# LAPORAN TUGAS IOT

# PROYEK MENAMPILKAN SUHU, KELEMBAPAN DAN

# INTENSITAS CAHAYA PADA OLED



**Dosen Pengampu:**

Ir. Subairi, ST., MT., IPM

**Disusun Oleh:**

Syafa Meilia Putri - 233140707111087

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2025**

**DAFTAR ISI**

[DAFTAR ISI 2](#_Toc192667757)

[ABSTRAK 4](#_Toc192667758)

[BAB I 5](#_Toc192667759)

[PENDAHULUAN 5](#_Toc192667760)

[1.1 Latar Belakang 5](#_Toc192667761)

[1.2 Tujuan Praktikum 5](#_Toc192667762)

[BAB II 6](#_Toc192667763)

[METODOLOGI 6](#_Toc192667764)

[2.1 Alat dan Bahan 6](#_Toc192667765)

[2.2 Langkah Implementasi 6](#_Toc192667766)

[BAB III 7](#_Toc192667767)

[HASIL DAN PEMBAHASAN 7](#_Toc192667768)

[3.1 Hasil Eksperimen 7](#_Toc192667769)

[3.2 Kode Program 7](#_Toc192667770)

[3.2.1 Main.cpp 7](#_Toc192667771)

[3.2.2 Diagram.json 10](#_Toc192667772)

[BAB IV 11](#_Toc192667774)

[KESIMPULAN 11](#_Toc192667775)

[4.1 Kesimpulan 11](#_Toc192667776)

**ABSTRAK**

Percobaan ini mengembangkan sistem berbasis mikrokontroler ESP32 yang menggunakan sensor DHT22, LDR, dan tampilan OLED untuk memantau suhu, kelembapan, dan tingkat pencahayaan. Tujuan dari penelitian ini adalah untuk mengembangkan sistem pemantauan waktu nyata (real-time) yang dapat menampilkan informasi lingkungan secara langsung. Hasil simulasi menunjukkan bahwa sistem dapat mengukur dan menampilkan suhu, kelembapan, dan intensitas cahaya dengan akurat; LDR mengukur intensitas cahaya. Data sensor diperbarui secara berkala pada selang waktu tertentu untuk memastikan sistem pemantauan yang konsisten dan stabil. Implementasi ini menunjukkan bahwa penggabungan LDR, ESP32, dan sensor DHT22 dapat menjadi solusi sederhana namun efektif untuk sistem pemantauan kondisi lingkungan.

Kata kunci: ESP32, sensor DHT22, LDR, layar OLED, pemantauan waktu nyata, suhu, kelembapan, intensitas cahaya.

**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Dengan berkembangnya teknologi Internet of Things (IoT), ada banyak peluang untuk membangun sistem yang dapat memantau dan merespons kondisi lingkungan secara otomatis dan real-time. Penggabungan berbagai jenis sensor ke dalam perangkat sehari-hari memungkinkan pengumpulan data terus-menerus yang bermanfaat untuk membantu pengambilan keputusan yang berbasis data. Mikrokontroler ESP32 digunakan sebagai unit kendali utama dalam proyek ini karena kemampuannya dalam mengolah data dan kemampuan konektivitasnya yang fleksibel. Sementara LDR (Resistor Bergantung Cahaya) digunakan untuk mengukur intensitas cahaya di lingkungan sekitar, sensor DHT22 mengukur suhu dan kelembapan udara. Selanjutnya, layar OLED menampilkan data dari kedua sensor secara langsung. Dengan demikian, pengguna dapat dengan mudah melihat kondisi lingkungan. Implementasi sistem ini menunjukkan bagaimana teknologi Internet of Things dapat digunakan untuk membuat solusi monitoring yang cerdas, hemat energi, dan mudah diterapkan dalam berbagai konteks.

**1.2 Tujuan Praktikum**

1. Mengembangkan dan menerapkan sistem pemantauan lingkungan yang digerakkan oleh Internet of Things (IoT) secara real-time.
2. Menggunakan mikrokontroler ESP32 sebagai pusat kendali untuk mengambil dan memproses data sensor.
3. Menggabungkan sensor DHT22 untuk mengukur suhu dan kelembapan udara dan LDR untuk melacak intensitas cahaya.

