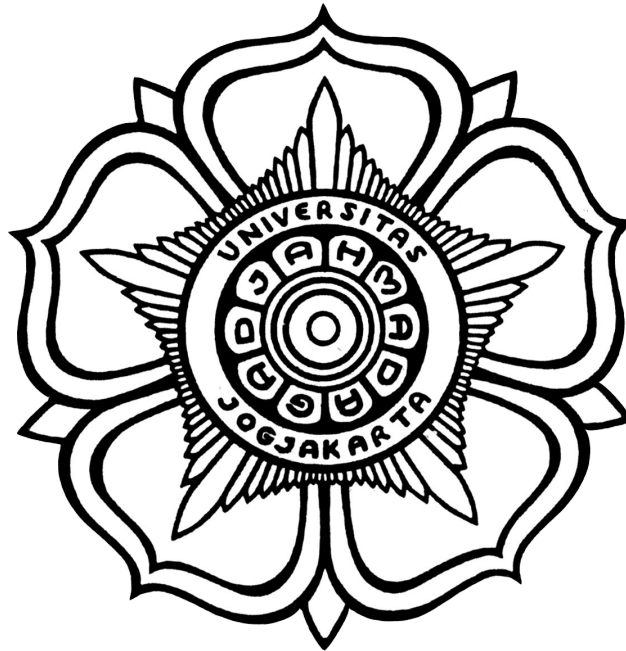


**LAPORAN PRAKTIKUM ELEKTRONIKA MESIN LISTRIK  
DAN TEKNIK KENDALI**

***“Esp32 Publishing Potentiometer Reading to SQL Database using  
Apps Python Flask”***

Dosen Pengampu: Irfan Bahiuddin, ST, M.Phil., Ph.D.



Disusun Oleh:

Kelompok 1

Catur Wardana (19/441194/SV/16546)

Santi Rahayu (19/441209/SV/16561)

Kelas: ARM 2

**TEKNOLOGI REKAYASA MESIN**

**DEPARTEMEN TEKNIK MESIN SEKOLAH VOKASI**

**UNIVERSITAS GADJAH MADA**

**YOGYAKARTA**

**2022**

## **BAB 1. Pengenalan Kasus**

Kasus pada tugas 1 ini adalah Esp32 Publishing Potentiometer Reading to SQL Database using Apps Python Flask.

Pengiriman menggunakan Fungsi dari HTTP POST dari esp32 ke python flask dan data yang diterima harus dimasukkan kedalam database menggunakan CRUD.

## **BAB 2. Komponen yang Digunakan**

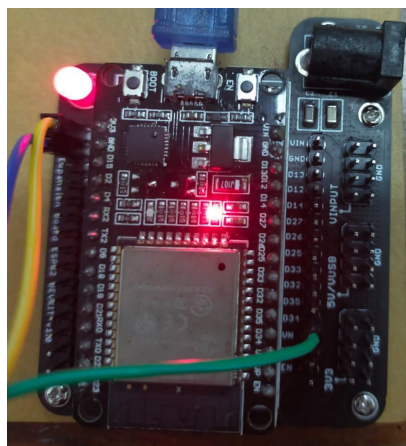
### **2.1 Potensiometer**



Gambar 2.1 Potensiometer

Potensiometer (POT) adalah salah satu jenis resistor yang nilai resistansinya dapat diatur sesuai dengan kebutuhan dari rangkaian elektronika ataupun kebutuhan pemakainya. Potensiometer merupakan Keluarga Resistor yang tergolong dalam Kategori Variable Resistor. Secara struktur, potensiometer terdiri dari 3 kaki Terminal dengan sebuah shaft atau tuas yang berfungsi sebagai pengaturnya (Fransisko, P., 2019).

