**Perancangan Sistem Pemantauan Suhu, Kelembapan, dan Intensitas Cahaya Menggunakan ESP32, DHT22, dan Sensor LDR**

*Ibnu Jaisyurrahman Faiz*

*Program Studi Teknologi Informasi, Fakultas Vokasi, Universitas Brawijaya*

*Email: (*[*ibnufaiz72ub@student.ub.ac.id*](mailto:ibnufaiz72ub@student.ub.ac.id)*)*

**Abstrak:** Pada proyek ini, dirancang sebuah sistem pemantauan suhu, kelembapan, dan intensitas cahaya menggunakan ESP32 sebagai mikrokontroler utama, sensor DHT22 sebagai alat pengukur suhu dan kelembapan, serta sensor LDR untuk mendeteksi intensitas cahaya. Hasil pengukuran ditampilkan secara real-time pada LCD 16x2. Sistem bekerja dengan membaca data dari sensor secara periodik dan menampilkannya dalam bentuk angka pada LCD. Implementasi dilakukan menggunakan bahasa pemrograman C++ dalam lingkungan Arduino IDE dan Visual Studio Code. Hasil pengujian menunjukkan bahwa sistem dapat menampilkan data suhu dalam satuan Celsius, kelembapan dalam bentuk persentase, serta tingkat cahaya berdasarkan nilai yang terbaca dari sensor LDR. Dengan adanya sistem ini, diharapkan pengguna dapat dengan mudah memantau kondisi lingkungan secara efisien dan praktis.

Kata Kunci : *ESP32, DHT22, LDR, LCD 16x2, Pemantauan Suhu, Kelembapan, Intensitas Cahaya, IoT, Arduino IDE*

**Abstract:** In this project, a monitoring system for temperature, humidity, and light intensity was designed using ESP32 as the main microcontroller, DHT22 sensor for measuring temperature and humidity, and LDR sensor for detecting light intensity. The measurement results are displayed in real-time on a 16x2 LCD. The system operates by periodically reading data from the sensors and displaying it numerically on the LCD. Implementation was carried out using the C++ programming language within the Arduino IDE and Visual Studio Code. Testing results show that the system successfully displays temperature in Celsius, humidity in percentage, and light intensity based on the values read from the LDR sensor. With this system, users can efficiently and practically monitor environmental conditions.

Key Word: *ESP32, DHT22, LDR, 16x2 LCD, Temperature Monitoring, Humidity, Light Intensity, IoT, Arduino IDE*

1. **Pendahuluan**
2. **Latar belakang**

Pemantauan suhu dan kelembaban merupakan aspek penting dalam berbagai bidang, seperti pertanian, kesehatan, industri, dan sistem kontrol lingkungan. Suhu dan kelembaban yang tidak stabil dapat berdampak negatif pada kondisi lingkungan, kualitas produk, serta kenyamanan manusia. Oleh karena itu, diperlukan sistem pemantauan yang dapat memberikan data secara real-time dan akurat.

Dalam perkembangan teknologi saat ini, mikrokontroler seperti ESP32 telah banyak digunakan dalam sistem pemantauan berbasis Internet of Things (IoT) karena memiliki konektivitas Wi-Fi dan kemampuan pemrosesan yang baik. Sensor DHT22 digunakan karena mampu mengukur suhu dan kelembaban dengan akurasi tinggi. Selain itu, LCD 16x2 digunakan sebagai media tampilan agar data dapat langsung terlihat oleh pengguna tanpa perlu perangkat tambahan.

Dengan mengembangkan sistem pemantauan suhu dan kelembaban menggunakan ESP32, sensor DHT22, dan LCD 16x2, pengguna dapat dengan mudah mengetahui kondisi lingkungan secara langsung. Eksperimen ini bertujuan untuk membangun dan menguji sistem yang dapat menampilkan informasi suhu dan kelembaban secara real-time dengan respon yang cepat dan akurat.

1. **Tujuan eksperimen**

* Merancang dan mengimplementasikan sistem pemantauan suhu, kelembaban dan intensitas cahaya menggunakan ESP32, DHT22, sensor LDR dan LCD 16x2.
* Mengukur dan menampilkan data suhu dan intensitas cahaya serta kelembaban secara real-time pada LCD.
* Menguji keakuratan sensor DHT22 dalam membaca suhu dan kelembaban lingkungan.
* Memahami prinsip kerja sensor DHT22 dan integrasinya dengan ESP32.
* Mengevaluasi efektivitas sistem dalam menampilkan data dengan cepat dan stabil.

