

LAMPIRAN

Lampiran. 1 Progam main.cpp (esp32)

```
#include <Arduino.h>
#include <WiFi.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <HTTPClient.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

HardwareSerial fromUno(2);

char c;
String dataIn;

const char *ssid = "KARAN";
const char *password = "12345679";

HTTPClient http;
String url = "http://192.168.20.45:3001/";

void displayData(String data)
{
    display.clearDisplay();

    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0, 10);
    display.println(data);
    display.display();
}

String ifElse(String data, String name, String stat)
{
    if (data != " NAN")
    {
        Serial.println(name + " : " + data + " " + stat);
        return name + " : " + data + " " + stat + "\n";
    }
    else
    {

```

```

    Serial.println("Error get " + name);
    return "Error get " + name+ '\n';
}
}

void seperator(String data)
{
    char charArray[data.length() + 1];          // Create a char array to store the string
    data.toCharArray(charArray, sizeof(charArray)); // Convert String to char array

    char *token = strtok(charArray, ","); // Tokenize the string based on comma delimiter
    int count = 0;

    String dataForDisplay = "";
    String jsonSend="{\"fasa\":\",";
    while (token != NULL)
    {
        switch (count)
        {
            case 1:
                dataForDisplay += String(token) + "\n";
                Serial.println(String(token));
                jsonSend += "\""+ String(token) + "\", ";
                break;
            case 2:
                dataForDisplay += ifElse(String(token), "Voltage", "V");
                jsonSend += "\"voltage\": \""+ String(token) + "\", ";
                break;
            case 3:
                dataForDisplay += ifElse(String(token), "Current", "A");
                jsonSend += "\"current\": \""+ String(token) + "\", ";
                break;
            case 4:
                dataForDisplay += ifElse(String(token), "Power", "W");
                jsonSend += "\"power\": \""+ String(token) + "\", ";
                break;
            case 5:
                dataForDisplay += ifElse(String(token), "Energy", "kWh");
                jsonSend += "\"energy\": \""+ String(token) + "\", ";
                break;
            case 6:
                dataForDisplay += ifElse(String(token), "Power Factor", "pf");
                jsonSend += "\"pF\": \""+ String(token) + "\"";
                break;
            default:
                dataForDisplay += ifElse(String(token), "Wifi", "");
                break;
        }
    }
}

```

```

    }
    token = strtok(NULL, ","); // Get the next token
    count++;
}
jsonSend += "}";

displayData(dataForDisplay);
Serial.println(jsonSend);
http.begin(url);
http.addHeader("Content-Type", "application/json");
int httpResponseCode = http.POST(jsonSend);

if (httpResponseCode > 0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    String payload = http.getString();
    Serial.println(payload);
}else{
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}
}
}

void setup()
{
    Serial.begin(19200);
    fromUno.begin(9600); // ini dari esp32
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
    { // Address 0x3D for 128x64
        Serial.println(F("SSD1306 allocation failed"));
        for (;;)
            ;
    }

    // Connect to WiFi
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(1000);
        Serial.println("Connecting to WiFi...\n");
        fromUno.print("Connecting to WiFi...\n");
        displayData("Connecting to WiFi...");
    }
    Serial.println("Connected to WiFi\n");
    fromUno.print("Connected to WiFi\n");
}

```

```

displayData("Connected to WiFi");
}

void loop()
{
  String wifiStatus = "";
  if (WiFi.status() == WL_CONNECTED)
  {
    Serial.println("WiFi Connected\n");
    fromUno.print("WiFi Connected\n");
    wifiStatus += "Connected";
    // delay(2000);
  }
  else
  {
    // toUnoWifiState.print("WiFi Disconnected\n");
    Serial.println("WiFi Disconnected\n");
    fromUno.print("WiFi Disconnected\n");
    wifiStatus += "Disconnected";
    // delay(2000);
  }

  while (fromUno.available())
  {
    c = fromUno.read();
    if (c != '\n')
    {
      dataIn += c;
    }
    else
    {
      break;
    }
  }

  if (c == '\n')
  {
    // Serial.println(dataIn);
    seperator(wifiStatus + "," + dataIn);
    // displayData(wifiStatus);
    c = 0;
    dataIn = "";
  }

  delay(2000);
}

