**PRAKTIKUM SIMULASI SISTEM MONITORING SUHU, KELEMBAPAN,**

**DAN KONTROL LED BERBASIS ESP32 DENGAN**

**PLATFORM BLYNK DI WOKWI**

*Fredlina Devhania Kholishah1*

*Fakultas Vokasi, Universitas Brawijaya*

*devhania88@gmail.com*

### **ABSTRAK**

Praktikum ini bertujuan untuk menerapkan sistem IoT dengan menggunakan ESP32 sebagai mikrokontroler utama. Sistem ini berfungsi untuk membaca data suhu dan kelembapan dari sensor DHT22 serta mengendalikan LED melalui platform Blynk. Pengujian dilakukan secara virtual menggunakan simulator Wokwi untuk mengevaluasi fungsi sistem tanpa perangkat keras nyata. Hasil yang diperoleh menunjukkan bahwa sistem berhasil menampilkan data suhu dan kelembapan di aplikasi Blynk serta mampu mengontrol LED dari jarak jauh. Keberhasilan komunikasi antara ESP32 dan Blynk sangat dipengaruhi oleh penerapan Virtual Pin yang tepat.

***Kata Kunci:*** *IoT, ESP32, DHT22, Blynk, Kontrol LED, Virtual Pin*

### **ABSTRACT**

This practical project applies an IoT system using the ESP32 as the main microcontroller to read temperature and humidity data from a DHT22 sensor and control an LED using the Blynk platform. The simulation is carried out in Wokwi to verify the system’s functionality without the need for physical hardware. The results show that the system successfully reads and displays temperature and humidity data on the Blynk app and can control the LED remotely via the interface. Proper use of Virtual Pins plays a key role in the communication between ESP32 and Blynk.

***Keywords:*** *IoT, ESP32, DHT22, Blynk, LED Control, Virtual Pin*

### **1. PENDAHULUAN**

#### 1.1 Latar Belakang

Kemajuan teknologi Internet of Things (IoT) telah memberikan peluang besar dalam menghubungkan perangkat fisik ke jaringan internet, sehingga memungkinkan pengguna untuk melakukan pemantauan dan pengendalian perangkat secara remote. Praktikum ini dirancang untuk mengimplementasikan sistem IoT sederhana yang mampu mendeteksi kondisi lingkungan dan mengendalikan perangkat secara daring. ESP32 dipilih sebagai otak dari sistem karena kemampuannya yang sudah terintegrasi dengan fitur WiFi dan Bluetooth, menjadikannya sangat cocok untuk digunakan dalam berbagai proyek IoT. Sementara itu, platform Blynk digunakan sebagai antarmuka pengguna karena kemudahannya dalam merancang aplikasi IoT tanpa harus menguasai pemrograman web atau mobile secara mendalam.

#### 1.2 Tujuan Praktikum

1. Memahami proses integrasi ESP32 dengan platform Blynk
2. Menerapkan pembacaan sensor DHT22 dan mengirimkan datanya ke cloud
3. Mengendalikan LED secara remote melalui aplikasi Blynk
4. Memahami konsep penggunaan Virtual Pin dalam komunikasi IoT
5. Mengidentifikasi dan menyelesaikan kendala dalam pengembangan sistem IoT

### **2. METODOLOGI**

#### 2.1 Alat dan Bahan

Alat:

* Laptop/PC dengan koneksi internet
* Platform simulasi online Wokwi
* Visual Studio Code (VSCode)
* Aplikasi dan web Blynk
* Browser (Chrome, Firefox, dsb.)
* Ekstensi PlatformIO untuk VSCode
* Ekstensi Wokwi untuk VSCode

Bahan:

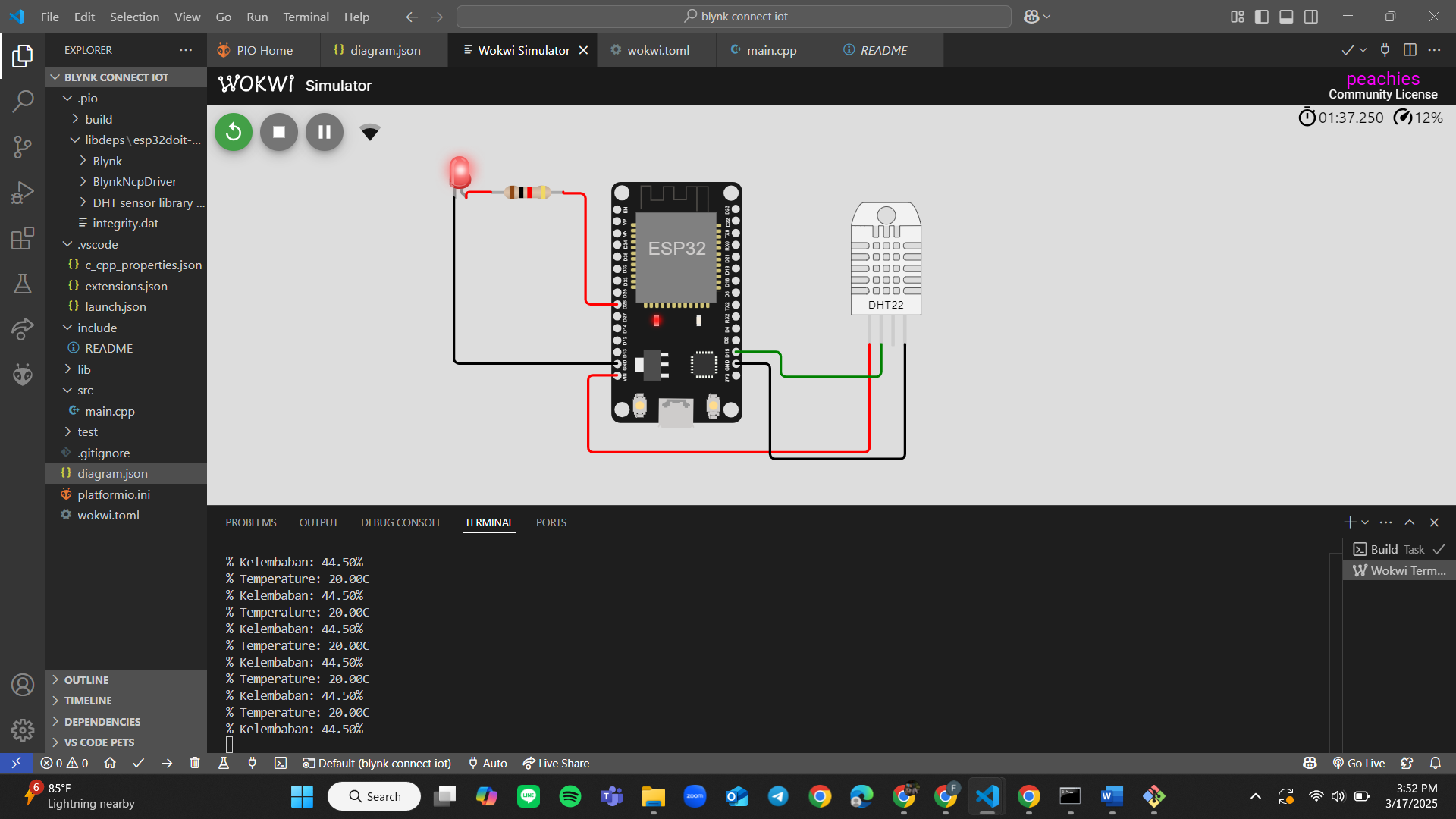
* ESP32 DevKit (virtual melalui Wokwi)
* LED virtual
* Sensor DHT22 virtual
* Resistor virtual 220 ohm
* Bahasa pemrograman C++ dengan Arduino
* Library: WiFi.h, BlynkSimpleEsp32.h, DHTesp.h
* Token autentikasi dari Blynk
* Template ID dan Template Name dari Blynk
* File diagram.json dari Wokwi

