LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

MENGIRIM DATA SUHU DAN KELEMBAPAN

DARI WOKWI SIMULATOR KE DATABASE

FAKULTAS VOKASI, UNIVERSITAS BRAWIJAYA

*Sholihul Fadjri Triwibowo*

*Fakultas Vokasi, Universitas Brawijaya*

*Email:* [*fadjritwibowo20@student.ub.ac.id*](mailto:fadjritwibowo20@student.ub.ac.id)

ABSTRAK

Pemantauan suhu dan kelembapan secara real-time menjadi aspek penting dalam berbagai aplikasi IoT. Penelitian ini bertujuan untuk mengimplementasikan sistem pemantauan berbasis ESP32 yang membaca data dari sensor DHT22 dalam Wokwi Simulator dan mengirimkannya ke database menggunakan protokol http. Data yang dikirim disimpan dalam database MySQL melalui server backend berbasis PHP Laravel. Pengujian menunjukkan bahwa data berhasil dikirim dan disimpan dengan baik, membuktikan bahwa system ini dapat digunakan sebagai Solusi pemantauan lingkungan yang efisien. Hasil ini dapat dikembangkan lebih lanjut dengan integrasi visualisasi data dan penggunaan protokol komunikasi yang lebih optimal.

Kata Kunci *—* Wokwi Simulator, ESP32, DHT22, IoT, MySQL

1. Pendahuluan
   1. Latar Belakang

Pemantauan suhu dan kelembapan secara real-time sangat penting dalam berbagai aplikasi IoT, seperti sistem pemantauan lingkungan, rumah pintar, dan pertanian cerdas. Wokwi Simulator menyediakan platform simulasi berbasis web yang memungkinkan pengguna untuk mengembangkan dan menguji perangkat berbasis mikrokontroler sebelum diimplementasikan secara fisik. Dalam praktik ini, data suhu dan kelembapan dari sensor DHT22 yang disimulasikan akan dikirim ke database untuk penyimpanan dan analisis lebih lanjut.

* 1. Tujuan Eksperimen

1. Memahami cara menghubungkan Wokwi Simulator dengan database.
2. Menggunakan ESP32 virtual untuk membaca data dari sensor DHT22
3. Mengirimkan data suhu dan kelembapan ke database menggunakan protokol http.
4. Menyimpan dan menampilkan data dalam sistem berbasis web.
5. Metodologi
   1. Alat dan Bahan
      1. Alat: Komputer, Wokwi Simulator, MySQL, Server backend (PHP Laravel)
      2. Bahan: ESP32 virtual, VSC dengan PlatformIO, phpMyAdmin
   2. Langkah Implementasi
      1. Jalankan API backend Laravel dengan parameter **php artisan serve --host=0.0.0.0 --port=8080**
      2. Buat proyek baru dengan PlatformIO dan ubah konfigurasi kode main.cpp menjadi seperti berikut

#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://6c45-175-45-191-11.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();

  }

}

* + 1. Edit file **platformio.ini** sebagai berikut

; PlatformIO Project Configuration File

;

;   Build options: build flags, source filter

;   Upload options: custom upload port, speed and extra flags

;   Library options: dependencies, extra library storages

;   Advanced options: extra scripting

;

; Please visit documentation for the other options and examples

; https://docs.platformio.org/page/projectconf.html

[env:esp32doit-devkit-v1]

platform = espressif32

board = esp32doit-devkit-v1

framework = arduino

monitor\_speed = 115200

lib\_deps =

    adafruit/DHT sensor library

* + 1. Lalu build **main.cpp**, setelah itu buat file dengan nama **wokwi.toml** dengan konfigurasi sebagai berikut.

[wokwi]

version = 1

firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'

elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'

* + 1. Buat file dengan nama **diagram.json** lalu konfigurasi kodenya sebagai berikut

{

  "version": 1,

  "author": "KAVITH BUDVIN",

  "editor": "wokwi",

  "parts": [

    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 86.4, "left": 24.04, "attrs": {} },

    { "type": "wokwi-dht22", "id": "dht1", "top": 19.5, "left": -91.8, "attrs": {} }

  ],

  "connections": [

    [ "esp:TX", "$serialMonitor:RX", "", [] ],

    [ "esp:RX", "$serialMonitor:TX", "", [] ],

    [ "dht1:VCC", "esp:3V3", "red", [ "v19.2", "h67.2", "v-67.2" ] ],

    [ "esp:GND.1", "dht1:GND", "black", [ "h0" ] ],

    [ "dht1:SDA", "esp:27", "green", [ "v0" ] ]

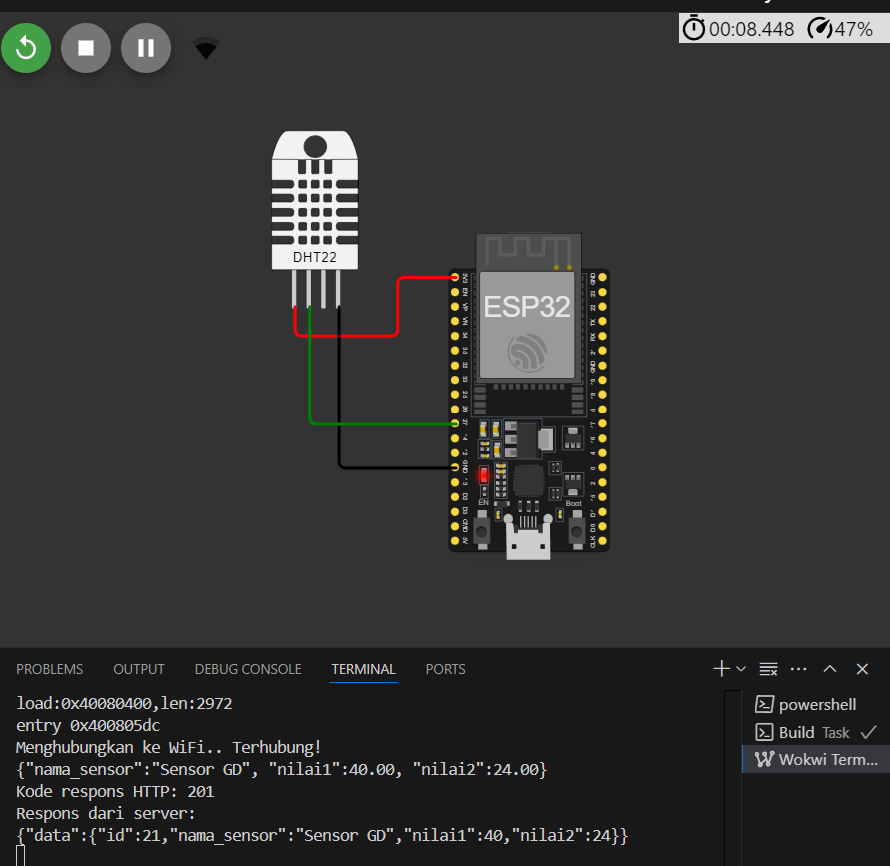
  ],

  "dependencies": {}

}

* + 1. Buka ngrok lalu jalankan dengan parameter **ngrok http --scheme=http 8080**
    2. Jalankan wokwi simulator di visual studio code lalu atur suhu dan kelembapannya

1. Hasil dan Pembahasan
   1. Hasil Eksperimen
      1. Wifi berhasil terhubung ke Wokwi-Guest, lalu data sensor suhu terkirim ke database



* + 1. Data tersimpan di dalam database

