Lampiran 1

```
#include <ThingerESP32.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "arduino_secrets.h"
#define MQ2 32
#define MQ9 34
#define MQ135 35
#define LEDFAN 17
#define LEDPIN 13
#define buzzerPin 12
#define LEDMODE 16
const int relayPin = 14; // Pin untuk mengontrol relay
const int modeButtonPin = 15; // Pin untuk tombol mode
#define LCD_ADDRESS 0x27 // Alamat I2C LCD 16x2
#define LCD_COLS 16
#define LCD_ROWS 2
#define THINGER_SERIAL_DEBUG
LiquidCrystal_I2C LCD(LCD_ADDRESS, LCD_COLS, LCD_ROWS);
ThingerESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
// Konstanta untuk kalibrasi sensor
const float VOLTAGE_REFERENCE = 3.3;
const float MQ2_SENSITIVITY = 0.8;  // Sensitivitas MQ-2
const float MQ9_SENSITIVITY = 0.9;  // Sensitivitas MQ-9
const float MQ135_SENSITIVITY = 0.8;  // Sensitivitas MQ-135
bool wifiConnected = false; // Status koneksi WiFi
bool relayStatus = false; // Status relay awal: mati (off)
bool automaticMode = false; // Mode awal: Manual
```

```
bool previousModeButtonState = false; // Status sebelumnya tombol mode

void setup() {
  Serial.begin(115200);
  pinMode(LEDPIN, OUTPUT);
  pinMode(LEDFAN, OUTPUT);
  pinMode(LEDMODE, OUTPUT);
  pinMode(buzzerPin, OUTPUT); // Mengatur pin buzzer sebagai OUTPUT
  pinMode(relayPin, OUTPUT); // Mengatur pin relay sebagai OUTPUT
  pinMode(modeButtonPin, INPUT_PULLUP); // Mengatur pin tombol mode
sebagai INPUT_PULLUP
  thing.add_wifi(SSID, SSID_PASSWORD);
  LCD.begin();
  LCD.setBacklight(HIGH);
  thing["relay"] << [](pson & in) {
    if (in.is_empty()) {
      in = relayStatus;
    } else {
      relayStatus = in;
      digitalWrite(relayPin, relayStatus ? HIGH : LOW);
    }
  };
}

void loop() {
  if (thing.is_connected()) {
    wifiConnected = true;
    digitalWrite(LEDPIN, HIGH);
```

```
} else {
  wifiConnected = false;
  digitalWrite(LEDPIN, LOW);
}
int gasValue2 = analogRead(MQ2); //sensor mq-2
// Konversi sensor MQ-2 ke PPM
float sensorVoltage2 = (gasValue2 / 4095.0) * VOLTAGE_REFERENCE;
float PPM2 = (sensorVoltage2 / MQ2_SENSITIVITY) * 1000.0;
thing["MQ2"] >> [](pson & out) {
  int gasValue2 = analogRead(MQ2);
  float sensorVoltage2 = (gasValue2 / 4095.0) * VOLTAGE_REFERENCE;
  float PPM2 = (sensorVoltage2 / MQ2_SENSITIVITY) * 1000.0;
  out["PPM2"] = PPM2;
};
// Tampilkan nilai sensor MQ-2 ke LCD
LCD.setCursor(0, 0);
if (PPM2 <= 500) {
  LCD.print("Butana: Aman    "); // Tampilkan keterangan "Aman"
} else {
  LCD.print("Butana: Bahaya  "); // Tampilkan keterangan "Bahaya"
}
LCD.setCursor(0, 1);
LCD.print("PPM: ");
LCD.print(PPM2);
delay(2000); // Tampilkan setiap nilai sensor selama 2 detik
LCD.clear();
int sensorValue9 = analogRead(MQ9); //sensor mq-9
// Konversi sensor MQ-9 ke PPM
```

```
float sensorVoltage9 = (sensorValue9 / 4095.0) * VOLTAGE_REFERENCE;
float PPM9 = (sensorVoltage9 / MQ9_SENSITIVITY) * 1000.0;
thing["MQ9"] >> [](pson & out) {
  int sensorValue9 = analogRead(MQ9);
  float sensorVoltage9 = (sensorValue9 / 4095.0) * VOLTAGE_REFERENCE;