1. **Metodologi**
2. **Alat dan Bahan**

Pada eksperimen ini, simulasi dilakukan menggunakan Wokwi dan implementasi lebih lanjut dilakukan di Visual Studio Code (VS Code). Berikut adalah daftar alat dan bahan yang digunakan:

* Wokwi Simulator – Untuk simulasi rangkaian secara virtual sebelum implementasi fisik.
* Visual Studio Code (VS Code) – Untuk menulis dan mengunggah kode ke ESP32.
* Arduino IDE / PlatformIO – Plugin tambahan untuk pengembangan kode pada VS Code.
* ESP32 – Mikrokontroler utama yang mengontrol sistem.
* Sensor DHT22 – Sensor untuk mengukur suhu dan kelembaban.
* LCD 16x2 – Tampilan untuk menampilkan hasil pengukuran.
* Sensor LDR

1. **Langkah Implementasi**

Langkah implementasi untuk membuat Monitoring Suhu dan Kelembaban Menggunakan ESP32, Sensor DHT22, dan LCD 16x2 sebagai berikut:

* Buka Wokwi di https://wokwi.com.
* Buat proyek baru dengan memilih ESP32 sebagai mikrokontroler.
* Tambahkan komponen: ESP32, Sensor DHT22 dan LCD 16x2
* Hubungkan kabel sesuai dengan rangkaian berikut:

DHT22

VCC → 3.3V ESP32

GND → GND ESP32

Data → GPIO 21 ESP32

* LCD 16x2

RS → GPIO 19

E → GPIO 23

D4 → GPIO 18

D5 → GPIO 17

D6 → GPIO 16

D7 → GPIO 15

VSS dan RW → GND

VDD dan A → 5V

K → GND

* LDR (Sensor Intensitas Cahaya)

