangan Sistem Pendeteksi Microsleep Untuk Peringatan Kelelahan Pada Pengemudi Kendaraan," *eProceedings of Engineering*, vol. 9, no. 4, Aug. 2022, Accessed: Nov. 20,2023.[Online]. Available:https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/18227

[2]A. Hertig-Godeschalk, J. Skorucak, A. Malafeev, P. Achermann, J. Mathis, and D. R. Schreier, "Microsleep episodes in the borderland between wakefulness and sleep," *Sleep*, Jul. 2019, doi: https://doi.org/10.1093/sleep/zsz163.

[3] A. Hartoko, Susijanto Tri Rasmana, and Heri Pratikno, "Implementasi Sistem Deteksi Mata Kantuk Berdasarkan Facial Landmarks Detection Menggunakan Metode Regression Trees," *Journal of Technology and Informatics (JoTI)*, vol. 1, no. 1, pp. 1–9,Oct.2019,doi: https://doi.org/10.37802/joti.v1i1.1.

[4] Sugeng Sugeng and Taufiq Nuzwir Nizar, "Deteksi Aktivitas Mata, Mulut Dan Kemiringan Kepala Sebagai Fitur Untuk Deteksi Kantuk Pada Pengendara Mobil," *Komputika*, vol. 12, no. 1, pp. 83–91, May 2023,doi: https://doi.org/10.34010/komputika.v12i1.9688.

[5] L. Elisana, N. W. Yanto, S. W. Lestari, and W. Handini, "Rancang Bangun Sistem Pendeteksi Kantuk Menggunakan Sensor IMU dan WeMos," *http://jurnalftijayabaya.ac.id?index.php/JTek*, Nov. 27, 2021. https://repo.jayabaya.ac.id/1655/1/Dokumen%20Publikasi%20211.pdf (accessed Nov. 26, 2023).

[6] Cahya AJi Saputra, Danang Erwanto, and Putri Nur Rahayu, "Deteksi Kantuk Pengendara Roda Empat Menggunakan Haar Cascade Classifier Dan Convolutional Neural Network," *Journal of Electrical Engineering and Computer*, vol. 3, no. 1, pp. 1–7, Apr. 2021, doi: https://doi.org/10.33650/jeecom.v3i1.1510.

[7] A. D.n, C. Nugraha, and H. Prassetiyo, "Perancangan Alat Deteksi Dini Kondisi Kantuk untuk Mengurangi Risiko Kecelakaan Kerja Berbasis Pengolahan Citra Digital," *e-Proceeding FTI*, 2022, Accessed: Jan. 06, 2024. [Online]. Available: https://eproceeding.itenas.ac.id/index.php/fti/article/view/1718

[8] F. Darmajati, "Model Mitigasi Kecelakaan Transportasi Menggunakan Pencegah *Microsleep*," *www.cyberarea.id*, 2022. http://www.cyberarea.id/index.php/cyberarea/article/view/218 (accessed Mar. 16, 2024).

[9] B. I. S. Duna, Y. Iashania, and N. M. A. Ganang, "Manajemen Fatigue untuk Mencegah *Microsleep* pada Driver Sarana: Fatigue Management to Prevent *Microsleep* in Facilities Drivers," *JURNAL TEKNIK PERTAMBANGAN*, vol. 23, no. 2, pp. 34–39, Aug. 2023, Accessed: Feb. 20, 2024. [Online]. Available: https://e-journal.upr.ac.id/index.php/JTP/article/view/10492

[10] R. Parulian, F. Napadap, J. Muhamad Jabbar, and M. Pamungkas, "Faktor Risiko *Microsleep* pada Driver Ojek Online Antapani Tahun 2023," 2023. Accessed: Mar. 12, 2024. [Online]. Available: https://siakad.stikesdhb.ac.id/repositories/400520/4005200058/ARTIKEL%20PDF.pdf

[11] H. Z. Ilmadina, D. Apriliani, and D. S. Wibowo, "Deteksi Pengendara Mengantuk dengan Kombinasi Haar Cascade Classifier dan Support Vector Machine," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 7, no. 1, pp. 1–7, Jan. 2022, doi: https://doi.org/10.30591/jpit.v7i1.3346.

