

PRAKTIKUM SIMULASI SISTEM MONITORING SUHU, KELEMBAPAN, DAN CAHAYA BERBASIS ESP32 DI WOKWI

TB Rangga Gilang Yanuari
Fakultas Vokasi, Universitas Brawijaya
gilangyanuarirangga@gmail.com

ABSTRAK

Pemanfaatan Internet of Things (IoT) semakin luas dalam berbagai bidang, salah satunya untuk sistem pemantauan lingkungan. Praktikum ini mensimulasikan sistem monitoring suhu, kelembapan, dan intensitas cahaya menggunakan ESP32, sensor DHT22, dan LDR dengan simulasi pada platform Wokwi. Data sensor ditampilkan pada LCD 16x2 melalui komunikasi I2C. Hasil simulasi menunjukkan bahwa ESP32 mampu membaca data suhu dan kelembapan dengan akurat serta menangkap perubahan intensitas cahaya melalui LDR. Wokwi memungkinkan pengujian tanpa perangkat keras fisik sehingga efisien untuk pembelajaran dan eksperimen. Pengembangan lebih lanjut dapat dilakukan dengan menyimpan data ke cloud untuk pemantauan jarak jauh.

Kata Kunci: IoT, ESP32, DHT22, LDR, Wokwi

ABSTRACT

The utilization of the Internet of Things (IoT) is expanding across various fields, including environmental monitoring systems. This practicum simulates a system for monitoring temperature, humidity, and light intensity using ESP32, a DHT22 sensor, and an LDR sensor with a simulation on the Wokwi platform. Sensor data is displayed on a 16x2 LCD via I2C communication. The simulation results show that ESP32 can accurately read temperature and humidity data and capture changes in light intensity through LDR. Wokwi enables testing without physical hardware, making it efficient for learning and experimentation. Future developments can include cloud storage for remote monitoring.

Kata Kunci: IoT, ESP32, DHT22, LDR, Wokwi

1. PENDAHULUAN

1.1 Latar Belakang

Internet of Things (IoT) telah mengubah cara manusia berinteraksi dengan lingkungan sekitar melalui sistem otomatisasi yang dapat mengumpulkan dan mengolah data secara real-time. Salah satu aplikasi utama IoT adalah sistem pemantauan lingkungan yang dapat mendeteksi perubahan suhu, kelembapan, dan intensitas cahaya. Dalam dunia industri, pertanian, dan rumah pintar, pemantauan parameter lingkungan sangat penting untuk mengoptimalkan efisiensi dan kenyamanan.

ESP32 merupakan salah satu mikrokontroler yang sering digunakan dalam sistem berbasis IoT karena memiliki kemampuan komputasi yang baik dan konektivitas WiFi serta Bluetooth. Dengan menggunakan sensor DHT22 untuk membaca suhu dan kelembapan, serta LDR untuk mendeteksi intensitas cahaya, sistem ini dapat memberikan informasi akurat tentang kondisi lingkungan sekitar. Dalam praktikum ini, digunakan simulator Wokwi untuk menguji sistem pemantauan tanpa memerlukan perangkat keras fisik.

1.2 Tujuan Praktikum

Praktikum ini bertujuan untuk:

- 1) Memahami konsep dasar IoT dalam pemantauan lingkungan.
- 2) Mempelajari cara membaca data dari sensor DHT22 dan LDR menggunakan ESP32.
- 3) Mengimplementasikan komunikasi I2C untuk menampilkan data sensor pada LCD 16x2.
- 4) Menggunakan Wokwi sebagai alat simulasi untuk menguji sistem tanpa perangkat keras fisik.
- 5) Mengevaluasi hasil simulasi dan menganalisis potensi pengembangan lebih lanjut.

2. METODOLOGI

2.1 Alat dan Bahan

- **Alat:**
 - a) Komputer/laptop sebagai perangkat utama untuk menjalankan simulasi.
 - b) Platform Wokwi sebagai simulator rangkaian tanpa perangkat fisik.
 - c) Visual Studio Code (VSCode) dengan ekstensi PlatformIO sebagai lingkungan pengembangan.
- **Bahan:**
 - a) **ESP32** sebagai mikrokontroler utama.
 - b) **Sensor DHT22** untuk mengukur suhu dan kelembapan.
 - c) **Sensor LDR** untuk mendeteksi intensitas cahaya.
 - d) **LCD 16x2 dengan I2C** untuk menampilkan data sensor.
 - e) **Bahasa Pemrograman C++ Arduino** untuk mengontrol sistem.

