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

**Fakultas Vokasi, Universitas Brawijaya**

### Praktikum Mengirim Data Sensor DHT22 Menggunakan ESP32 ke API Laravel

### Wildan Aridh Takhfif *Fakultas Vokasi, Universitas Brawijaya Email: apipkph@gmail.com*

### **Abstract**

This experiment aims to implement an IoT system that sends temperature and humidity data from a DHT22 sensor using ESP32 to a Laravel-based REST API. The system utilizes the Wokwi simulator for prototyping and NGROK to tunnel the local Laravel API. Results show successful transmission and storage of environmental data into a MySQL database with accurate and real-time updates. This practical activity demonstrates the interoperability between embedded devices and web servers.

Keywords— Internet of Things, ESP32, Laravel API, NGROK, DHT22

## 1. Introduction

### 1.1 Latar Belakang

Internet of Things (IoT) memungkinkan perangkat fisik untuk saling terhubung dan bertukar data melalui jaringan internet. Dalam praktikum ini, digunakan ESP32 sebagai mikrokontroler untuk membaca data dari sensor suhu dan kelembaban DHT22 dan mengirimkannya ke REST API Laravel untuk disimpan di database.

### 1.2 Tujuan

Membangun sistem IoT sederhana yang menghubungkan ESP32 dengan sensor DHT22 dan mengirim data ke backend Laravel menggunakan NGROK sebagai tunneling service.

## 2. Methodology

### 2.1 Tools & Materials

* ESP32 (via Wokwi Simulator)
* Sensor DHT22
* Wokwi (simulator)
* Laravel 11
* NGROK
* MySQL

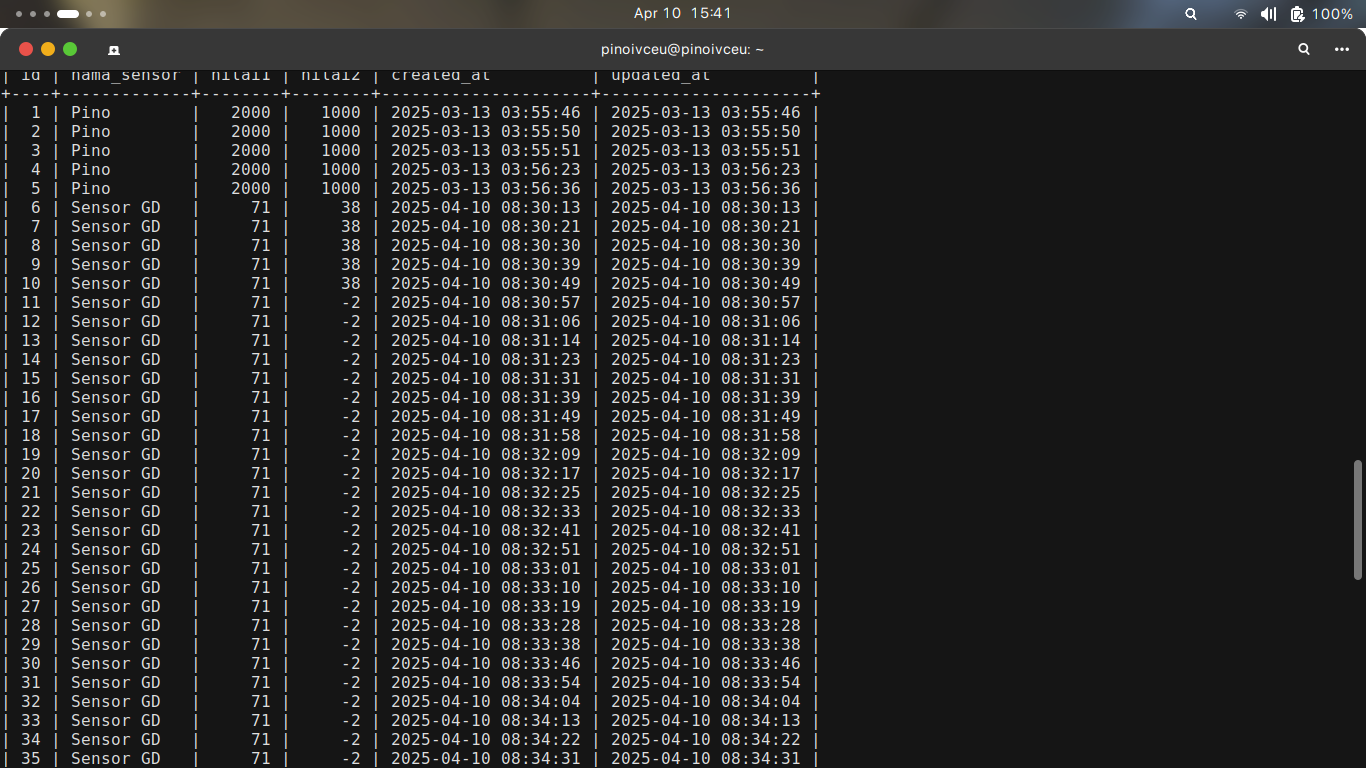
### 2.2 Implementation Steps

1. Membuat rangkaian ESP32 dan DHT22 di Wokwi
2. Membuat project Laravel dan endpoint API untuk menerima data
3. Menyambungkan Laravel ke database MySQL
4. Menyambungkan ESP32 ke endpoint dengan HTTP POST menggunakan NGROK
5. Mengirim data suhu dan kelembaban secara berkala
6. Memastikan data tersimpan di database

## 3. Results and Discussion

### 3.1 Experimental Results

Data suhu dan kelembaban berhasil dikirim setiap 5 detik dari ESP32 ke API Laravel. Data terlihat masuk ke database dan API merespons status "success".



Diskusi:  
Hasil eksperimen menunjukkan bahwa sistem dapat berjalan dengan baik dan responsif. Tantangan yang dihadapi adalah memastikan konektivitas NGROK stabil serta pengaturan format JSON agar sesuai dengan backend Laravel.

## 4. Appendix

### A. Source Code Arduino ESP32

#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 = "Wokwi-GUEST";

const char\* password = "";

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://f0d5-175-45-191-253.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();

}

}