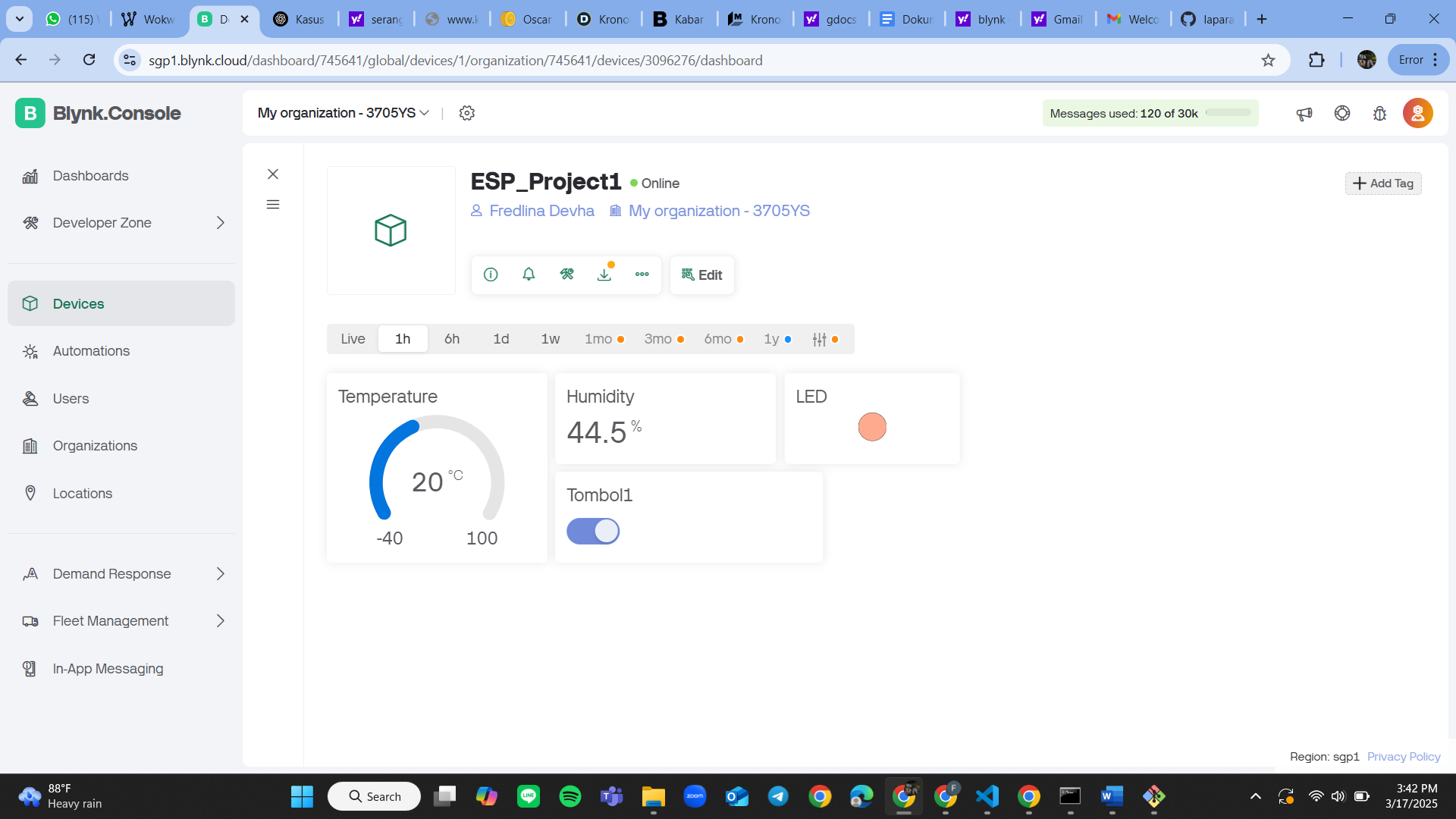
#### 2.2 Langkah Implementasi

1. Membuat proyek ESP32 di PlatformIO dan Wokwi melalui VSCode
2. Mengatur konfigurasi pada Wokwi: menggunakan template bawaan, menambahkan sensor DHT22 dan LED, menghubungkan DHT22 (data) ke pin 15 dan LED ke pin 26 melalui resistor
3. Menyalin file diagram.json ke VSCode dan membuat file wokwi.toml
4. Membuat akun Blynk dan template baru dengan nama ESP Project1
5. Menambahkan datastream: suhu (V4), kelembapan (V3), kontrol LED (V5), dan status LED (V6)
6. Mendesain dashboard: gauge untuk suhu dan kelembapan, switch untuk kontrol LED, dan widget LED untuk menampilkan status
7. Mengambil token autentikasi Blynk dan memasukkannya ke dalam kode
8. Mengimplementasikan kode dengan library WiFi.h, BlynkSimpleEsp32.h, dan DHTesp.h
9. Memastikan fungsi sendSensor() digunakan untuk membaca dan mengirim suhu serta kelembapan
10. Memastikan fungsi BLYNK\_WRITE() digunakan untuk mengontrol LED
11. Melakukan pengujian dengan menjalankan simulasi di Wokwi dan memverifikasi hasil monitoring serta kontrol LED melalui aplikasi Blynk

**3. HASIL DAN PEMBAHASAN**

3.1 Hasil Eksperimen





3.2 Pembahasan

Hasil simulasi menunjukkan bahwa sistem yang dikembangkan dapat berfungsi dengan baik. ESP32 berhasil terhubung ke jaringan WiFi Wokwi dan mengirimkan data suhu serta kelembaban ke aplikasi Blynk secara real-time. Nilai yang ditampilkan di aplikasi sesuai dengan pembacaan dari sensor DHT22, dengan tingkat akurasi yang cukup baik.Selain itu, kontrol LED melalui aplikasi Blynk juga bekerja dengan optimal. Ketika pengguna menekan tombol toggle, LED pada Wokwi dapat menyala dan mati sesuai perintah yang diberikan. Hal ini menunjukkan bahwa komunikasi antara Blynk dan ESP32 melalui WiFi berjalan dengan baik, tanpa adanya delay yang signifikan.

