LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Transfer Data Suhu dan Kelembaban ke Database**

**Menggunakan ESP32 , DHT 11**

*Fauzan Taufikurohman - 233140700111033*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: fauzantaufikurohman123@gmail.com*

**Abstract** (Abstrak)

Dalam praktik ini, dilakukan proses pengumpulan dan pengiriman data suhu serta kelembaban secara real-time menggunakan sensor DHT11 yang dikendalikan oleh mikrokontroler ESP32. Data hasil pembacaan dikirim ke server Laravel melalui endpoint API yang diekspos menggunakan Ngrok. Server Laravel bertindak sebagai perantara antara perangkat keras dan database MySQL, di mana data akan disimpan untuk keperluan monitoring dan analisis lebih lanjut. Praktik ini menunjukkan bagaimana sistem IoT dapat diintegrasikan dengan layanan backend secara langsung menggunakan pendekatan REST API dan pengiriman data melalui jaringan lokal maupun internet.

*Kata kunci: ESP32, DHT11, Suhu dan Kelembaban, mysql, WiFi.*

**1. Pendahuluan**

**1.1 Latar belakang**

Perkembangan teknologi IoT memungkinkan perangkat fisik seperti sensor dapat terhubung langsung ke aplikasi server untuk mengirimkan data secara real-time. Salah satu tantangan dalam implementasi sistem ini adalah menghubungkan perangkat IoT yang berjalan pada jaringan lokal ke server atau database yang berjalan di komputer lokal lainnya. Dalam praktik ini, digunakan Ngrok sebagai jembatan untuk mengekspos API Laravel ke internet, sehingga ESP32 dapat mengirim data meskipun berada dalam jaringan berbeda. Praktik ini sangat relevan dalam pengembangan sistem monitoring suhu dan kelembaban berbasis IoT dengan komunikasi client-server.

**1.2 Tujuan eksperimen**

1. Menghubungkan sensor DHT11 ke mikrokontroler ESP32 untuk membaca data suhu dan kelembaban.
2. Mengirimkan data ke server Laravel menggunakan HTTP POST dengan format JSON.
3. Menyimpan data ke database MySQL melalui API yang dikembangkan menggunakan Laravel.
4. Menggunakan Ngrok untuk mengekspos endpoint API Laravel agar dapat diakses oleh ESP32 secara publik.

**2. Methodology (Metodologi)**

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

Alat dan bahan yang di gunakan dalam membuat praktik ini adalah sebagai berikut:

* Laptop dengan Visual Studio Code, ekstensi PlatformIO, laravel, mysql
* Mikrokontroler ESP32
* Sensor DHT11
* Kabel jumper
* Breadboard
* Jaringan WiFi
* Ngrok (untuk men-tunnel API ke publik)
* XAMPP  
  1. **Implementation Steps (Langkah Implementasi)**

**2.1. Perakitan Perangkat Keras (Hardware Setup)**

* Siapkan rangkaian ESP32 dan sensor DHT11 di atas breadboard (jika tersedia untuk kemudahan koneksi).
* Sambungkan pin VCC dari DHT11 ke 3.3V ESP32, GND ke ground, dan pin DATA ke salah satu pin digital ESP32, misalnya GPIO27.
* Pastikan koneksi kabel jumper kuat dan tidak longgar, serta posisi pin tidak tertukar.

**2.2. Pemrograman Mikrokontroler (ESP32)**

* Jalankan API dengan perintah “**php artisan serve --host=0.0.0.0 --port=8080**” laravel terlebih dahulu agar nanti bis mengakses databse nya
* Setelah itu jalankan ngrok pada cmd dengan perintah “**ngrok http --scheme=http 8080**”.
* Buka Visual Studio Code dan buat proyek baru menggunakan PlatformIO dengan board ESP32 yang sesuai.
* Instal library yang dibutuhkan, seperti:
  + DHT untuk komunikasi dengan sensor DHT11.
  + WiFi.h untuk koneksi jaringan.
  + HTTPClient.h untuk pengiriman data ke cloud.
* Tulis program sebagai 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* **=** "mbayar";

const char**\*** *password* **=** "12345678";

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://88ed-2001-448a-c0b0-19cf-7494-3b5b-d15a-8c1d.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**();

  }

}

* Setelah program berhasil di buat lalu lakukan proses upload dan tunggu hingga “succes”.
* Setelah upload “succes” klik tombol serial monitor
* Lakukan pengujian berulang dan pantau kestabilan koneksi serta akurasi pembacaan data sensor

**3. Hasil Praktik dan lampiran**

**3.1 Hasil Eksperimen**

#### ****1. Koneksi Perangkat Berjalan dengan Baik****

* Mikrokontroler **ESP32** berhasil membaca data dari **sensor DHT11** secara berkala setiap 5 detik.
* Nilai suhu dan kelembaban berhasil ditampilkan pada **Serial Monitor** melalui komunikasi UART dengan laptop.

#### ****2. Koneksi WiFi Stabil****

* ESP32 berhasil terhubung ke jaringan WiFi yang telah ditentukan (ssid = "Zzzz").
* Tidak terdapat gangguan signifikan pada koneksi selama eksperimen berlangsung.

#### ****3. Integrasi Ngrok dan Laravel API Berhasil****

* Laravel API berhasil dijalankan secara lokal pada port 8080, lalu ditunnel menggunakan **Ngrok** untuk mendapatkan **public URL** (misalnya https://abcd1234.ngrok.io).
* URL Ngrok berhasil digunakan dalam kode ESP32 untuk mengirimkan data sensor secara realtime.

#### ****4. Pengiriman Data dari ESP32 ke API Berhasil****

* Payload JSON yang dikirim dari ESP32 berhasil diterima oleh endpoint API Laravel (/api/posts) dan dicetak ke terminal Laravel sebagai log debug.
* Contoh payload:

json

SalinEdit

{

"nama\_sensor": "Sensor GD",

"nilai1": 74,

"nilai2": 29

}

#### ****5. Data Tersimpan di Database MySQL****

* Data yang dikirim berhasil disimpan ke dalam database MySQL melalui model Laravel.
* Isi tabel posts menampilkan data sensor dengan field seperti:
  + nama\_sensor
  + nilai1 (kelembaban)
  + nilai2 (suhu)
  + created\_at, updated\_at

**3.2 Lampiran**







