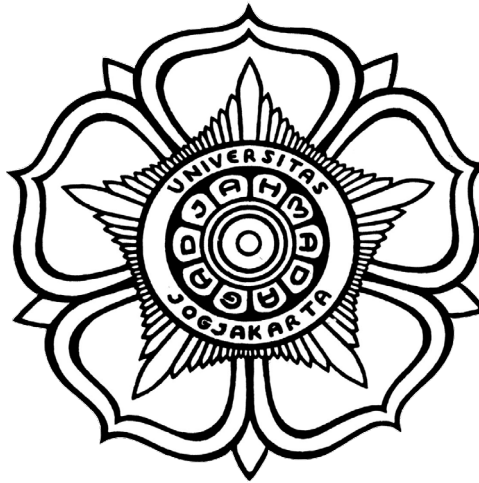


**LAPORAN TUGAS KELOMPOK**  
**PRAKTIK TEKNIK KENDALI DAN MESIN LISTRIK**

*“Gesture Classification With Esp32 and TinyML”*



Disusun Oleh:

Kelompok 3/ARM 2

Ade Surya Pramudya (19/441188/SV/16540)

Lukman Aulia Rahman (19/441201/SV/16553)

Mahesa Audriansyah Agatha (19/441203/SV/16555)

**DEPARTEMEN TEKNIK MESIN**  
**SEKOLAH VOKASI**  
**UNIVERSITAS GADJAH MADA**  
**YOGYAKARTA**

**2022**

## KATA PENGANTAR

Puji dan syukur kami panjatkan kehadiran Tuhan Yang Maha Esa karena atas berkat rahmat-Nya kami dapat menyelesaikan Project Praktikum Teknik Kendali beserta laporan akhir dan *demo* program dengan baik dan tepat pada waktunya.

Adapun isi dari laporan akhir ini adalah bentuk dokumentasi atas *project* yang dilakukan. Laporan ini disusun untuk melengkapi tugas akhir mata kuliah Praktikum Teknik Kendali dan Mesin Listrik.

Penyusun mengucapkan terimakasih sebesar-besarnya kepada Bapak Andhi Akhmad Ismail, ST, M.Eng, dan Bapak Irfan Bahiuddin, S.T., M.Phil., Ph. D. selaku dosen mata kuliah Praktikum Teknik Kendali dan Mesin Listrik beserta Tenaga Kependidikan yang selalu membimbing dan memberikan ilmunya dalam pembelajaran hingga penyusunan laporan ini. Dan juga kepada semua pihak yang telah membantu.

Penyusun menyadari bahwa sepenuhnya laporan ini masih sangat jauh dari kesempurnaan. Oleh karena itu, kritik serta saran yang membangun sangat diharapkan untuk penyempurnaan laporan akhir ini.

Atas perhatian dari semua pihak maka kami ucapkan terimakasih. Semoga laporan Project Praktikum Teknik Kendali dapat bermanfaat bagi penyusun dan semua pembaca serta dapat dipergunakan seperlunya.

Yogyakarta, 30 Mei 2021

Penyusun

## DAFTAR ISI

KATA PENGANTAR .....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Deskripsi Kasus .....	3
BAB II ISI.....	4
A. Tahap Persiapan .....	4
B. Tahap Pengerjaan.....	5
BAB III PENUTUP .....	16
C. Kesimpulan .....	16
D. Saran .....	16

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

#### **1. ESP32 WiFi Module**

ESP32 merupakan sebuah mikrokontroller yang dibuat oleh perusahaan bernama Espressif Systems. Salah satu kelebihan yang dimiliki oleh ESP32 yaitu sudah terdapat Wi-Fi dan Bluetooth di dalamnya. Selain itu, ESP32 memiliki biaya yang relatif lebih terjangkau. ESP32 memiliki kemampuan yang lebih powerfull dibandingkan dengan mikrokontroller sejenis dikelasnya. Berikut merupakan spesifikasi mikrokontroller ESP32 :

<b>Number of cores</b>	2 (dual core)
<b>Wi-Fi</b>	2.4 GHz up to 150 Mbits/s
<b>Bluetooth</b>	BLE (Bluetooth Low Energy) and legacy Bluetooth
<b>Architecture</b>	32 bits
<b>Clock frequency</b>	Up to 240 MHz
<b>RAM</b>	512 KB
<b>Pins</b>	30 or 36 (depends on the model)
<b>Peripherals</b>	Capacitive touch, ADC (analog to digital converter), DAC (digital to analog converter), I2C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Network), SPI (Serial Peripheral Interface), I2S (Integrated Inter-IC Sound), RMII (Reduced Media-Independent Interface), PWM (pulse width modulation), and more.
<b>Processor</b>	Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz

