DAFTAR PUSTAKA

A. Nursyahid, H. Helmy, A. I. Karimah, dan T. A. Setiawan, "Nutrient Film Technique (NFT) hydroponic nutrition controlling system using linear regression method," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1108, no. 1, hal. 012033, 2021, doi: 10.1088/1757-899x/1108/1/012033.

Anissa, D. L. F., & Andryani, R. (2022). Penerapan Cloud Computing Dalam Aplikasi Panggil Teknisi Berbasis Android Menggunakan Google Cloud Platform. *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, *6*(2), 1292-1300.

A.Peryanto, A. Yudhana, R. Umar, "*Convolutional Neural Network and Support Vector Machine in Classification of Flower Images*" Khazanah informatia: Jurnal ilmu computer dan informatika Vol. 8 No. 1 | April 2022

D. Palacios, D. Gregor, M. Arzamendia, K. Cikel, dan M. Santacruz, "Determination of a computational parameter of quantitative production of crops in hydroponic greenhouses through digital image processing," *IEEE Chil. Conf. Electr. Electron. Eng. Inf. Commun. Technol. CHILECON 2019*, Nov 2019, doi: 10.1109/CHILECON47746.2019.8987719.

D. J. De Beer dan T. H. Joubert, "Impedance Spectroscopy for Determination of Total Dissolved Solids in Aqueous Solutions of Sodium Chloride and Magnesium Sulphate," *Proc. IEEE Sensors*, vol. 2019-October, Okt 2019, doi: 10.1109/SENSORS43011.2019.8956627.

D. T. , S. Yuga Hadfridar Putra, "SISTEM PEMANTAUAN DAN PENGENDALIAN NUTRISI, SUHU, DAN TINGGI AIR PADA PERTANIAN HIDROPONIK BERBASIS WEBSITE," *Coding J. Komput. dan Apl.*, vol. 6, no. 3, Sep 2018, doi: 10.26418/CODING.V6I3.29041.

E. Rasywir, R. Sinaga, and Y. Pratama, "Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode *Convolutional Neural Network* (CNN)," Paradig. - J. Komput. dan Inform., vol. 22, no. 2, pp. 117–123, 2020, doi: 10.31294/p.v22i2.8907.

Fonda, H. (2020). Klasifikasi w Batik Riau Dengan Menggunakan Convolutional Neural Networks (Cnn): Klasifikasi Batik Riau Dengan Menggunakan Convolutional Neural Networks (Cnn). *Jurnal Ilmu Komputer*, *9*(1), 7-10.

I. L. Fajari, A. Salsabila, dan T. Tohir, "Rancang Bangun Sistem Hidroponik Nutrient Film Technique (NFT) Sebagai Media Terobosan Penanaman Tanaman Menggunakan Wemos Mega + WiFi R3 Atmega2560," hal. 26–27, 2020.

I. Denata, T. Rismawan, I. Ruslianto, "Implementasi *Deep Learning* untuk Klasifikasi Jenis Jeruk dengan Menggunakan Metode *Convolutional Neural Network,*" Telematika: Jurnal Informatika dan Teknologi Informasi Vol. 18, No. 3, Oktober 2021, pp.297-307. DOI:10.31515/telematika.v18i3.5541

P. Megantoro, dan A. Ma'Arif, "Nutrient Film Technique for Automatic ydroponic System Based on Arduino," *Proceeding - 2020 2nd Int. Conf. d. Electr. Electron. ICIEE 2020*, hal. 84–86, 2020, doi: .1109/ICIEE49813.2020.9276920.

vongsai, "Automatic control and management system for tropical

hydroponic cultivation," *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2019-May, 2019, doi: 10.1109/ISCAS.2019.8702572.

J. M. Kim, D. G. Lee, dan S. W. Lee, "Temperature-Dependent Behavioral Model of Active-Matrix Liquid Crystal Displays for All Types of Liquid Crystals," *J. Disp. Technol.*, vol. 12, no. 9, hal. 881–887, Sep 2016, doi: 10.1109/JDT.2016.2547459.

Kholik, A. (2021). Klasifikasi Menggunakan Convolutional Neural Network (Cnn) Pada Tangkapan Layar Halaman Instagram. *Jurnal Data Mining dan Sistem Informasi*, *2*(2), 10-20.

