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

**PRAKTIK REAL HARDWARE ESP32**

****

*Amelya Eka Wulandari*

233140700111005

[amelyaaeka@gmail.com](mailto:amelyaaeka@gmail.com)

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**DEPARTEMEN INDUSTRI KREATIF DAN INOVASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**2025**

**ABSTRACTS**

Praktikum ini membahas implementasi langsung (real hardware) mikrokontroler ESP32 untuk pengendalian LED, pemindaian jaringan WiFi, dan pengiriman data sensor ke API berbasis Laravel. Praktikum dilakukan untuk menguji kemampuan ESP32 dalam aplikasi Internet of Things (IoT) secara nyata. Langkah-langkah mencakup pengenalan perangkat, instalasi driver, konfigurasi PlatformIO, coding, serta integrasi dengan server menggunakan NGROK. Hasil menunjukkan bahwa ESP32 dapat dikenali sistem, menjalankan logika LED, mendeteksi jaringan WiFi, serta mengirim data suhu dan kelembapan ke database melalui koneksi internet.

**Kata kunci:** ESP32, IoT, LED, WiFi Scanner, API Laravel, NGROK, DHT22.

**BAB I**

**INTRODUCTION**

* 1. Latar Belakang

Dalam era Internet of Things (IoT), mikrokontroler seperti ESP32 memainkan peran penting dalam menghubungkan perangkat fisik ke jaringan. Praktikum Bab 14 ini bertujuan untuk mengimplementasikan penggunaan hardware ESP32 secara nyata. Sebelumnya, banyak eksperimen dilakukan melalui simulator seperti WOKWI, namun pada tahap ini mahasiswa dituntut untuk menguasai aspek riil dari perangkat. Penerapan meliputi kendali output (LED), pemindaian jaringan WiFi, dan koneksi ke server API. Pengalaman langsung ini menjadi bekal penting dalam merancang dan membangun sistem IoT yang lebih kompleks dan aplikatif.

* 1. Tujuan Eksperimen

1. Mengenali dan mengintalasi perangkat keras ESP32 pada komputer
2. Menguji program pengendalian LED melalui ESP32 secara langsung
3. Melakukan pemindaian jaringan WiFi dengan ESP32
4. Mengirim data sensor suhu dan kelembapan (DHT22) ke server API
5. Melatih mahasiswa untuk dapat mengintegrasikan hardware, software, dan jaringan secara menyeluruh

**BAB II**

**METHODOLOGY**

2.1 Tools & Materials

Alat dan bahan yang digunakan untuk praktikum ini:

* Software : PlatformIO, Visual Studio Code, Arduino Framework, Serial Monitor, Laravel
* Hardware : ESP32, Breadboard, Kabel Jumper, LED, DHT22
* Driver : Silicon Labs CP210x

2.2 Implementation Steps

1. Pastikan ESP32 dikenali oleh komputer melalui Device Manager.
2. Instal driver Silicon Labs CP210x jika ESP32 belum terdeteksi.
3. Buat project baru di PlatformIO dan lakukan wiring LED ke GPIO ESP32.
4. Masukkan kode kendali LED dan upload ke ESP32, lalu uji nyala LED secara fisik.
5. Upload kode pemindaian WiFi menggunakan library WiFi.h ke ESP32.
6. Cek hasil pemindaian SSID melalui Serial Monitor dengan monitor\_speed = 115200.
7. Hubungkan sensor DHT22 ke GPIO ESP32 dan baca data suhu serta kelembapan.
8. Buat kode HTTP POST untuk mengirim data sensor ke API Laravel.
9. Jalankan server Laravel dengan perintah php artisan serve --host=0.0.0.0 --port=8080.
10. Buka akses publik menggunakan ngrok http --scheme=http 8080.
11. Uji apakah ESP32 berhasil mengirim data dan status HTTP menunjukkan 200/201.
12. Pastikan data dari sensor berhasil masuk ke database melalui endpoint API.

**BAB III**

**RESULTS AND DISCUSSION**

3.1 Experimental Results

|  |  |  |
| --- | --- | --- |
| No | Tahapan | Hasil |
| 1 | Instalasi driver CP210x | ESP32 dikenali dengan baik di Device Manager (COM3) |
| 2 | Upload program LED | LED berkedip sesuai interval 1 detik ON/OFF |
| 3 | Pemindaian WiFi | Terdeteksi >3 SSID dengan kekuatan sinyal terukur di Serial Monitor |
| 4 | Kirim data sensor ke API | Payload suhu dan kelembapan terkirim dan muncul di database API |

3.2 Kode Program

* LED

#include <Arduino.h>

int lampu = 33;

int lampu2 = 25;

void setup() {

Serial.begin(115200);

pinMode(lampu, OUTPUT);

pinMode(lampu2, OUTPUT);

