# EARLY ANOMALY PREDICTION OF MACHINE DAMAGE USING C4.5 ALGORITHM BASED ON IOT

**[1]Praseta Citra Bhaskara, [2]Y.b. Dwi Setianto**
[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Katolik Soegijapranata
[2]setianto@unika.ac.id

## ABSTRACT

*Machine condition is a problem that is difficult to predict, prediction of machine condition is an important aspect in the application of maintenance because the occurrence of damage can result in a decrease in the productivity of a company. Measurement of vibration, temperature, and machine displacement is a fairly good method to determine the condition of the machine because it is an indicator of mechanical conditions and an early indicator of damage to the machine as a whole, Application of Algorithms in processing vibration, temperature and machine displacement data to improve the prognosis of damage. In this project, predictions of machine condition will be carried out using the C4.5 algorithm. Data taken using sensors at a certain time will be used to predict the decline in the performance of a machine. This data will be used for training and testing data. This project was concluded that the C45 algorithm obtained accuracy with the difference between Training data and Test data, 60.8% for Training data and 76.4% for Testing data. Proving that the C45 algorithm is effective for predicting the initial anomaly of damage to the machine. It is necessary to re-calibrate the sensor limits, and replace the sensors used because this project uses sensors for prototypes.ds yang berhubungan dengan artikel ini. Tuliskan paling tidak tiga buah keywords.*

**Keywords:** Machine condition, Vibration, Temperature, Machine Displacement, C4.5

## INTRODUCTION

Machine condition prediction plays an important role in the process various industries in the world. The condition of machine components is a special concern for the company, because if there is damage it can result in a decrease in productivity. Therefore, it is very important to diagnose and diagnose damage to the rotating machine before the damage occurs, so that a more appropriate method and time of maintenance can be planned.

There are various methods that can be used to measure machine condition, such as vibration analysis, oil analysis, temperature measurement, pressure measurement, flow rate measurement, and others. Measurement of vibration, temperature, and machine displacement are several methods that are quite good for determining machine condition because vibration and temperature are indicators of mechanical conditions and early indicators of defects in a machine as a whole. The higher the vibration and the resulting temperature, this indicates a decrease in the condition of the machine. Productivity will not be reduced because the machine is still running during data retrieval. From the results of periodic data measurements, data will be obtained which will then be trained with the C4.5 algorithm.

This project aims to predict machine condition based on vibration, temperature, and machine displacement. From there we can improve the maintenance that needs to be applied to the machine and how long the machine will last. Based on the problems, it can be formulated into,

1. How to build a C45 structure that can be used for predicting the condition of the machine based on data in the form of temperature, vibration, and engine displacement?
2. Is the C45 algorithm effective for predicting machine condition?

Such that this research will try to propose an automatic system that is able to predict the condition of the machine by using the C45 algorithm and develop research on the application of artificial intelligence in the field of maintenance, especially for condition monitoring.

## LITERATURE STUDY

In the late 1970s to early 1980s J. Ross Quinlan, a researcher in the field of machine learning, making a decision tree algorithm known as ID3 (Iterative dichotomizer). Quinlan then developed the C4.5 algorithm decision) which is the development of the ID3 algorithm [2]. This algorithm has advantages, namely easy to understand, flexible, and attractive because it can be visualized in the form of images [3]. Decision Tree is Knowledge Discovery In Database (KDD). The result of this method will get knowledge from database processed. The database will be extracted so that produce information or knowledge that useful [9]. The steps taken in KDD is data mining. Data mining is an algorithm for extracting valuable information hidden in data collection (database) is huge, so found previously unknown interesting pattern [10].

One of the algorithms used in data mining is the C4.5 algorithm. The C4.5 algorithm is an algorithm used to form a decision tree and its rules. The C4.5 algorithm maps attribute values into classes that can be applied to the new classification [7]. The C4.5 algorithm is a tree structure where there are nodes that describe the attributes, each branch describes the results of the tested attributes, and each leaf describes the class. The C4.5 algorithm recursively visits each decision node, choosing the optimal division, until it can no longer be divided. The C4.5 algorithm uses the concept of information gain or entropy reduction to choose the optimal division [2].

