**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)**

**Praktik Koneksi Sensor Suhu dan Kelembaban ke Blynk Cloud Menggunakan ESP32**

**Salma Salsabila**  
Fakultas Vokasi Universitas Brawijaya  
Email: salmasalsabila@student.ub.ac.id

**Abstrak**

Praktikum ini bertujuan untuk merancang dan mengimplementasikan sistem monitoring suhu dan kelembaban berbasis Internet of Things (IoT) menggunakan mikrokontroler ESP32, sensor DHT22, dan layanan cloud Blynk. Sistem ini dirancang agar dapat membaca data lingkungan secara berkala, kemudian mengirimkan data tersebut ke Blynk Cloud melalui koneksi WiFi. Seluruh proses pengembangan dilakukan menggunakan Visual Studio Code dengan PlatformIO sebagai sistem manajemen proyek. Hasil eksperimen menunjukkan bahwa data suhu dan kelembaban dapat ditampilkan secara real-time pada dashboard Blynk, baik melalui aplikasi web maupun mobile. Pendekatan ini membuktikan efisiensi integrasi perangkat keras dan cloud dalam pengembangan sistem IoT sederhana.

**Kata Kunci**: IoT, ESP32, Blynk, Sensor DHT22, VS Code, PlatformIO

**Pendahuluan**

**Latar Belakang**

Perkembangan teknologi Internet of Things (IoT) membuka banyak peluang dalam pengembangan sistem berbasis data real-time, khususnya dalam monitoring lingkungan. Salah satu komponen penting dari sistem monitoring adalah kemampuan untuk membaca parameter lingkungan, seperti suhu dan kelembaban, dan mengirimkannya ke cloud untuk dipantau dari jarak jauh.

IoT memadukan antara sensor, mikrokontroler, koneksi internet, dan layanan cloud. Dalam eksperimen ini, digunakan ESP32 sebagai mikrokontroler yang memiliki kemampuan koneksi WiFi bawaan, serta sensor DHT22 untuk mengukur suhu dan kelembaban. Platform Blynk dipilih sebagai media pengiriman dan visualisasi data karena mendukung pengembangan cepat dan efisien.

Pendekatan ini menghilangkan kebutuhan akan perangkat pemroses lokal yang kompleks dan menggantinya dengan pengolahan data di cloud. Data dapat dilihat dari mana saja, kapan saja, melalui perangkat apapun yang terhubung internet. Dengan demikian, eksperimen ini sekaligus menjadi implementasi nyata konsep IoT dalam konteks pengukuran lingkungan secara nirkabel dan real-time.

**Tujuan Eksperimen**

1. Mengembangkan sistem monitoring suhu dan kelembaban berbasis ESP32 dan sensor DHT22.
2. Menghubungkan ESP32 ke layanan Blynk Cloud menggunakan koneksi WiFi.
3. Mengirim dan menampilkan data sensor secara real-time di dashboard Blynk.
4. Menguji kestabilan dan akurasi pengiriman data dari mikrokontroler ke cloud.
5. Mengetahui alur kerja integrasi perangkat keras dan cloud dalam proyek IoT sederhana.

**Metodologi**

**Alat dan Bahan**

1. Mikrokontroler: ESP32 DevKit V1
2. Sensor: DHT22 (untuk pengukuran suhu dan kelembaban)
3. Jaringan: WiFi lokal
4. Platform Cloud: Blynk (<https://blynk.cloud>)
5. Software:
   * Visual Studio Code
   * PlatformIO Extension
   * Library Arduino: Blynk, DHTesp

**Langkah-Langkah Implementasi**

**1. Konfigurasi Blynk Cloud**

1. Membuat akun di <https://blynk.cloud>.
2. Membuat Template baru dengan:
   1. Board: ESP32 Dev Board
   2. Connection: WiFi
3. Membuat *Datastream*:
   1. V4: suhu (type: Double)
   2. V5: kelembaban (type: Double)
4. Membuat Device dari template, lalu mencatat Auth Token untuk digunakan dalam kode ESP32.

**2. Pembuatan Proyek di VS Code**

* Membuat project baru menggunakan PlatformIO di Visual Studio Code.
* Menambahkan dependensi di platformio.ini:

ini

CopyEdit

lib\_deps =

blynkkk/Blynk@^1.1.0

beegee-tokyo/DHTesp@^1.2.3

* Menulis kode Arduino untuk membaca data dari sensor DHT22 dan mengirimkan ke virtual pin Blynk (V4 dan V5).
* Memasukkan kredensial WiFi dan Auth Token.

