**Praktik Implementasi Penggunaan Platform Bylnk Menggunakan Wokwi**

**oleh**

*Nadia Alya Paramitha Erwanto1*

*Fakultas Vokasi, Universitas Brawijaya*

*Email:*[*nadiaalya2729@student.ub.ac.id*](nadiaalya2729@student.ub.ac.id)

**Abstrak**

Internet of Things (IoT) semakin berkembang dan memungkinkan berbagai perangkat untuk saling terhubung dan dikendalikan secara jarak jauh melalui internet. Salah satu platform yang mendukung pengembangan IoT adalah Blynk, yang memungkinkan pengguna untuk membuat antarmuka pemantauan dan kontrol dengan mudah. Dalam bab ini, dibahas praktik implementasi penggunaan platform Blynk dengan simulator Wokwi untuk mengembangkan sistem berbasis IoT. Wokwi digunakan sebagai media simulasi perangkat keras sebelum implementasi di perangkat fisik, sehingga mempermudah proses pengujian dan pengembangan. Studi kasus yang diangkat meliputi koneksi antara perangkat mikrokontroler dengan Blynk, pemrograman antarmuka, serta pemantauan data sensor secara real-time. Hasil dari praktik ini menunjukkan bahwa kombinasi Blynk dan Wokwi dapat menjadi solusi efektif dalam perancangan dan simulasi sistem IoT sebelum diterapkan secara nyata.

***Kata kunci****: IoT, Blynk, Wokwi, simulasi, pemantauan data.*

**Abstract**

The Internet of Things (IoT) is growing and allows various devices to be connected and controlled remotely via the internet. One of the platforms that support IoT development is Blynk, which allows users to create monitoring and control interfaces easily. In this chapter, the implementation practice of using the Blynk platform with the Wokwi simulator to develop an IoT-based system is discussed. Wokwi is used as a hardware simulation medium before implementation on a physical device, making the testing and development process easier. The case studies include the connection between microcontroller devices and Blynk, interface programming, and real-time monitoring of sensor data. The results of this practice show that the combination of Blynk and Wokwi can be an effective solution in designing and simulating IoT systems before real implementation.

***Keywords:*** *IoT, Blynk, Wokwi, simulation, data monitoring.*

**Pendahuluan**

Internet of Things (IoT) merupakan konsep yang memungkinkan berbagai perangkat untuk saling terhubung dan berkomunikasi melalui jaringan internet. Dengan semakin berkembangnya teknologi, IoT telah banyak diterapkan dalam berbagai bidang seperti otomatisasi rumah, pemantauan lingkungan, kesehatan, dan industri manufaktur. Salah satu tantangan dalam pengembangan sistem IoT adalah bagaimana cara menghubungkan perangkat keras dengan aplikasi pemantauan yang mudah digunakan dan memiliki antarmuka yang interaktif.

Untuk menjawab tantangan tersebut, Blynk hadir sebagai platform yang memungkinkan pengguna untuk memantau dan mengontrol perangkat IoT dari jarak jauh menggunakan smartphone atau komputer. Blynk mendukung berbagai jenis mikrokontroler seperti ESP8266, ESP32, dan Arduino, serta menawarkan konektivitas berbasis WiFi, Bluetooth, maupun jaringan seluler. Dengan adanya Blynk, pengguna dapat mengembangkan aplikasi IoT tanpa perlu membangun dashboard atau server sendiri, karena Blynk telah menyediakan antarmuka berbasis cloud yang intuitif.

Namun, sebelum sistem IoT diterapkan di perangkat fisik, diperlukan simulasi dan pengujian untuk memastikan bahwa perangkat dan aplikasi berjalan dengan baik. Oleh karena itu, dalam proyek ini digunakan Wokwi, yaitu simulator berbasis web yang memungkinkan pengembang untuk menguji program mikrokontroler secara virtual tanpa memerlukan perangkat keras. Dengan menggunakan Wokwi, ESP32 dapat diprogram dan disimulasikan langsung di browser, sehingga memudahkan proses pengujian sebelum implementasi sebenarnya.

Dalam proyek ini, Blynk dan Wokwi digunakan untuk membuat sistem pemantauan dan kontrol IoT yang menampilkan data temperature, humidity, serta kontrol LED dan toggle switch melalui koneksi WiFi. Sensor DHT22 digunakan untuk membaca suhu dan kelembaban, sementara LED akan dikendalikan menggunakan tombol pada aplikasi Blynk. Dengan implementasi ini, pengguna dapat memantau kondisi lingkungan secara real-time dan mengontrol perangkat elektronik dari jarak jauh, tanpa perlu berada di lokasi yang sama dengan perangkat.

**Metodologi**

Penelitian ini dilakukan dengan pendekatan eksperimental, di mana ESP32 digunakan sebagai mikrokontroler utama yang bertugas membaca data dari sensor DHT22 dan mengirimkan informasi tersebut ke Blynk melalui koneksi WiFi. Selain itu, ESP32 juga dikonfigurasi untuk menerima perintah dari aplikasi Blynk untuk mengontrol LED secara nirkabel.