[12] Aries Suprayogi, Hurriyatul Fitriyah, and Tibyani Tibyani, "Sistem Pendeteksi Kecelakaan Pada Sepeda Motor Berdasarkan Kemiringan Menggunakan Sensor Gyroscope Berbasis Arduino," *Jurnal Pengembangan*

Teknologi Informasi dan Ilmu Komputer, vol. 3, no. 3, pp. 3079–3085, 2019, Accessed: Mar. 09, 2024. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4872

[13] Ir. E. Supriyadi, MT. and A. Hamzah, MT., "Rancang Bangun Prototype Cruise Control pada Kendaraan Listrik dengan Metode Kendali PID," 2023. http://repository.istn.ac.id/7162/1/STugas%20FTI%2BLap%20Teliti%20RBangun%20Cruise%20Control.pdf (accessed Jun. 27, 2024).

[14]Fauzi Adi Rafrastaraa, Ricardus Anggi Pramunendar, Dwi Puji Prabowo, Etika Kartikadarma, and Usman Sudibyo, "Optimasi Algoritma *Random Forest* menggunakan Principal Component Analysis untuk Deteksi Malware," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5 , no. 3, pp. 217–223, Jul. 2023, doi: https://doi.org/10.47233/jteksis.v5i3.854.

[15]I. Luthfiana Mulyahati, "Implementasi Machine Learning Prediksi Harga Sewa Apartemen Menggunakan Algoritma *Random Forest* Melalui Framework Website Flask Python (Studi Kasus: Apartemen di DKI Jakarta Pada Website mamikos.com)," *dspace.uii.ac.id*, Jul. 2020, Accessed: Jun. 27, 2024. [Online]. Available: https://dspace.uii.ac.id/handle/123456789/23970

[16]F. Mangkusasmito, D. Y. Tadeus, H. Winarno, and E. Winarno, "Peningkatan Akurasi Sensor GY-521 MPU-6050 dengan Metode Koreksi Faktor Drift," *Ultima Computing : Jurnal Sistem Komputer*, vol. 12, no. 2, pp. 91–95, Nov. 2020, doi: https://doi.org/10.31937/sk.v12i2.1791.

[17]G. G. Putra and D. U. Suwarno, "Pembaca Aktivitas Manusia Dengan Sensor Gyro," *Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia (SENASTINDO)*, vol. 1, pp. 139–146, 2019, Accessed: Jun. 29, 2024. [Online]. Available: https://aau.e-journal.id/senastindo/article/view/97/96

[18] G. Artheg Sandag, "View of Prediksi Rating Aplikasi App Store Menggunakan Algoritma Random Forest," *Unklab.ac.id*, 2024. Available: https://cogito.unklab.ac.id/index.php/cogito/article/view/270/153. [Accessed: Jul. 18, 2024]

[19]M. Amirullah, H. Kusuma, and D. Tasripan, "Sistem Peringatan Dini Menggunakan Deteksi Kemiringan Kepala pada Pengemudi Kendaraan Bermotor yang Mengantuk," *JURNAL TEKNIK ITS*, vol. 7, no. 2, 2018, Available: https://ejurnal.its.ac.id/index.php/teknik/article/viewFile/31011/5495. [Accessed: Jul. 15, 2024]

Syiva Awaliyah Maqdis

ORIGINALITY REPORT

8%	6%	3%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ejournal.unkhair.ac.id Internet Source	1%
2	www.atlantis-press.com Internet Source	1%
3	Submitted to Universitas Khairun Student Paper	1%
4	Iveren Blessing Iorliam, Barnabas Achakpa Ikyo, Aamo Iorliam, Emmanuel Odeh Okube, Kenneth Dekera Kwaghtyo, Yahaya I. Shehu. "Application of Machine Learning Techniques for Okra Shelf Life Prediction", Journal of Data Analysis and Information Processing, 2021 Publication	1%
5	www.ncbi.nlm.nih.gov Internet Source	1%
6	Submitted to University of Hertfordshire Student Paper	<1%
7	dev.to Internet Source	<1%

8	www-users.cse.umn.edu Internet Source	<1 %
9	Submitted to Colorado School of Mines Student Paper	<1 %
10	publications.aap.org Internet Source	<1 %
11	www.frontiersin.org Internet Source	<1 %
12	www.scielo.br Internet Source	<1 %
13	"Innovative Computing and Communications", Springer Science and Business Media LLC, 2024 Publication	<1 %
14	dspace.umkt.ac.id Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		