**BAB II**

**METODOLOGI**

**2.1 Alat dan Bahan**

* Mikrokontroler ESP32
* Sensor suhu dan kelembapan DHT22
* Sensor cahaya LDR
* Layar OLED dengan resolusi 128x64 piksel
* Arduino IDE atau PlatformIO
* Simulator Wokwi

**2.2 Langkah Implementasi**

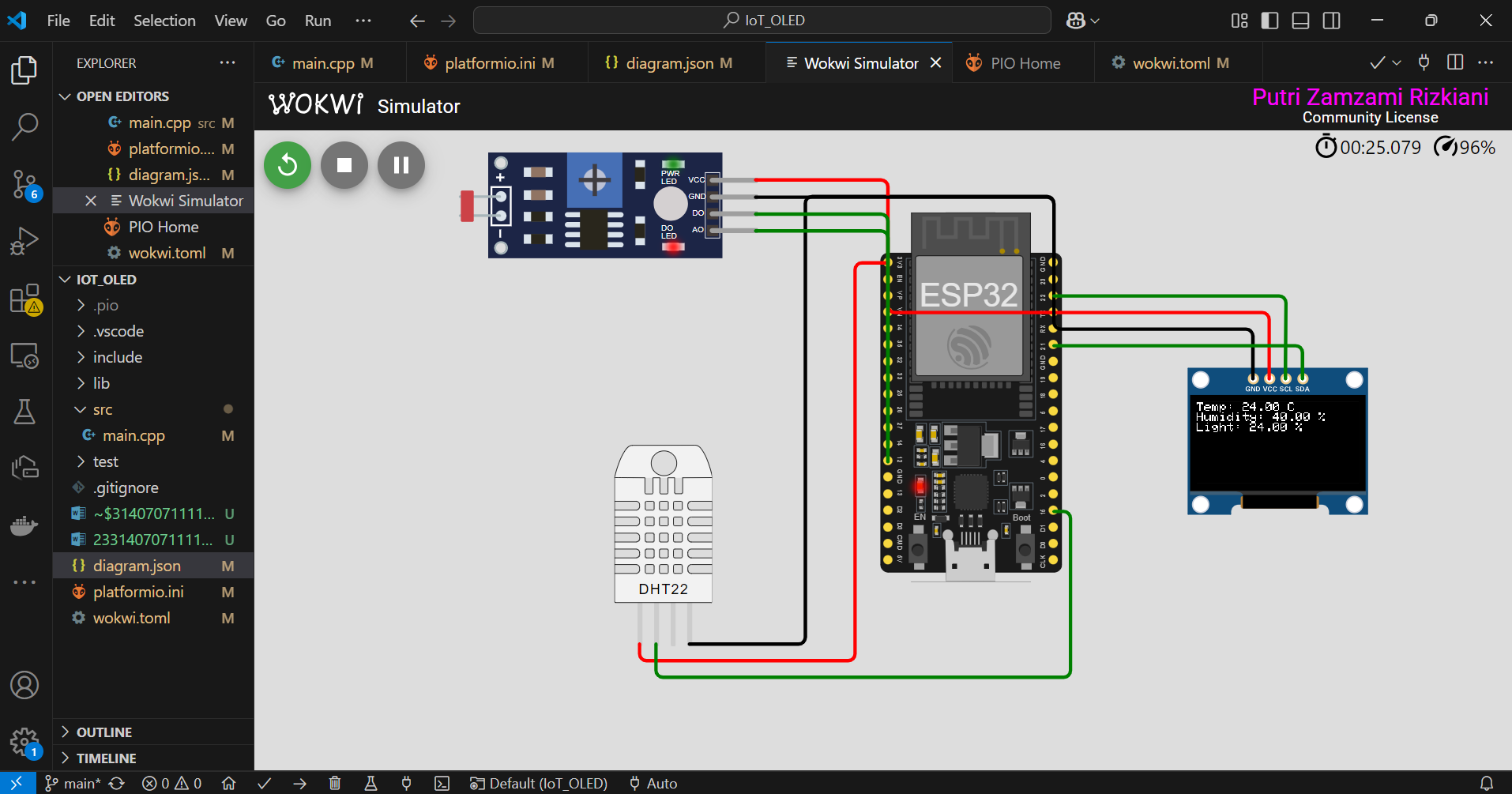
1. Merancang dan menyusun koneksi antara ESP32, DHT22, LDR, dan OLED berdasarkan diagram rangkaian.
2. Menggunakan pustaka (library) khusus untuk DHT22 dan OLED di dalam Arduino IDE atau PlatformIO.
3. Melakukan inisialisasi perangkat dan menampilkan hasil pengukuran suhu, kelembapan, serta pencahayaan ke layar OLED dalam interval waktu tertentu.
4. Mengunggah kode program ke ESP32 dan melakukan pengujian sistem. Sebelum uji coba langsung, simulasi dapat dilakukan menggunakan platform Wokwi untuk memastikan sistem berjalan dengan benar.