Research conducted by Khafiizh Hastuti [4] Inactive students are students who stopped learning and did not do administrative registration. Students with non-active status have a tendency to drop out of school. The high percentage of students with non-active status affects the university's accreditation score. It is necessary to know the factors that cause students to be inactive. Data mining classification techniques can be used to predict inactive students. Many data mining classification algorithms can be used, so comparisons need to be made to determine the level of accuracy of each algorithm. The algorithm used is logistic regression, decision tree, nave bayes and neural network. The data used are 3861 students of the Informatics Engineering, Information Systems and Visual Communication Design study program at Dian Nuswantoro University. The results of the classification process were evaluated using cross-validation, confusion matrix, ROC curve and T-Test to determine the most accurate data mining classification algorithm for predicting inactive students.

Research conducted by Gaol, Nur Yanti Lumban [1] success and The failure of student studies is a reflection of the quality of a university. Non-active student are students who do not register at the beginning of the semester or do not follow study for at least one semester. The existence of these non-active students, of course, has an effect on students graduating not on time, the more non-active students then more and more students graduate not on time other than that non-active students can increase the number of students who have the potential to drop out. Graduating on time is an element of assessment college accreditation. Therefore, to overcome this problem, data mining with the C.4.5 method to find the characteristics of potentially non-active students. Design This research uses CRISP-DM and the research uses data from Faculty of Undergraduate students Dian Nuswantoro University Computer Science. The validation process used is split validation. and for testing the model using the confusion matrix. The results show the best accuracy value is 97.60% with a training data ratio of 90%. Based on the experiments conducted, students with Social Studies Semester 3 is below 2.60, semester 3 credits taken are <20 credits, SKS semester 4 is empty and semester 4 social studies is empty, the greater the potential to be non-active students.

Research conducted by Mohammad Taufan [5] conducted research on Life Time Prediction in Rounding Machine with Anfis Method (Adaptive Neuro-Fuzzy Inference Systems). From the vibration data taken using a vibration sensor at a certain time will be used to predict the decline in the performance of a machine. It was concluded that in the process of training and testing anfis using a hybrid learning system the average error was better than using backpropagation. Mohammad Taufan's research still needs to be developed, where performance in predicting the remaining useful life is considered important for Improved.

Research conducted by Devila Mustika Franceca [6] conducted a study on Bearing Remaining Life Prediction Using Exponential Regression to determine the RUL model on bearings based on the exponential regression method and to determine the effect of cutting data by ignoring normal conditions starting from the time of failure. It is concluded that exponential regression can be used to determine or predict Remaining Useful Life (RUL) with very satisfactory results.

Ginting, Selvia Lorena Br et al [8] used the C4.5 Algorithm to predict the student's study period based on academic score data. The results obtained through this study have a major influence on the percentage of fit and accuracy in predicting the student's study period based on academic value data.

## RESEARCH METHODOLOGY

### *Collecting Data*

In this final project used data in the form of temperature data, vibration, engine displacement data from a machine consisting of a motorcycle, a new fan, an old fan. Sensor value limits use personal assumptions. Here are 10 examples of data obtained from each machine :

**Table 1.** Motorcycle Data

| id | Temperature | Vibration | X | Y | Life |
|---|---|---|---|---|---|
| 0 | 32 | 17 | 636 | 757 | <5Tahun |
| 1 | 31 | 3 | 592 | 736 | <5Tahun |
| 2 | 32 | 23 | 577 | 771 | <5Tahun |
| 3 | 32 | 17 | 575 | 770 | <5Tahun |
| 4 | 32 | 13 | 585 | 747 | <5Tahun |
| 5 | 31 | 14 | 589 | 793 | >5Tahun |
| 6 | 30 | 9 | 620 | 732 | >5Tahun |
| 7 | 32 | 14 | 607 | 707 | <5Tahun |
| 8 | 32 | 4 | 588 | 779 | <5Tahun |
| 9 | 30 | 25 | 587 | 683 | <5Tahun |
| 10 | 33 | 23 | 589 | 775 | >5Tahun |