  float PPM9 = (sensorVoltage9 / MQ9_SENSITIVITY) * 1000.0;
  out["PPM9"] = PPM9;
};
// Tampilkan nilai sensor MQ-9 ke LCD
LCD.setCursor(0, 0);
if (PPM9 <= 500) {
  LCD.print("Smoke: Aman    "); // Tampilkan keterangan "Aman"
} else {
  LCD.print("Smoke: Bahaya  "); // Tampilkan keterangan "Bahaya"
}
LCD.setCursor(0, 1);
LCD.print("PPM: ");
LCD.print(PPM9);
delay(2000); // Tampilkan setiap nilai sensor selama 2 detik
LCD.clear();
int value135 = analogRead(MQ135); //sensor mq-135
// Konversi sensor MQ-135 ke PPM
float sensorVoltage135 = (value135 / 4095.0) * VOLTAGE_REFERENCE;
float PPM135 = (sensorVoltage135 / MQ135_SENSITIVITY) * 1000.0;
thing["MQ135"] >> [](pson & out) {
  int value135 = analogRead(MQ135);
  float sensorVoltage135 = (value135 / 4095.0) * VOLTAGE_REFERENCE;
  float PPM135 = (sensorVoltage135 / MQ135_SENSITIVITY) * 1000.0;
```

```
  out["PPM135"] = PPM135;
 };
 LCD.setCursor(0, 0);
 if (PPM135 <= 500) {
  LCD.print("CO2: Aman    "); // Tampilkan keterangan "Aman"
 } else {
  LCD.print("CO2: Bahaya  "); // Tampilkan keterangan "Bahaya"
 }
 LCD.setCursor(0, 1);
 LCD.print("PPM: ");
 LCD.print(PPM135);
 delay(2000); // Tampilkan setiap nilai sensor selama 2 detik
 LCD.clear();
 // Baca status tombol mode
 bool modeButtonState = digitalRead(modeButtonPin);
 // Jika tombol mode ditekan (status LOW) dan status sebelumnya adalah HIGH,
// toggle antara mode otomatis dan mode semi otomatis
if (modeButtonState == LOW && previousModeButtonState == HIGH) {
 automaticMode = !automaticMode;
 // Ketika mode berubah dari otomatis ke manual dan relay sedang menyala,
matikan relay
 if (!automaticMode && relayStatus) {
  relayStatus = false;
  digitalWrite(relayPin, LOW);
 }
}
 // Simpan status tombol mode saat ini untuk digunakan pada iterasi berikutnya
 previousModeButtonState = modeButtonState;
```

```
// Mode Otomatis
if (automaticMode) {
  digitalWrite(LEDMODE, HIGH);
  if (PPM2 > 500 || PPM9 > 500 || PPM135 > 500) {
    digitalWrite(buzzerPin, HIGH);
    digitalWrite (relayPin, HIGH);
    digitalWrite(LEDFAN, HIGH);
    LCD.setCursor(0, 0);
    LCD.print("Fan Aktif ");
    LCD.setCursor(0, 1);
    LCD.print("Mode: Otomatis");
    delay(4000);
    LCD.clear();
  } else {
    digitalWrite(buzzerPin, LOW);
    digitalWrite(relayPin, LOW);
    digitalWrite(LEDFAN, LOW);
    LCD.setCursor(0, 0);
    LCD.print("Fan Nonaktif ");
    LCD.setCursor(0, 1);
    LCD.print("Mode: Otomatis");
    delay (4000);
    LCD.clear();
  }
}
// Mode manual
else {
  digitalWrite(relayPin, LOW);
```

```
   digitalWrite(LEDMODE, LOW);
   digitalWrite(buzzerPin, LOW);
   digitalWrite(LEDFAN, LOW);
   // Tampilkan status relay pada LCD
   LCD.setCursor(0, 0);
   LCD.print("Fan ");
   if (relayStatus) {
     LCD.print("Aktif ");
     digitalWrite(LEDFAN, HIGH);
   } else {
     LCD.print("Nonaktif");
     digitalWrite(LEDFAN, LOW);
   }
   LCD.setCursor(0, 1);
   LCD.print("Mode: Manual");
   delay(2000);
   LCD.clear();
   thing.handle();
  }
  digitalWrite(LEDPIN, wifiConnected ? HIGH : LOW);
}
```