### **2.2 ESP32**



Gambar 2.2 ESP32

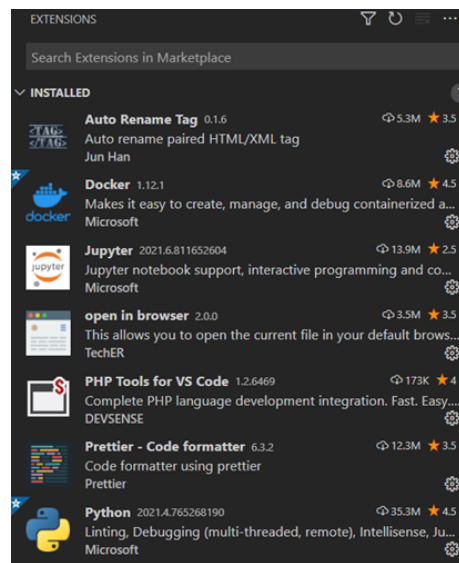
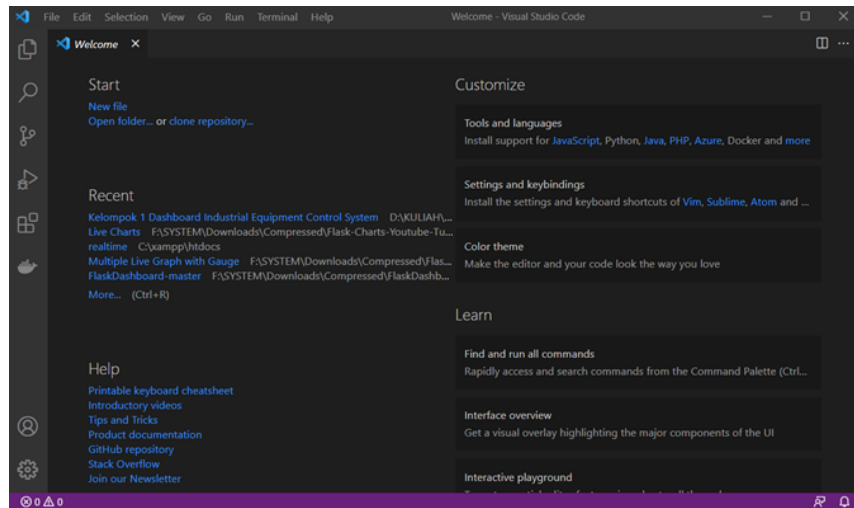
ESP32 adalah nama dari mikrokontroler yang dirancang oleh perusahaan yang berbasis di Shanghai, China yakni Espressif Systems. ESP32 menawarkan solusi jaringan WiFi yang mandiri sebagai jembatan dari mikrokontroler yang ada ke jaringan WiFi. ESP32 menggunakan prosesor dual core yang berjalan di instruksi Xtensa LX16 [3], ESP32 memiliki spesifikasi seperti yang ditampilkan pada tabel 1.

| No  | Atribut            | Detail                     |
|-----|--------------------|----------------------------|
| 1.  | Tegangan           | 3.3 Volt                   |
| 2.  | Prosesor           | Tensilica L108 32 bit      |
| 3.  | Kecepatan Prosesor | Dual 160 Mhz               |
| 4.  | RAM                | 520K                       |
| 5.  | GPIO               | 34                         |
| 6.  | ADC                | 7                          |
| 7.  | Dukungan 802.11    | 11b/g/n/e/i                |
| 8.  | Bluetooth          | BLE (Bluetooth Low Energy) |
| 9.  | SPI                | 3                          |
| 10. | I2C                | 2                          |
| 11. | UART               | 3                          |

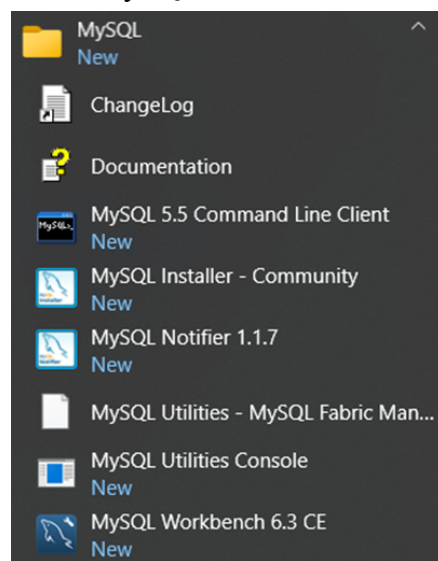
Jika dilihat dari spesifikasi pada tabel maka mikrokontroler ESP32 dapat dijadikan pilihan untuk digunakan pada alat peraga interface mikrokontroler karena mikrokontroler ini memiliki interface yang lengkap, juga memiliki WiFi yang sudah tertanam pada mikrokontroler sehingga tepat untuk digunakan pada alat peraga atau trainer Internet of Things. Pada gambar 1 merupakan pin out dari GPIO pada ESP32 (Kusumah, H. dan Pradana, R.A., 2019).

