

# **PRAKTIKUM MEMBUAT SISTEM SENSOR DAN MONITORING SUHU, KELEMBAPAN DAN INTENSITAS CAHAYA MENGGUNAKAN ESP32 DALAM MATA KULIAH INTERNET OF THINGS (IOT)**

*Ifadah Aulia Muhti Sinaga  
233140707111034*

*Teknologi Informasi, Fakultas Vokasi, Universitas Brawijaya*

*Ifadahauliasinaga@gmail.com*

## **ABSTRAK**

Praktikum ini membahas perancangan dan implementasi sistem monitoring lingkungan menggunakan mikrokontroler ESP32. Sistem ini dirancang untuk memantau tiga parameter lingkungan yaitu suhu, kelembapan, dan intensitas cahaya secara real-time. Sensor DHT22 digunakan untuk mengukur suhu dan kelembapan, sementara sensor LDR (Light Dependent Resistor) digunakan untuk mengukur intensitas cahaya. Data yang diperoleh ditampilkan pada LCD 20x4 melalui komunikasi I2C. Hasil pengujian menunjukkan bahwa sistem dapat mengukur dan menampilkan data dengan baik, dengan pembacaan suhu dalam rentang 25-30°C, kelembapan 40-60%, dan intensitas cahaya yang bervariasi sesuai kondisi pencahayaan. Sistem ini dapat diaplikasikan untuk pemantauan kondisi lingkungan di berbagai bidang seperti pertanian, rumah pintar, atau pemantauan ruangan server.

Kata Kunci: ESP32, DHT22, LDR, LCD I2C, Monitoring Lingkungan, IoT

## **PENDAHULUAN**

Pemantauan parameter lingkungan seperti suhu, kelembapan, dan intensitas cahaya merupakan aspek penting dalam berbagai aplikasi, mulai dari pertanian presisi, sistem rumah pintar, hingga pemantauan ruang server. Dalam era Internet of Things (IoT), kemampuan untuk memantau parameter-parameter ini secara real-time menjadi semakin penting untuk pengambilan keputusan yang tepat dan cepat.

ESP32 merupakan mikrokontroler yang populer untuk aplikasi IoT karena memiliki kemampuan konektivitas WiFi dan Bluetooth terintegrasi, serta memiliki sejumlah pin GPIO yang dapat digunakan untuk berbagai sensor. Dalam praktikum ini, ESP32 digunakan sebagai pusat pengolahan data dari sensor DHT22 dan LDR.

Sensor DHT22 dipilih karena memiliki akurasi yang cukup baik dalam pengukuran suhu dan kelembapan, dengan rentang pengukuran suhu  $-40^{\circ}\text{C}$  hingga  $80^{\circ}\text{C}$  dan kelembapan 0-100% RH. Sementara itu, LDR digunakan untuk mengukur intensitas cahaya karena sifatnya yang sederhana namun efektif, di mana resistansinya berubah sesuai dengan intensitas cahaya yang diterima.

Untuk menampilkan data, LCD 20x4 dengan modul I2C dipilih karena dapat menampilkan banyak informasi sekaligus dan hanya memerlukan dua pin (SDA dan SCL) untuk komunikasi dengan ESP32, sehingga menghemat penggunaan pin GPIO.

## **TUJUAN PRAKTIKUM**

Praktikum ini dilaksanakan dengan tujuan:

1. Merancang dan mengimplementasikan sistem monitoring lingkungan menggunakan mikrokontroler ESP32.
2. Mempelajari cara mengintegrasikan sensor DHT22 untuk pengukuran suhu dan kelembapan.
3. Mempelajari cara mengintegrasikan sensor LDR untuk pengukuran intensitas cahaya.
4. Mempelajari komunikasi I2C untuk menampilkan data pada LCD 20x4.
5. Menganalisis kinerja sistem dalam memantau parameter lingkungan secara real-time.