#### **2. Sensor MPU6050**

DFRobot 6 DOF Sensor - MPU6050 merupakan perangkat motiontracking 6-sumbu pertama yang memiliki kebutuhan daya rendah, biaya rendah, kinerja tinggi ponsel cerdas, tablet, dan sensor yang dapat dikenakan. Perangkat MPU-6000/6050 menggabungkan giroskop 3-sumbu dan akselerometer 3-sumbu dengan Digital Motion Processor (DMP) onboard yang mampu memproses algoritma 9-sumbu motionfusion yang kompleks. Algoritma motion fusion 9-sumbu terintegrasi bagian mengakses magnetometer eksternal atau sensor lain melalui bus

master I2C tambahan, yang memungkinkan perangkat mengumpulkan set data sensor lengkap tanpa intervensi dari prosesor sistem. Berikut merupakan spesifikasi sensor MPU6050 :

- a. Working voltage: 3~5v
- b. I2C Digital-output of 6 or 9-axis MotionFusion data in rotation matrix, quaternion, Euler Angle, or raw data format
- c. Tri-Axis angular rate sensor (gyro) with a sensitivity up to 131 LSBs/dps and a full-scale range of  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , and  $\pm 2000$ dps
- d. Tri-Axis accelerometer with a programmable full scale range of  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  and  $\pm 16g$
- e. Digital Motion Processing (DMP) engine offloads complex MotionFusion, sensor timing synchronization and gesture detection
- f. Embedded algorithms for run-time bias and compass calibration. No user intervention required
- g. Dimensions: 14 x 21mm(0.55x0.83")

### 3. Kabel Jumper

Kabel Jumper adalah kabel elektrik yang memiliki pin konektor di setiap ujungnya dan memungkinkanmu untuk menghubungkan dua komponen yang melibatkan arduino tanpa memerlukan solder. Kegunaan kabel jumper ini adalah sebagai konduktor listrik untuk menyambungkan rangkaian listrik. Biasanya kabel jumper digunakan pada breadboard atau alat prototyping lainnya agar lebih mudah untuk mengutak-atik rangkaian. Konektor yang ada pada ujung kabel terdiri atas dua jenis yaitu konektor jantan (male connector) dan konektor betina (female connector). Prinsip kerja kabel jumper yaitu menghantarkan arus listrik dari satu komponen ke komponen lainnya yang dihubungkan.



## **B. Deskripsi Kasus**

Pada kasus ini kelompok kami melakukan klasifikasi gerakan menggunakan ESP32 dan TinyML. Klasifikasi gerakan adalah contoh sederhana dan sekaligus bagus tentang apa yang dapat dilakukan oleh Machine Learning. Dengan menggunakan banyak data tidak beraturan untuk mengklasifikasikan berbagai hal. Setelah itu kami membuat empat kelas klasifikasi, yaitu idle, kanan kiri, atas bawah dan melingkar. Pada prosesnya dilakukan training data pada masing-masing kelas tersebut. Setelah itu dilakukan *test data* untuk melihat apakah berhasil atau tidak. Jika berhasil maka akan didapatkan nilai akurasi dan nilai *loss* atas proses pengklasifikasian yang dilakukan.

## BAB II

### ISI

#### A. Tahap Persiapan

##### 1. Komponen Yang Digunakan

- a. ThingSoC ESP32 WiFi Module
- b. DFRobot 6 DOF Sensor - MPU6050
- c. Jumper wires (generic)

##### 2. Persiapan Aplikasi Yang Digunakan

###### a. Arduino IDE

Pada kasus ini kami menggunakan Arduino IDE untuk melakukan pemrograman terhadap sensor MPU6050 yang dirangkaikan dengan ESP32. Sensor MPU6050 tersebut diprogram sedemikian sehingga dapat terhubung dan dapat mendeteksi gestur-gestur gerakan.

###### b. Edge Impulse Studio

Edge Impulse Studio merupakan sebuah web developer yang dapat berfungsi sebagai solusi cerdas dalam pembelajaran *machine learning*. Pada Edge Impulse Studio sendiri dapat melakukan training data dan testing data. Dan juga mendapatkan nilai keakuratan pada proses *machine learning*.

###### c. Install Python3

Python adalah bahasa pemrograman interpretatif yang dapat digunakan di berbagai platform dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode dan merupakan salah satu bahasa populer yang berkaitan dengan Data Science, Machine Learning, dan Internet of Things (IoT). Keunggulan python yang bersifat interpretatif juga banyak digunakan untuk prototyping, scripting dalam pengelolaan infrastruktur, hingga pembuatan website berskala besar.