M. Singgih, K. Prabawati, dan D. Abdulloh, "Bercocok Tamam Mudah Dengan Sistem Hidroponik NFT," *J. Abdikarya J. Karya Pengabdi. Dosen dan Mhs.*, vol. 03, no. 1, hal. 21–24, 2019.

O. Gandhi, M. Ramdhani, M. A. Murti, dan C. Setianingsih, "Water flow control system based on context aware algorithm and IoT for hydroponic," *Proc. - 2019 IEEE Int. Conf. Internet Things Intell. Syst. IoTaIS 2019*, hal. 212–217, Nov 2019, doi: 10.1109/IOTAIS47347.2019.8980373.

M. T. Niculescu, A. Florescu, dan P. Sever, "Accuracy improvement for the biochemistry assays by using total dissolved solids and turbidimetry measurements," *2020 8th E-Health Bioeng. Conf. EHB 2020*, Okt 2020, doi: 10.1109/EHB50910.2020.9280270.

Muliadi, M., Imran, A., & Rasul, M. (2020). Pengembangan tempat sampah pintar menggunakan ESP32. *Jurnal Media Elektrik*, *17*(2), 73-79.

M. Muslihudin, W. Renvillia, T. Taufiq, A. Andoyo, dan F. Susanto, "IMPLEMENTASI APLIKASI RUMAH PINTAR BERBASIS ANDROID DENGAN ARDUINO MICROCONTROLLER," *J. Keteknikan dan Sains*, vol. 1, no. 1, hal. 23–31, Jun 2018.

O. Kulce dan L. Onural, "Analysis of the longitudinal component of the electric field generated by flat and pixelated liquid crystal displays," *3DTV-Conference*, vol. 2016-August, Agu 2016, doi: 10.1109/3DTV.2016.7548957.

O. M. Sinaulan, "Perancangan Alat Ukur Kecepatan Kendaraan Menggunakan ATMega 16," *J. Tek. Elektro dan Komput.*, vol. 4, no. 3, hal. 60–70, 2015.

Rahmadhani, L. E., Widuri, L. I., & Dewanti, P. (2020). Kualitas mutu sayur kasepak (kangkung, selada, dan pakcoy) dengan sistem budidaya akuaponik dan hidroponik. *Jurnal Agroteknologi*, *14*(01), 33-43.

R. S. (Ritha) Simbar dan A. (Alfi) Syahrin, "Prototype Sistem Monitoring Temperatur Menggunakan Arduino Uno R3 Dengan Komunikasi Wireless," *J. Teknol. Elektro*, vol. 8, no. 1, hal. 143288, Apr 2017, doi: 10.22441/JTE.V8I1.1381.

S. Mohammad Noviansyah, Hafdiarsya, "PERANCANGAN ALAT KONTROL RELAY LAMPU RUMAH VIA MOBILE," *J. Akrab Juara*, vol. 4, no. 4, hal. 1–14, 2019.

h and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis mbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural twork," *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia)*, l. 3, no. 2, pp. 49–56, 2018.

Lampiran 1

Program untuk training model

```
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.utils import plot_model

from sklearn.model_selection import train_test_split
```

Import data set

format dataset mau diubah jadi:

input: {hari,php,nutrisi}

output: {pompa nutrisi,pompa air, pompa ph_up, pompa ph_down}

```
# import dataset
df = pd.read  csv('dataset  hidroponik.csv')


#split train test
train, testa = train  test  split(df, test  size=0.3)
valid, test = train_test_split(testa, test_size=0.1)


#masukkan ke variable
x_train = train.iloc[:, 0:3]
y  train = train.iloc[:, 3:]

x_test = test.iloc[:, 0:3]
y  test = test.iloc[:, 3:]

x  valid = test.iloc[:, 0:3]
yvalid = test.iloc[:, 3:]

print("train data")
print(x_train)
in)
data")
t)
t)
d data")
```

```
print(x_test)
print(y_test)

-----------------------------------------------------------------------------------------------
# Bikin model


model = keras.Sequential([
    #keras.layers.Flatten(input_shape=[3]),

    keras.layers.Conv1D(input_shape=(1,3),filters=3, kernel_size=3, activation='relu'),
    #keras.layers.BatchNormalization(),
    #keras.layers.Activation('relu'),

    #keras.layers.Conv1D(filters=16, kernel  size=3, activation='relu'),
    #keras.layers.BatchNormalization(),
    #keras.layers.Activation('relu'),

    keras.layers.Dropout(0.5),
    keras.layers.MaxPooling1D(),
    keras.layers.Flatten(),

    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(2) #TODO ganti output jadi 4 (atau 3 kalau nilai pompa ph digabung)
])

model.compile(loss=tf.keras.metrics.mean_squared_error,
              optimizer="rmsprop",          #TODO ganti parameter optimizer
              metrics=[tf.keras.metrics.RootMeanSquaredError(name='rmse')])

print(model.summary())
plot_model(model, to_file='model.png', show_shapes=True, show_layer_names=True)


-----------------------------------------------------------------------------------------------
#TODO plot
import matplotlib.pyplot as plt

# summarize history for accuracy
tory.history['rmse'])
tory.history['val_rmse'])
odel accuracy')
accuracy')
epoch')
```

```
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val  loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

----------------------------------------------------------------------------------------------------
#test predict
prediction_test = [1,3,200]
y = model.predict([prediction  test])
print(y)
----------------------------------------------------------------------------------------------------
# convert model ke tensorflow lite
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

# Guardar modelo en el disco
open("nutrisi_sensor_model.tflite", "wb").write(tflite_model)
----------------------------------------------------------------------------------------------------
from tinymlgen import port
# Ubah model tflite jadi c header array
c_code = port(model, pretty_print=True)
print(c_code)
```