## 2.3 Persiapan Aplikasi dan Extensi yang akan digunakan

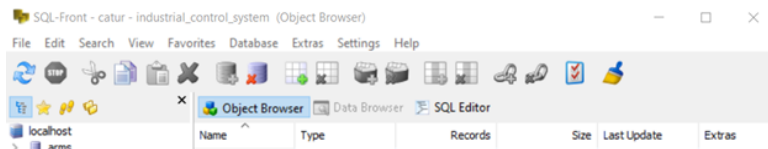
1. Langkah pertama dalam pembuatan project ini adalah membuka aplikasi – aplikasi penunjang pembuatan project seperti Visual Studio Code.



## 2. MySQL Installer



### 3. My-Sql Front



### 4. Python

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\uq>python --version
Python 3.7.9

C:\Users\uq>
```

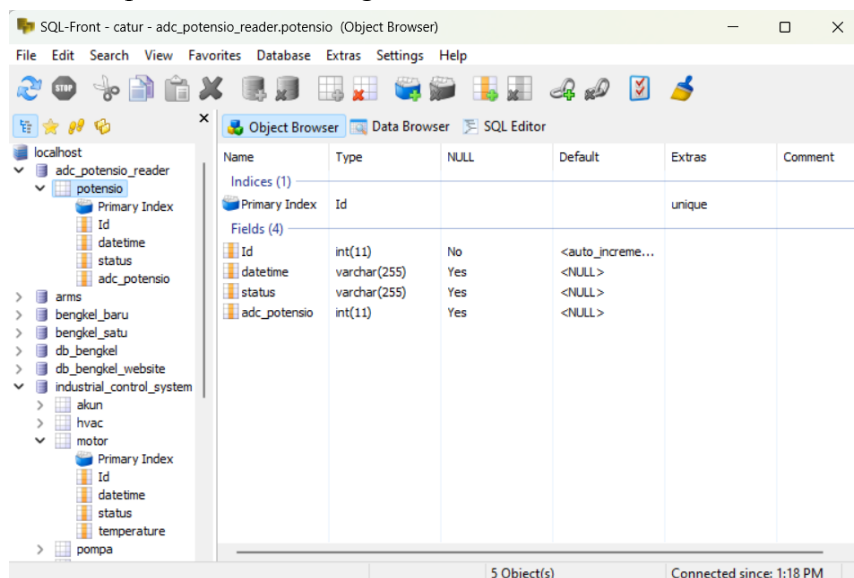
### 5. Extensi Python (Flask, flask\_mysqldb, datetime, time)

```
C:\Users\uq>pip install flask
Collecting flask
  Downloading flask-1.1.2-py2.py3-none-any.whl (94 kB)
    94 kB 59 kB/s
Collecting Jinja2>=2.10.1
  Downloading Jinja2-2.11.3-py2.py3-none-any.whl (125 kB)
    125 kB 57 kB/s
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting click>=5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
    82 kB 78 kB/s
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
    298 kB 91 kB/s
Collecting MarkupSafe>=0.23
  Downloading MarkupSafe-1.1.1-cp37-cp38-win_amd64.whl (16 kB)
Installing collected packages: MarkupSafe, Jinja2, itsdangerous, click, Werkzeug, flask
Successfully installed Jinja2-2.11.3 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 flask-1.1.2 itsdangerous-1.1.0
WARNING: You are using pip version 20.1.1; however, version 21.1 is available.
You should consider upgrading via the 'c:\users\uq\appdata\local\programs\python\python37\python.exe -m pip install --upgrade pip' command.

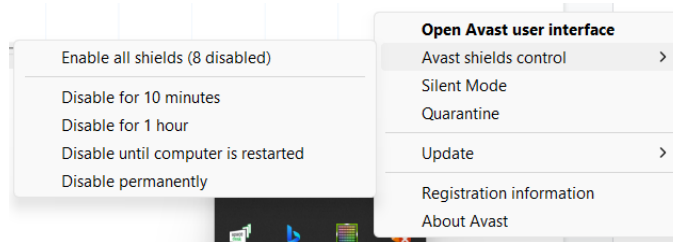
C:\Users\uq>pip install mysql-connector
Requirement already satisfied: mysql-connector in c:\users\uq\appdata\local\programs\python\python37\site-packages (2.2.9)
WARNING: You are using pip version 20.1.1; however, version 21.1 is available.
You should consider upgrading via the 'c:\users\uq\appdata\local\programs\python\python37\python.exe -m pip install --upgrade pip' command.