Pada kasus kali ini fungsi python bertujuan untuk koneksi antara web server dan chip ESP32. Program yang telah diupload pada ESP32 kita flask dengan bantuan python menuju web server yang diinginkan.

d. Install Node.js

Node.js adalah runtime environment untuk JavaScript yang bersifat open-source dan cross-platform. Dengan Node.js kita dapat menjalankan kode JavaScript di mana pun, tidak hanya terbatas pada lingkungan browser.

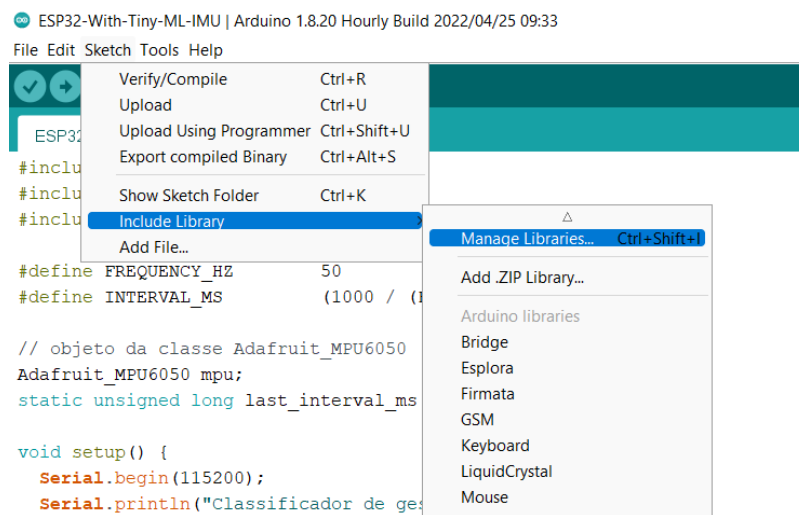
Pada kasus ini runtime javascript digunakan untuk menginstall beberapa library dengan command pada terminal node.js.

## B. Tahap Pengerjaan

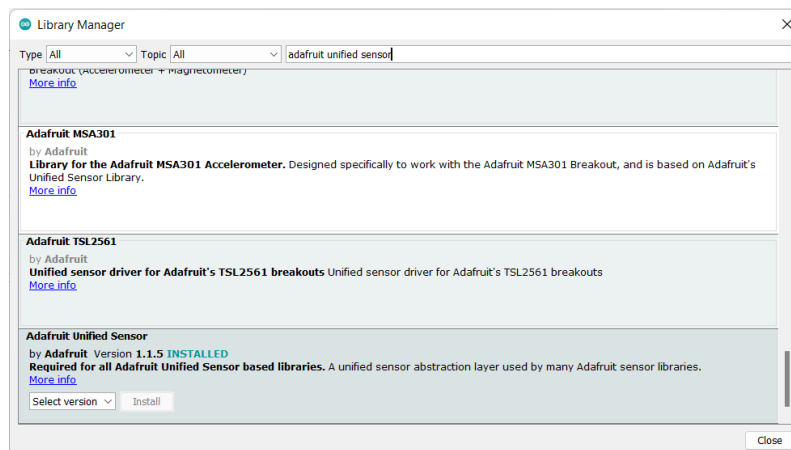
1. Arduino IDE data sketch to ESP32-WROOM-32.

a. Install Adafruit Unified Sensor Library di Arduino IDE.

- Pergi ke sketch > include library > manage libraries



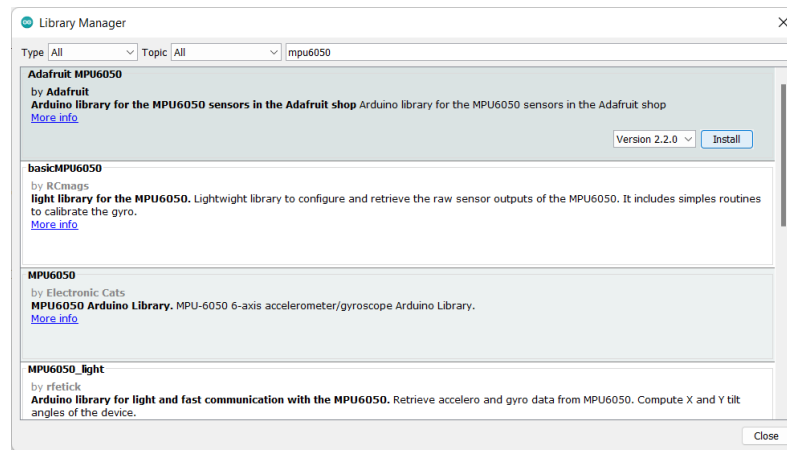
- Cari library “Adafruit Unified Sensor”



b. Install MPU6050 Sensor Library di Arduino IDE.

