

# **Smart Sensor System: Monitoring Suhu, Kelembaban & Cahaya dengan ESP32, DHT 22, LDR, dan OLED Display**

***Fransiska Natasya Desyanti***

*Program Studi Teknologi Informasi, Fakultas Vokasi Universitas Brawijaya*

[francenats@gmail.com](mailto:francenats@gmail.com)

## **Abstract**

*This research develops an IoT Smart Sensor System for real-time environmental monitoring using ESP32 as the main microcontroller. The system integrates DHT22 sensor for temperature and humidity measurement, and Light Dependent Resistor (LDR) for light intensity detection, with results displayed on a 0.96-inch OLED Display. Implementation is carried out through Arduino IDE programming with a 2-second data sampling interval. Test results show measurement accuracy of  $\pm 0.5^{\circ}\text{C}$  for temperature,  $\pm 2\%$  RH for humidity, and LDR readings in ADC values convertible to lux units.*

**Keywords:** *IoT, Smart Sensor, ESP32, DHT22, LDR, OLED Display, Environmental Monitoring.*

## **Abstrak**

Penelitian ini bertujuan mengembangkan IoT Smart Sensor System untuk memantau parameter lingkungan secara real-time menggunakan ESP32 sebagai mikrokontroler utama. Sistem ini mengintegrasikan sensor DHT22 untuk pengukuran suhu dan kelembaban, serta Light Dependent Resistor (LDR) untuk mendeteksi intensitas cahaya, dengan hasil yang ditampilkan pada OLED Display 0.96 inci. Implementasi dilakukan melalui pemrograman berbasis Arduino IDE dengan metode pengambilan data setiap 2 detik. Hasil pengujian menunjukkan akurasi pengukuran suhu  $\pm 0.5^{\circ}\text{C}$ , kelembaban  $\pm 2\%$  RH, dan pembacaan LDR dalam nilai ADC yang dapat dikonversi ke satuan lux.

**Kata Kunci:** IoT, Smart Sensor, ESP32, DHT22, LDR, OLED Display, Monitoring Lingkungan.

## 1. PENDAHULUAN

### 1.1. LATAR BELAKANG

Perkembangan teknologi Internet of Things (IoT) telah membawa perubahan signifikan di berbagai bidang, termasuk pemantauan lingkungan. Dengan memungkinkan perangkat elektronik terhubung dan bertukar data secara otomatis melalui internet, IoT menciptakan sistem yang cerdas, efisien, serta responsif terhadap perubahan lingkungan. Salah satu contoh penerapannya adalah sistem pemantauan berbasis sensor yang mampu mengukur suhu, kelembaban, dan intensitas cahaya secara real-time. Data yang akurat ini memungkinkan tindakan pencegahan atau penyesuaian dilakukan dengan lebih cepat. Namun, metode konvensional yang mengandalkan pengukuran manual dianggap kurang praktis, terutama ketika dibutuhkan pemantauan berkelanjutan.

Untuk mengatasi keterbatasan tersebut, dikembangkanlah *Smart Sensor System* berbasis mikrokontroler ESP32, yang terintegrasi dengan sensor DHT22 (pengukur suhu dan kelembaban) serta sensor LDR (pendeteksi cahaya). Data yang dikumpulkan dapat langsung ditampilkan pada layar OLED, memudahkan pengguna dalam memantau kondisi lingkungan secara real-time. Proyek ini juga memanfaatkan platform simulasi Wokwi, memungkinkan pengujian dan pengembangan sistem secara virtual. Dengan demikian, solusi ini tidak hanya hemat biaya tetapi juga fungsional dan siap diimplementasikan dalam skala nyata.