**BAB III**

**HASIL DAN PEMBAHASAN**

**3.1 Hasil Eksperimen**

Hasil pelaksanaan praktikum menunjukkan bahwa ESP32 mampu memproses data dari sensor DHT22 untuk mengukur suhu serta kelembapan, dan juga dari sensor LDR untuk mengukur tingkat cahaya. Data yang dikumpulkan kemudian ditampilkan secara langsung pada layar OLED. Contohnya, dalam simulasi diperoleh nilai suhu sebesar 24.00°C, kelembapan 40.00%, dan intensitas cahaya sebesar 24.00%. Pembaruan data berlangsung secara berkala, misalnya setiap lima detik. Sensor DHT22 menghasilkan nilai yang stabil dan akurat dalam pengukuran suhu serta kelembapan, sedangkan LDR mampu mendeteksi perubahan pencahayaan dengan responsif. Hasil ini menunjukkan bahwa sistem dapat digunakan untuk melakukan pemantauan kondisi lingkungan secara efektif dan berkelanjutan.



Gambar 1. (Simulasi)

## Kode Program

### 3.2.1 Main.cpp

#include <Wire.h>

#include <Adafruit\_SSD1306.h>

#include <Adafruit\_GFX.h>

#include <DHT.h>

#define SCREEN\_WIDTH 128 // OLED width, in pixels

#define SCREEN\_HEIGHT 64 // OLED height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

// DHT22 setup

#define DHTPIN 15      // Pin where DHT22 is connected

#define DHTTYPE DHT22  // DHT 22 (AM2302)

DHT dht(DHTPIN, DHTTYPE);

// LDR setup

#define LDR\_PIN 12     // Analog pin for LDR (AO pin)

void setup() {

  // Initialize the serial monitor

  Serial.begin(115200);

  // Initialize the DHT22 sensor

  dht.begin();

  // Initialize the OLED display

  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {  // Address 0x3C for 128x64

    Serial.println(F("SSD1306 allocation failed"));

    for (;;); // Don't proceed, loop forever

  }

  // Clear the display buffer

  display.clearDisplay();

  // Set text size and color

  display.setTextSize(1);

  display.setTextColor(SSD1306\_WHITE);

}

void loop() {

  // Read temperature and humidity from DHT22

  float temperature = dht.readTemperature();

  float humidity = dht.readHumidity();

  // Read light intensity from LDR

  int lightValue = analogRead(LDR\_PIN);

  // Convert LDR value to a percentage (optional, for display)

  float lightPercentage = map(lightValue, 0, 4095, 0, 100);

  // Check if any reading failed, and exit early (to try again).

  if (isnan(temperature) || isnan(humidity)) {

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

    return;

  }

  // Print the values to the serial monitor (optional)

  Serial.print(F("Temperature: "));

  Serial.print(temperature);

  Serial.print(F(" °C  Humidity: "));

  Serial.print(humidity);

  Serial.print(F("%  Light Intensity: "));

  Serial.print(lightPercentage);

  Serial.println(F("%"));

  // Display the values on OLED

  display.clearDisplay();  // Clear the display buffer

  display.setCursor(0, 0); // Set the cursor to (0,0)

  display.print(F("Temp: "));

  display.print(temperature);

  display.println(F(" C"));

  display.print(F("Humidity: "));

  display.print(humidity);

  display.println(F(" %"));

  display.print(F("Light: "));

  display.print(lightPercentage);

  display.println(F(" %"));

  display.display();  // Send the buffer to the display

  delay(2000);  // Wait for 2 seconds before the next reading

}

# 3.2.2 Diagram.json

# 

# BAB IV

# KESIMPULAN

## Kesimpulan

Projek ini menunjukkan cara yang efektif untuk menggunakan teknologi Internet of Things (IoT) untuk memantau kondisi lingkungan secara real-time dan efisien. Sistem menggunakan mikrokontroler ESP32 sebagai pusat kontrol untuk membaca data dari sensor DHT22 tentang suhu dan kelembapan, serta intensitas cahaya dari sensor LDR. Semua data ditampilkan secara langsung pada layar OLED, sehingga pengguna dapat dengan mudah melihat kondisi sekitar. Implementasi ini menunjukkan bahwa sensor dan perangkat mikrokontroler dapat digabungkan untuk membuat sistem monitoring yang sederhana tetapi efektif. Sistem ini dapat dikembangkan lebih lanjut untuk memenuhi berbagai kebutuhan pemantauan otomatis.