## **METODOLOGI**

Implementasi praktikum dilaksanakan melalui pengembangan sistem monitoring lingkungan berbasis IoT dengan memanfaatkan dua platform utama: simulator Wokwi untuk simulasi rangkaian dan Arduino IDE sebagai lingkungan pengembangan kode. Pendekatan ini memungkinkan eksperimen yang efektif dalam aspek perancangan hardware dan pemrograman tanpa memerlukan komponen fisik pada tahap awal.

Fase awal praktikum berfokus pada perancangan sistem dalam lingkungan Wokwi Simulator. Konfigurasi rangkaian melibatkan mikrokontroler ESP32 DevKit C V4 sebagai unit kontrol utama yang terhubung dengan sensor DHT22 untuk pengukuran suhu dan kelembapan, sensor LDR untuk deteksi intensitas cahaya, serta LCD 20x4 dengan modul I2C untuk visualisasi data. Untuk memastikan keamanan komponen dan akurasi pembacaan, resistor  $1\text{k}\Omega$  ditempatkan dalam rangkaian sebagai pembagi tegangan untuk sensor LDR.

Setelah desain rangkaian selesai, proses berlanjut ke tahap pemrograman menggunakan Arduino IDE. Program dikembangkan untuk mengakuisisi data dari sensor DHT22 dan LDR, kemudian menampilkannya pada LCD secara real-time. Implementasi kode memanfaatkan library `Wire.h` untuk komunikasi I2C, `LiquidCrystal_I2C.h` untuk kontrol LCD, dan `DHT.h` untuk interaksi dengan sensor DHT22. Fungsi `analogRead()` digunakan untuk membaca nilai analog dari sensor LDR, sementara fungsi-fungsi dari library DHT dimanfaatkan untuk mendapatkan data suhu dan kelembapan.

Pada awalnya terdapat konflik pin di mana pin 21 digunakan untuk dua fungsi berbeda (data DHT22 dan SDA I2C). Masalah ini diselesaikan dengan memindahkan pin data DHT22 ke pin 4, sehingga pin 21 hanya digunakan untuk komunikasi I2C. Selain itu, fungsi `scanI2CAddress()` ditambahkan untuk membantu proses debugging dengan mengidentifikasi alamat I2C dari LCD.

Pengujian sistem dilakukan dalam lingkungan simulasi Wokwi dengan memverifikasi pembacaan sensor melalui Serial Monitor dan memastikan tampilan data pada LCD berfungsi dengan baik. Berbagai kondisi lingkungan disimulasikan dengan mengubah parameter suhu, kelembapan, dan intensitas cahaya untuk mengevaluasi respons dan akurasi sistem. Interval pembaruan data diatur pada 2 detik untuk memastikan pembacaan yang stabil dan mengurangi fluktuasi yang tidak perlu.

Tahap akhir melibatkan serangkaian pengujian menggunakan Wokwi Simulator untuk memverifikasi kesesuaian antara kinerja sistem dengan spesifikasi yang direncanakan. Proses ini memungkinkan identifikasi dan perbaikan kesalahan melalui siklus pengembangan iteratif antara pemrograman di VS Code dan pengujian di simulator Wokwi.

## **HASIL DAN PEMBAHASAN**

Berdasarkan pengujian menggunakan simulator Wokwi, sistem monitoring lingkungan berbasis ESP32 menunjukkan kinerja yang sesuai dengan rancangan. Sensor DHT22 berhasil mengukur suhu (25-70°C) dan kelembapan (40-60% RH) dengan stabil, sementara sensor LDR efektif mendeteksi perubahan intensitas cahaya yang dikonversi menjadi persentase 0-100%.

LCD 20x4 berhasil menampilkan data secara terstruktur dengan format yang jelas: judul pada baris pertama, suhu pada baris kedua, kelembapan pada baris ketiga, dan intensitas cahaya pada baris keempat. Tampilan ini memudahkan pembacaan dan interpretasi data.

