**Pemantauan Suhu dan Kelembapan Berbasis ESP32 dan Blynk**

*Difa Aqilah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email :* [*difaaqilah04@student.ub.ac.id*](mailto:difaaqilah04@student.ub.ac.id)

**Abstrak**

Seiring berkembangnya teknologi Internet of Things (IoT), pemantauan kondisi lingkungan kini bisa dilakukan secara real-time dan jarak jauh. Praktikum ini bertujuan untuk membuat sistem pemantauan suhu dan kelembapan menggunakan sensor DHT22 yang dikendalikan oleh mikrokontroler ESP32, serta terhubung ke platform Blynk melalui koneksi WiFi. Sistem ini tidak hanya memungkinkan pengguna untuk melihat data suhu dan kelembapan dari smartphone, tapi juga mengontrol LED secara digital. Proyek ini dikembangkan menggunakan simulator Wokwi dan dikodekan di Visual Studio Code dengan PlatformIO. Hasil akhir menunjukkan sistem dapat bekerja stabil, dengan pembaruan data setiap satu detik. Meski masih ada keterbatasan dari sisi sensor dan jaringan, sistem ini memberikan dasar yang kuat untuk pengembangan lebih lanjut seperti penambahan sensor lain atau fitur notifikasi otomatis.

Kata Kunci — *ESP32, DHT22, Blynk, IoT, Wokwi, PlatformIO*

**Pendahuluan**

Dalam kehidupan sehari-hari, suhu dan kelembapan adalah dua parameter penting yang perlu dipantau, baik di rumah, kantor, peternakan, hingga pertanian. Melalui teknologi IoT, data-data ini bisa dipantau langsung dari perangkat pintar hanya dengan koneksi internet. Untuk itulah, praktikum ini dirancang.

Kami menggunakan sensor DHT22 untuk membaca suhu dan kelembapan lingkungan, yang datanya diproses oleh mikrokontroler ESP32 dan dikirim ke aplikasi Blynk. Di Blynk, pengguna bisa melihat nilai suhu dan kelembapan secara langsung, serta mengontrol LED yang tersambung ke board.

**Metodologi**

**Alat dan Bahan**

* ESP32 DevKit V1
* Sensor DHT22
* LED dan resistor
* Koneksi WiFi
* Visual Studio Code + PlatformIO
* Akun Blynk dan aplikasi Blynk
* Platform Wokwi (untuk simulasi)

**Langkah-Langkah**

1. Buat akun dan login ke <https://blynk.cloud>
2. Buat Template baru, pilih hardware: ESP32, dan connection type: WiFi
3. Tambahkan Datastream:
   * V0: Temperature
   * V1: Humidity
   * V2: Switch untuk LED
4. Buat **New Device** dari template → dapatkan **Auth Token**
5. Di aplikasi Blynk, tambahkan widget **Gauge**, **Value Display**, dan **Switch**
6. Buat proyek di VSCode dengan Board esp32doit-devkit-v1, install library:
   * Blynk
   * DHTesp
7. Sambungkan rangkaian di Wokwi seperti pada file diagram.json
8. Tulis kode pada main.cpp (lihat bagian bawah)
9. Build & upload program ke ESP32
10. Buka Serial Monitor dan pantau data di aplikasi Blynk

**Konfigurasi Diagram (diagram.json)**

{

"version": 1,

"author": "Difa Aqilah",

"editor": "wokwi",

"parts": [

{ "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -283.3, "left": 52.6, "attrs": {} },

{

"type": "wokwi-led",

"id": "led1",

"top": -224.4,

"left": -101.8,

"attrs": { "color": "red" }

},

{

"type": "wokwi-resistor",

"id": "r5",

"top": -168.85,

"left": -48,

"attrs": { "value": "1000" }

},

{

"type": "wokwi-dht22",

"id": "dht1",

"top": -260.42,

"left": 247.56,

"attrs": { "temperature": "58.7", "humidity": "77" }

}

],

"connections": [

[ "esp:TX0", "$serialMonitor:RX", "", [] ],

[ "esp:RX0", "$serialMonitor:TX", "", [] ],

[ "led1:A", "r5:1", "red", [ "v0" ] ],

[ "r5:2", "esp:D26", "red", [ "v1.2", "h17.93", "v81.46" ] ],

[ "dht1:VCC", "esp:VIN", "red", [ "v135.92", "h-214.56", "v-115.2" ] ],

[ "dht1:GND", "esp:GND.1", "black", [ "v126.32", "h-128.16", "v-115.1" ] ],

[ "dht1:SDA", "esp:D15", "green", [ "v145.52", "h-99.26", "v-143.8" ] ],

[ "led1:C", "esp:GND.2", "black", [ "v48", "h144.4" ] ]

],

"dependencies": {}

}

**Kode Program (main.cpp)**

#define BLYNK\_PRINT Serial

#define BLYNK\_TEMPLATE\_ID "TMPL6Odp-RXEs"

#define BLYNK\_TEMPLATE\_NAME "Monitor Suhu"

#define BLYNK\_AUTH\_TOKEN "Mq7Cyzgu5ozhsFOX0WGlWSbiGnc4\_UGF"

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#include <DHTesp.h>

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "Wokwi-GUEST";

char pass[] = "";

const int DHT\_PIN = 15;

int value0, value1, value2, value3, value6;

byte LED\_R = 26;

byte LED\_Y = 27;

byte LED\_G = 14;

byte LED\_B = 12;

DHTesp dht;

BlynkTimer timer;

void sendSensor() {

TempAndHumidity data = dht.getTempAndHumidity();

Serial.print("% Temperature: ");

Serial.print(data.temperature);

Serial.println("C ");

Serial.print("% Kelembaban: ");

Serial.print(data.humidity);

Serial.println("% ");

Blynk.virtualWrite(V0, data.temperature);

Blynk.virtualWrite(V1, data.humidity);

}

BLYNK\_WRITE(V2) {

int nilaiBacaIO = param.asInt();

digitalWrite(LED\_R, nilaiBacaIO);

Blynk.virtualWrite(V3, nilaiBacaIO);

}

void setup() {

Serial.begin(115200);

dht.setup(DHT\_PIN, DHTesp::DHT22);

pinMode(LED\_R, OUTPUT);

pinMode(LED\_Y, OUTPUT);

pinMode(LED\_G, OUTPUT);

pinMode(LED\_B, OUTPUT);

Blynk.begin(auth, ssid, pass);

#ifdef WOKWI

Blynk.connectWiFi(ssid, pass);

Blynk.config(auth);

#endif

timer.setInterval(1000, sendSensor);

}

void loop() {

Blynk.run();

timer.run();

}

**Hasil dan Pembahasan**

Setelah semua konfigurasi selesai, sistem berhasil membaca suhu dan kelembapan melalui sensor DHT22 dan menampilkannya secara real-time di aplikasi Blynk. Widget Value Display dan Gauge berhasil menampilkan data, sedangkan tombol Switch dapat digunakan untuk menghidupkan dan mematikan LED merah yang terhubung ke pin D26. Respons sensor cukup stabil dan cepat, dengan pembaruan tiap detik. Namun, karena sistem bergantung pada koneksi WiFi, terkadang bisa terjadi delay atau putus koneksi, terutama saat sinyal lemah. Selain itu, sensor DHT22 memiliki akurasi yang cukup, walau tidak setinggi sensor kelas industri seperti BME280.

**Dokumentasi hasil uji coba:**