1. Perancangan Sistem

Sistem ini terdiri dari tiga komponen utama, yaitu:

* Mikrokontroler ESP32: Bertindak sebagai otak sistem yang membaca data sensor dan berkomunikasi dengan Blynk melalui WiFi.
* Sensor DHT22: Berfungsi untuk membaca suhu dan kelembaban udara yang kemudian dikirimkan ke aplikasi Blynk.
* LED dan Toggle Switch: LED digunakan sebagai indikator yang dikendalikan melalui tombol pada aplikasi Blynk.

2. Konfigurasi Blynk

Langkah pertama dalam implementasi adalah membuat proyek di aplikasi Blynk dan menambahkan widget yang diperlukan, yaitu:

* Gauge atau Label untuk menampilkan suhu dan kelembaban.
* Tombol Toggle untuk mengontrol LED secara jarak jauh.
* WiFi Connection untuk menghubungkan ESP32 dengan server Blynk.

Blynk akan menyediakan Auth Token, yang harus dimasukkan ke dalam kode ESP32 agar perangkat dapat terhubung dengan aplikasi.

3. Pemrograman ESP32

Kode program ditulis menggunakan Arduino IDE, dengan pustaka yang digunakan meliputi:

* BlynkSimpleEsp32.h untuk menghubungkan ESP32 dengan server Blynk.
* WiFi.h untuk konektivitas WiFi.
* DHT.h untuk membaca data dari sensor DHT22.

Kode ESP32 dikembangkan agar dapat:

1. Menghubungkan mikrokontroler ke jaringan WiFi Wokwi.
2. Membaca data suhu dan kelembaban dari sensor DHT22.
3. Mengirimkan data ke aplikasi Blynk secara berkala.
4. Menerima perintah dari aplikasi Blynk untuk mengontrol LED.

4. Simulasi di Wokwi

Setelah kode program selesai, simulasi dilakukan di Wokwi, dengan langkah-langkah sebagai berikut:

* Menambahkan komponen ESP32, DHT22, dan LED dalam simulator Wokwi.
* Mengunggah kode ESP32 ke dalam Wokwi dan menjalankan simulasi.
* Memeriksa konektivitas antara ESP32 dan Blynk, serta memastikan data suhu dan kelembaban tampil di dashboard aplikasi.
* Mengontrol LED melalui aplikasi Blynk dan mengevaluasi respons sistem.