2.2 Langkah Implementasi

Berikut adalah langkah-langkah implementasi secara rinci dalam melakukan simulasi sistem monitoring suhu, kelembapan, dan intensitas cahaya menggunakan ESP32 di Wokwi dan PlatformIO:

1. Mengunduh dan Menginstal Perangkat Lunak

- 1) Unduh dan instal **Visual Studio Code (VSCode)** sebagai lingkungan pengembangan.
- 2) Instal **PlatformIO** sebagai ekstensi di VSCode.
- 3) Buka situs **Wokwi** (<https://wokwi.com>) untuk melakukan simulasi tanpa perangkat keras fisik.

2. Membuat Rangkaian Wokwi

- 1) Buka **Wokwi** dan buat proyek baru.
- 2) Tambahkan komponen **ESP32**, **DHT22**, **LDR**, dan **LCD 16x2 (I2C)** ke dalam workspace.
- 3) Hubungkan setiap komponen sesuai dengan konfigurasi berikut:

A. DHT22

- i. VCC → **3V3 ESP32**
- ii. GND → **GND ESP32**
- iii. SDA → **Pin 4 ESP32**

B. LDR (Photoresistor Sensor)

- i. VCC → **3V3 ESP32**
- ii. GND → **GND ESP32**
- iii. AO (Analog Output) → **Pin 32 ESP32**

C. LCD 16x2 (I2C)

- i. VCC → **5V ESP32**
- ii. GND → **GND ESP32**
- iii. SDA → **Pin 21 ESP32**
- iv. SCL → **Pin 22 ESP32**

- 4) Simpan proyek dan pastikan semua koneksi sudah sesuai dengan diagram di Wokwi.
- 5) Buat kode program agar proyek bisa Menampilkan Suhu, kelembapan dan intensitas Cahaya pada LCD

3. Menghubungkan Wokwi dengan PlatformIO di VSCode

- 1) Buka **VSCode** dan buat proyek baru di **PlatformIO** dengan board **ESP32 DOIT**.
- 2) Salin seluruh isi file **diagram.json** dari Wokwi.
- 3) Di dalam proyek **PlatformIO**, buat file baru bernama **diagram.json**.
- 4) Tempelkan hasil salinan **diagram.json** ke dalam file tersebut.
- 5) Buat file konfigurasi **wokwi.toml** di dalam proyek PlatformIO dan isi dengan:

```
[wokwi]
version = 1
firmware = '.pio/build/esp32doit-devkit-v1/firmware.bin'
elf = '.pio/build/esp32doit-devkit-v1/firmware.elf'
```

- 6) Simpan semua perubahan di dalam VSCode.

4. Menulis dan Menunggah Kode Program

- 1) Buat file **main.cpp** di dalam folder **src** di proyek PlatformIO.
- 2) Menyalin kode program dari situs wokwi untuk membaca data dari sensor dan menampilkannya di LCD
- 3) Simpan file dan pastikan tidak ada error dalam kode.