**Hasil dan Pembahasan**

Seluruh tahapan pengembangan dan implementasi berhasil dilakukan tanpa kendala yang berarti. ESP32 mampu terhubung ke jaringan WiFi dengan stabil dan mulai mengirimkan data ke server Blynk kurang dari 5 detik setelah dinyalakan. Auth Token yang dimasukkan juga valid, sehingga komunikasi antara perangkat dan server berlangsung dengan baik.

Data kelembaban yang ditampilkan di Serial Monitor sesuai dengan nilai default simulasi. Nilai tersebut kemudian muncul di dashboard Blynk secara berkala tiap satu detik, sesuai interval yang diatur pada BlynkTimer.

Fitur visualisasi pada Blynk juga sangat membantu. Dengan widget Gauge atau Value Display, pengguna dapat langsung melihat kondisi lingkungan secara real-time tanpa perlu membuka log teknis. Hal ini sangat bermanfaat dalam konteks smart monitoring atau sistem peringatan dini.

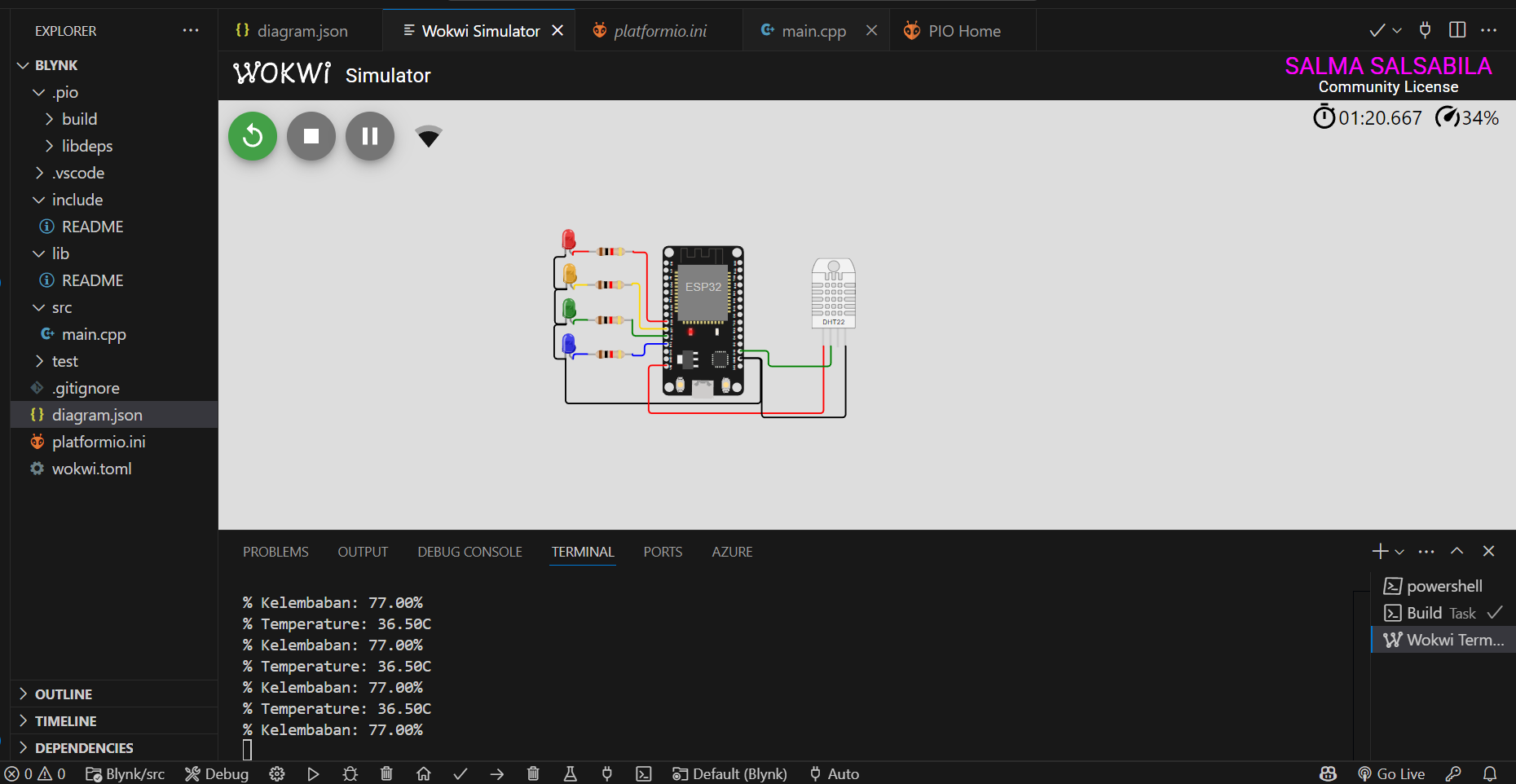
Dalam eksperimen ini, tidak terjadi kehilangan data (data loss) ataupun disconnect mendadak. Ini menunjukkan kestabilan koneksi dan efisiensi protokol yang digunakan oleh Blynk. Praktikum ini juga memberikan pemahaman tentang pentingnya sinkronisasi antara kode program, datastream, dan widget visual agar sistem berjalan dengan sempurna.

