

Lampiran 3 Source Code Arduino Ke ESP32

```
#include <ESP32Servo.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <DHT.h>

#include <WiFi.h>

#include <HTTPClient.h>

const char* ssid = "66666";

const char* password = "12345678";

const String serverName = "https://rizkymmaulanaa.tif-  
skripsi1.my.id/kirimdata.php";

// Paket 1

const int pirPin = 26; // Pin sensor PIR terhubung

const int relayPin = 18; // Pin relay lampu

// Paket 2

const int listrikUtama = 23; // Pin relay listrik utama

const int buzzer = 12; // Pin relay listrik utama

const int gas = 33;

const int servoPin = 4;

Servo myservo;

const int GAS_THRESHOLD = 2000;

// Paket 3

LiquidCrystal_I2C lcd(0x27, 16, 2);

#define DHTPIN 19 // Pin data DHT22

#define DHTTYPE DHT22 // Jenis sensor DHT22

DHT dht(DHTPIN, DHTTYPE); // Setup pin dan jenis DHT ke  
library
```

```

float humi, suhu;

const int kipas = 13;

// Variabel daya listrik

float totalDaya = 0.0;

// Variabel untuk melacak energi dalam kWh

float totalEnergy = 0.0;

// Variabel untuk melacak waktu

unsigned long previousMillis = 0;

const unsigned long interval = 5000; // Interval dalam
milidetik (5 detik)

// Timer untuk lampu

unsigned long lampuTimer = 0;

const unsigned long lampuDelay = 5000; // Menjaga lampu tetap
nyala selama 5 detik setelah deteksi terakhir

// Variabel untuk timer gas

unsigned long gasDetectTime = 0;

const unsigned long gasDelay = 15000; // Delay 15 detik
sebelum mematikan listrik utama

// Variabel untuk debounce PIR

unsigned long lastPirMillis = 0;

const unsigned long pirDebounceDelay = 200; // Delay debounce
PIR 200ms

int displayState = 0; // Variabel untuk melacak tampilan saat
ini

void sendData(String tempVal, String pirStateVal, String
gasValueVal, String totalEnergyVal) {

    if (WiFi.status() == WL_CONNECTED) {

        HTTPClient http;

```

```

http.begin(serverName);

    http.addHeader("Content-Type", "application/x-www-form-
urlencoded");

    String httpRequestData = "suhu=" + tempVal + "&gerak=" +
pirStateVal + "&gas=" + gasValueVal + "&daya=" +
totalEnergyVal;

    Serial.println("Sending data: " + httpRequestData); //
Debugging

    int httpResponseCode = http.POST(httpRequestData);

    if (httpResponseCode > 0) {

        String response = http.getString();

        Serial.println(httpResponseCode);

        Serial.println(response);

    } else {

        Serial.print("Error on sending POST: "); ★

        Serial.println(httpResponseCode);

    }

    http.end();

} else {

    Serial.println("WiFi Disconnected");

}

}

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 WiFi");

Serial.println(WiFi.localIP());

// Paket 1

pinMode(pirPin, INPUT_PULLDOWN); // Set pin sensor PIR
sebagai input dengan pull-down

pinMode(relayPin, OUTPUT); // Set pin relay sebagai
output

digitalWrite(relayPin, LOW); // Pastikan relay dalam
kondisi mati saat awal

// Paket 2

pinMode(listrikUtama, OUTPUT); // Set pin relay sebagai
output

pinMode(buzzer, OUTPUT); // Set pin relay sebagai output

pinMode(kipas, OUTPUT); // Set pin kipas sebagai output

digitalWrite(relayPin, LOW); // Pastikan relay dalam kondisi
mati saat awal

digitalWrite(listrikUtama, HIGH);

myservo.attach(servoPin);

myservo.write(0);

// Paket 3

dht.begin();

lcd.init();

lcd.backlight();

}

void loop() {

    unsigned long currentMillis = millis();

```

```

Serial.println("Connecting to WiFi...");

}

Serial.println("Connected to WiFi");

Serial.println(WiFi.localIP());

// Paket 1

pinMode(pirPin, INPUT_PULLDOWN); // Set pin sensor PIR
sebagai input dengan pull-down

pinMode(relayPin, OUTPUT); // Set pin relay sebagai
output

digitalWrite(relayPin, LOW); // Pastikan relay dalam
kondisi mati saat awal

// Paket 2

pinMode(listrikUtama, OUTPUT); // Set pin relay sebagai
output

pinMode(buzzer, OUTPUT); // Set pin relay sebagai output
pinMode(kipas, OUTPUT); // Set pin kipas sebagai output

digitalWrite(relayPin, LOW); // Pastikan relay dalam kondisi
mati saat awal

digitalWrite(listrikUtama, HIGH);

myservo.attach(servoPin);

myservo.write(0);

// Paket 3

dht.begin();

lcd.init();

lcd.backlight();

}

void loop() {

    unsigned long currentMillis = millis();