- Pergi ke sketch > include library > manage libraries
- Cari library “Adafruit MPU6050”





c. Pembuatan program data di Arduino IDE.

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

#define FREQUENCY_HZ    50
#define INTERVAL_MS     (1000 / (FREQUENCY_HZ + 1))

// objeto da classe Adafruit_MPU6050
Adafruit_MPU6050 mpu;
static unsigned long last_interval_ms = 0;

void setup() {
  Serial.begin(115200);
  Serial.println("Classificador de gestos com TinyML");

  // Try to initialize!
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
}
```

```
Serial.println("MPU6050 Found!");  
mpu.setAccelerometerRange(MPU6050_RANGE_8_G);  
Serial.print("Accelerometer range set to: ");
```

```
switch (mpu.getAccelerometerRange()) {  
  case MPU6050_RANGE_2_G:  
    Serial.println("+2G");  
    break;  
  case MPU6050_RANGE_4_G:  
    Serial.println("+4G");  
    break;  
  case MPU6050_RANGE_8_G:  
    Serial.println("+8G");  
    break;  
  case MPU6050_RANGE_16_G:  
    Serial.println("+16G");  
    break;  
}
```

```
mpu.setGyroRange(MPU6050_RANGE_500_DEG);  
Serial.print("Gyro range set to: ");  
switch (mpu.getGyroRange()) {  
  case MPU6050_RANGE_250_DEG:  
    Serial.println("+ 250 deg/s");  
    break;  
  case MPU6050_RANGE_500_DEG:  
    Serial.println("+ 500 deg/s");  
    break;  
  case MPU6050_RANGE_1000_DEG:  
    Serial.println("+ 1000 deg/s");  
    break;  
  case MPU6050_RANGE_2000_DEG:  
    Serial.println("+ 2000 deg/s");
```

```

        break;
    }
    mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
    Serial.print("Filter bandwidth set to: ");
    switch (mpu.getFilterBandwidth()) {
        case MPU6050_BAND_260_HZ:
            Serial.println("260 Hz");
            break;
        case MPU6050_BAND_184_HZ:
            Serial.println("184 Hz");
            break;
        case MPU6050_BAND_94_HZ:
            Serial.println("94 Hz");
            break;
        case MPU6050_BAND_44_HZ:
            Serial.println("44 Hz");
            break;
        case MPU6050_BAND_21_HZ:
            Serial.println("21 Hz");
            break;
        case MPU6050_BAND_10_HZ:
            Serial.println("10 Hz");
            break;
        case MPU6050_BAND_5_HZ:
            Serial.println("5 Hz");
            break;
    }
    Serial.println("");
    delay(100);
}

void loop() {
    sensors_event_t a, g, temp;
    if (millis() > last_interval_ms + INTERVAL_MS) {
        last_interval_ms = millis();
    }
}

```

```

mpu.getEvent(&a, &g, &temp);

Serial.print(a.acceleration.x);

Serial.print(",");

Serial.print(a.acceleration.y);

Serial.print(",");

Serial.println(a.acceleration.z);

}

}

```

- d. Upload data program dari Arduino IDE ke ESP32-WROOM-32.

```

sketch_may30a | Arduino 1.8.13
File Edit Sketch Tools Help

sketch_may30a.g
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

#define FREQUENCY_HZ 50
#define INTERVAL_MS (1000 / (FREQUENCY_HZ + 1))

// object da classe Adafruit_MPU6050
Adafruit_MPU6050 mpu;
static unsigned long last_interval_ms = 0;

void setup() {
  Serial.begin(115200);
  Serial.println("Classificador de gestos com TinyML");
  // Try to initialize!
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");
  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
  Serial.print("Accelerometer range set to: ");

  switch (mpu.getAccelerometerRange()) {
    case MPU6050_RANGE_8_G:
      Serial.println("8-G");
      break;
  }