Lampiran 2

Program untuk kendali nutrisi

```
/**
 * Program utama dengan menerapkan sensor pedeteksi PH dan TDS pada Hidroponik Tanaman Pakcoy
 * BUG: Program crash (ditandai dengan restart) saat mendeteksi TDS dengan nilai terlalu tinggi
 */

// Librari Awal Arduino
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
// Untuk Library ML
#define ESP32 1
#include <EloquentTinyML.h>
#include <eloquent_tinyml/tensorflow.h> //comment it if you're using 0.0.5 version
// untuk http request firebase
// include wifi untuk konek internet
#include "WiFi.h"
#include <HTTPClient.h>
#include <Firebase_ESP_Client.h>
// Provide the token generation process info.
#include <addons/TokenHelper.h>
// untuk ambil jam dari internet
// Pastikan memakai NTPClient dari https://github.com/taranais/NTPClient
#include <NTPClient.h>
#include <WiFiUdp.h>
// Math dan String
#include <stdlib.h>
#include <string.h>

// OLED Setup
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 oled(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
// ADC Resultion untuk konversi analog
 ADC_RESOLUTION = 4096.0;
ample pengukuran dan mengambil nilai rata-rata
umSamplePH = 30;
umSampleTDS = 40;
```

```
// Delay pengukuran dalam ms
const int delaySamplePH = 100;
const int delaySampleTDS = 30;
// Init pH
#define PIN_PH 34
// Kalibrasi saat short
const float BASE_PH = 1.53;
// NOTE: Kalibrasi saat pengukuran
const float ph1x = 0.88; // Ganti sesuai voltage pada ph = 4.01
const float ph1y = 3.7;
const float ph2x = 0.68; // Ganti sesuai voltage pada ph = 1.49
const float ph2y = 7.3;
// Var util
float phSense;
// Init TDS
#define PIN_TDS 35
float tdsSense = 0;
float temperature = 25; // Temperature sekarang untuk TDS Ideal
// Init Pompa
#define PIN_P1 32
#define PIN_P2 33
#define PIN_P3 25
#define PIN_P4 26
// Status Pompa
bool statusP1 = false;
bool statusP2 = false;
bool statusP3 = false;
bool statusP4 = false;

// Get Status Server URL
String serverPath = "https://us-central1-arduinoann.cloudfunctions.net/getStatus";

/* 1. Define the WiFi credentials */
#define WIFI_SSID "sdd"
#define WIFI_PASSWORD "111222333"
// #define WIFI_SSID "GUEST HOUSE 710"
// #define WIFI_PASSWORD "bahar710"
// #define WIFI_SSID "Unlisted-Hotspot"
// #define WIFI_PASSWORD "anukuji123456"

e the API Key */
PI_KEY "AIzaSyAv6GjW3ML3FNO1LABO9gaMn_60w1o-UPA"

e the project ID */
```

```
#define FIREBASE_PROJECT_ID "arduinoann"

/* 4. Define the user Email and password that alreadey registerd or added in your project */
#define USER_EMAIL "saya@wasdlabs.com"
#define USER_PASSWORD "canceled"

// Define Firebase Data object
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long dataMillis = 0;
int count = 0;

bool taskCompleted = false;

// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

// Variables to save date and time
String formattedDate;
String dayStamp;
String timeStamp;

// sensor_model.h contains the array you exported from the previous step with xxd or
tinymlgen
#include "sensormodel.h"

#define NUMBER_OF_INPUTS 3
#define NUMBER_OF_OUTPUTS 2
// in future projects you may need to tweak this value: it's a trial and error process
#define TENSOR_ARENA_SIZE 2 * 1024 // default 2

// Eloquent::TinyML::TfLite<NUMBER_OF_INPUTS, NUMBER_OF_OUTPUTS,
TENSOR_ARENA_SIZE> ml; //uncomment it if you're using version 0.0.5

Eloquent::TinyML::TensorFlow::TensorFlow<NUMBER_OF_INPUTS,
NUMBER_OF_OUTPUTS, TENSOR_ARENA_SIZE> tf; // comment it if you're using 0.0.5
```


