**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)   
MEMBUAT RANGKAIAN SENSOR SUHU KELEMBAPAN**

Fakultas Vokasi, Universitas Brawijaya

*Farhan Febrianto**Fakultas Vokasi, Universitas Brawijaya**Email: farhan05@student.ub.ac.id*

## Abstract (Abstrak)

Eksperimen ini bertujuan untuk mengimplementasikan sistem pengukuran suhu dan kelembapan menggunakan sensor DHT22 dan mikrokontroler ESP32 dengan pemrograman C++ di Visual Studio Code. Sensor DHT22 digunakan untuk mendeteksi nilai suhu dan kelembapan, yang kemudian dibaca oleh ESP32 dan dikirim ke cloud untuk pemantauan jarak jauh. Pemrograman menggunakan library DHT memudahkan proses pembacaan data sensor dengan fungsi readTemperature() dan readHumidity(). Hasil eksperimen menunjukkan bahwa sistem berhasil membaca dan mengirimkan data suhu serta kelembapan secara real-time ke platform cloud, yang memungkinkan pemantauan jarak jauh. Sistem ini dapat diperluas dengan fitur tambahan seperti tampilan LCD atau sistem peringatan berbasis data yang diterima di cloud. Kesimpulannya, ESP32 dapat digunakan sebagai pengontrol untuk sistem monitoring suhu dan kelembapan, dengan integrasi cloud yang memungkinkan pemantauan dari jarak jauh.

**Kata kunci**: Sensor suhu dan kelembapan, DHT22, ESP32, cloud, Visual Studio Code, pemrograman.

## 1. Pendahuluan

### 1.1 Latar belakang

Pemantauan suhu dan kelembapan sangat penting dalam berbagai bidang seperti pertanian, peternakan, dan industri untuk menjaga kualitas lingkungan dan mencegah kerusakan. Dengan kemajuan teknologi, sistem monitoring dapat dilakukan secara otomatis menggunakan mikrokontroler seperti ESP32 yang mampu membaca data dari sensor DHT22 secara real-time dan mengirimkan data tersebut ke cloud untuk pemantauan jarak jauh. Sistem ini memungkinkan pengguna untuk mengakses data suhu dan kelembapan dari lokasi yang berbeda menggunakan platform cloud. Praktikum ini bertujuan untuk mengimplementasikan sistem monitoring suhu dan kelembapan menggunakan ESP32 dan DHT22 dengan pemrograman C++ di Visual Studio Code, serta mengirimkan data ke platform cloud.

### 1.2 Tujuan eksperimen

1. Mempelajari cara menghubungkan dan mengkonfigurasi sensor DHT22 dengan mikrokontroler ESP32 untuk membaca data suhu dan kelembapan.
2. Mengimplementasikan pemrograman C++ di Visual Studio Code untuk membaca, memproses, dan mengirimkan data sensor ke cloud.
3. Mengintegrasikan sistem monitoring suhu dan kelembapan dengan platform cloud untuk pemantauan jarak jauh.

## 2. Methodology (Metodologi)

### 2.1 Tools & Materials (Alat dan Bahan)

Alat dan Perangkat Lunak:

1. **Visual Studio Code** – Editor kode untuk menulis dan mengunggah program ke ESP32.
2. **Arduino Framework** – Digunakan sebagai dasar pemrograman mikrokontroler ESP32 dalam bahasa C++.
3. **Platform Cloud** (Google) – Platform untuk menerima dan menampilkan data yang dikirimkan oleh ESP32.

Bahan (Simulasi Komponen Elektronik):

1. **ESP32** – Mikrokontroler yang digunakan sebagai pengendali utama dalam simulasi.
2. **Sensor DHT22** – Digunakan untuk mengukur suhu dan kelembapan udara.
3. **Kabel Penghubung (dalam simulasi Wokwi)** – Menghubungkan komponen dalam desain rangkaian.

### 2.2 Implementation Steps (Langkah Implementasi)

1. **Perancangan Rangkaian Hardware**
2. Menghubungkan sensor DHT22 ke ESP32, dengan pin data DHT22 dihubungkan ke pin GPIO
3. Menghubungkan pin VCC sensor ke 3.3V ESP32 dan pin GND ke ground ESP32.
4. **Pemrograman Sistem Monitoring Suhu dan Kelembapan**
5. Membuka Visual Studio Code dan menulis kode dalam bahasa C++ menggunakan Arduino framework.
6. Mengimpor library DHT dan mendefinisikan pin serta tipe sensor yang digunakan.
7. Menghubungkan ESP32 dengan platform cloud
8. Menulis logika pembacaan data suhu dan kelembapan pada fungsi loop(), kemudian mengirimkan data ke cloud menggunakan koneksi Wi-Fi ESP32.
9. **Pengujian Sistem**
10. Mengunggah program ke ESP32 dan menguji pembacaan suhu dan kelembapan pada serial monitor.
11. Memastikan data berhasil dikirim ke platform cloud dan dapat diakses secara real-time.
12. Melakukan debugging dan perbaikan pada kode jika terjadi kesalahan dalam pengiriman data atau pembacaan sensor.

## 3. Hasil dan Pembahasan

### 3.1 Hasil Eksperimen

* 1. **Rangkaian Berhasil Dijalankan**
  2. Rangkaian sensor DHT22 berhasil terhubung dengan ESP32, dan ESP32 mampu membaca data suhu dan kelembapan menggunakan fungsi readTemperature() dan readHumidity().
  3. Koneksi Wi-Fi ESP32 berhasil terhubung dengan platform cloud untuk mengirimkan data..
  4. **Pembacaan Data Suhu dan Kelembapan Berjalan dengan Baik**
  5. Sensor DHT22 berhasil mengukur suhu dan kelembapan secara real-time.
  6. Data suhu dan kelembapan berhasil dikirimkan dan ditampilkan pada platform cloud setiap interval waktu tertentu.
  7. Nilai yang ditampilkan sesuai dengan rentang normal suhu dan kelembapan yang diukur.
  8. **Hasil Simulasi di VsCode**
  9. Program yang dijalankan di VsCode menampilkan data suhu dan kelembapan secara visual pada serial monitor.
  10. Tidak ditemukan error atau kesalahan dalam eksekusi kode.
  11. Serial monitor menampilkan pesan "Reading temperature and humidity..." sebagai indikasi bahwa sistem berjalan dengan baik.

## 4. Lampiran

Source Code   
#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 = "5min=1k";

const char \*password = "sosiologi";

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 pio device monitor*

    HTTPClient http;

    String url = "https://script.google.com/macros/s/AKfycbwfMnOmqIkDggDkjv0GfKXbVzQEFbI1dLev2nRiVoa135VeqnXrNKqNtWSurh\_CBM8/exec"; *// Ganti dengan URL ngrok yang benar*

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

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

    String payload = "{\"nama\_sensor\":\"Sensor DADA\", \"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();

  }

}



