

**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)**  
**AKSES API MELALUI SIMULASI WOKWI**



*Amelya Eka Wulandari*

233140700111005

[amelyaaeka@gmail.com](mailto:amelyaaeka@gmail.com)

**PROGRAM STUDI TEKNOLOGI INFORMASI**  
**DEPARTEMEN INDUSTRI KREATIF DAN INOVASI**  
**FAKULTAS VOKASI**  
**UNIVERSITAS BRAWIJAYA**

**2025**

## **ABSTRACTS**

Pada praktikum ini, dilakukan simulasi pengiriman data sensor suhu dan kelembaban menggunakan ESP32 melalui platform simulasi Wokwi. Data yang diperoleh dari sensor DHT22 dikirimkan ke API Laravel menggunakan metode HTTP POST. Simulasi ini bertujuan untuk memahami proses komunikasi antara perangkat IoT dan Server melalui protokol HTTP serta menguji konektivitas API dalam menerima data dari sensor IoT secara real-time. Hasil praktikum menunjukkan bahwa ESP32 dapat mengirimkan data sensor ke server dengan sukses melalui API yang dikembangkan menggunakan framework Laravel.

**Kata Kunci:** IoT, ESP32, Wokwi, API

# **BAB I**

## **INTRODUCTION**

### **1.1 Latar Belakang**

Internet of Things (IoT) memungkinkan berbagai perangkat terhubung ke internet dan saling berkomunikasi. Salah satu implementasi IoT adalah pengiriman data sensor ke server untuk dianalisis lebih lanjut. Dalam praktikum ini, sensor DHT22 digunakan untuk membaca suhu dan kelembaban, kemudian data tersebut dikirim ke server Laravel melalui API menggunakan metode HTTP POST.

### **1.2 Tujuan Praktikum**

1. Memahami cara kerja API dalam menerima data dari perangkat IoT
2. Mengimplementasikan pengiriman data dari ESP32 ke API menggunakan HTTP POST
3. Melakukan simulasi menggunakan platform Wokwi

## **BAB II**

### **METHODOLOGY**

#### 2.1 Tools & Materials

1. Hardware : ESP32, SensorDHT22, Breadboard dan Kabel Jumper
2. Software : Wokwi Simulator, PlatformIO, Visual Studio Code, Laravel

#### 2.2 Implementation Steps

1. Menjalankan server Laravel dengan perintah “ `php artisan serve --host=0.0.0.0 --port=8080` ”
2. Membuat endpoint API di Laravel untuk menerima data sensor
3. Menulis kode pada ESP32 untuk membaca data dari sensor DHT22
4. Mengirimkan data ke API menggunakan HTTP
5. Mengamati respons dari server melalui serial monitor ESP32 ( wokwi simulator pada Visual Studio Code )

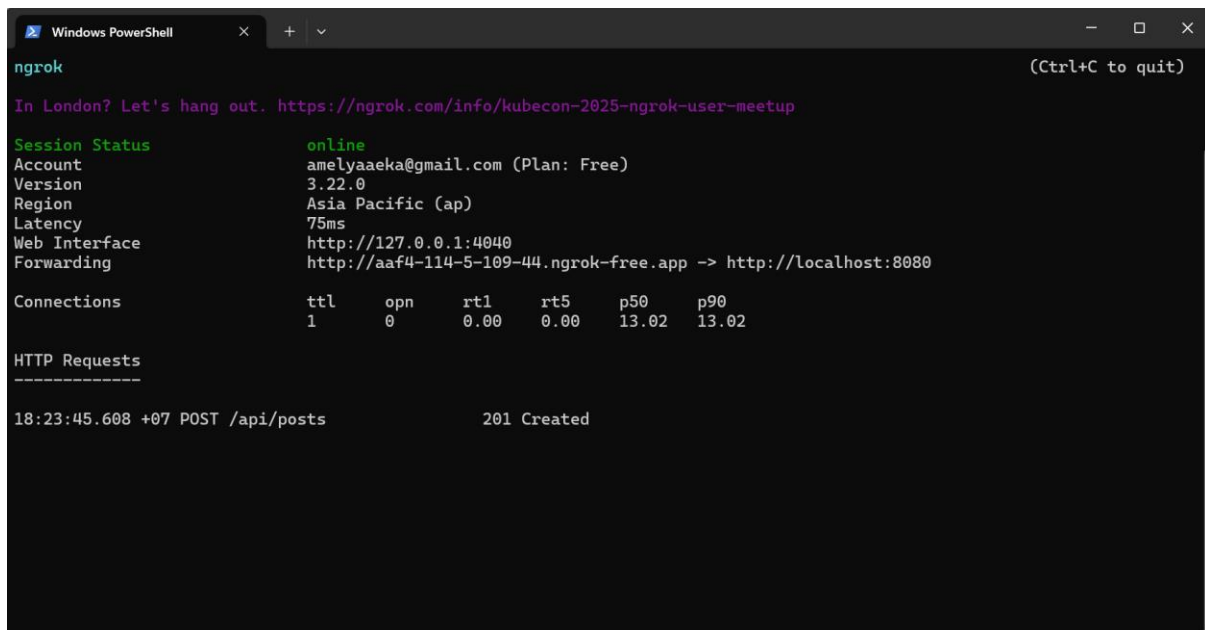
## BAB III

### RESULTS AND DISCUSSION

#### 3.1 Experimental Results

1. Data sensor suhu dan kelembaban berhasil dikirim ke API Laravel
2. Server Laravel dapat menerima dan menyimpan data dengan baik
3. Respons API dapat dilihat melalui Serial Monitor ESP32

HASIL PRAKTIKUM :