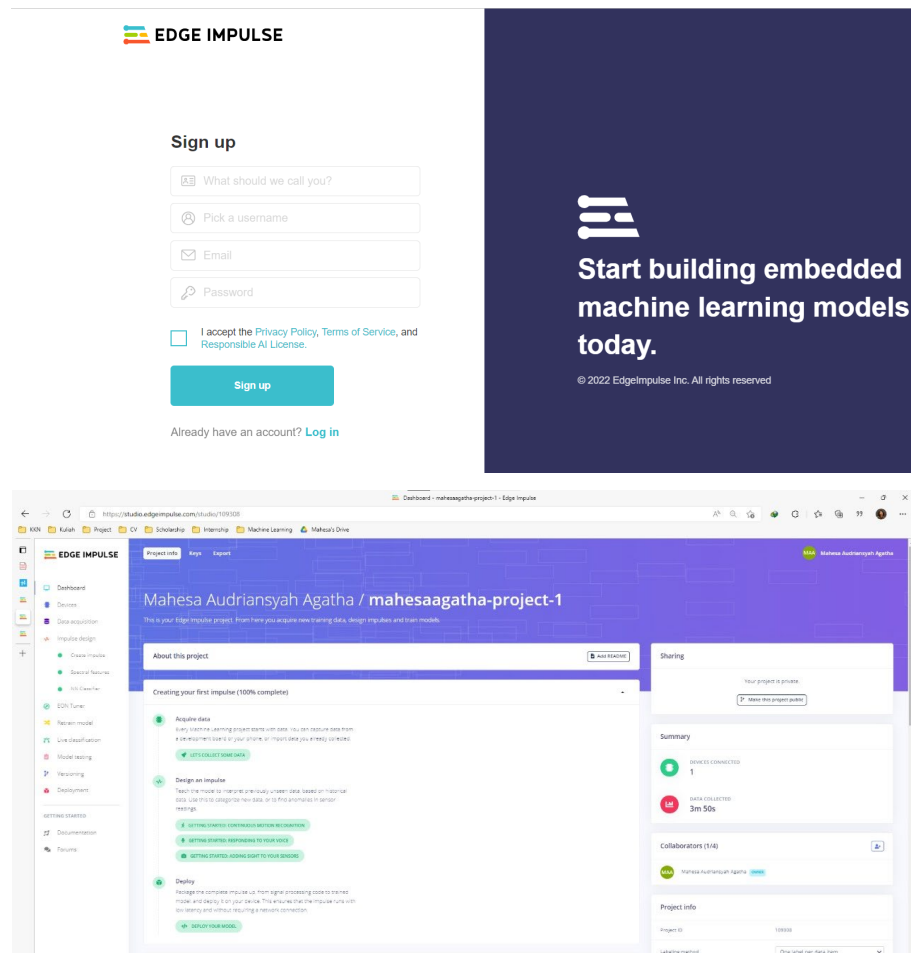
  Serial.println("Done Uploading");
  Uploading 25600 bytes to 4/...
  Writing at 0x00000000... (100 %)
  wrote 25600 bytes (at 0x00000000 in 0.1 seconds (effective 329.0 kbit/s)...)
  Check of data verified.
  Compressed 25600 bytes to 141256...
  Writing at 0x00010000... (11 %)
  Writing at 0x00010001... (22 %)
  Writing at 0x00024e98... (23 %)
  Writing at 0x00024e99... (44 %)
  Writing at 0x00024e9a... (55 %)
  Writing at 0x00024e9b... (66 %)
  Writing at 0x00040000... (77 %)
  Writing at 0x00040001... (88 %)
  Writing at 0x00040002... (100 %)
  wrote 25600 bytes (141256 compressed) at 0x00010000 in 12.0 seconds (effective 140.0 kbit/s)...)
  Check of data verified.
  Uploading...
  Read awaiting via RTS pin...

```

2. Edge Impuls CLI Installation.

```
npm install -g edge-impulse-cli --force
```

3. Membuat akun dan channel board pada edgeimpuls.com.



4. Connect ESP32-WROOM-32 board ke Edge Impuls Web.

- a. Buka CMD lalu ketikkan command dibawah ini.

```
edge-impulse-daemon
```

- b. Hasil Running Command edge-impulse-daemon.

Command ini bertujuan untuk membuat channel baru sekaligus sebagai tanda koneksi port pada esp32.