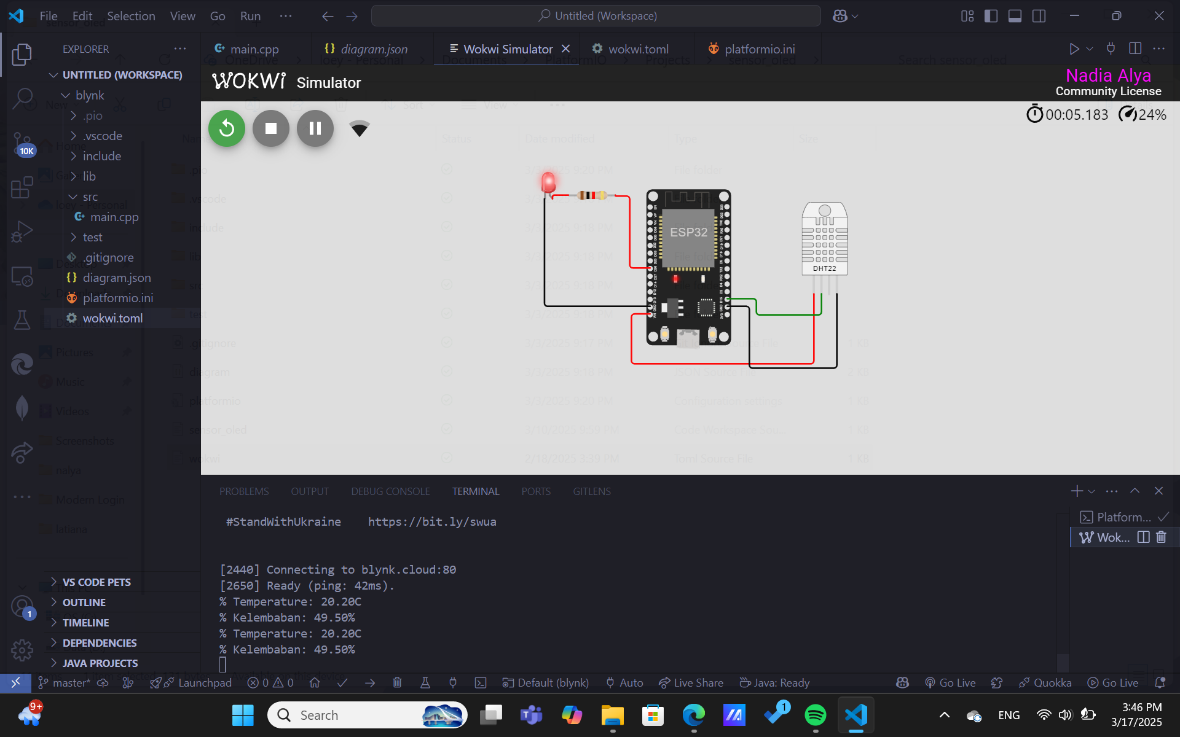
5. Pengujian dan Evaluasi

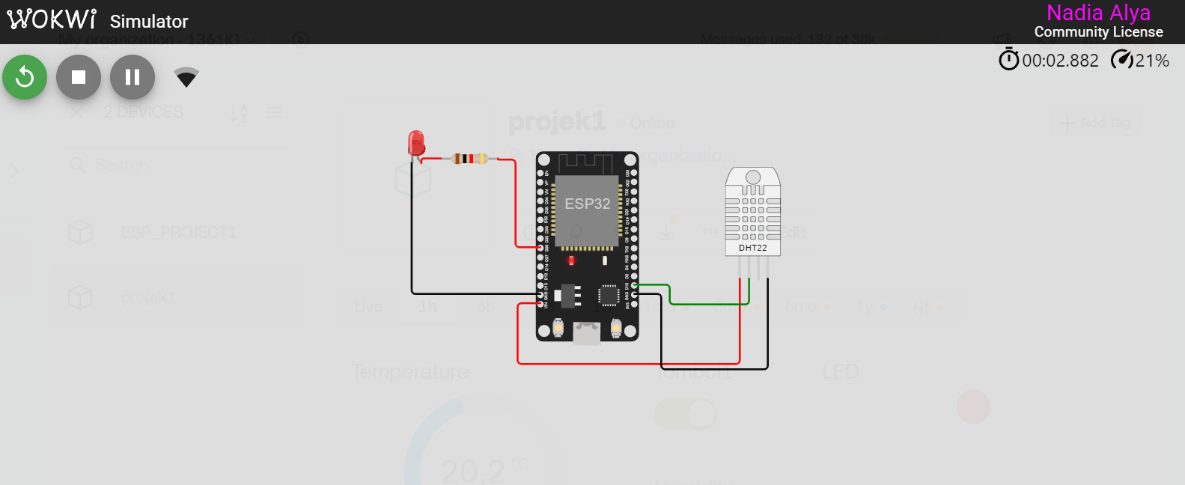
Pengujian dilakukan untuk memastikan bahwa:

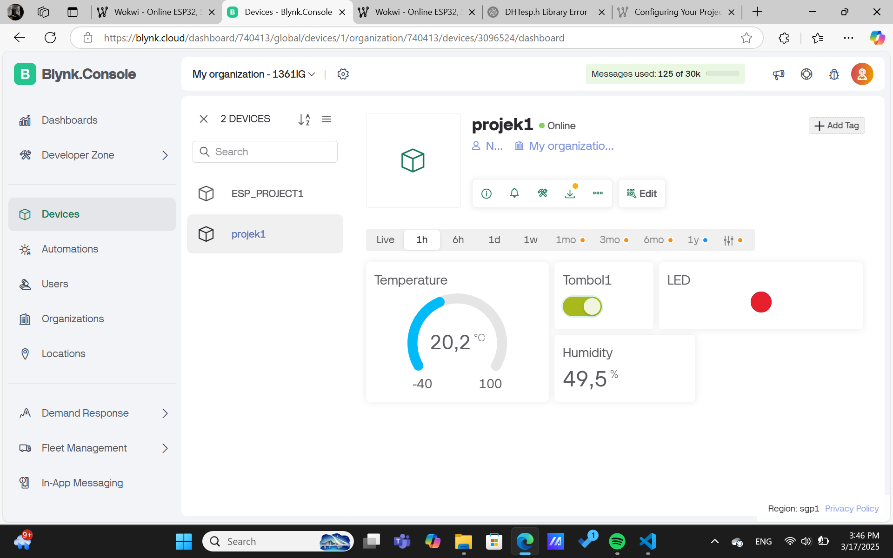
1. ESP32 berhasil terhubung ke jaringan WiFi dan server Blynk.
2. Data suhu dan kelembaban terbaca dengan benar di aplikasi Blynk.
3. LED dapat dikendalikan melalui tombol toggle di aplikasi Blynk.
4. Sistem berjalan stabil tanpa gangguan koneksi.

**Hasil dan 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.

**Lampiran**

Source code **main.cpp :**

#include <Arduino.h>

#define BLYNK\_DEVICE\_NAME "Esp32IoT"

#define BLYNK\_PRINT Serial

#define BLYNK\_AUTH\_TOKEN "Tn3qPXriQXP8PkVg7PH7aqjwzoGntWHK"

#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

}

**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": {}

}