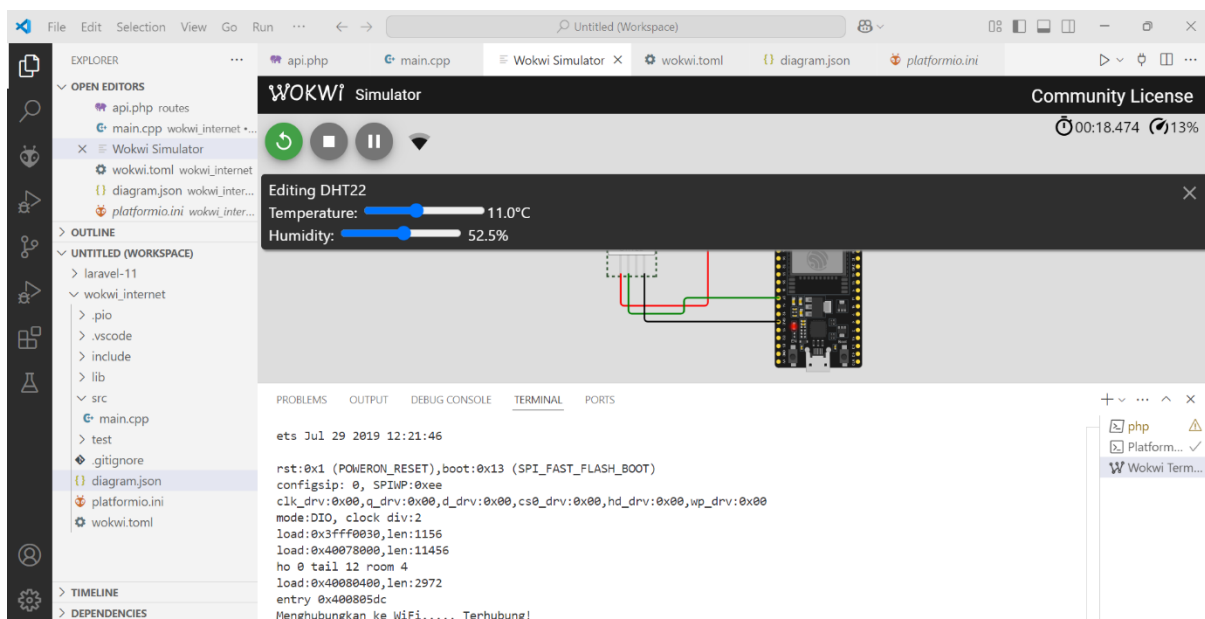


```
Windows PowerShell
ngrok
In London? Let's hang out. https://ngrok.com/info/kubecon-2025-ngrok-user-meetup

Session Status      online
Account             amelyaaeka@gmail.com (Plan: Free)
Version             3.22.0
Region              Asia Pacific (ap)
Latency              75ms
Web Interface        http://127.0.0.1:4040
Forwarding            http://aaf4-114-5-109-44.ngrok-free.app -> http://localhost:8080

Connections          ttl    opn    rt1    rt5    p50    p90
                     1      0      0.00   0.00   13.02   13.02

HTTP Requests
-----
18:23:45.608 +07 POST /api/posts      201 Created
```



## APPENDIX

```
#include <Arduino.h>

#include <WiFi.h>

#include <HTTPClient.h>

#include "DHT.h"

#define DHTPIN 27

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);


// Ganti dengan kredensial WiFi Anda

const char* ssid = "username wifi yang digunakan";

const char* password = "sesuai settingan";


unsigned long previousMillis = 0;

const long interval = 5000; // Interval 5 detik (5000 ms)


void setup() {

    Serial.begin(115200);


    // Hubungkan ke WiFi

    WiFi.begin(ssid, password);

    Serial.print("Menghubungkan ke WiFi");

    while (WiFi.status() != WL_CONNECTED) {

        delay(500);
```

```
    Serial.print(".");  
  
}  
  
Serial.println(" Terhubung!");  
  
  
dht.begin();  
  
  
  
// Tunggu sebentar agar koneksi stabil  
  
delay(1000);  
  
}  
  
  
void loop() {  
  
    unsigned long currentMillis = millis();  
  
  
    // Lakukan POST setiap interval yang telah ditentukan  
  
    if (currentMillis - previousMillis >= interval) {  
  
        previousMillis = currentMillis;  
  
  
  
        float h = round(dht.readHumidity());  
  
        // Read temperature as Celsius (the default)  
  
        float t = round(dht.readTemperature());  
  
  
  
        // Check if any reads failed and exit early (to try again).  
  
        if (isnan(h) || isnan(t)) {  
  
            Serial.println(F("Failed to read from DHT sensor!"));
```

```

    return;
}

// Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

// Inisialisasi HTTPClient

HTTPClient http;

String url = "http://aaf4-114-5-109-44.ngrok-free.app/api/posts"; // Ganti dengan URL ngrok
yang benar

http.begin(url); // Menggunakan HTTP, bukan HTTPS

http.addHeader("Content-Type", "application/json");

String payload = "{\"nama_sensor\":\"Sensor GD\", \"nilai1\":\"" + String(h) + ", \"nilai2\":\"" +
String(t) + "\"}";

Serial.println(payload); // Untuk melihat apakah payload sudah terbentuk dengan benar

// Kirim POST request

int httpResponseCode = http.POST(payload);

// Tampilkan kode respons HTTP

Serial.print("Kode respons HTTP: ");

Serial.println(httpResponseCode);

```



```
// Tampilkan respons dari server jika request berhasil

if (httpResponseCode == 200 || httpResponseCode == 201) {

    String response = http.getString();

    Serial.println("Respons dari server:");

    Serial.println(response);

} else {

    Serial.println("Gagal mengirim data");

}

// Tutup koneksi HTTP

http.end();

}

}
```