**Table 2.** New Fan Data

| id | Temperature | Vibration | X | Y | Life |
|---|---|---|---|---|---|
| 1 | 30 | 4 | 190 | 76 | >5Tahun |
| 2 | 31 | 5 | 207 | 61 | >5Tahun |
| 3 | 31 | 0 | 157 | 56 | >5Tahun |
| 4 | 31 | 3 | 168 | 63 | >5Tahun |
| 5 | 31 | 4 | 210 | 76 | <5Tahun |
| 6 | 30 | 1 | 180 | 83 | <5Tahun |
| 7 | 30 | 2 | 209 | 79 | >5Tahun |
| 8 | 30 | 2 | 157 | 89 | <5Tahun |
| 9 | 31 | 4 | 165 | 60 | >5Tahun |
| 10 | 29 | 0 | 187 | 61 | <5Tahun |

**Table 3.** Old Fan Data

| id | Temperature | Vibration | X | Y | Life |
|---|---|---|---|---|---|
| 1 | 31 | 8 | 180 | 79 | <5Tahun |
| 2 | 30 | 3 | 170 | 94 | <5Tahun |
| 3 | 29 | 6 | 175 | 75 | >5Tahun |
| 4 | 29 | 2 | 198 | 88 | <5Tahun |
| 5 | 31 | 14 | 192 | 78 | >5Tahun |
| 6 | 30 | 14 | 172 | 73 | >5Tahun |
| 7 | 30 | 12 | 196 | 70 | >5Tahun |
| 8 | 29 | 10 | 187 | 100 | >5Tahun |
| 9 | 30 | 10 | 180 | 81 | >5Tahun |
| 10 | 31 | 0 | 196 | 87 | >5Tahun |

### *Prototype Schematic*

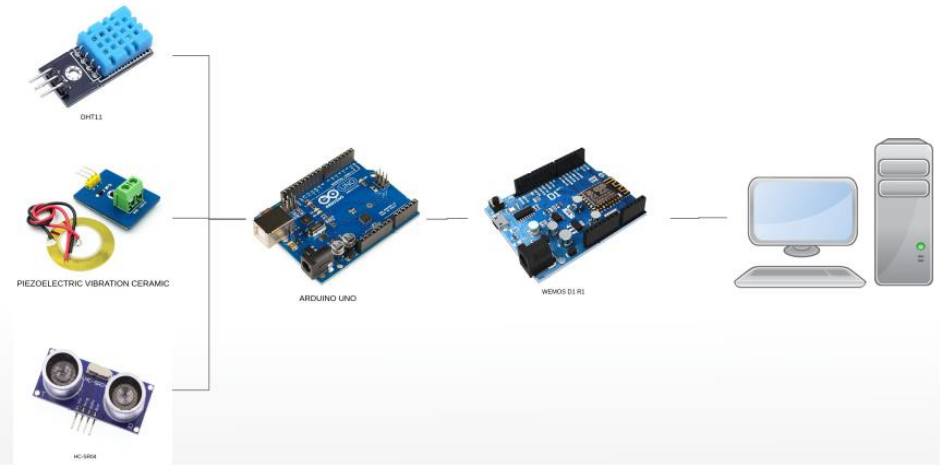


**Figure 1.** Prototype Schematic

## Age Machine Sensor

| No | Temperature | Vibration | X | Y | Life |
|---|---|---|---|---|---|
| 1 | 0 | 8 | 274 | 35 | >5Tahun |
| 2 | 30 | 10 | 178 | 11 | >5Tahun |
| 3 | 0 | 6 | 203 | 36 | >5Tahun |
| 4 | 0 | 7 | 179 | 36 | >5Tahun |
| 5 | 30 | 9 | 280 | 11 | >5Tahun |
| 6 | 0 | 0 | 204 | 4 | >5Tahun |
| 7 | 30 | 4 | 204 | 11 | >5Tahun |
| 8 | 30 | 0 | 280 | 19 | >5Tahun |
| 9 | 30 | 0 | 255 | 9 | >5Tahun |
| 10 | 30 | 3 | 179 | 10 | >5Tahun |
| 11 | 0 | 13 | 256 | 104 | >5Tahun |
| 12 | 0 | 0 | 154 | 4 | >5Tahun |
| 13 | 0 | 7 | 179 | 99 | >5Tahun |
| 14 | 30 | 1 | 154 | 81 | >5Tahun |
| 15 | 30 | 0 | 179 | 32 | >5Tahun |