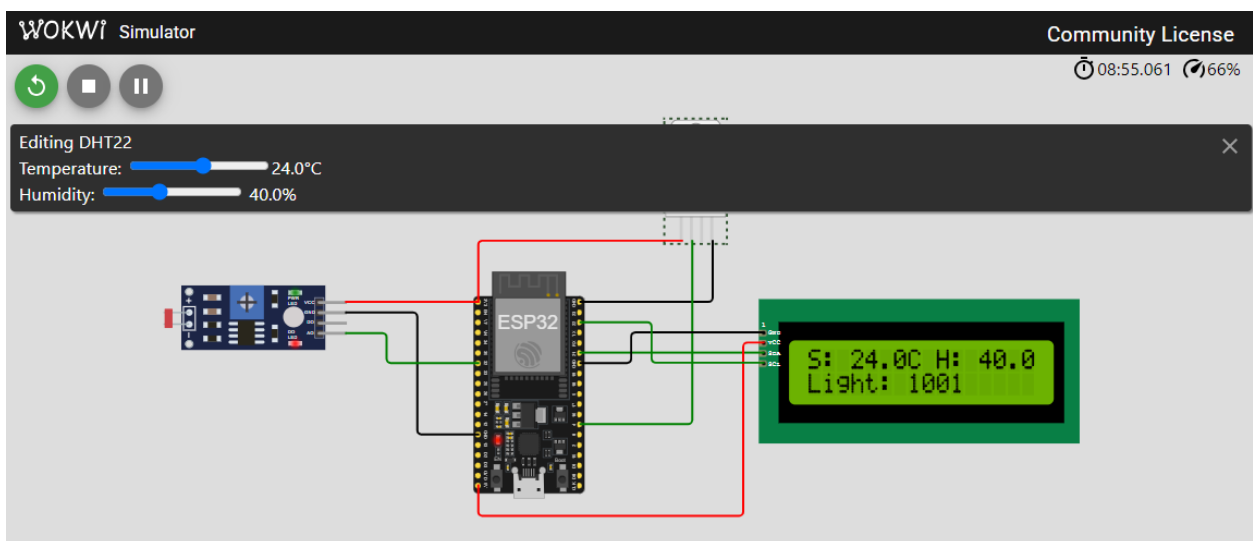
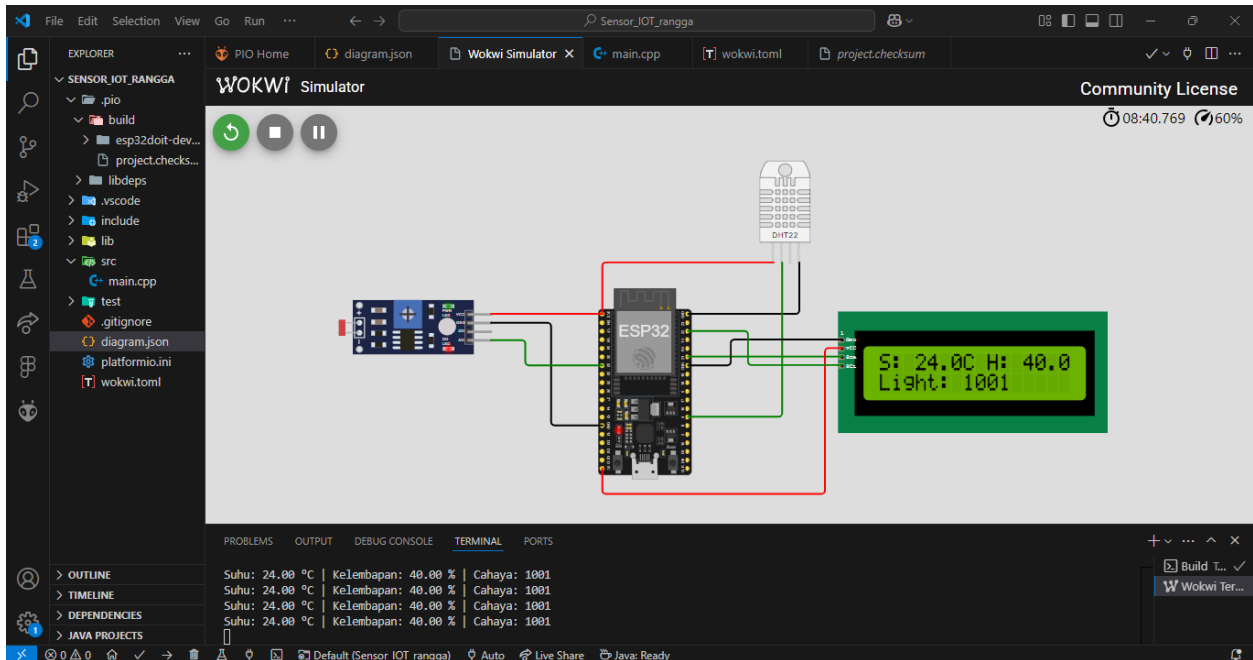
5. Menjalankan Simulasi di Wokwi

- 1) Klik **tanda centang** di pojok kanan atas **PlatformIO** untuk melakukan **kompilasi** kode.

- 2) Setelah kompilasi berhasil, klik **Run** untuk menjalankan simulasi.
- 3) Amati hasil tampilan di **LCD 16x2** dan pastikan pembacaan sensor sesuai dengan ekspektasi.

3. HASIL DAN PEMBAHASAN

3.1 Hasil



3.2 Pembahasan

Hasil dari percobaan ini adalah tampilan di LCD yang menunjukkan suhu, kelembapan, dan intensitas cahaya secara real-time. Hasil yang Ditampilkan di LCD Pada LCD 16x2, yaitu tampilan data sensor terlihat menampilkan sebagai berikut: **S: 24.0C H: 40.0 Light: 1001**

- S menunjukkan suhu dalam derajat Celsius (°C), yang dalam contoh ini adalah 24.0°C.
- H menunjukkan kelembapan dalam persen (%), yaitu 40.0%.
- Light menunjukkan intensitas cahaya dari sensor LDR, yaitu 1001 (nilai analog).

Simulasi Pengaturan Suhu dan Kelembapan Gambar kedua menunjukkan pengaturan manual pada DHT22, yang memungkinkan kita mengubah nilai Temperature dan Humidity untuk melihat perubahan tampilan di LCD secara langsung. Dalam simulasi ini, suhu diatur ke 24.0°C dan kelembapan ke 40.0%, yang sesuai dengan tampilan di LCD. dan untuk intensitas cahaya nilainya bergantung pada cahaya sekitar yang diterima oleh LDR. Jika lingkungan lebih terang, nilai intensitas cahaya akan lebih kecil, dan jika lebih gelap, nilainya akan lebih besar.