C:\Users\uq>
```

Selanjutnya adalah membuat database di localhost agar nantinya sistem dapat terkoneksi dengan database secara otomatis.

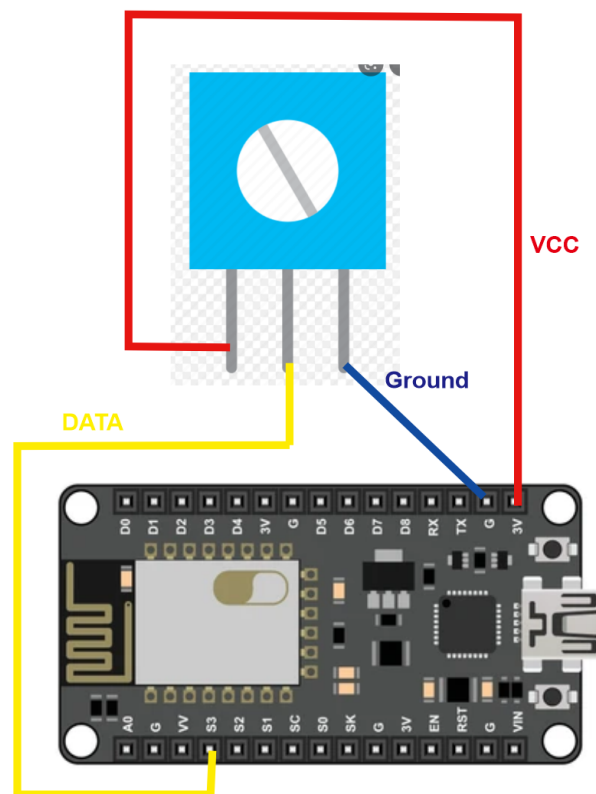


Firewall atau antivirus harus dimatikan



## BAB 3. Pengerjaan

### 3.1 Rangkaian ESP dan Potensio



- Data dari Potensio dibaca pada pin 34
- Kedua kaki di ujung kanan dan kiri potensio adalah pin VCC dan GND

### 3.2 ESP32 CODE

```
// KELOMPOK 1 ARM 2 (CATUR WARDANA, SANTI RAHAYU)
#include <WiFi.h>
#include <HTTPClient.h>
const int potPin = 34;
int potValue = 0;
```

```

const char* ssid = "hghh";
const char* password = "11111111";

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the WiFi network");
}

void loop() {

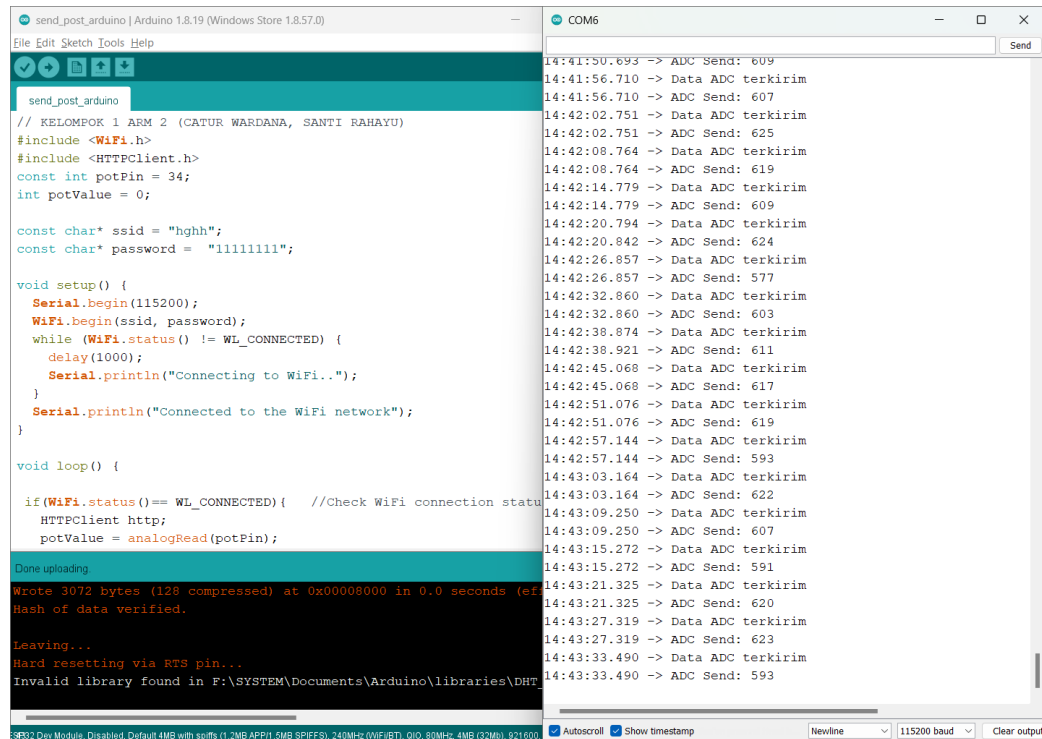
  if(WiFi.status()== WL_CONNECTED){ //Check WiFi connection status
    HTTPClient http;
    potValue = analogRead(potPin);
    int adc;
    adc = potValue;