**Figure 2.** Website View

## IMPLEMENTATION AND TESTING

### *Arduino Code*

This program was created using the Arduino programming language, below will explain the program code.

```
1. #include <DHT.h>
2. #include <HCSR04.h>
3. #define DHT_PIN A0
4. #define DHTTYPE DHT11
5. DHT dht(DHT_PIN, DHTTYPE);
6. int Temp;
7. int y;
8. int x;
9. int vib = A5;
10.String life;
11.const int TriggerPin = 2;
12.const int TriggerPinx = 4;
13.const int EchoPin = 6;
14.const int EchoPinx = 5;
15.long duration, distance;
16.long durationx, distancex;
```

Lines 1-2 to add the library to be used, lines 3-5 to set DHT11 reading on pin A0 and determining the type of DHT sensor used, line 10 to set Piezoelectric Vibration Ceramic sensor reading at pin A5, lines 12-15 to set 2 The HC-SR04 reads on pins D2, D4, D6, D5, lines 7-9 and 16-17 are used to add variables.

```
1. void setup() {
2.   Serial.begin(9600);
3.   dht.begin();
4.   pinMode(TriggerPin,OUTPUT);
5.   pinMode(EchoPin,INPUT);
6.   pinMode(TriggerPinx,OUTPUT);
7.   pinMode(EchoPinx,INPUT);
8. }
```

Line 2 to start Serial with 9600 baudrate, line 3 to start sensor

DHT, lines 4-7 to set TriggerPin, TriggerPinx as OUTPUT and EchoPin, EchoPinx as INPUT.

```
1. void loop() {
2.   String req = "";
3.   while(Serial.available()>0){
4.     req += char(Serial.read());
5.   }
6.   req.trim();
7.   if(req=="Yes"){
8.     sendata();
9.   }
10.        req = "";
11.        delay(1000);
12.      }
```

Lines 3-5 to receive the serial communication process from NodeMCU and make the req variable into a String filled with character data, line 6 to delete the received data space, lines 7-9 when Arduino receives serial data from NodeMCU in the form of "Yes" then Arduino will sending data to NodeMCU.

```
1. if ((Temp <= 30) && (VibFix <= 50) && (x <= 250) && (y <= 150))
2.     {
3.       life=">5Tahun";
4.     }
5.     else if ((Temp <= 30) && (VibFix <= 50) && (x <= 250) && (y > 150))
6.     {
7.       life=">5Tahun";
8.     }
9. .....
10.       else
11.           {
12.             life="<5Tahun";
13.           }
```

Lines 1-13 make a statement to fill in the value of life, where the constraints used in the sample code are personal assumptions. For application to the machine, calibration needs to be carried out according to the standard machine used.

```
1. String                                data                                =
   String(Temp)+"#"+String(VibFix)+"#"+String(x)+"#"+String(y)+"#"+Strin
   g(life);
2.   Serial.println(data);
```

Line 1 is used to accommodate all sensor data on 1 string variable 'data' because in serial communication each sending process can only send 1 data, line 2 is used to display String data on a serial monitor.

### *NodeMCU Code*

```
1. #include <SoftwareSerial.h>
2. #include <ESP8266WiFi.h>
3. #include <ESP8266HTTPClient.h>
```

Line 1 is used to accommodate all sensor data on 1 string variable 'data' because in serial communication each sending process can only send 1 data, line 2 is used to display String data on a serial monitor.

```
1. SoftwareSerial DataSerial(12,13);
2. unsigned long previousMillis = 0;
3. const long interval = 3000;
4. String arrData[5];
```

Line 1 digunakan untuk membuat variable Software serial (Rx, Tx), Line 2-3 untuk menggantikan delay karena NodeMCU akan restart dengan sendirinya apabila terlalu lama delay, Line

```
1. const char *ssid = "Bhaskara";
2. const char *password = "cobatanyampos";
3. const char *host = "192.168.1.12";
```

