**LAPORAN PRAKTIKUM**

**IOT MINGGU KE-3**

**Indikator Suhu, Kelembapan & Intensitas Cahaya**

**Menggnakan Esp32**

****

**Dosen Pengampu :**

**Ir. Subairi, ST., MT., IPM**

**Disusun Oleh:**

**Muhammad Alif Aris**

**(233140707111077)**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**2025**

**Abstrak**

Banyak aplikasi, seperti manajemen ruangan dan pertanian cerdas, membutuhkan pemantauan lingkungan. Tujuan dari penelitian ini adalah untuk membuat sistem monitoring berbasis ESP32 yang dapat mengukur suhu, kelembapan, dan intensitas cahaya secara real-time. Sistem ini menggunakan sensor DHT22 untuk mengukur suhu dan kelembapan, dan sensor LDR untuk mengukur intensitas cahaya. Hasilnya ditampilkan pada serial monitor. Untuk memastikan stabilitas dan akurasi sistem, simulasi dilakukan menggunakan platform Wokwi. Hasil eksperimen menunjukkan bahwa sistem berjalan seperti yang direncanakan. Menunjukkan integrasi yang berhasil antara perangkat keras dan perangkat lunak, data sensor ditampilkan dengan akurasi tinggi secara real-time. Selain menawarkan alat untuk mengajarkan pemahaman praktis tentang teknologi IoT, protokol ini membantu mengembangkan sistem berbasis IoT yang lebih kompleks seperti pertanian cerdas dan otomatisasi rumah pintar.

*Keywords - ESP32, monitoring lingkungan, suhu, kelembapan, intensitas cahaya, sensor DHT22, sensor LDR, IoT, simulasi Wokwi, otomatisasi, rumah pintar, pertanian cerdas, serial monitor.*

**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Pemantauan suhu, kelembapan, dan intensitas cahaya sangat penting untuk berbagai aplikasi, seperti manajemen ruangan dan aplikasi pertanian kontemporer. Ketiga parameter ini saling berhubungan dan berdampak pada kondisi lingkungan umum. Faktor-faktor seperti suhu yang tidak stabil, kelembapan yang terlalu tinggi atau rendah, dan pencahayaan yang tidak memadai dapat menyebabkan penurunan produktivitas, kenyamanan, dan efisiensi energi. Oleh karena itu, untuk memungkinkan penyesuaian lingkungan yang cepat, diperlukan alat pemantau yang dapat memberikan data secara real-time.

Selain menawarkan solusi untuk monitoring, proyek ini menawarkan kesempatan untuk meningkatkan pemahaman kita tentang pengelolaan data sensor dan teknologi Internet of Things. Sistem ini dapat dikembangkan lebih lanjut untuk otomatisasi, seperti mengatur lampu atau pendingin ruangan berdasarkan kondisi lingkungan, dengan kemampuan ESP32 untuk memproses dan mengirimkan data. Inovasi seperti ini meningkatkan efisiensi sumber daya dan membuat lingkungan lebih nyaman dan terkontrol.

**1.2 Tujuan Eksperimen**

1. Uji sensor suhu, kelembapan, dan intensitas cahaya untuk mendeteksi perubahan kondisi lingkungan dalam waktu nyata.
2. Dengan menggunakan program IDE Arduino, pelajari cara mengkodekan mikrokontroler ESP32 dan mengendalikan data sensor.
3. Mengembangkan prototipe sistem monitoring yang hemat energi dan efisien yang dapat diterapkan untuk berbagai aplikasi, seperti rumah pintar dan pertanian cerdas.

**BAB II**

**METODOLOGI**

**2.1 Alat dan Bahan**

* Mikrokontroler ESP32 (virtual)
* Sensor suhu dan kelembapan (DHT11/DHT22)
* Sensor intensitas cahaya (LDR)
* Resistor (1kΩ dan 10kΩ)
* Breadboard
* Kabel penghubung

**2.3 Implementasi Sistem**

1. Menghubungkan ESP32, DHT11/DHT22, LDR, dan resistor ke breadboard virtual untuk menyusun rangkaian virtual di Wokwi.
2. Dengan menulis kode Arduino, Anda dapat membaca suhu, kelembapan, dan intensitas cahaya dan menampilkan hasilnya pada monitor serial.
3. Untuk menjalankan simulasi, unggah kode ke ESP32 secara virtual di platform Wokwi.
4. Monitor hasil pembacaan sensor di Wokwi secara real-time.
5. Jika terdapat ketidaksesuaian hasil, periksa logika program atau koneksi rangkaian virtual.