```

1 Edge Impulse serial daemon v1.1.0
2 ? To which server do you want to connect? (you can override this by setting EI_HC
3 ? What is your user name or e-mail address? jan@edgeimpulse.com
4 ? What is your password? [hidden]
5 Endpoints:
6   Websocket: wss://remote-mgmt.edgeimpulse.com
7   API:       https://studio.edgeimpulse.com
8   Ingestion: https://ingestion.edgeimpulse.com
9
10 [SER] Connecting to /dev/tty.usbmodem401203
11 [SER] Serial is connected, trying to read config...
12 [SER] Retrieved configuration
13 ? To which project do you want to add this device? accelerometer-demo-1
14 Configuring API key in device... OK
15 Configuring HMAC key in device... OK
16 ? What name do you want to give this device? Jan's DISCO-L475VG
17 Setting upload host in device... OK
18 Configuring remote management settings... OK
19 ? WiFi is not connected, do you want to set up a WiFi network now? Yes
20 Scanning WiFi networks... OK
21 ? Select WiFi network SSID: edgeimpulse-office, Security: WPA2 (3), RSSI: -60 dBm
22 ? Enter password for network "edgeimpulse-office" 0624710192
23 Connecting to "edgeimpulse-office"... OK
24 [SER] Device is connected over WiFi to remote management API, no need to run the

```

- c. Connect dan Upload Data dari ESP32-WROOM-32 ke Edge Impulse menggunakan CMD.

edge-impulse-data-forwarder

- d. Hasil Running edge-impulse-data-forwarder.

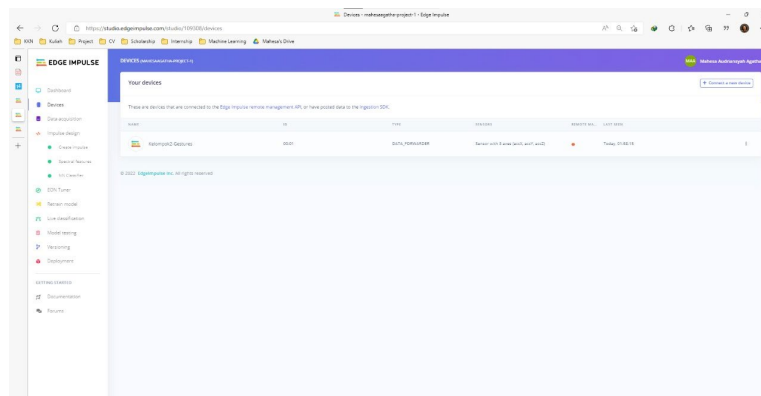
Command data forwarder bertujuan untuk menghubungkan data yang ada pada esp32 agar bisa terbaca pada edge impuls website.

```

1 Edge Impulse data forwarder v1.5.0
2 ? What is your user name or e-mail address (edgeimpulse.com)? jan@edgeimpulse.com
3 ? What is your password? [hidden]
4 Endpoints:
5   Websocket: wss://remote-mgmt.edgeimpulse.com
6   API:       https://studio.edgeimpulse.com
7   Ingestion: https://ingestion.edgeimpulse.com
8
9 [SER] Connecting to /dev/tty.usbmodem401203
10 [SER] Serial is connected
11 [WS ] Connecting to wss://remote-mgmt.edgeimpulse.com
12 [WS ] Connected to wss://remote-mgmt.edgeimpulse.com
13 ? To which project do you want to add this device? accelerometer-demo-1
14 ? 3 sensor axes detected. What do you want to call them? Separate the names with
15 ? What name do you want to give this device? Jan's DISCO-L475VG
16 [WS ] Authenticated