Line 1-3 to declare the ssid, password, ip used.

```
1. WiFi.mode(WIFI_STA);
2.   WiFi.begin(ssid, password);
3.   while (WiFi.status() != WL_CONNECTED){
4.     delay(500);
5.   }
```

Line 1 to set wifi mode as STA, line 2 to start connection on stated SSID and password, line 3-5 to check if NodeMCU is connected to wifi.

```
1. unsigned long currentMillis = millis();
2.   if(currentMillis - previousMillis >= interval){
3.     previousMillis = currentMillis;
4.     String data = "";
5.     while(DataSerial.available()>0){
6.     data += char(DataSerial.read());
7.     }
8.     data.trim();
9.     if(data!=""){
10.           int index = 0;
11.             for(int i=0; i<data.length(); i++){
12.             char delimiter = '#';
13.             if(data[i] != delimiter)
14.               arrData[index] += data[i];
15.             else
16.               index++;
17.             }
18.             if(index == 4){
19.             Serial.println("Temperature : " + arrData[0]);
20.             Serial.println("Vibration : " + arrData[1]);
21.             Serial.println("X : " + arrData[2]);
22.             Serial.println("Y : " + arrData[3]);
23.             Serial.println("Life : " + arrData[4]);
24.             Serial.println();
25.             }
26.             Temperature = arrData[0].toInt();
27.             Vibration = arrData[1].toInt();
28.             X = arrData[2].toInt();
29.             Y = arrData[3].toInt();
30.             Life = arrData[4];
31.             arrData[0]="";
32.             arrData[1]="";
33.             arrData[2]="";
34.             arrData[3]="";
35.             arrData[4]="";
36.           }
37.           DataSerial.println("Yes");
38.         }
```

Line 1 to read the current mailing list time, line 2 to reread data, lines 11-17 to receive data from Arduino Uno, lines 18-25 to display data values to serial monitor, lines 26-29 to convert String data to INT, lines 31-35 to empty the arrData[] variable is used to hold new data, line 37 sends the data string “Yes” to Arduino Uno.

```
1. WiFiClient client;
2.   const int httpPort = 80;
3.   if (!client.connect(host, httpPort)){
4.     Serial.println("Failed");
5.     return;
6.   }
7.   String url = "/Project/write-data.php?Temperature=";
8.   url += Temperature;
9.   url += "&Vibration=";
10.        url += Vibration;
11.        url += "&X=";
12.        url += X;
13.        url += "&Y=";
14.        url += Y;
15.        url += "&Life=";
16.        url += Life;
17.        client.print(String("GET ") + url + " HTTP/1.1\r\n" +
18.                     "Host: " + host + "\r\n" +
19.                     "Connection: close\r\n\r\n");
20.        unsigned long timeout = millis();
21.        while (client.available() == 0) {
22.          if (millis() - timeout > 1000) {
23.            Serial.println(">>> Client Timeout !");
24.            client.stop();
25.            return;
26.          }
27.        }
28.         while (client.available()) {
29.          String line = client.readStringUntil('\r');
30.          Serial.print(line);
31.        }
```

Lines 3-6 to connect NodeMCU on port 80, lines 7-25 to send data that has been received from Arduino Uno to localhost database.

### *C45 Algorithm Code*

This program was created using the MySql programming language, below will explain the program code.

```
1. CREATE TABLE tblData(
2. id INT,
3. Temperature INT,
4. Vibration INT,
5. X INT,
6. Y INT,
7. Life varchar(20)
8. );
9.
10.      LOAD DATA LOCAL INFILE 'myFile0.csv'
11.      INTO TABLE tblData
12.      FIELDS TERMINATED BY ','
13.      ENCLOSED BY '"'
14.      LINES TERMINATED BY '\n'
15.      IGNORE 1 ROWS;
```

Lines 1-8 to create a table to store csv data, lines 10-15 to import csv data to tblData.

```
1. select @amountdata:=count(*)
2. from tblData;
3.
4. select @kurang5tahun:=count(*)
5. from tblData
6. where Life LIKE ('%<5Tahun%');
7.
8. select @lebih5tahun:=count(*)
9. from tblData
10.       where Life LIKE ('%>5Tahun%');
```