Dari hasil evaluasi, kelebihan utama dari implementasi ini adalah:

* Kemudahan dalam pemantauan suhu dan kelembaban melalui aplikasi Blynk tanpa perlu menggunakan layar tambahan pada perangkat keras.
* Kontrol LED yang responsif melalui koneksi WiFi, sehingga pengguna dapat mengontrol perangkat elektronik dari mana saja.
* Penggunaan Wokwi sebagai simulator memungkinkan pengujian sistem sebelum diterapkan ke perangkat fisik, mengurangi risiko kesalahan pada implementasi sebenarnya.

Namun, terdapat beberapa keterbatasan dalam simulasi ini, antara lain:

* Konektivitas WiFi di Wokwi tidak selalu stabil, sehingga terkadang ESP32 perlu direstart untuk kembali terhubung ke server Blynk.
* Tidak semua sensor dan aktuator dapat disimulasikan di Wokwi, sehingga beberapa fitur tambahan mungkin perlu diuji langsung pada perangkat keras.

Secara keseluruhan, hasil dari praktik ini menunjukkan bahwa kombinasi Blynk dan Wokwi dapat menjadi solusi efektif dalam pengembangan dan simulasi sistem IoT sebelum diterapkan dalam perangkat fisik. Dengan menggunakan Blynk, pengguna dapat dengan mudah membuat dashboard pemantauan dan kontrol berbasis cloud, sementara Wokwi memungkinkan pengujian sistem tanpa harus membeli perangkat keras terlebih dahulu.

**4. Lampiran**

a. Kode Program Main.cpp

#include <Arduino.h>

#define BLYNK\_DEVICE\_NAME "Esp32IoT"

#define BLYNK\_PRINT Serial

#define BLYNK\_AUTH\_TOKEN "HTQYzD\_N7WEYU2SSZ2WrL178XbrHOLMR"

#define BLYNK\_TEMPLATE\_ID "TMPL65IBPXVXd"

#define BLYNK\_TEMPLATE\_NAME "Template1"

#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;

//function untuk pengiriman sensor

void sendSensor()

{

TempAndHumidity data = dht.getTempAndHumidity();

//menampilkan temperature pada Serial monitor

Serial.print("% Temperature: ");

Serial.print(data.temperature);

Serial.println("C ");

Serial.print("% Kelembaban: ");

Serial.print(data.humidity);

Serial.println("% ");

Blynk.virtualWrite(V0, data.temperature); //mengirimkan data temperatur ke Virtual pin VO di Blynk Cloud

Blynk.virtualWrite(V1, data.humidity); //mengirimkan data kelemaban ke Virtual pin V1 di Blynk Cloud

}

BLYNK\_WRITE(V2)

{

int nilaiBacaIO =param.asInt();

digitalWrite(LED\_R, nilaiBacaIO);

Blynk.virtualWrite(V3, nilaiBacaIO);

}

void setup()

{

// Debug console

Serial.begin(115200); //serial monitor menggunakan bautrate 9600

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

pinMode(LED\_R, OUTPUT);

Blynk.begin(auth, ssid, pass); //memulai Blynk

timer.setInterval(1000, sendSensor); //Mengaktifkan timer untuk pengiriman data 1000ms

}

void loop()

{

Blynk.run(); //menjalankan blynk

timer.run(); //menjalankan timer

}

b. Kode Program diagram.json

{

"version": 1,

"author": "Anonymous maker",

"editor": "wokwi",

"parts": [

{ "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -278.9, "left": 52.76, "attrs": {} },

{

"type": "wokwi-led",

"id": "led1",

"top": -306.4,

"left": -89.47,

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

},

{

"type": "wokwi-resistor",

"id": "r5",

"top": -274.74,

"left": -44.52,

"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", [ "v87.6", "h-228.22", "v-54.65" ] ],

[ "dht1:GND", "esp:GND.1", "black", [ "v93.06", "h-109.48", "v-76.5" ] ],

[ "dht1:SDA", "esp:D15", "green", [ "v26.39", "h-81.44", "v-19.67" ] ],

[ "led1:C", "esp:GND.2", "black", [ "v0" ] ]

],

"dependencies": {}

}