Filter Algorithm

```
int getMedianNum(int bArray[], int iFilterLen)
{
  int bTab[iFilterLen];
  for (byte i = 0; i < iFilterLen; i++)
    bTab[i] = bArray[i];
  int i, j, bTemp;
  for (j = 0; j < iFilterLen - 1; j++)
  {
    for (i = 0; i < iFilterLen - j - 1; i++)
    {
      if (bTab[i] > bTab[i + 1])
      {
        bTemp = bTab[i];
        bTab[i] = bTab[i + 1];
        bTab[i + 1] = bTemp;
      }
    }
  }
  if ((iFilterLen & 1) > 0)
  {
    bTemp = bTab[(iFilterLen - 1) / 2];
  }
  else
  {
    bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
  }
  return bTemp;
}

/**
 * Fungsi untuk mengukur PH
 */
float ph()
{
  float phVoltage;
  float sumSample = 0;
  // Iterasi sebanyak jumlah sample berdasarkan delay
  for (int i = 0; i < numSamplePH; i++)
  {
    phVoltage = (3.3 / ADC  RESOLUTION) * analogRead(PIN  PH);
                ample += phVoltage;
                (delaySamplePH);


                ge = sumSample / numSamplePH;
```

```
    float phMeasure = (((phVoltage - ph1x) * (ph2y - ph1y)) / (ph2x - ph1x)) + ph1y;
    Serial.printf("Tegangan Ph = %.2f\n", phVoltage);
    Serial.printf("Nilai Ph    = %.2f\n", phMeasure);
    return phMeasure;
}

/**
 * Fungsi untuk mengukur TDS
 */
float tds()
{
    int analogTDSArray[numSampleTDS];
    float tdsMeasure;
    // Iterasi sebanyak jumlah sample berdasarkan delay
    for (int i = 0; i < numSampleTDS; i++)
    {
        analogTDSArray[i] = analogRead(PIN_TDS);
        delay(delaySampleTDS);
    }
    // Median Filtering Supaya Pengukuran Lebih Stabil dan convert ke nilai voltage
    float averageVoltage = getMedianNum(analogTDSArray, numSampleTDS) * (float)3.3 /
ADC_RESOLUTION;
    // temperature compensation formula: fFinalResult(25^C) =
fFinalResult(current)/(1.0+0.02*(fTP-25.0));
    float compensationCoefficient = 1.0 + 0.02 * (temperature - 25.0);
    // temperature compensation
    float compensationVoltage = averageVoltage / compensationCoefficient;
    // convert voltage value to tds value
    tdsMeasure = (133.42 * compensationVoltage * compensationVoltage *
compensationVoltage - 255.86 * compensationVoltage * compensationVoltage + 857.39 *
compensationVoltage) * 0.5;
    return tdsMeasure;
}

/**
 * Ganti status kontrol pompa
 */
void pumpControl()
{

                                    sP1)

                                    lWrite(PIN_P1, LOW);
                                    rintln("on");
```

```
    else
    {
      digitalWrite(PIN_P1, HIGH);
      oled.println("off");
    }
    oled.setCursor(30, 50);
    if (statusP2)
    {
      digitalWrite(PIN_P2, LOW);
      oled.println("on");
    }
    else
    {
      digitalWrite(PIN_P2, HIGH);
      oled.println("off");
    }
    oled.setCursor(60, 50);
    if (statusP3)
    {
      digitalWrite(PIN_P3, LOW);
      oled.println("on");
    }
    else
    {
      digitalWrite(PIN_P3, HIGH);
      oled.println("off");
    }
    oled.setCursor(90, 50);
    if (statusP4)
    {
      digitalWrite(PIN_P4, LOW);
      oled.println("on");
    }
    else
    {
      digitalWrite(PIN_P4, HIGH);
      oled.println("off");
    }
  }
```