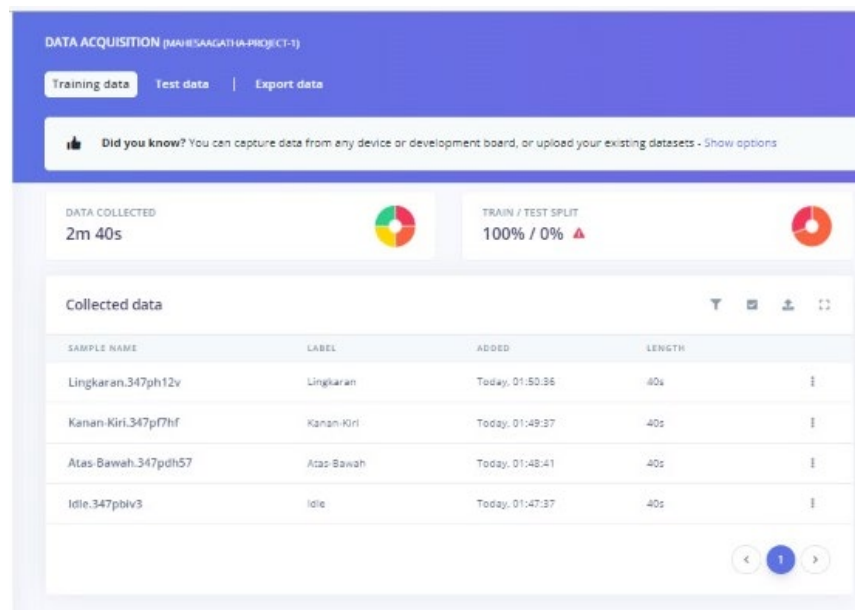
```

- e. Tampilan port yang sudah didefinisikan.



5. Pengambilan sampel data gerakan pada Edge Impulse Website.

- a. Data akuisisi dengan Melakukan 4 gerakan untuk sampel data klasifikasi dengan masing - masing gerakan selama 40 detik.



- b. Pengecekan input sampel data.

Disini input axis kami menggunakan 3 koordinat yang sudah kita atur di dalam arduino IDE code.

### Spectral Analysis

Name

Spectral features

Input axes (3)

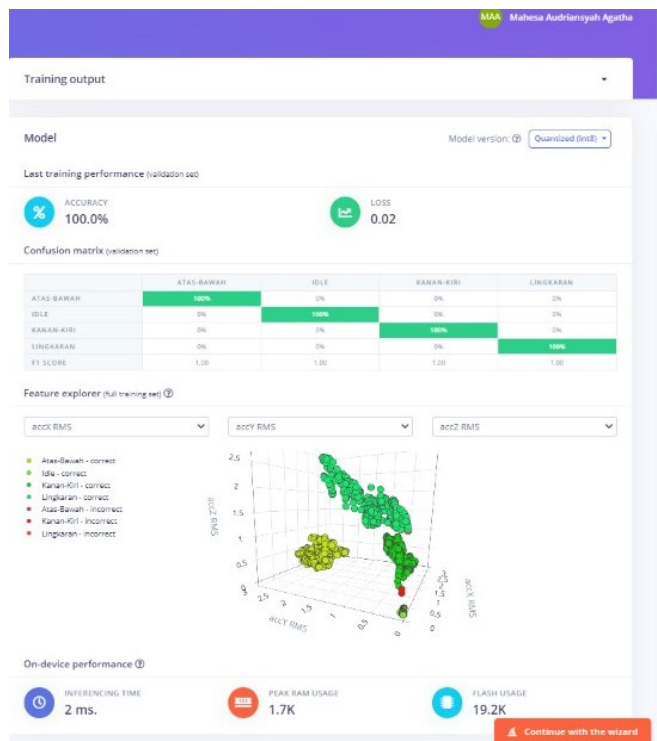
☒ accX

☒ accY

☒ accZ

c. Hasil Training 4 data yang sudah diinput.

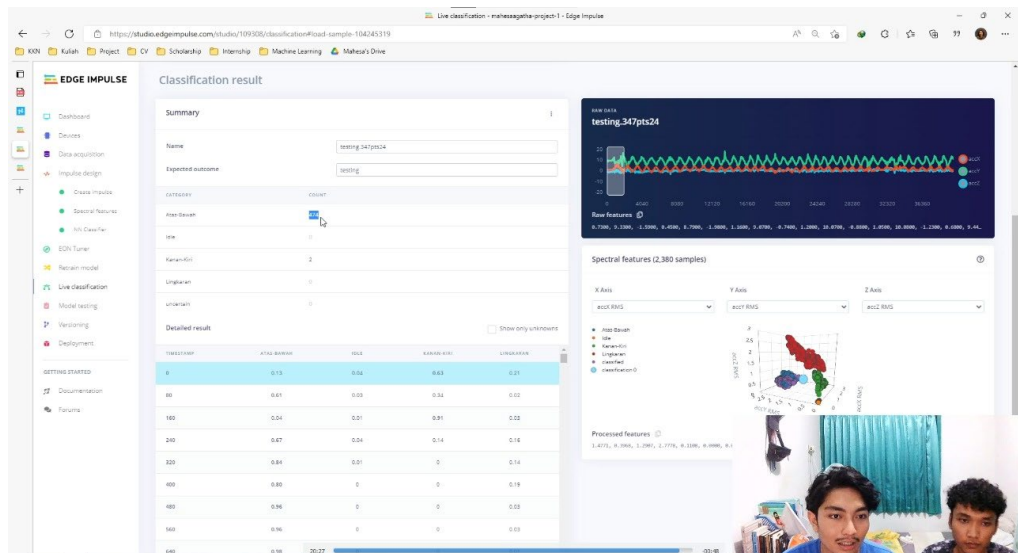
Dari keempat data yang diambil dengan cara gerakan yang berbeda dan dilakukan training menyatakan keakurasian data sebesar 100%.



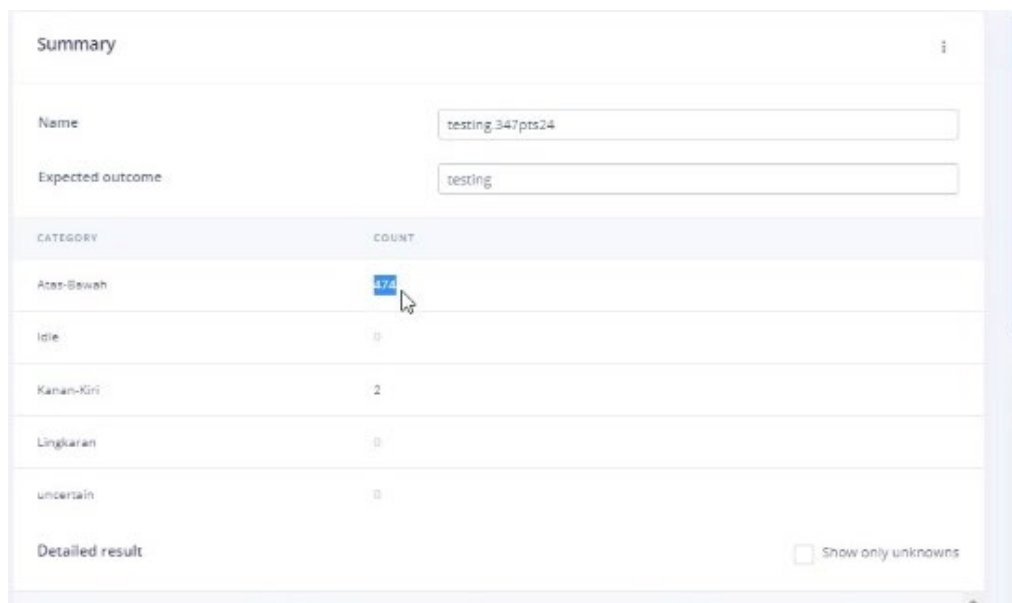
6. Live Classification data testing.

a. Pengambilan data testing dan melakukan pengecekan terhadap gerakan testing sudah sesuai dengan data training kita atau belum.





- b. Data testing menyatakan 474 data terbaca mengacu pada gerakan atas-bawah yang mana sesuai dengan gerakan yang sudah dilakukan.



- c. Selain itu apabila angka testing mendekati nilai 1 maka gerakan dianggap mendekati gerakan training sekaligus mengklasifikasikan gerakan tersebut.

TIMESTAMP	ATAS-BAWAH	IDLE	KANAN-KIRI	LINGKARAN
0	0.13	0.04	0.63	0.21
80	0.61	0.03	0.34	0.02
160	0.04	0.01	0.91	0.03