Lines 1-2 to calculate the amount of data in tblData, lines 4-6 to calculate the age of the data with the value '<5Tahun', lines 8-10 to calculate the age of the data with the value '>5Tahun'.

```
1. insert into tblCount
2.   (info, amountdata, kurang5tahun, lebih5tahun)
3.   select distinct(A.Temperature) as TEMPERATURE, count(A.Temperature)
   as JUMLAHDATA,
4.   (
5.     select COUNT(*)
6.     from tblData as B
7.     where B.Life LIKE ('%<5Tahun%') and
8.     B.Temperature = A.Temperature
9.   )AS RATINGLOW,
10.        (
11.          select COUNT(*)
12.          from tblData as C
13.          where C.Life LIKE ('%>5Tahun%') and
14.          C.Temperature = A.Temperature
15.        )as RATINGHIGH
16.      from tblData as A
17.      group by A.Temperature;
18.      ....
19.      insert into tblCount
20.          (info, amountdata, kurang5tahun, lebih5tahun)
21.        select distinct(A.Y) as Y, count(A.Y) as JUMLAHDATA,
22.          (
23.            select COUNT(*)
24.            from tblData as B
25.            where B.Life LIKE ('%<5Tahun%') and
26.            B.Y = A.Y
27.          )AS RATINGLOW,
28.          (
29.            select COUNT(*)
30.            from tblData as C
31.            where C.Life LIKE ('%>5Tahun%') and
32.            C.Y = A.Y
33.          )as RATINGHIGH
34.        from tblData as A
35.        group by A.Y;
```

Lines 1-35 are used to calculate the sum of '<5Tahun' and '>5Tahun' for each attribute, then put in tblCount.

```
1. update tblCount set entropy =
2. (-(kurang5tahun/amountdata) *log2(kurang5tahun/amountdata))
```

```
3. +
4. (-(lebih5tahun/amountdata) *log2(lebih5tahun/amountdata));
5.
6. insert into tblTampung(atribut, gain)
7. select atribut, @entropy - SUM((amountdata/@amountdata)*entropy) as
   COUNTGAIN
8. from tblCount
9. group by atribut;
```

Lines 1-4 to calculate the entropy value, lines 6-9 to calculate the gain value.

### *Testing*

In the C-45 algorithm, after calculating all the data and getting the results from each iteration, then get the final result. Confusion matrix used to calculate the accuracy of the data.

**Table 4.** The results of the iteration process with 700 data

| Iteration | Attribute |
|---|---|
| 1 | Y |
| 2 | X |
| 3 | Vibration |

**Table 5.** Condition training results after iteration process with 700 data

| Atribute | Info | Damage | GoodCondition |
|---|---|---|---|
| Vibration | 1 | 0 | 1 |
| Vibration | 9 | 1 | 0 |

**Table 6.** Condition training master data

| Atribute | Info | Damage | GoodCondition | Amount Data |
|---|---|---|---|---|
| Vibration | 1 | 14 | 24 | 38 |
| Vibration | 9 | 15 | 21 | 36 |

**Table 7.** Confusion matrix data with 700 dataset

| | Predicted GoodCondition | Predicted Damage |
|---|---|---|
| **Reality GoodCondition** | 24 | 14 |
| **Reality Damage** | 15 | 21 |

$$Accuracy = \frac{21 + 24}{74} = 0.608x100 = 60.8\%$$

After calculating the accuracy using 700 Training data, then compare the accuracy with 300 Testing data to show whether the results of calculating the accuracy of the data are correct.

**Table 8.** Condition testing results after iteration process with 300 data

| Atribute | Info | Damage | GoodCondition |
|---|---|---|---|
| Y | 219 | 1 | 0 |
| Y | 239 | 0 | 1 |
| Y | 242 | 1 | 0 |

**Table 9.** Condition testing master data

| Atribute | Info | Damage | GoodCondition | Amount Data |
|---|---|---|---|---|
| Y | 219 | 2 | 3 | 5 |
| Y | 239 | 0 | 6 | 6 |
| Y | 242 | 2 | 4 | 6 |

**Table 10.** Confusion matrix data with 300 dataset

| | Predicted GoodCondition | Predicted Damage |
|---|---|---|
| **Reality GoodCondition** | 6 | 0 |
| **Reality Damage** | 4 | 7 |

$$Accuracy = \frac{6+7}{17} = 0.764x100 = 76.4\%$$

After calculating the accuracy of the Training data and Testing data there is a difference in accuracy. And the accuracy of the final result is the accuracy of the Testing data.

