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

**Fakultas Vokasi, Universitas Brawijaya**

**Implementasi Pengendalian LED dan Pengiriman Data Sensor Menggunakan ESP32 dalam Sistem IoT**

## *Author(s):*

*Wildan Aridh Takhfif  
 Fakultas Vokasi, Universitas Brawijaya  
 Email: apipkph@studen.ub.ac.id*

## **Abstract**

## This experiment focuses on implementing the ESP32 microcontroller to control LEDs and send sensor data using HTTP POST requests in an Internet of Things (IoT) system. The system involves connecting an ESP32 to a Wi-Fi network, controlling two LEDs based on programmed logic, and sending data from a DHT22 sensor (temperature and humidity) to an API hosted on a Laravel server. The results show that the ESP32 can effectively control the LEDs and send real-time sensor data to the server. The experiment demonstrates the integration of basic IoT concepts with practical hardware components for real-world applications.

*Keywords—Internet of Things, ESP32, DHT22, HTTP POST, Sensor Data*

## **1. Introduction**

### **1.1 Latar belakang praktikum IoT yang dilakukan**

Praktikum ini bertujuan untuk mengimplementasikan konsep dasar Internet of Things (IoT) dengan menggunakan ESP32 sebagai mikrokontroler untuk mengendalikan perangkat dan mengirimkan data melalui jaringan internet. IoT semakin berkembang dalam berbagai sektor, salah satunya adalah dalam pengendalian perangkat secara jarak jauh dan pemantauan data secara real-time. Dalam praktikum ini, kami menggunakan sensor DHT22 untuk membaca suhu dan kelembaban serta mengendalikan LED melalui pemrograman dengan ESP32.

### **1.2 Tujuan eksperimen**

Tujuan dari eksperimen ini adalah untuk mempelajari cara menghubungkan ESP32 ke jaringan Wi-Fi, mengendalikan perangkat seperti LED, serta mengirimkan data sensor ke server menggunakan protokol HTTP. Kami juga bertujuan untuk menguji dan memastikan sistem IoT ini berfungsi dengan baik dalam mengirimkan data suhu dan kelembaban ke API yang telah disediakan.

## **2. Methodology**

### 2.1 Tools & Materials

* Mikrokontroler: ESP32  
  Sensor: DHT22 (sensor suhu dan kelembaban)
* Software: PlatformIO (VSCode), Arduino IDE, Laravel API Server, NGROK  
  Kabel dan Breadboard: Untuk pengkabelan antara ESP32 dan sensor DHT22

### **2.2 Implementation Steps**

1. Menyiapkan Hardware:  
   * Menghubungkan ESP32 ke komputer dan menginstal driver yang diperlukan.
   * Menghubungkan sensor DHT22 ke pin ESP32 untuk membaca suhu dan kelembaban.
   * Menyambungkan dua LED ke pin GPIO ESP32 untuk diuji.
2. Menyiapkan Software:  
   * Membuat project baru di PlatformIO dan memodifikasi file platformio.ini untuk menyesuaikan dengan board ESP32.
   * Menulis program di file main.cpp untuk mengendalikan LED dan membaca data dari sensor DHT22.
   * Menyusun program untuk menghubungkan ESP32 ke jaringan Wi-Fi dan mengirimkan data melalui HTTP POST ke server API.
3. Testing dan Verifikasi:  
   * Melakukan upload program ke ESP32 dan memverifikasi apakah LED menyala sesuai dengan program.
   * Memverifikasi koneksi Wi-Fi dan pemindaian jaringan di ESP32 melalui serial monitor.
   * Mengirimkan data sensor ke API dan memverifikasi data yang diterima di server.

## **3. Results and Discussion**

### **3.1 Experimental Results**

* Kontrol LED:

Kedua LED berhasil dikendalikan untuk menyala dan mati secara bergantian sesuai dengan logika program yang telah diunggah ke ESP32.

* Koneksi Wi-Fi:

ESP32 berhasil terhubung ke jaringan Wi-Fi dan dapat memindai jaringan Wi-Fi sekitar. Serial monitor menampilkan daftar Access Point yang tersedia beserta informasi kekuatan sinyalnya.

* Data Sensor:

Sensor DHT22 berhasil membaca data suhu dan kelembaban. Data ini kemudian dikirimkan melalui HTTP POST request ke server API yang dihosting di Laravel. Data yang diterima oleh server berhasil disimpan dalam database.

Tabel 1: Data Sensor yang Dikirimkan ke Server

| Nama Sensor | Suhu (°C) | Kelembaban (%) |
| --- | --- | --- |
| Sensor GD | 23.5 | 60 |

Gambar 1: Tampilan Serial Monitor ESP32  
.

### **3.2 Discussion**

Eksperimen ini membuktikan bahwa ESP32 dapat digunakan dengan efektif dalam aplikasi IoT untuk mengendalikan perangkat dan mengirimkan data sensor melalui internet. Penggunaan HTTP POST untuk mengirim data sensor ke API sangat efisien dan memungkinkan integrasi dengan berbagai platform backend. Meskipun demikian, untuk aplikasi IoT yang lebih kompleks, penggunaan protokol lain seperti MQTT bisa dipertimbangkan untuk efisiensi yang lebih baik, terutama dalam hal penghematan bandwidth dan waktu respons.

## **4. Appendix**

### **4.1 Kode Program untuk Pengendalian LED dan Pengiriman Data Sensor**

#include <WiFi.h>

#include <HTTPClient.h>

#include "DHT.h"

#define DHTPIN 27

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

const char\* ssid = "Wokwi-GUEST";

const char\* password = "";

unsigned long previousMillis = 0;

const long interval = 5000;

void setup() {

Serial.begin(115200);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println(" Terhubung!");

dht.begin();

delay(1000);

}

void loop() {

unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {

previousMillis = currentMillis;

float h = round(dht.readHumidity());

float t = round(dht.readTemperature());

if (isnan(h) || isnan(t)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

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

HTTPClient http;

String url = "http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app/api/posts";

http.begin(url);

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

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

Serial.println(payload);

int httpResponseCode = http.POST(payload);

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

Serial.println(httpResponseCode);

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

String response = http.getString();

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

Serial.println(response);

} else {

Serial.println("Gagal mengirim data");

}

http.end();

}

}