240	0.67	0.04	0.14	0.16
320	0.84	0.01	0	0.14
400	0.80	0	0	0.19
480	0.96	0	0	0.03
560	0.96	0	0	0.03
640	0.98	0	0	0.01

### **BAB III**

### **PENUTUP**

#### **C. Kesimpulan**

Pada kasus ini mikrokontroller ESP32 dirangkaikan dengan sensor MPU6050. Dengan tujuan sebagai *machine learning* yang mendeteksi gestur gerakan. Tetapi sebelumnya kami menghimpun data untuk dilakukan training sebanyak empat klasifikasi dengan masing-masing selama 40s. Data tersebut diproses sedemikian rupa sehingga menjadi data siap *testing*. Kemudian rangkaian ESP32 dengan sensor MPU6050 dilakukan pengujian gestur gerakan. Didapatkan sebanyak 474 gerakan terdeteksi untuk selanjutnya didapatkan nilai akurasi 100% dengan nilai *loss* sebesar 0.02. Ini menunjukkan hasil proses klasifikasi gerakan yang sangat optimum.

#### **D. Saran**

Kasus *machine learning gesture classification* with Esp32 and TinyML pada kelompok kami masih memiliki kekurangan, yaitu kurangnya penambahan empat buah LED sebagai lampu indikator. LED tersebut akan berfungsi sebagai penanda *data testing* yang diujikan tersebut diklasifikasikan kedalam gestur seperti apa. Dengan warna-warna LED yang berbeda maka dapat menandakan *data testing* sesuai dengan banyaknya klasifikasi *data training*. Dengan ini selanjutnya diharapkan kasus *machine learning gesture classification* bisa menghasilkan data-data yang lebih akurat dan optimum.