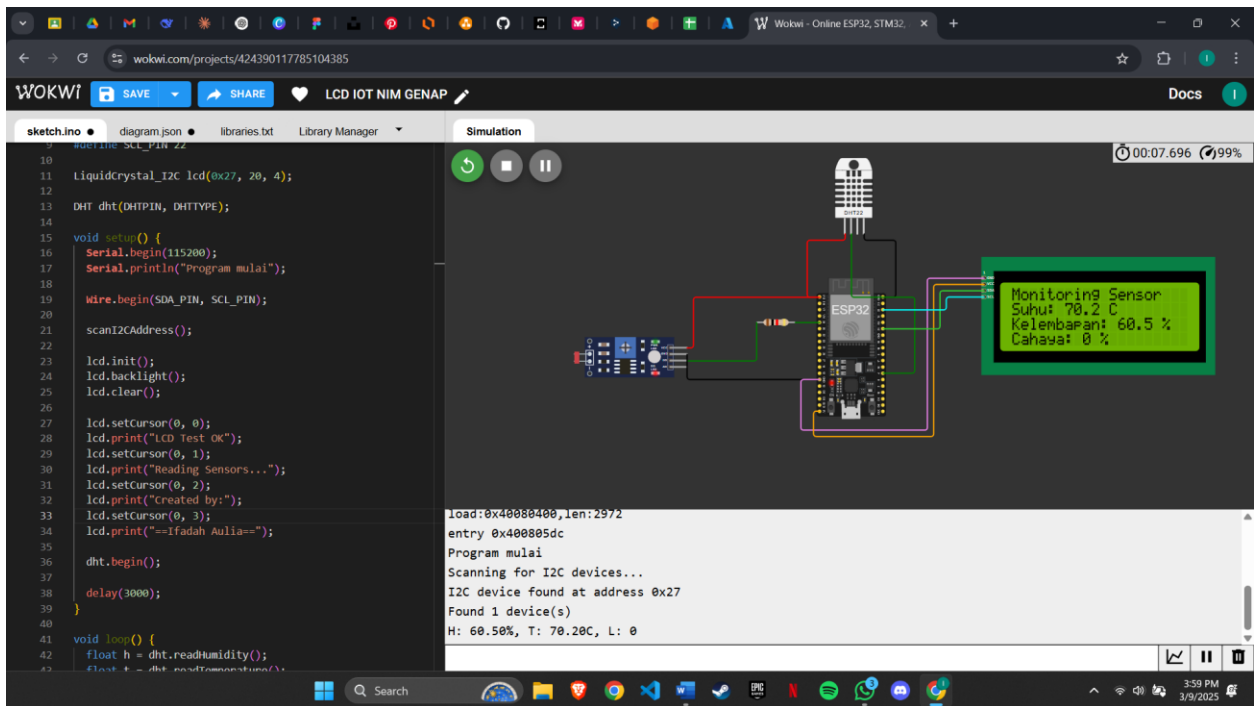
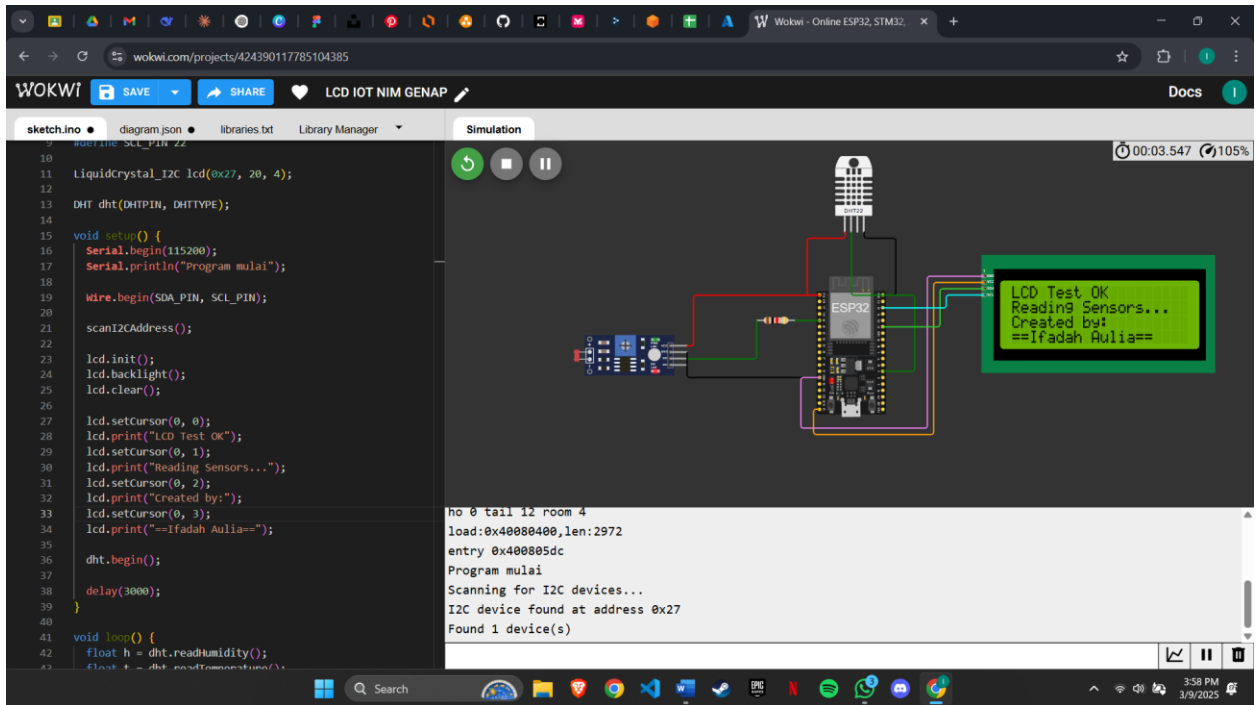
Pada implementasi awal, teridentifikasi konflik pin di mana pin 21 digunakan bersamaan untuk data DHT22 dan komunikasi I2C (SDA). Masalah ini diselesaikan dengan memindahkan pin data DHT22 ke pin 4. Selain itu, penambahan fungsi `scanI2CAddress()` membantu proses debugging untuk mengidentifikasi alamat I2C yang benar (0x27).