### 1.2. TUJUAN PRAKTIKUM

Praktikum ini bertujuan untuk merancang dan menerapkan sistem pemantauan lingkungan berbasis Internet of Things (IoT) yang dapat mengukur serta menampilkan data suhu, kelembaban, dan intensitas cahaya secara real-time. Sistem ini dikembangkan dengan menggunakan mikrokontroler ESP32 yang terhubung ke sensor DHT22 (untuk suhu dan kelembaban) dan LDR (untuk intensitas cahaya), serta menampilkan hasil pengukuran melalui OLED Display.

## 2. METODOLOGI

### 2.1. ALAT DAN BAHAN

Dalam pelaksanaan praktikum ini, beberapa alat dan bahan digunakan untuk mengimplementasikan sistem lampu lalu lintas berbasis IoT menggunakan ESP32. Alat dan bahan tersebut dibagi menjadi dua kategori utama, yaitu perangkat lunak (software) dan komponen virtual yang digunakan dalam simulasi Wokwi. Berikut adalah penjelasan detailnya:

- Wokwi Simulator  
Platform simulasi online yang memungkinkan pengguna untuk merancang dan menguji sirkuit elektronik secara virtual. Wokwi digunakan untuk membuat sirkuit lampu lalu lintas dengan ESP32 sebagai mikrokontroler utama.
- Visual Studio Code (VSCode)  
Editor kode yang digunakan untuk menulis, mengelola, dan mengembangkan program. VSCode dipilih karena fleksibilitas dan dukungannya yang luas terhadap berbagai bahasa pemrograman.

- PlatformIO  
Ekstensi di VSCode yang menyediakan lingkungan pengembangan terintegrasi untuk pemrograman mikrokontroler, termasuk ESP32. PlatformIO memudahkan proses kompilasi, upload, dan debugging kode.
- Library ESP32  
Beberapa library seperti WiFi.h digunakan untuk mengimplementasikan fitur IoT, seperti konektivitas Wi-Fi dan kontrol jarak jauh.
- ESP32  
Mikrokontroler virtual yang berfungsi sebagai otak dari sistem lampu lalu lintas. ESP32 dipilih karena kemampuannya dalam menghubungkan perangkat ke jaringan IoT.
- OLED Display  
OLED Display pada Wokwi merupakan salah satu komponen output yang berfungsi untuk menampilkan data dari mikrokontroler secara langsung dalam bentuk visual.
- DHT 22  
DHT22 pada Wokwi adalah sensor digital yang digunakan untuk mengukur suhu dan kelembaban udara secara akurat. Sensor ini terhubung langsung dengan mikrokontroler ESP32 dan mengirimkan data dalam format digital, sehingga mudah diproses dan ditampilkan.
- Photoresistor (LDR)  
Photoresistor atau LDR pada Wokwi merupakan sensor cahaya yang digunakan untuk mengukur intensitas pencahayaan di lingkungan sekitar. Sensor ini bekerja dengan mengubah nilai resistansi berdasarkan jumlah cahaya yang diterima; semakin terang cahaya, semakin kecil resistansinya.
- Kabel Virtual  
Digunakan untuk menghubungkan ESP32 dengan OLED, DHT 22, dan Photoresistor (LDR) dan dalam sirkuit virtual.
- Koneksi Internet  
Diperlukan untuk mengakses Wokwi Simulator dan menguji fitur IoT pada kontrol jarak jauh.

### 2.2. LANGKAH IMPLEMENTASI

Implementasi sistem lampu lalu lintas berbasis IoT menggunakan ESP32 pada Wokwi Simulator dengan PlatformIO di Visual Studio Code (VSCode) dilakukan melalui beberapa tahapan. Adapun tahapan yang akan dilakukan dalam praktikum ini adalah sebagai berikut:

#### A. Persiapan Lingkungan Pengembangan

- Instalasi Visual Studio Code (VSCode), unduh dan instal VSCode dari situs resmi
- Instalasi PlatformIO, tambahkan ekstensi PlatformIO di VSCode untuk memudahkan pengembangan kode ESP32.
- Buat Project Baru, membuat project baru di PlatformIO dengan memilih board ESP32 dan framework Arduino.
- Akses Wokwi Simulator, membuka platform Wokwi melalui browser dan pilih ESP32 sebagai mikrokontroler untuk simulasi.