tan Nilai PH dan TDS pada OLED

Sensor(float ph, float tds)

```
    oled.setCursor(0, 0);
    oled.println("PH");
    oled.setCursor(50, 0);
    oled.println("TDS");
    oled.setCursor(0, 10);
    oled.println("-----------------");
    oled.setCursor(0, 20);
    char stringPH[5];
    sprintf(stringPH, "%.2f", ph);
    oled.println(stringPH);
    oled.setCursor(50, 20);
    char stringTDS[10];
    sprintf(stringTDS, "%.2f ppm", tds);
    oled.println(stringTDS);
    oled.display();
}

/**
 * Tampilan Selamat Datang OLED
 */
void welcomeOled()
{
    // Welcome OLED
    oled.clearDisplay();
    oled.setTextSize(1);
    oled.setTextColor(WHITE);
    oled.setCursor(0, 0);
    oled.println("Hai Selamat Datang");
    oled.setCursor(0, 10);
    oled.println("My Smart Hidroponik");
    oled.setCursor(0, 20);
    oled.println(":)");
    oled.display();
    delay(2000);
}

void setup()
{
    Serial.begin(115200);
    // OLED Init
    .begin(SSD1306_SWITCHCAPVCC, 0x3C))

    .println(F("failed to start SSD1306 OLED"));
    (1)
```

```
}
// Pin Mode TDS
pinMode(PIN_PH, INPUT);
pinMode(PIN_TDS, INPUT);
// Pin Mode Pompa
pinMode(PIN_P1, OUTPUT);
pinMode(PIN_P2, OUTPUT);
pinMode(PIN  P3, OUTPUT);
pinMode(PIN_P4, OUTPUT);
digitalWrite(PIN  P1, HIGH);
digitalWrite(PIN_P2, HIGH);
digitalWrite(PIN_P3, HIGH);
digitalWrite(PIN  P4, HIGH);
// Tampilkan welcome pada OLED
welcomeOled();

Serial.println("Loading model...");
tf.begin(model  data);
Serial.println("Model dah keload");
Serial.println("Konek ke wifi");
WiFi.begin(WIFI  SSID, WIFI  PASSWORD);
Serial.print("Connecting to Wi-Fi");
while (WiFi.status() != WL_CONNECTED)
{
   Serial.print(".");
   delay(300);
}
Serial.println();
Serial.print("Connected with IP:");
Serial.print(WiFi.localIP());
Serial.printf("\nFirebase Client v%s \n", FIREBASE_CLIENT_VERSION);

/* Assign the api key (required) */
config.api  key = API  KEY;
/* Assign the user sign in credentials */
auth.user.email = "mr.unlistedi@gmail.com";
auth.user.password = "canceled123";
/* Assign the callback function for the long running token generation task */
config.token_status_callback = tokenStatusCallback; // see addons/TokenHelper.h
// Limit the size of response payload to be collected in FirebaseData
 ResponseSize(2048);
.begin(&config, &auth);
.reconnectWiFi(true);
 ambil jam dari internet
lize a NTPClient to get time
```

```
    timeClient.begin();
    timeClient.setTimeOffset(28800); // SET time offset di sini
}

void loop()
{
    // Tarik data sensor
    phSense = ph();
    Serial.printf("PH = %.2f \n", phSense);
    tdsSense = tds();
    Serial.printf("TDS = %.2f ppm \n", tdsSense);
    // Show measurement PH and TDS
    oled.clearDisplay();
    showSensor(phSense, tdsSense);
    oled.display();

    // Throttle
    if(phSense > 9 || phSense < 3){
        phSense = 7;
    }

    // Ambil status hari keberapa pada database
    HTTPClient http;
    int hariKe = 0;
    bool pumpTDSStatus = false;
    http.begin(serverPath.c  str());
    int httpResponseCode = http.GET();
    if (httpResponseCode > 0)
    {
        Serial.print("HTTP Response code: ");
        Serial.println(httpResponseCode);
        String payload = http.getString();
        // Cari posisi karakter ";"
        int indexT = 0;
        for (int i = 0; i < payload.length(); i++)
        {
            if (payload.charAt(i) == ';')
            {
                indexT = i;
                break;


                str hari
                al.println(payload.substring(indexT - 3));
                e = payload.substring(0, indexT).toInt();
```

```
    // Jika substr sudah ";" true atau false
    if (payload.substring(indexT + 1) == "true")
    {
      pumpTDSStatus = true;
    }
}
else
{