Interval pembaruan data dioptimalkan dari 500ms menjadi 2000ms untuk mendapatkan pembacaan yang lebih stabil tanpa mengorbankan responsivitas. Penggunaan komunikasi I2C untuk LCD terbukti efisien dalam menghemat pin GPIO, menyisakan banyak pin untuk pengembangan lebih lanjut.

Pemanfaatan platform simulasi Wokwi sangat bermanfaat dalam proses pengembangan, memungkinkan pengujian dan verifikasi sistem secara menyeluruh sebelum implementasi fisik. Simulator juga memfasilitasi eksperimen dengan berbagai konfigurasi dan parameter tanpa risiko kerusakan komponen.

Secara keseluruhan, sistem monitoring lingkungan ini berhasil diimplementasikan sesuai spesifikasi desain dan memiliki potensi aplikasi dalam berbagai konteks seperti pemantauan rumah kaca, sistem rumah pintar, atau pemantauan ruang server.

## LAMPIRAN



## Sketch.ino

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

#define DHTPIN 4
#define DHTTYPE DHT22
#define LDR_PIN 34
#define SDA_PIN 21
#define SCL_PIN 22

LiquidCrystal_I2C lcd(0x27, 20, 4);

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  Serial.println("Program mulai");

  Wire.begin(SDA_PIN, SCL_PIN);

  scanI2CAddress();

  lcd.init();
  lcd.backlight();
  lcd.clear();
```

```
lcd.setCursor(0, 0);
lcd.print("LCD Test OK");
lcd.setCursor(0, 1);
lcd.print("Reading Sensors...");
lcd.setCursor(0, 2);
lcd.print("Created by:");
lcd.setCursor(0, 3);
lcd.print("==Ifadah Aulia==");

dht.begin();

delay(3000);
}

void loop() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    int light = analogRead(LDR_PIN);