```

Lampiran. 2 Program main.cpp (arduino uno)

```
#include <Arduino.h>
#include <WiFi.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <HTTPClient.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

HardwareSerial fromUno(2);

char c;
String dataIn;

const char *ssid = "KARAN";
const char *password = "12345679";

HTTPClient http;
String url = "http://192.168.20.45:3001/";

void displayData(String data)
{
    display.clearDisplay();

    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0, 10);
    display.println(data);
    display.display();
}

String ifElse(String data, String name, String stat)
{
    if (data != " NAN")
    {
        Serial.println(name + " : " + data + " " + stat);
        return name + " : " + data + " " + stat + "\n";
    }
    else
    {
        Serial.println("Error get " + name);
        return "Error get " + name+ "\n";
    }
}
```

```

    }
}

void seperator(String data)
{
    char charArray[data.length() + 1];          // Create a char array to store the string
    data.toCharArray(charArray, sizeof(charArray)); // Convert String to char array

    char *token = strtok(charArray, ","); // Tokenize the string based on comma delimiter
    int count = 0;

    String dataForDisplay = "";
    String jsonSend = "{\"fasa\"";
    while (token != NULL)
    {
        switch (count)
        {
            case 1:
                dataForDisplay += String(token) + "\n";
                Serial.println(String(token));
                jsonSend += "\"" + String(token) + "\", ";
                break;
            case 2:
                dataForDisplay += ifElse(String(token), "Voltage", "V");
                jsonSend += "\"voltage\": \"" + String(token) + "\", ";
                break;
            case 3:
                dataForDisplay += ifElse(String(token), "Current", "A");
                jsonSend += "\"current\": \"" + String(token) + "\", ";
                break;
            case 4:
                dataForDisplay += ifElse(String(token), "Power", "W");
                jsonSend += "\"power\": \"" + String(token) + "\", ";
                break;
            case 5:
                dataForDisplay += ifElse(String(token), "Energy", "kWh");
                jsonSend += "\"energy\": \"" + String(token) + "\", ";
                break;
            case 6:
                dataForDisplay += ifElse(String(token), "Power Factor", "pf");
                jsonSend += "\"pf\": \"" + String(token) + "\"";
                break;
            default:
                dataForDisplay += ifElse(String(token), "Wifi", "");
                break;
        }
        token = strtok(NULL, ","); // Get the next token
    }
}

```

```

    count++;
}
jsonSend += "}";

displayData(dataForDisplay);
Serial.println(jsonSend);
http.begin(url);
http.addHeader("Content-Type", "application/json");
int httpResponseCode = http.POST(jsonSend);

if (httpResponseCode > 0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    String payload = http.getString();
    Serial.println(payload);
}else{
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}

}

}

void setup()
{
    Serial.begin(19200);
    fromUno.begin(9600); // ini dari esp32
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
    { // Address 0x3D for 128x64
        Serial.println(F("SSD1306 allocation failed"));
        for (;;)
            ;
    }

    // Connect to WiFi
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(1000);
        Serial.println("Connecting to WiFi...\n");
        fromUno.print("Connecting to WiFi...\n");
        displayData("Connecting to WiFi...");
    }
    Serial.println("Connected to WiFi\n");
    fromUno.print("Connected to WiFi\n");
    displayData("Connected to WiFi");
}

```

```

void loop()
{
  String wifiStatus = "";
  if (WiFi.status() == WL_CONNECTED)
  {
    Serial.println("WiFi Connected\n");
    fromUno.print("WiFi Connected\n");
    wifiStatus += "Connected";
    // delay(2000);
  }
  else
  {
    // toUnoWifiState.print("WiFi Disconnected\n");
    Serial.println("WiFi Disconnected\n");
    fromUno.print("WiFi Disconnected\n");
    wifiStatus += "Disconnected";
    // delay(2000);
  }

  while (fromUno.available())
  {
    c = fromUno.read();
    if (c != '\n')
    {
      dataIn += c;
    }
    else
    {
      break;
    }
  }

  if (c == '\n')
  {
    // Serial.println(dataIn);
    seperator(wifiStatus + "," + dataIn);
    // displayData(wifiStatus);
    c = 0;
    dataIn = "";
  }

  delay(2000);
}