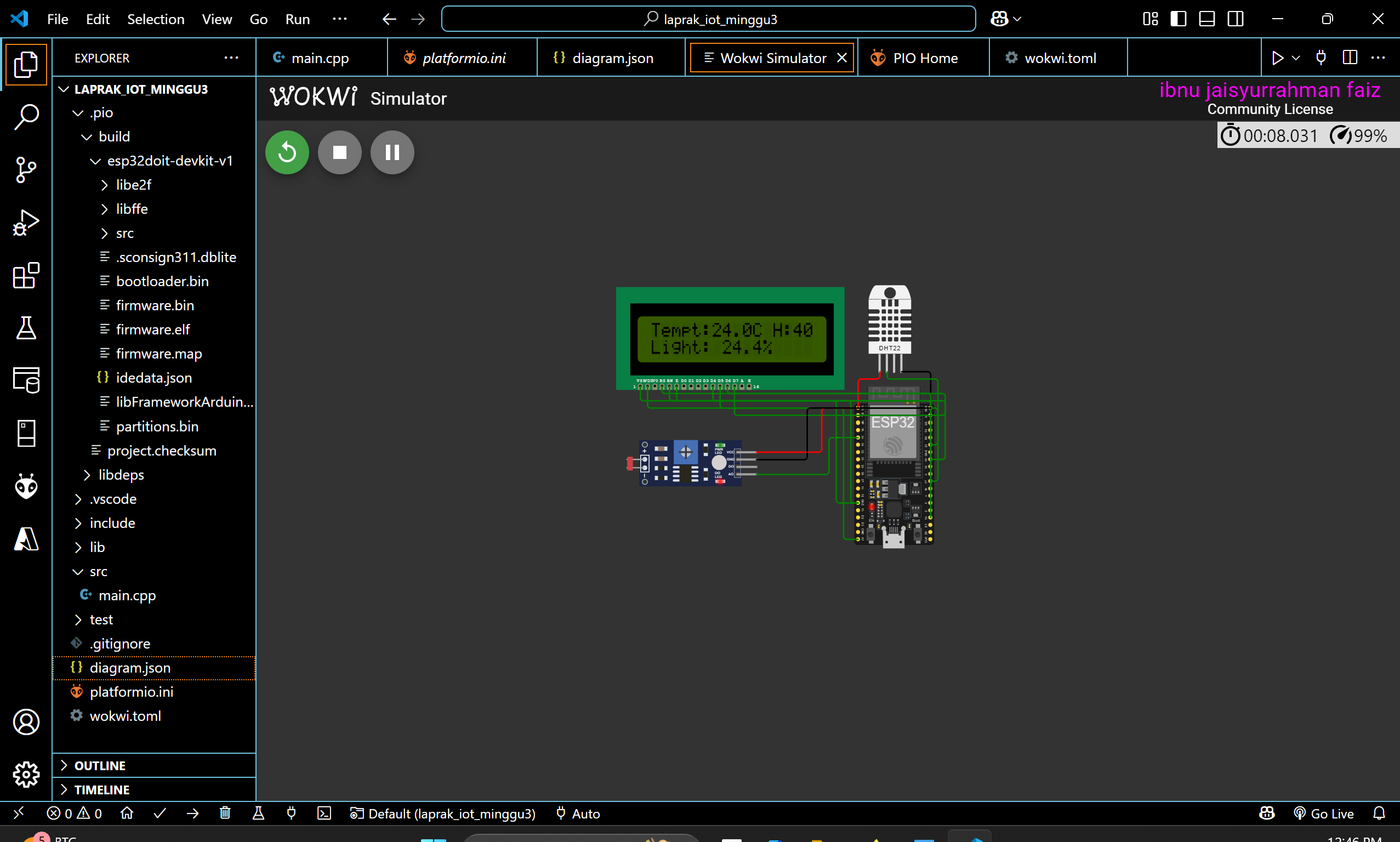
Salah satu kaki LDR → 3.3V ESP32

Kaki LDR lainnya → GPIO 34 ESP32

* Masukkan kode program ke editor Wokwi, lalu jalankan simulasi.
* #include <DHT.h>
* #include <LiquidCrystal.h>
* // Konfigurasi LCD (RS, E, D4, D5, D6, D7)
* LiquidCrystal lcd(19, 23, 18, 17, 16, 15);
* // Konfigurasi DHT22
* #define DHTPIN 21
* #define DHTTYPE DHT22
* #define TIMEDHT 1000
* float humidity, celsius;
* uint32\_t timerDHT = TIMEDHT;
* DHT dht(DHTPIN, DHTTYPE);
* // Konfigurasi LDR
* #define LDRPIN 34  // Pin LDR pada ESP32
* void getSensorData() {
* if ((millis() - timerDHT) > TIMEDHT) {
* timerDHT = millis();
* // Membaca data dari sensor DHT22
* humidity = dht.readHumidity();
* celsius = dht.readTemperature();
* if (isnan(humidity) || isnan(celsius)) {
* Serial.println("Failed to read from DHT sensor!");
* return;
* }
* // Membaca data dari sensor LDR
* int lightValue = analogRead(LDRPIN);
* float intensity = (lightValue / 4095.0) \* 100.0; // Konversi ke persen
* // Menampilkan data di Serial Monitor
* Serial.print("Temp: "); Serial.print(celsius, 1); Serial.println("C");
* Serial.print("Humidity: "); Serial.print(humidity, 1); Serial.println("%");
* Serial.print("Light: "); Serial.print(intensity, 1); Serial.println("%");
* // Menampilkan data di LCD
* lcd.clear();
* lcd.setCursor(0, 0);
* lcd.print("Tempt:"); 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(intensity, 1); lcd.print("%  ");
* }
* }
* void setup() {
* Serial.begin(115200);
* Serial.println("Hello, ESP32!");
* dht.begin();
* lcd.begin(16, 2);
* lcd.print("Starting...");
* delay(2000);
* lcd.clear();
* }
* void loop() {
* getSensorData();
* delay(500);
* }

1. **Hasil dan Pembahasan**
2. **Hasil Eksperimen**

Setelah kode diunggah ke ESP32 menggunakan VS Code (PlatformIO/Arduino IDE), ESP32 mampu mengolah dan mengirim data suhu, kelembapan, dan intensitas cahaya ke LCD serta Serial Monitor tanpa kendala.