## B. Merancang Sirkuit Virtual di Wokwi

- Tambahkan Komponen: Tambahkan komponen virtual ESP32, DHT 22, Photoresistor (LDR), dan OLED Display, ke dalam workspace Wokwi.
- Hubungkan Komponen: Rangkai komponen sesuai dengan desain sirkuit.
- Verifikasi Sirkuit: Pastikan sirkuit virtual telah terhubung dengan benar sebelum melanjutkan ke tahap pengkodean.

## C. Menulis dan Mengembangkan Kode Program

- Buat File Kode: Buat file main.cpp di dalam folder src pada project PlatformIO.
- Tulis Kode Program: Gunakan bahasa C/C++ untuk menulis kode program. Berikut adalah contoh kode sederhana untuk mengontrol lampu lalu lintas:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

// Definisi pin
#define DHTPIN 4 // Pin GPIO untuk DHT22
#define DHTTYPE DHT22 // Jenis sensor DHT
#define LDR_PIN 34 // Pin analog untuk LDR
#define SCREEN_WIDTH 128 // Lebar OLED
#define SCREEN_HEIGHT 64 // Tinggi OLED
#define OLED_RESET -1 // Reset pin OLED

// Inisialisasi OLED
Adafruit_SSD1306 display(SCREEN_WIDTH,
SCREEN_HEIGHT, &Wire, OLED_RESET);

// Inisialisasi DHT
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  // Inisialisasi Serial Monitor
  Serial.begin(9600);

  // Inisialisasi OLED
  If (!display.begin(SSD1306_SWITCHCAPVCC,
0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for (;;)
  }
  display.display();
  delay(2000);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("Initializing...");
  display.display();

  // Inisialisasi DHT
  dht.begin();
}
```

```
void loop() {
  // Baca suhu dan kelembapan dari DHT22
  float humidity = dht.readHumidity();
```

```
float temperature = dht.readTemperature();
```

```
// Baca intensitas cahaya dari LDR
int lightIntensity = analogRead(LDR_PIN);
```

```
// Periksa apakah pembacaan DHT gagal
if (isnan(humidity) || isnan(temperature)) {
  Serial.println("Failed to read from DHT sensor!");
  return;
}
```

```
// Tampilkan data di Serial Monitor
Serial.print("Temperature: ");
Serial.print(temperature);
Serial.print(" °C\t");
Serial.print("Humidity: ");
Serial.print(humidity);
Serial.print(" %\t");
Serial.print("Light Intensity: ");
Serial.println(lightIntensity);
```

```
// Tampilkan data di OLED
display.clearDisplay();
display.setCursor(0, 0);
display.print("Temp: ");
display.print(temperature);
display.println(" C");
display.print("Humidity: ");
display.print(humidity);
display.println(" %");
display.print("Light: ");
display.print(lightIntensity);
display.println(" lx");
display.display();
```