**BAB III**

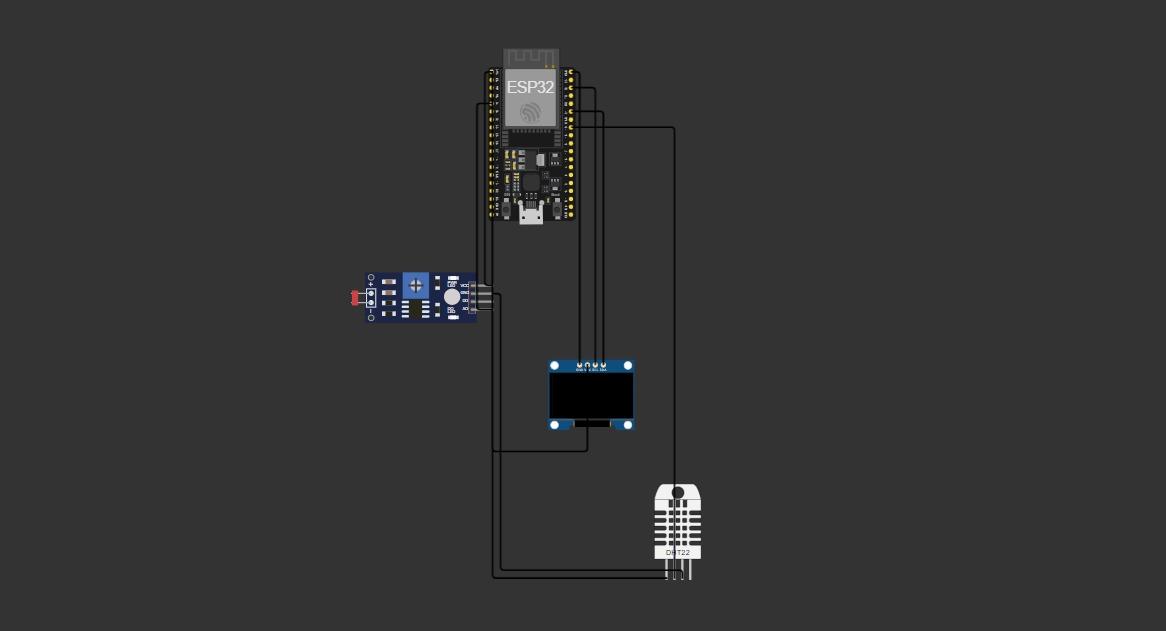
**HASIL DAN PEMBAHASAN**

**3.1 Hasil Eksperimen**

Hasil eksperimen menunjukkan bahwa sistem pemantauan suhu, kelembapan, dan intensitas cahaya berbasis ESP32 berjalan sesuai dengan desain pada simulasi Wokwi. Serial monitor dapat membaca dan menampilkan data suhu, kelembapan, dan intensitas cahaya dari LDR secara real-time. Kode pemrograman dan rangkaian virtual bekerja dengan benar, memberikan hasil yang konsisten sesuai skenario simulasi.

Untuk memastikan respons sensor yang lebih cepat dan stabil, beberapa optimasi dilakukan, terutama untuk pengambilan data terus-menerus. Selain itu, sistem ini menunjukkan kemampuan untuk mengintegrasikan berbagai sensor dengan ESP32 untuk aplikasi pemantauan. Eksperimen ini membantu kita memahami bagaimana perangkat Internet of Things sederhana berfungsi, dan menawarkan peluang untuk pengembangan lebih lanjut untuk otomatisasi berbasis lingkungan.

**3.2 Dokumentasi eksperimen meliputi screenshoot simulasi ESP32 :**

****

**Lampiran**

**Kode Program 1 :**

{

"version": 1,

"author": "Anonymous maker",

"editor": "wokwi",

"parts": [

{

"type": "board-esp32-devkit-c-v4",

"id": "esp",

"top": 28.8,

"left": -110.36,

"rotate": 9000,

"attrs": {}

},

{ "type": "wokwi-dht22", "id": "dht1", "top": 557.1, "left": 90.6, "attrs": {} },

{

"type": "wokwi-photoresistor-sensor",

"id": "ldr1",

"top": 300.8,

"left": -277.6,

"attrs": {}

},

{

"type": "board-ssd1306",

"id": "oled1",

"top": 406.34,

"left": -38.17,

"attrs": { "i2cAddress": "0x3c" }

}

],

"connections": [

[ "esp:TX", "$serialMonitor:RX", "", [] ],

[ "esp:RX", "$serialMonitor:TX", "", [] ],

[ "dht1:DATA", "esp:19", "green", [ "v0" ] ],

[ "oled1:SDA", "esp:21", "black", [ "v0" ] ],

[ "oled1:SCL", "esp:22", "black", [ "v0" ] ],

[ "oled1:GND", "esp:GND.2", "black", [ "v0" ] ],

[ "oled1:VCC", "esp:3V3", "black", [ "v105.6", "h-105.45" ] ],

[ "dht1:SDA", "esp:19", "black", [ "v0" ] ],

[ "dht1:VCC", "esp:3V3", "black", [ "v0", "h-259.2" ] ],

[ "ldr1:AO", "esp:34", "black", [ "h-19.2", "v-0.7" ] ],

[ "ldr1:VCC", "esp:3V3", "black", [ "h-9.6", "v96" ] ],

[ "ldr1:GND", "dht1:NC", "black", [ "h9.6", "v335.6", "h259.2" ] ]

],

"dependencies": {}

}

**Kode Program 2 :**

#include <Arduino.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <DHT.h>

#define DHTPIN 19

#define DHTTYPE DHT22

#define TIMEDHT 1000

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

#define LDR\_PIN 34

float humidity, celsius;

uint32\_t timerDHT = TIMEDHT;

DHT dht(DHTPIN, DHTTYPE);

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

void getTemperature() {

if ((millis() - timerDHT) > TIMEDHT) {

timerDHT = millis();

humidity = dht.readHumidity();

celsius = dht.readTemperature();

if (isnan(humidity) || isnan(celsius)) {

Serial.println("Gagal membaca sensor DHT!");

return;

}

}

}

void setup() {

Serial.begin(115200);

dht.begin();

if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

Serial.println("OLED gagal diinisialisasi");

for (;;);

}

display.clearDisplay();

}

void loop() {

getTemperature();

int lightValue = analogRead(LDR\_PIN);

Serial.print("Temp: ");

Serial.print(celsius);

Serial.print(" C, Humidity: ");

Serial.print(humidity);

Serial.print("%, Light: ");

Serial.println(lightValue);

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(0, 0);

display.print("Temp: ");

display.print(celsius);

display.println(" C");

display.setCursor(0, 16);

display.print("Humidity: ");

display.print(humidity);

display.println(" %");

display.setCursor(0, 32);

display.print("Light: ");

display.print(lightValue);

display.display();

delay(1000);

}