```

Lampiran. 3 Index JS

```

const express = require('express');
const http = require('http');
const cors = require('cors');
const dotenv = require('dotenv');
const socket = require('./socket');
const app = express();
const server = http.createServer(app);
const io = socket.init(server);
const iosend = socket.getIO();

dotenv.config();

app.use(express.json());
app.use(express.urlencoded({ extended: true }));
// app.use(fileUpload());
app.options('*', cors());
app.use(cors());

// Handle POST requests to '/post'
app.post('/', (req, res) => {
  // Access the POST data sent in the request body
  const postData = req.body;

  // Process the received data
  console.log('Received data:', postData);
  iosend.emit('datanya', postData);

  // Respond with a simple message
  res.send('Received your POST request!');
});

// app error handler
app.use((err, req, res, next) => {
  console.log(err);
  res.status(500).send('Something broke!');
});

io.on('connection', (socket) => {
  let userID = socket.id;
  console.log('A user connected: ' + userID);

  socket.on('scan_dia', (data) => {
    console.log('Received scan_dia event: ' + data);
  });
});

```

```
socket.on('disconnect', () => {
  console.log("User disconnected: " + userID);
});
});

module.exports = {
  app,
  server,
  io
};

const port = process.env.PORT || 3001;

// Start the server
server.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```



KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : Muh. Fathur R. Alimuddin	Pembimbing I : Muhammad Basri, ST., MT.
NIM : 217180064	Pembimbing II : Alauddin Y., ST., M.Kom.
Judul Skripsi : Perancangan Prototype Sistem Monitoring Gardu Distribusi berbasis Mikrokontroler	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1		Konsultasi 1 - Bab II tinjauan pustaka.	 19/8/23
Konsultasi 2		Konsultasi 2 - Bab III bagian Akhir - Alat dan bahan yg digunakan - pustaka	 19/8/23
Konsultasi 3		Konsultasi 3 - Sensor arus 3 - modul cv - display - komputer (maka) tampilan	 20/1/24
Konsultasi 4		Konsultasi 4 - narasi flowchart - gbr desain papan komponen elektronik beserta hobengannya	 24/2/24
Konsultasi 5		Konsultasi 5 	 24/2/24

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna merah muda dan dicetak timbal balik

Lanjutan ...

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 6		Konsultasi 6 1 - Abstrak diperbaiki 2 - Rumusan lengkap 3 - disubikan 4 - bab 3, 4, 5. 5 - Data logger.	16/05/2021
Konsultasi 7		Konsultasi 7	
Konsultasi 8		Konsultasi 8	
Konsultasi 9 <i>acc Haril</i>		Konsultasi 9 <i>Wahid Nurzi</i>	22/05/2021
Konsultasi 10		Konsultasi 10 	25/05/2021

Mengetahui
Ketua Program Studi

Asrul, ST., MT.
NBM. 986 836

Parepare,

Mahasiswa

Muh. Fathur R. Alimuddin
NIM. 217180064

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna merah muda dan dicetak timbal balik



KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : Muh. Fathur R. Alimuddin	Pembimbing I : Muhammad Basri, ST., MT.
NIM : 217180064	Pembimbing II : Alauddin Y., ST., M.Kom.
Judul Skripsi : Perancangan Prototype Sistem Monitoring Gardu Distribusi berbasis Mikrocontroller	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1		Konsultasi 1	
Konsultasi 2		Konsultasi 2	
Konsultasi 3		Konsultasi 3	
Konsultasi 4 hasri	Basri	Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna merah muda dan dicetak timbal balik

Lanjutan...

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 6		Konsultasi 6	
Konsultasi 7		Konsultasi 7	
Konsultasi 8		Konsultasi 8	
Konsultasi 9		Konsultasi 9	
Konsultasi 10		Konsultasi 10	



Parepare, 3/8/2024.

Mahasiswa

Muh. Fathur R. Alimuddin
NIM. 217180064

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna merah muda dan dicetak timbal balik