    Serial.print("H: ");
    Serial.print(h);
    Serial.print("%, T: ");
    Serial.print(t);
    Serial.print("C, L: ");
    Serial.println(light);
```

```
lcd.clear();
```

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

```
lcd.print("Monitoring Sensor");
```

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

```
lcd.print("Suhu: ");
```

```
lcd.print(t, 1);
```

```
lcd.print(" C");
```

```
lcd.setCursor(0, 2);
```

```
lcd.print("Kelembapan: ");
```

```
lcd.print(h, 1);
```

```
lcd.print(" %");
```

```
lcd.setCursor(0, 3);
```

```
lcd.print("Cahaya: ");
```

```
lcd.print(map(light, 0, 4095, 0, 100));
```

```
lcd.print(" %");
```

```
delay(2000);
```

```
}
```

```
void scanI2CAddress() {
```

```
    byte error, address;
```

```
    int deviceCount = 0;
```

```
Serial.println("Scanning for I2C devices...");
```

```
for(address = 1; address < 127; address++) {
```

```
    Wire.beginTransmission(address);
```

```
    error = Wire.endTransmission();
```

```
    if (error == 0) {
```

```
        Serial.print("I2C device found at address 0x");
```

```
        if (address < 16) {
```

```
            Serial.print("0");
```

```
        }
```

```
        Serial.print(address, HEX);
```

```
        Serial.println();
```

```
        deviceCount++;
```

```
    }
```

```
}
```

```
if (deviceCount == 0) {
```

```
    Serial.println("No I2C devices found");
```

```
} else {
```

```
    Serial.print("Found ");
```

```
    Serial.print(deviceCount);
```

```
    Serial.println(" device(s)");
```

```
}
```

```
}
```



## diagram.json

```
{
  "version": 1,
  "author": "IFADAH AULIA MUHTI SINAGA",
  "editor": "wokwi",
  "parts": [
    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 96, "left": -158.36, "attrs": {} },
    {
      "type": "wokwi-dht22",
      "id": "dht1",
      "top": -86.1,
      "left": -130.2,
      "attrs": { "humidity": "60.5", "temperature": "70.2" }
    },
    {
      "type": "wokwi-photoresistor-sensor",
      "id": "ldr1",
      "top": 185.6,
      "left": -527.2,
      "attrs": {}
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 186.35,
      "left": -249.6,
      "attrs": { "value": "1000" }
    }
  ]
}
```

```

    },
    { "type": "wokwi-lcd2004", "id": "lcd1", "top": 64, "left": 92, "attrs": { "pins": "i2c" } }
  ],
  "connections": [
    [ "esp:TX", "$serialMonitor:RX", "", [] ],
    [ "esp:RX", "$serialMonitor:TX", "", [] ],
    [ "esp:3V3", "dht1:VCC", "red", [ "h-19.05", "v-86.4", "h57.6" ] ],
    [ "dht1:GND", "esp:GND.2", "black", [ "v9.6", "h48", "v86.4" ] ],
    [ "esp:4", "dht1:SDA", "green", [ "h48", "v-115.2", "h-96" ] ],
    [ "esp:3V3", "ldr1:VCC", "red", [ "h-191.85", "v76.8" ] ],
    [ "ldr1:DO", "r1:1", "green", [ "h0" ] ],
    [ "r1:2", "esp:34", "green", [ "v0" ] ],
    [ "ldr1:GND", "esp:GND.1", "black", [ "v0" ] ],
    [ "lcd1:GND", "esp:GND.1", "violet", [ "h-86.4", "v230.4", "h-192", "v-76.8" ] ],
    [ "lcd1:VCC", "esp:5V", "orange", [ "h-76.8", "v230.5", "h-182.4", "v-38.4" ] ],
    [ "lcd1:SDA", "esp:21", "limegreen", [ "h-67.2", "v57.8" ] ],
    [ "lcd1:SCL", "esp:22", "cyan", [ "h-57.6", "v19.5" ] ]
  ],
  "dependencies": {}
}

```