1. **Lampiran**

* Kode program main.cpp
* #include <DHT.h>
* #include <LiquidCrystal.h>
* // Konfigurasi LCD (RS, E, D4, D5, D6, D7)
* LiquidCrystal lcd(19, 23, 18, 17, 16, 15);
* // Konfigurasi DHT22
* #define DHTPIN 21
* #define DHTTYPE DHT22
* #define TIMEDHT 1000
* float humidity, celsius;
* uint32\_t timerDHT = TIMEDHT;
* DHT dht(DHTPIN, DHTTYPE);
* // Konfigurasi LDR
* #define LDRPIN 34  // Pin LDR pada ESP32
* void getSensorData() {
* if ((millis() - timerDHT) > TIMEDHT) {
* timerDHT = millis();
* // Membaca data dari sensor DHT22
* humidity = dht.readHumidity();
* celsius = dht.readTemperature();
* if (isnan(humidity) || isnan(celsius)) {
* Serial.println("Failed to read from DHT sensor!");
* return;
* }
* // Membaca data dari sensor LDR
* int lightValue = analogRead(LDRPIN);
* float intensity = (lightValue / 4095.0) \* 100.0; // Konversi ke persen
* // Menampilkan data di Serial Monitor
* Serial.print("Temp: "); Serial.print(celsius, 1); Serial.println("C");
* Serial.print("Humidity: "); Serial.print(humidity, 1); Serial.println("%");
* Serial.print("Light: "); Serial.print(intensity, 1); Serial.println("%");
* // Menampilkan data di LCD
* lcd.clear();
* lcd.setCursor(0, 0);
* lcd.print("Tempt:"); 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(intensity, 1); lcd.print("%  ");
* }
* }
* void setup() {
* Serial.begin(115200);
* Serial.println("Hello, ESP32!");
* dht.begin();
* lcd.begin(16, 2);
* lcd.print("Starting...");
* delay(2000);
* lcd.clear();
* }
* void loop() {
* getSensorData();
* delay(500);
* }
* Kode program diagram.json
* {
* "version": 1,
* "author": "Anonymous maker",
* "editor": "wokwi",
* "parts": [
* { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 105.6, "left": 168.04, "attrs": {} },
* { "type": "wokwi-lcd1602", "id": "lcd1", "top": -25.37, "left": -147.2, "attrs": {} },
* {
* "type": "wokwi-photoresistor-sensor",
* "id": "ldr1",
* "top": 176,
* "left": -133.6,
* "attrs": {}
* },
* { "type": "wokwi-dht22", "id": "dht1", "top": -28.5, "left": 186.6, "attrs": {} }
* ],
* "connections": [
* [ "esp:TX", "$serialMonitor:RX", "", [] ],
* [ "esp:RX", "$serialMonitor:TX", "", [] ],
* [ "dht1:VCC", "esp:3V3", "red", [ "v9.6", "h-28.8", "v38.4" ] ],
* [ "dht1:GND", "esp:GND.2", "black", [ "v0", "h38.4" ] ],
* [ "dht1:SDA", "esp:21", "green", [ "v9.6", "h67.3", "v9.6" ] ],
* [ "lcd1:VSS", "esp:GND.1", "green", [ "v19.2", "h278.4", "v134.4" ] ],
* [ "lcd1:VDD", "esp:5V", "green", [ "v28.8", "h259.3", "v172.8" ] ],
* [ "lcd1:RS", "esp:19", "green", [ "v9.6", "h374.7", "v86.4" ] ],
* [ "lcd1:RW", "esp:GND.1", "green", [ "v19.2", "h220.7", "v134.4" ] ],
* [ "lcd1:E", "esp:23", "green", [ "v19.2", "h355.2", "v19.2" ] ],
* [ "lcd1:D4", "esp:18", "green", [ "v19.2", "h316.8" ] ],
* [ "lcd1:D5", "esp:17", "green", [ "v28.8", "h288.1", "v96" ] ],
* [ "lcd1:D6", "esp:16", "green", [ "v9.6", "h297.8" ] ],
* [ "lcd1:D7", "esp:15", "green", [ "v38.4", "h288.3" ] ],
* [ "ldr1:AO", "esp:34", "green", [ "h96", "v-48.7" ] ],
* [ "ldr1:VCC", "esp:3V3", "red", [ "h86.4", "v-57.6" ] ],
* [ "ldr1:GND", "esp:GND.2", "black", [ "h67.2", "v9.2" ] ]
* ],
* "dependencies": {}
* }