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}
http.end();
Serial.println("================");
Serial.print("Hari ke-");
Serial.print(hariKe);
Serial.print("\nStatus Nutrisi = ");
Serial.print(pumpTDSStatus);
Serial.print("\n");

// Masukkan ke input ML
// Format {Hari, PH, Nutrisi}
float input[3] = {(float) hariKe, phSense, tdsSense};
// float input[3] = {1, 6.77, 40};
// {ph, TDS} dalam detik
// NOTE: pH > 0 berarti ph mesti up dan pH < 0 berarti pH mesti down
float output[2] = {};
// Prediksi ML
tf.predict(input, output);
Serial.println("================");
Serial.printf("Predicted PH      : %.2f\n", output[0]);
Serial.printf("Predicted Nutrisi : %.2f\n", output[1]);
Serial.println("================");
// Jika waktu tidak terupdate
while (!timeClient.update())
{
    timeClient.forceUpdate();
}

// Upload hasil prediksi ke Firebase
// TODO: masukkan juga nilai catatan sensor di sini
irebase(output[0], output[1], phSense, tdsSense);
TDSStatus)

P3 = true;
Control();
```

```
        delay(output[1] * 1000);
        statusP3 = false;
        pumpControl();
        // Pompa 4 = 25 - output[1] (pompa 3)
        statusP4 = true;
        pumpControl();
        delay(abs(25000 - (output[1] * 1000)));
        statusP4 = false;
        pumpControl();
    }

    // Kontrol pompa PH sesuai dengan durasi
    if (output[0] > 0)
    {
        statusP1 = true;
        pumpControl();
        delay(output[0] * 1000);
    }
    if (output[0] < 0)
    {
        statusP2 = true;
        pumpControl();
        delay(abs(output[0]) * 1000);
    }

    statusP1 = false;
    statusP2 = false;
    pumpControl();

    // TODO: Production Ubah pengukuran ke rentang waktu jauh misal setiap 3 Jam (108000)
    delay(108000);
}

void sendToFirebase(double nilaiPH, double nilaiNutrisi, double sensorPh, double
sensorTDS)
{
    if (Firebase.ready())
    {
        dataMillis = millis();
        formattedDate = timeClient.getFormattedDate();

        aseJson content;
        g documentPath = "data";
        rite data ke field document **/
        nt.set("fields/nutrisi/doubleValue", nilaiNutrisi);
```

```
        content.set("fields/ph/doubleValue", nilaiPH);
        content.set("fields/sensorph/doubleValue", sensorPh);
        content.set("fields/sensornutrisi/doubleValue", sensorTDS);
        content.set("fields/myTimestamp/timestampValue", formattedDate); // RFC3339 UTC
"Zulu" format

        count++;

        Serial.print("Create a document... ");

        if (Firebase.Firestore.createDocument(&fbdo, FIREBASE_PROJECT_ID, "" /*
databaseId can be (default) or empty */, documentPath.c_str(), content.raw()))
            Serial.printf("ok\n%s\n\n", fbdo.payload().c_str());
        else
            Serial.println(fbdo.errorReason());
    }
}

// The Firestore payload upload callback function
void fcsUploadCallback(CFS_UploadStatusInfo info)
{
    if (info.status == fb_esp_cfs_upload_status_init)
    {
        Serial.printf("\nUploading data (%d)...\n", info.size);
    }
    else if (info.status == fb_esp_cfs_upload_status_upload)
    {
        Serial.printf("Uploaded %d%s\n", (int)info.progress, "%");
    }
    else if (info.status == fb_esp_cfs_upload_status_complete)
    {
        Serial.println("Upload completed ");
    }
    else if (info.status == fb_esp_cfs_upload_status_process_response)
    {
        Serial.print("Processing the response... ");
    }
    else if (info.status == fb_esp_cfs_upload_status_error)
    {
        Serial.printf("Upload failed, %s\n", info.errorMsg.c_str());
```

If this is not the case and you expect multiple output from your model, you have to declare an output array.

```
  float input[10] = { ... };
  float output[5] = { 0 };

  tf.predict(input, output);
*/
```