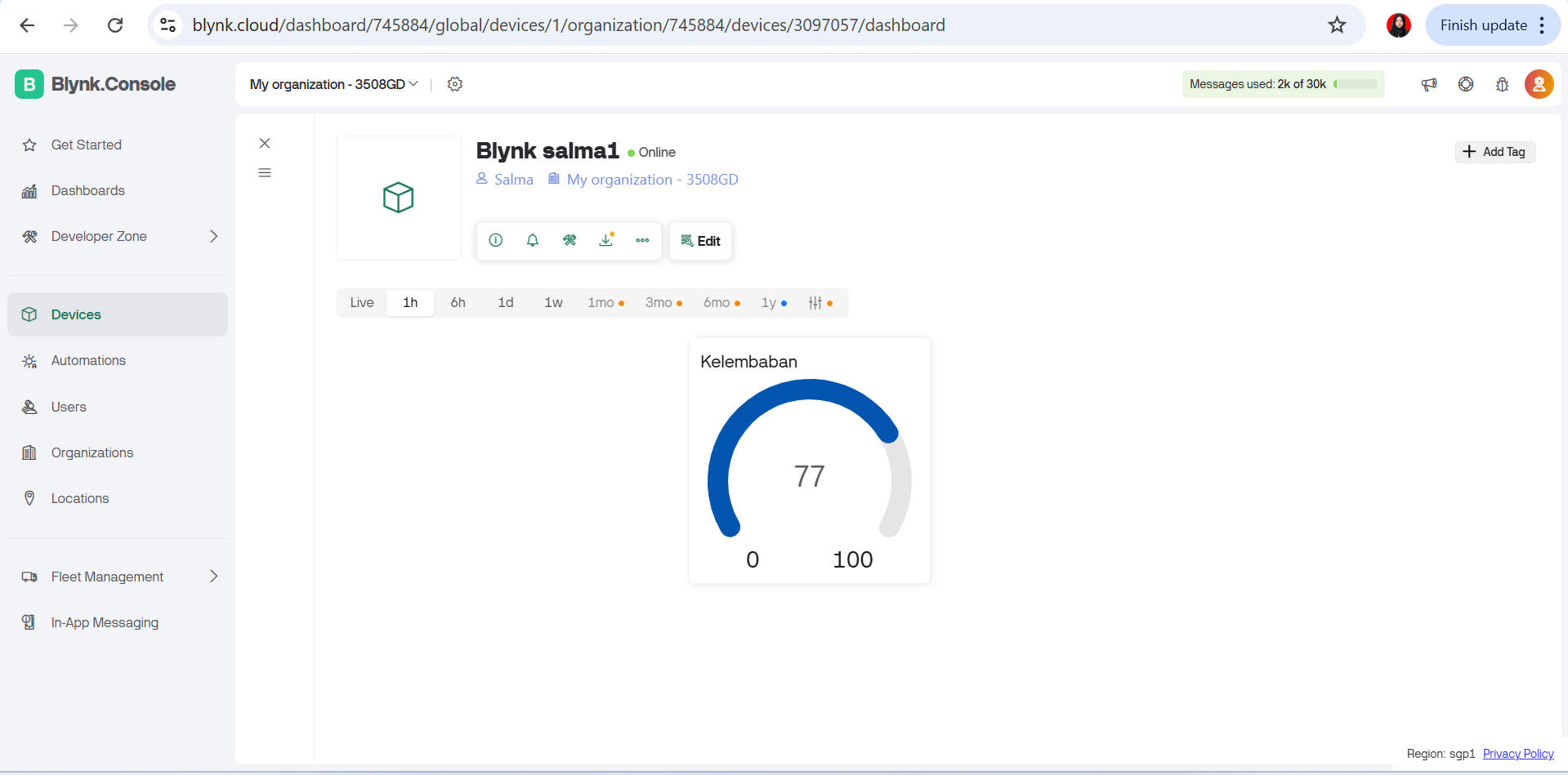
**Kesimpulan**

Dari eksperimen yang dilakukan, dapat disimpulkan bahwa:

1. Sistem monitoring suhu dan kelembaban berbasis ESP32 dan DHT22 dapat dibangun dengan efisien menggunakan VS Code dan PlatformIO.
2. Koneksi ke Blynk Cloud berhasil dilakukan menggunakan WiFi dengan autentikasi token.
3. Data dari sensor berhasil dikirim dan divisualisasikan secara real-time pada dashboard Blynk.
4. Platform Blynk mempermudah integrasi perangkat IoT tanpa perlu pengaturan server atau backend secara manual.

**Lampiran**

****

****

**Kode Program**

#define BLYNK\_TEMPLATE\_ID "TMPL6MdVmMxPI"

#define BLYNK\_TEMPLATE\_NAME "Blynk salma1"

#define BLYNK\_AUTH\_TOKEN "UA0SYM87CcMOosdcrGKi9kg7zGaUzeAL"

#define BLYNK\_PRINT Serial

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#include <DHTesp.h> // Library untuk DHT

char auth[] = BLYNK\_AUTH\_TOKEN; // Auth Token

char ssid[] = "Wokwi-GUEST"; // Nama hotspot yang digunakan

char pass[] = ""; // Password hotspot yang digunakan

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 cekAllLed(); // Tambahkan ini sebelum BLYNK\_WRITE() pertama

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(V4, data.temperature); // Kirim data ke Blynk

    Blynk.virtualWrite(V5, data.humidity);

}

BLYNK\_WRITE(V0) {

    value0 = param.asInt();

    digitalWrite(LED\_R, value0);

    Blynk.virtualWrite(V7, value0);

    cekAllLed();

}

BLYNK\_WRITE(V1) {

    value1 = param.asInt();

    digitalWrite(LED\_Y, value1);

    Blynk.virtualWrite(V8, value1);

    cekAllLed();

}

BLYNK\_WRITE(V2) {

    value2 = param.asInt();

    digitalWrite(LED\_G, value2);

    Blynk.virtualWrite(V9, value2);

    cekAllLed();

}

BLYNK\_WRITE(V3) {

    value3 = param.asInt();

    digitalWrite(LED\_B, value3);

    Blynk.virtualWrite(V10, value3);

    cekAllLed();

}

BLYNK\_WRITE(V6) {

    value6 = param.asInt();

    digitalWrite(LED\_R, value6);

    Blynk.virtualWrite(V7, value6);

    Blynk.virtualWrite(V0, value6);

    digitalWrite(LED\_Y, value6);

    Blynk.virtualWrite(V8, value6);

    Blynk.virtualWrite(V1, value6);

    digitalWrite(LED\_G, value6);

    Blynk.virtualWrite(V9, value6);

    Blynk.virtualWrite(V2, value6);

    digitalWrite(LED\_B, value6);

    Blynk.virtualWrite(V10, value6);

    Blynk.virtualWrite(V3, value6);

}

void cekAllLed() {

    if (value0 == 1 && value1 == 1 && value2 == 1 && value3 == 1) {

        Blynk.virtualWrite(V6, 1);

    } else {

        Blynk.virtualWrite(V6, 0);

    }

}

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);

    // Jika menggunakan Wokwi, gunakan mode simulasi WiFi

    #ifdef WOKWI

        Blynk.connectWiFi(ssid, pass);

        Blynk.config(auth);

    #endif

    timer.setInterval(1000, sendSensor); // Kirim data setiap 1000ms

}

void loop() {

    Blynk.run();

    timer.run();

}

**platformio.ini**

ini

CopyEdit

[env:esp32dev]

platform = espressif32

board = esp32dev

framework = arduino

monitor\_speed = 115200

lib\_deps =

blynkkk/Blynk@^1.1.0

beegee-tokyo/DHTesp@^1.2.3