```

```

// Membaca nilai dari pir dengan debounce

int pirState = digitalRead(pirPin); // Membaca status sensor
PIR

if (pirState == HIGH) {

    digitalWrite(relayPin, LOW); // Nyalakan lampu saat ada
gerakan

    Serial.println("Gerakan terdeteksi! Lampu: Nyala");

} else {

    digitalWrite(relayPin, HIGH); // Matikan lampu saat tidak
ada gerakan

    Serial.println("Tidak ada gerakan. Lampu: Mati");

}

// Membaca nilai dari pin digital
int gasValue = analogRead(gas);

// Paket 2 (gas, servo, buzzer, listrik)
Serial.print("Nilai gas: ");

Serial.println(gasValue);

if (gasValue >= GAS_THRESHOLD) {

    if (gasDetectTime == 0) {

        gasDetectTime = currentMillis; // Mulai timer untuk gas

    }

    digitalWrite(buzzer, HIGH); // Bunyi buzzer

    if (currentMillis - gasDetectTime >= 2000) { // Jika sudah
2 detik sejak gas terdeteksi

        myservo.write(115);          // Buka pintu

        delay(2000);                  // Tambahkan delay 2 detik
setelah membuka pintu

        digitalWrite(listrikUtama, LOW); // Matikan listrik

```

```

utama

    }

    } else {

        digitalWrite(buzzer, LOW);    // Matikan buzzer

        myservo.write(0);              // Tutup pintu

        digitalWrite(listrikUtama, HIGH); // Nyalakan listrik
utama

        gasDetectTime = 0;             // Reset timer gas

    }

    // Hitung daya listrik dalam watt

    float lampuDaya = (digitalRead(relayPin) == HIGH) ? 0.3 * 5
: 0.0; // Lampu: 0.3A × 5V

    float buzzerDaya = (gasValue >= GAS_THRESHOLD) ? 0.03 * 5 :
0.0; // Buzzer: 0.03A × 5V

    float lcdDaya = 0.001 * 5; // LCD: 0.001A × 5V

    float kipasDaya = 0.0;

    // Paket 3 (DHT22 dan Fan)

    humi = dht.readHumidity(); // Baca kelembaban

    suhu = dht.readTemperature() - 4; // Baca suhu

    if (isnan(humi) || isnan(suhu)) { // Jika tidak berhasil

        Serial.println("Failed to read from DHT sensor!");

        return; // Kembali

    } else { // Jika berhasil membaca sensor

        if (suhu >= 29) { // Jika suhu di atas 29

            digitalWrite(kipas, HIGH);

            kipasDaya = 0.2 * 5; // Kipas: 0.2A × 5V

            Serial.println("Kipas: Nyala");

        } else {

```

```

digitalWrite(kipas, LOW);

    kipasDaya = 0.0;

    Serial.println("Kipas: Mati");

}

}

// Total daya listrik dalam watt

totalDaya = lampuDaya + kipasDaya + buzzerDaya + lcdDaya;

// Update energi dalam kWh setiap interval

if (currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    // Hitung energi dalam kWh berdasarkan daya

    totalEnergy += (totalDaya / 1000.0 / 3600.0) * (interval /

1000.0);

    // Tampilkan daya listrik dan energi ke serial monitor

    Serial.print("Total Daya: ");

    Serial.print(totalDaya);

    Serial.println(" W");

    Serial.print("Total Energi: ");

    Serial.print(totalEnergy, 6); // Menampilkan hingga 6

desimal untuk kWh

    Serial.println(" kWh");

    // Mengirim data ke server

    sendData(String(suhu), String(pirState), String(gasValue),

String(totalEnergy, 6));

}

// Update tampilan LCD dengan informasi yang berbeda

lcd.clear();

```

```

switch (displayState) {

    case 0:

        lcd.setCursor(0, 0);

        lcd.print("Suhu: ");

        lcd.print(suhu);

        lcd.print(" C");

        lcd.setCursor(0, 1);

        lcd.print("Kelembaban: ");

        lcd.print(humi);

        lcd.print(" %");

        delay(3000); // Delay 3 detik

        break;

    case 1:

        lcd.setCursor(0, 0);

        lcd.print("Lampu: ");

        lcd.print(digitalRead(relayPin) == HIGH ? "Mati" :
"Hidup");

        lcd.setCursor(0, 1);

        lcd.print("Kipas: ");

        lcd.print(digitalRead(kipas) == HIGH ? "Hidup" : "Mati");

        delay(3000); // Delay 3 detik

        break;

    case 2:

        lcd.setCursor(0, 0);

        lcd.print("Gas: ");

        lcd.print(gasValue);

        lcd.print(" ppm");

```

```
lcd.setCursor(0, 1);

    lcd.print("Daya: ");

    lcd.print(totalEnergy, 6);

    lcd.print(" kWh");

    delay(3000); // Delay 3 detik

    break;

}

displayState = (displayState + 1) % 3; // Siklus tampilan
delay(1000); // Tunda untuk mencegah pembacaan terlalu cepat
}
```



Lampiran 4 Source Code Koneksi Kirim Data PHP dengan Database

```
<?php
// Database connection
$conn = new mysqli("localhost", "root", "", "kontrol");
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Get the input values
$suhu = $_POST['suhu'];
$gerak = $_POST['gerak'];
$gas = $_POST['gas'];
$daya = $_POST['daya'];
// Validate inputs
if (is_numeric($suhu) && is_numeric($gerak) &&
    is_numeric($gas) && is_numeric($daya)) {
    // Prepare the statement to avoid SQL injection
    $stmt = $conn->prepare("INSERT INTO status (suhu, gerak,
gas, daya) VALUES (?, ?, ?, ?)");
    $stmt->bind_param("dddd", $suhu, $gerak, $gas, $daya);
    if ($stmt->execute()) {
        echo "Berhasil disimpan";
    } else {
        echo "Gagal tersimpan: " . $stmt->error;
    }
    $stmt->close();
} else {
    echo "Invalid input";
}

// Close the connection
$conn->close();
?>
```