```
// Tunggu 2 detik sebelum membaca lagi
delay(2000);
}
```

- Integrasi IoT (Opsional): Tambahkan fitur IoT seperti konektivitas Wi-Fi menggunakan library WiFi.h untuk memungkinkan kontrol jarak jauh.

## D. Menguji Sistem di Wokwi Simulator

- Upload Kode ke Wokwi: Salin kode yang telah ditulis di PlatformIO dan tempelkan ke editor kode di Wokwi.
- Jalankan Simulasi: Klik tombol "Run" di Wokwi untuk menjalankan simulasi.
- Debugging: Jika terdapat kesalahan, periksa kode dan sirkuit virtual untuk memastikan semuanya berfungsi dengan benar.

## 3. HASIL DAN PEMBAHASAN

### 3.1. HASIL EKSPERIMEN

**Table 1.** Hasil Implementasi IoT Smart Sensor System

(ESP32, DHT22, LDR, dan OLED Display)

Aspek	Hasil	Keterangan
Pembacaan Suhu (DHT22)	24.00 °C	Nilai stabil, akurat sesuai

		spesifikasi DHT22 ( $\pm 0.5^{\circ}\text{C}$ )
Pembacaan kelembaban (DHT 22)	40.00 %RH	Kelembaban relatif rendah, indikasi lingkungan kering
Pembacaan Cahaya (LDR)	1001 (nilai ADC)	Nilai mentah ADC, perlu dikonversi ke Lux untuk interpretasi yang lebih bermakna
Respons Sistem Real-time	Update setiap 2000 ms	Sesuai dengan delay(2000) dalam kode
Tampilan OLED	Data ditampilkan dengan jelas	Format: Temp, Humidity, Light dalam 3 baris terpisah
Error Handling DHT22	Berfungsi dengan baik	Terdapat pengecekan isnan() untuk nilai humidity dan temperature
Komunikasi Serial	Data tercetak di Serial Monitor	Format: Temperature: 24.00 $^{\circ}\text{C}$ Humidity: 40.00 % Light Intensity: 1001

### 3.2. PEMBAHASAN

Pembahasan hasil eksperimen menunjukkan sistem IoT berhasil memantau parameter lingkungan secara real-time dengan baik, dimana DHT22 memberikan pembacaan suhu ( $24.0^{\circ}\text{C}$ ) dan kelembaban (40% RH) yang stabil sesuai spesifikasi, sementara LDR mendeteksi intensitas cahaya (nilai ADC 1001) yang perlu dikonversi ke satuan Lux untuk interpretasi lebih akurat. Secara keseluruhan, sistem bekerja optimal dengan update data setiap 2 detik, tampilan OLED yang jelas, dan kemampuan deteksi error, meski masih dapat ditingkatkan dengan kalibrasi LDR dan penambahan fitur notifikasi jika melebihi threshold tertentu.

## 4. LAMPIRAN

### 4.1. KODE PROGRAM

```

A. #include <Wire.h>
B. #include <Adafruit_GFX.h>
C. #include <Adafruit_SSD1306.h>
D. #include <DHT.h>
E.
F. // Definisi pin
G. #define DHTPIN 4           // Pin GPIO
                               untuk DHT22

```

```

H. #define DHTTYPE DHT22     // Jenis
                               sensor DHT
I. #define LDR_PIN 34        // Pin
                               analog untuk LDR
J. #define SCREEN_WIDTH 128 // Lebar
                               OLED
K. #define SCREEN_HEIGHT 64 // Tinggi
                               OLED
L. #define OLED_RESET -1    // Reset
                               pin OLED
M.
N. // Inisialisasi OLED
O. Adafruit_SSD1306
   display(SCREEN_WIDTH, SCREEN_HEIGHT,
   &Wire, OLED_RESET);
P.
Q. // Inisialisasi DHT
R. DHT dht(DHTPIN, DHTTYPE);
S.
T. void setup() {
U.   // Inisialisasi Serial Monitor
V.   Serial.begin(9600);
W.
X.   // Inisialisasi OLED
Y.   if
   (!display.begin(SSD1306_SWITCHCAPVCC,
   0x3C)) {
Z.     Serial.println(F("SSD1306
   allocation failed"));
AA.   for (;;)
BB.   }
CC.   display.display();
DD.   delay(2000);
EE.   display.clearDisplay();
FF.   display.setTextSize(1);
GG.   display.setTextColor(SSD1306_WHITE)
   ;
HH.   display.setCursor(0, 0);
II.   display.println("Initializing...");
JJ.   display.display();
KK.
LL.   // Inisialisasi DHT
MM.   dht.begin();
NN. }
OO.
PP. void loop() {
QQ. // Baca suhu dan kelembapan dari
   DHT22

```