## CONCLUSION

It was concluded that the C45 algorithm obtained accuracy with the difference between Training data and Test data, 60.8% for Training data and 76.4% for Testing data. Proving that the C45 algorithm is effective for predicting the initial anomaly of damage to the machine.

The sensor value limit used in this project uses personal assumptions, it is necessary to re-calibrate the machine to be tested to get the official sensor value limit, and replace the sensors used because this project uses sensors for prototypes.

## REFERENCES


[1] G. N. Y. Lumban, "Prediksi Mahasiswa Berpotensi Non Aktif Menggunakan Data Mining dalam Decision Tree dan Algoritma C4.5," *Jurnal Informasi & Teknologi*, 2020. https://doi.org/10.37034/jidt.v2i1.22.

[2] H. Jiawei and K. Micheline,"Data Mining: Concepts and Techniques. Morgan Kaufmann," 2006. [Online]. Available: http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-

Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf. Accessed: December 27, 2020.

[3] G. Florin,"Data Mining,"Vol. 12, *Intelligent Systems Reference Library*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. https://doi.org/10.1007/978-3-642-19721-5.

[4] H. Khafizih,"Analisis Komparasi Algoritma Klasifikasi Data Mining Untuk Prediksi Mahasiswa Non Aktif," 2012. [Online]. Available: http://publikasi.dinus.ac.id/index.php/semantik/article/download/132/87. Accessed: December 27, 2020.

[5] T. Mohammad,"Prediksi Sisa Umur Pada Rotating Machinery Dengan Metode ANFIS (Adaptive Neuro-Fuzzy Inference Systems)," 2010. [Online]. Available: https://dokumen.tips/reader/f/prediksi-sisa-umur-pada-rotating-machinery-i-tugas-akhir-tm-091486-prediksi. Accessed: December 27, 2020.

[6] P. D. Mustika and D. Sutawanir,"Prediksi Sisa Umur Bearing Menggunakan Regresi Eksponensial," 2021. [Online]. Available: https://journals.unisba.ac.id/0337661f-6b6b-4ef7-863b-ed9ec6381d61. Accessed: December 28, 2020.

[7] N. Andhika and O. Isni,"Penerapan Algoritma Klasifikasi Data Mining C4.5 Pada Dataset Cuaca Wilayah Bekasi," *Konferensi Nasional Ilmu Sosial dan Teknologi. Vol 1, no 1*, 2017. [Online]. Available: https://media.neliti.com/media/publications/224664-penerapan-algoritma-klasifikasi-data-min-e8105f77.pdf. Accessed: December 28, 2020.

[8] G. S. Lorena Br, W. Zarman, and I. Hamidah,"Analisis Dan Penerapan Algoritma C4.5 Dalam Data Mining Untuk Memprediksi Masa Studi Mahasiswa Berdasarkan Data Nilai Akademik," 2014. [Online]. Available: https://ejournal.akprind.ac.id/1388b545-6412-44e3-8a27-964a4a14db11. Accessed: December 29, 2020.

[9] J. Arta, I. Kadek, G. Indrawan, and G. R. Dantes," Data Mining Rekomendasi Calon Mahasiswa Berprestasi Di STMIK Denpasar Menggunakan Metode Technique For Others Reference By Similarity To Ideal Solution," *JST (Jurnal Sains dan Teknologi)* 5, 2017. https://doi.org/10.23887/jst-undiksha.v5i2.8549.

[10] E. Yusuf and W. P. Pudjo," Pemilihan Criteria Splitting Dalam Algoritma Iterative Dichotomiser 3 (ID3) Untuk Penentuka Kualitas Beras : Studi Kasus Pada Perum Bulog Drive Lampung," 2012. [Online]. Available: https://journal.budiluhur.ac.id/index.php/telematika/article/download/159/153. Accessed: December 29, 2020.