    http.begin("http://192.168.232.50:5000/post");
    http.addHeader("Content-Type", "text/plain");

    int httpResponseCode = http.POST(String(adc));

    if(httpResponseCode >=0 || httpResponseCode <0);
    {
      Serial.print("Data ADC terkirim");
    }
    Serial.print('\n');
    Serial.print("ADC Send: ");
    Serial.println(adc);
    http.end();
  }
  else{
    Serial.println("Error in WiFi connection");
  }
  delay(1000);
}

```



The screenshot shows the Arduino IDE interface. The left pane displays the sketch 'send\_post\_arduino' with the following code:

```
// KELOMPOK 1 ARM 2 (CATUR WARDANA, SANTI RAHAYU)
#include <WiFi.h>
#include <HTTPClient.h>
const int potPin = 34;
int potValue = 0;

const char* ssid = "hghh";
const char* password = "11111111";

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the WiFi network");
}

void loop() {

  if (WiFi.status() == WL_CONNECTED) { //Check WiFi connection status
    HTTPClient http;
    potValue = analogRead(potPin);
  }
}
```

The right pane shows the serial monitor output for COM6, displaying a series of timestamps and data readings:

```
14:41:50.693 -> ADC Send: 609
14:41:56.710 -> Data ADC terkirim
14:41:56.710 -> ADC Send: 607
14:42:02.751 -> Data ADC terkirim
14:42:02.751 -> ADC Send: 625
14:42:08.764 -> Data ADC terkirim
14:42:08.764 -> ADC Send: 619
14:42:14.779 -> Data ADC terkirim
14:42:14.779 -> ADC Send: 609
14:42:20.794 -> Data ADC terkirim
14:42:20.842 -> ADC Send: 624
14:42:26.857 -> Data ADC terkirim
14:42:26.857 -> ADC Send: 577
14:42:32.860 -> Data ADC terkirim
14:42:32.860 -> ADC Send: 603
14:42:38.874 -> Data ADC terkirim
14:42:38.921 -> ADC Send: 611
14:42:45.068 -> Data ADC terkirim
14:42:45.068 -> ADC Send: 617
14:42:51.076 -> Data ADC terkirim
14:42:51.076 -> ADC Send: 619
14:42:57.144 -> Data ADC terkirim
14:42:57.144 -> ADC Send: 593
14:43:03.164 -> Data ADC terkirim
14:43:03.164 -> ADC Send: 622
14:43:09.250 -> Data ADC terkirim
14:43:09.250 -> ADC Send: 607
14:43:15.272 -> Data ADC terkirim
14:43:15.272 -> ADC Send: 591
14:43:21.325 -> Data ADC terkirim
14:43:21.325 -> ADC Send: 620
14:43:27.319 -> Data ADC terkirim
14:43:27.319 -> ADC Send: 623
14:43:33.490 -> Data ADC terkirim
14:43:33.490 -> ADC Send: 593
```

The status bar at the bottom indicates the board is 'R302 Dev Module, Disabled, Default 4MB with splits (1.2MB APP, 1.2MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32M), 921600' and the serial monitor is set to 'Autoscroll', 'Show timestamp', 'Newline', '115200 baud', and 'Clear output'.

## 3.2 PYTHON FLASK CODE

