# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Pengiriman Data Sensor DHT22 ke API Laravel Menggunakan ESP32 dan Postman**

*Rifqi Naufal Nazhir*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: rifqinaufaln@ub.ac.id*

**Abstract**

Praktikum ini bertujuan untuk membaca data suhu dan kelembaban dari sensor DHT22 yang terhubung ke ESP32, dan mengirimkannya ke server API berbasis Laravel melalui metode HTTP POST. Hasil data dikirimkan dalam format JSON dan dapat diuji menggunakan Postman. Praktik ini menunjukkan bagaimana perangkat fisik dapat berkomunikasi dengan server web melalui protokol HTTP, salah satu komponen utama dalam sistem cloud.

**Kata kunci:** ESP32, DHT22, API, Laravel, Postman, Internet of Things, HTTP POST

**1. Introduction**

**1.1 Latar belakang**

Internet of Things (IoT) merupakan konsep penggabungan antara perangkat fisik dengan kemampuan konektivitas jaringan untuk mengumpulkan dan bertukar data. Dalam praktik ini, digunakan ESP32 sebagai mikrokontroler yang membaca data dari sensor suhu dan kelembaban DHT22, kemudian mengirimkannya ke server API berbasis Laravel. Proses pengiriman dilakukan melalui HTTP POST, dan data dapat diuji melalui aplikasi Postman sebagai alat bantu validasi komunikasi client-server.

**1.2 Tujuan eksperimen**

1. Membaca suhu dan kelembaban dari sensor DHT22.
2. Menghubungkan ESP32 ke jaringan WiFi.
3. Mengirimkan data sensor ke API Laravel melalui HTTP POST.
4. Menguji dan memverifikasi data menggunakan Postman.

**2. Methodology (Metodologi)**

**2.1 Tools & Materials**

* ESP32
* Sensor DHT22
* Breadboard dan kabel jumper
* Visual Studio Code + PlatformIO
* Laravel 11 (API Backend)
* Postman
* WiFi

**2.2 Implementation Steps (Langkah Implementasi)**

1. Hubungkan DHT22 ke GPIO 15 pada ESP32.
2. Jalankan server Laravel lokal di port 8080 dan pastikan endpoint POST /api/posts aktif.
3. Buat project PlatformIO dan tambahkan kode berikut ke file main.cpp::

#include <Arduino.h>

#include <WiFi.h>

#include <HTTPClient.h>

#include "DHT.h"

#define DHTPIN 15

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

const char\* ssid = "ninja";

const char\* password = "11111111";

unsigned long previousMillis = 0;

const long interval = 5000;

void setup() {

Serial.begin(115200);

WiFi.begin(ssid, password);

Serial.print("Menghubungkan ke WiFi");

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://192.168.161.190:8080/api/posts"; // Ganti dengan alamat Laravel API

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();

}

1. Uji endpoint Laravel menggunakan Postman dengan metode POST dan header Content-Type: application/json.
2. Pastikan data berhasil masuk ke database Laravel dan dapat ditampilkan melalui endpoint API.

**3. Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

ESP32 berhasil membaca data suhu dan kelembaban dari sensor DHT22 dan mengirimkannya dalam format JSON ke API Laravel. Postman digunakan untuk menguji endpoint dan memastikan data yang dikirim valid. Respons server menunjukkan bahwa data berhasil diproses dan disimpan.

Contoh Payload:

{

"nama\_sensor": "Sensor GD",

"nilai1": 62,

"nilai2": 28

}

Contoh output Serial Monitor:

{"nama\_sensor":"Sensor GD", "nilai1":62, "nilai2":28}

Kode respons HTTP: 201

Respons dari server:

{

"message": "Data berhasil disimpan",

"data": {

"id": 1,

"nama\_sensor": "Sensor GD",

"nilai1": 62,

"nilai2": 28

}

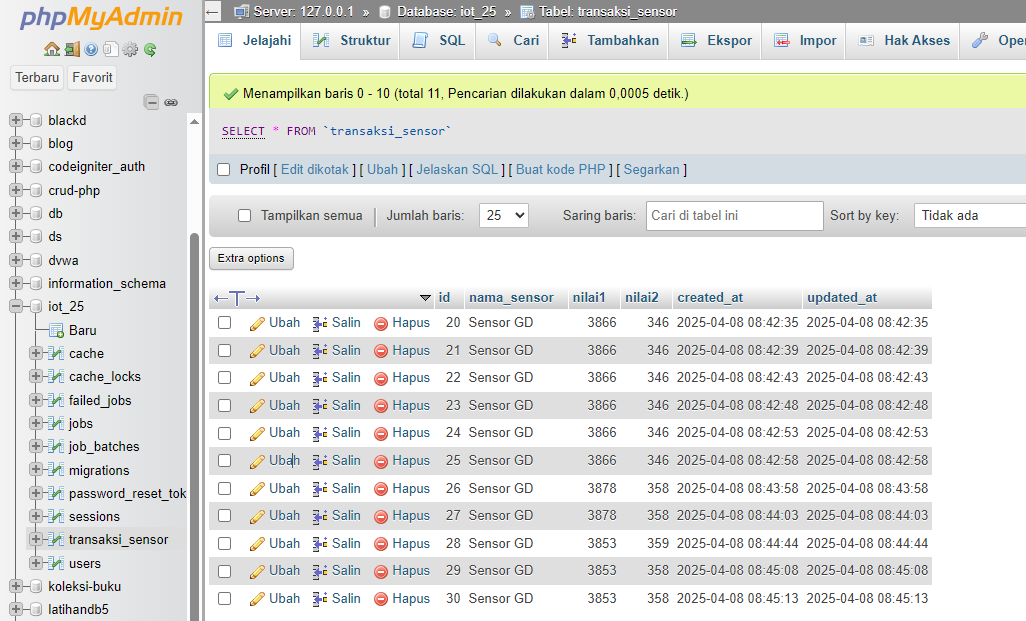
}

**Conclusion**

ESP32 dapat digunakan untuk mengirimkan data sensor ke server API menggunakan protokol HTTP POST. Dalam praktik ini, data berhasil dikirim ke API Laravel dan diverifikasi menggunakan Postman. Ini membuktikan integrasi antara perangkat IoT dan cloud backend sangat memungkinkan dan dapat diimplementasikan dengan infrastruktur sederhana.

**4. Appendix (Lampiran, jika diperlukan)**

**Hasil praktikum**

****