4. LAMPIRAN

4.1 Kode program untuk Menampilkan Suhu, kelembapan dan intensitas Cahaya

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>

#define DHTPIN 4
#define DHTTYPE DHT22
#define LDRPIN 32 // Pin Analog untuk LDR
#define TIMEDHT 2000 // Update setiap 2 detik

// Inisialisasi sensor dan LCD
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2); // Jika tidak tampil, coba ganti 0x27 ke 0x3F

// Variabel untuk menyimpan data sensor
float humidity, celsius;
int lightIntensity;
uint32_t timerDHT = 0;

// Fungsi untuk membaca suhu dan kelembapan
void getTemperature() {
    humidity = dht.readHumidity();
    celsius = dht.readTemperature();

    if (isnan(humidity) || isnan(celsius)) {
        Serial.println("Failed to read from DHT sensor!");
        humidity = 0.0;
        celsius = 0.0;
    }
}
```

```

// Fungsi untuk membaca intensitas cahaya
void getLightIntensity() {
    lightIntensity = analogRead(LDRPIN);
}

// Cek apakah LCD terdeteksi di I2C
void checkI2C() {
    Wire.begin();
    Serial.println("Scanning for I2C devices...");
    for (byte address = 1; address < 127; address++) {
        Wire.beginTransmission(address);
        if (Wire.endTransmission() == 0) {
            Serial.print("Found I2C device at address 0x");
            Serial.println(address, HEX);
        }
    }
}

void setup() {
    Serial.begin(115200);
    Serial.println("Monitoring Started...");

    checkI2C(); // Cek alamat I2C LCD

    dht.begin();
    lcd.init(); // Pastikan menggunakan lcd.init() bukan lcd.begin();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("System Starting...");
    delay(2000);
    lcd.clear();
}

void loop() {
    getTemperature();
    getLightIntensity();

    Serial.print("Suhu: "); Serial.print(celsius); Serial.print(" °C | ");
    Serial.print("Kelembapan: "); Serial.print(humidity); Serial.print(" % | ");
    Serial.print("Cahaya: "); Serial.println(lightIntensity);

    // Tampilkan di LCD tanpa flicker
    lcd.setCursor(0, 0);
    lcd.print("S: "); lcd.print(celsius, 1); lcd.print("C ");

```

```

    lcd.print("H: "); lcd.print(humidity, 1); lcd.print("% ");

    lcd.setCursor(0, 1);
    lcd.print("Light: "); lcd.print(lightIntensity);
    lcd.print("    "); // Tambahkan spasi untuk menghapus digit lama

    delay(2000);
}

```

4.2 Kode diagram.json

```

{
  "version": 1,
  "author": "Anonymous maker",
  "editor": "wokwi",
  "parts": [
    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": -76.8, "left": 129.64, "attrs": {} },
    { "type": "wokwi-dht22", "id": "dht1", "top": -220.5, "left": 311.4, "attrs": {} },
    {
      "type": "wokwi-photoresistor-sensor",
      "id": "ldr1",
      "top": -64,
      "left": -162.4,
      "attrs": {}
    },
    {
      "type": "wokwi-lcd1602",
      "id": "lcd1",
      "top": -51.2,
      "left": 399.2,
      "attrs": { "pins": "i2c" }
    }
  ],
  "connections": [
    [ "esp:TX", "$serialMonitor:RX", "", [ ] ],
    [ "esp:RX", "$serialMonitor:TX", "", [ ] ],
    [ "lcd1:SDA", "esp:21", "green", [ "h0" ] ],
    [ "dht1:VCC", "esp:3V3", "red", [ "v0", "h-96" ] ],
    [ "dht1:GND", "esp:GND.2", "black", [ "v48" ] ],
    [ "dht1:SDA", "esp:4", "green", [ "v0" ] ],
    [ "ldr1:VCC", "esp:3V3", "red", [ "h0" ] ],
    [ "ldr1:GND", "esp:GND.1", "black", [ "h67.2", "v114.8" ] ],

```

```
    [ "lcd1:VCC", "esp:5V", "red", [ "h-19.2", "v144.1", "h0", "v19.2", "h-  
259.2" ] ],  
    [ "ldr1:AO", "esp:32", "green", [ "h38.4", "v37.7" ] ],  
    [ "lcd1:GND", "esp:GND.3", "black", [ "h-124.8", "v38.4" ] ],  
    [ "lcd1:SCL", "esp:22", "green", [ "h-105.6", "v0.3" ] ]  
  ],  
  "dependencies": {}  
}
```