```
# app.py
```

```
# KELOMPOK 1 ARM 2 (CATUR WARDANA, SANTI RAHAYU)
```

```
from flask import Flask, render_template, request, redirect, url_for, session
```

```
from flask_mysql import MySQL
```

```
import datetime, time
```

```
app = Flask(__name__)
```

```
app.config['MYSQL_HOST'] = 'localhost'
```

```
app.config['MYSQL_USER'] = 'root'
```

```
app.config['MYSQL_PASSWORD'] = '1111'
```

```
app.config['MYSQL_DB'] = 'adc_potensio_reader'
```

```
app.config['MYSQL_CURSORCLASS'] = 'DictCursor'
```

```
mysql = MySQL(app)
```

```
@app.route('/post', methods=["GET", "POST"])
```

```
def post():
```

```
    adc_potensio = (request.data)
```

```
    print(adc_potensio)
```

```
    cur = mysql.connection.cursor()
```



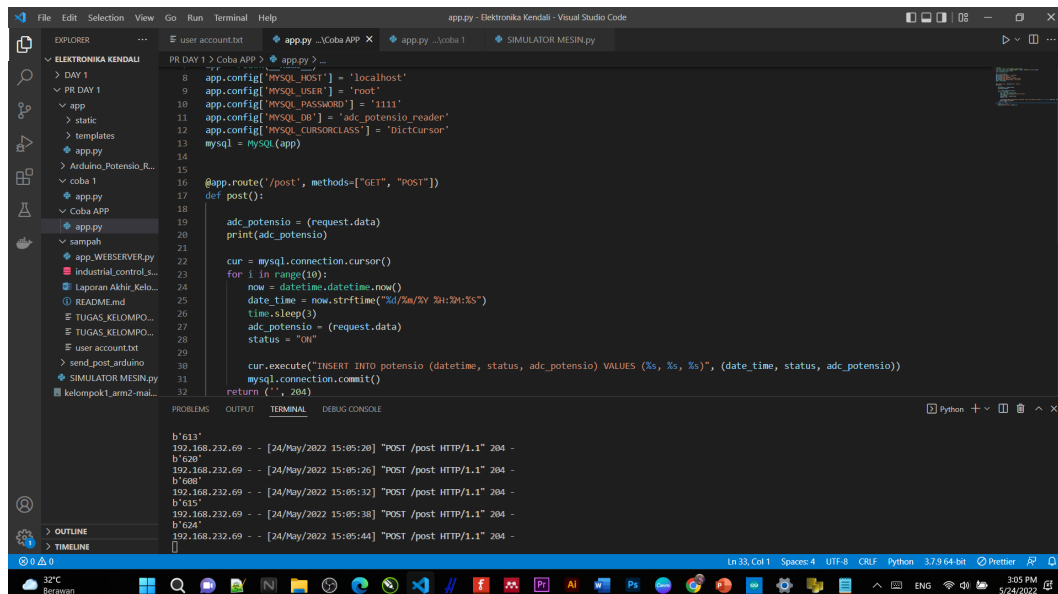
```

for i in range(10):
    now = datetime.datetime.now()
    date_time = now.strftime("%d/%m/%Y %H:%M:%S")
    time.sleep(3)
    adc_potensio = (request.data)
    status = "ON"

    cur.execute("INSERT INTO potensio (datetime, status, adc_potensio)
VALUES (%s, %s, %s)", (date_time, status, adc_potensio))
    mysql.connection.commit()
    return ('', 204)

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)

```

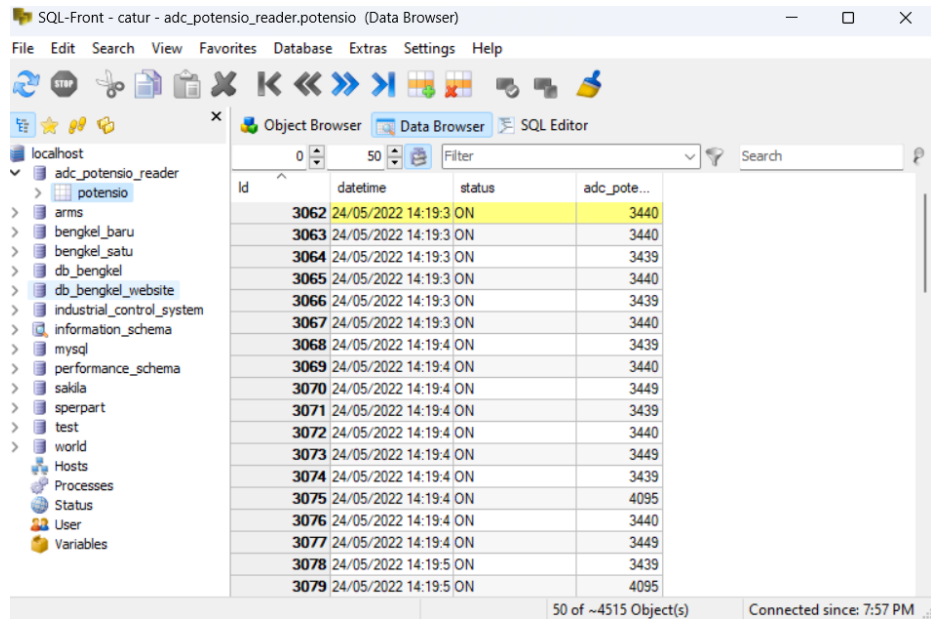


## BAB 4. Kesimpulan

Kesimpulan proses kerja program:

1. Fungsi Code yang ada di ESP adalah mengambil data ADC dari sebuah tegangan yang mengalir ke pin 34.
2. Pembacaan ADC dikirimkan menggunakan HTTP Post ke Python Flask dengan metode local network dengan mengkoneksikan komputer dan esp ke satu router.
3. Python Flask menerima data HTTP Post dari ESP32 menggunakan fungsi Python Flash GET.
4. Python Flask melakukan fungsi CRUD data yang diperoleh dari esp ke MYSQL database menggunakan Fungsi Python POST.

5. Nilai ADC dari ESP berhasil dikirimkan ke Python Flask dan berhasil di masukkan kedalam Database Local.



SQL-Front - catur - adc\_potensio\_reader.potensio (Data Browser)

File Edit Search View Favorites Database Extras Settings Help

Object Browser Data Browser SQL Editor

localhost

- adc\_potensio\_reader
  - potensio
- arms
- bengkel\_baru
- bengkel\_satu
- db\_bengkel
- db\_bengkel\_website
- industrial\_control\_system
- information\_schema
- mysql
- performance\_schema
- sakila
- sperpart
- test
- world
- Hosts
- Processes
- Status
- User
- Variables

| Id   | datetime           | status | adc_pote... |
|------|--------------------|--------|-------------|
| 3062 | 24/05/2022 14:19:3 | ON     | 3440        |
| 3063 | 24/05/2022 14:19:3 | ON     | 3440        |
| 3064 | 24/05/2022 14:19:3 | ON     | 3439        |
| 3065 | 24/05/2022 14:19:3 | ON     | 3440        |
| 3066 | 24/05/2022 14:19:3 | ON     | 3439        |
| 3067 | 24/05/2022 14:19:3 | ON     | 3440        |
| 3068 | 24/05/2022 14:19:4 | ON     | 3439        |
| 3069 | 24/05/2022 14:19:4 | ON     | 3440        |
| 3070 | 24/05/2022 14:19:4 | ON     | 3449        |
| 3071 | 24/05/2022 14:19:4 | ON     | 3439        |
| 3072 | 24/05/2022 14:19:4 | ON     | 3440        |
| 3073 | 24/05/2022 14:19:4 | ON     | 3449        |
| 3074 | 24/05/2022 14:19:4 | ON     | 3439        |
| 3075 | 24/05/2022 14:19:4 | ON     | 4095        |
| 3076 | 24/05/2022 14:19:4 | ON     | 3440        |
| 3077 | 24/05/2022 14:19:4 | ON     | 3449        |
| 3078 | 24/05/2022 14:19:5 | ON     | 3439        |
| 3079 | 24/05/2022 14:19:5 | ON     | 4095        |

50 of ~4515 Object(s) Connected since: 7:57 PM

## Daftar Pustaka

Fransisko, P., 2019. Media Promosi Elektronik untuk Produk-Produk di Supermarket Menggunakan Arduino Nano. Jurnal Sistem Cerdas dan Rekayasa (JSCR), 1(1).

Kusumah, H. and Pradana, R.A., 2019. Penerapan Trainer Interfacing Mikrokontroler Dan Internet of Things Berbasis Esp32 Pada Mata Kuliah Interfacing. Journal Cerita, 5(2), pp.120-134.