}

void loop() {

digitalWrite(lampu, HIGH);

digitalWrite(lampu2, HIGH);

delay(1000);

digitalWrite(lampu, LOW);

digitalWrite(lampu2, LOW);

delay(1000);

}

* Scanner

#include <WiFi.h>

void setup() {

Serial.begin(115200);

WiFi.mode(WIFI\_STA);

WiFi.disconnect();

delay(100);

}

void loop() {

int n = WiFi.scanNetworks();

for (int i = 0; i < n; ++i) {

Serial.print(WiFi.SSID(i));

Serial.print(" (");

Serial.print(WiFi.RSSI(i));

Serial.println(" dBm)");

}

delay(5000);

}

* Kirim data sensor ke API

#include <Arduino.h>

#include <WiFi.h>

#include <HTTPClient.h>

#include "DHT.h"

#define DHTPIN 27

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

const char\* ssid = "HotspotKamu";

const char\* password = "passwordKamu";

unsigned long previousMillis = 0;

const long interval = 5000;

void setup() {

Serial.begin(115200);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

}

dht.begin();

}

void loop() {

if (millis() - previousMillis >= interval) {

previousMillis = millis();

float h = round(dht.readHumidity());

float t = round(dht.readTemperature());

if (!isnan(h) && !isnan(t)) {

HTTPClient http;

String url = "http://xxx-xxx.ngrok-free.app/api/posts";

http.begin(url);

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

String payload = "{\"nama\_sensor\":\"Sensor GD\",\"nilai1\":" + String(h) + ",\"nilai2\":" + String(t) + "}";

http.POST(payload);

http.end();

}

}

}

* Platform Io

[env:esp32doit-devkit-v1]

platform = espressif32

board = esp32doit-devkit-v1

framework = arduino

upload\_port = COM3

monitor\_port = COM3

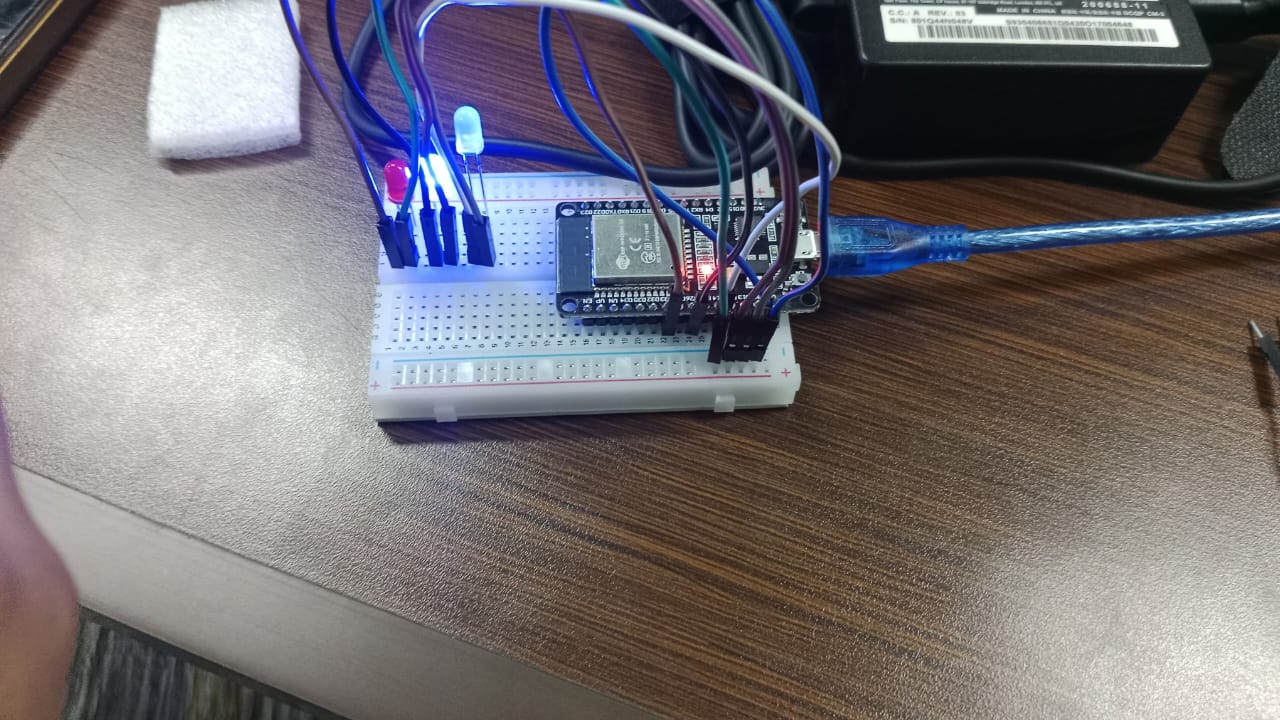
monitor\_speed = 115200

lib\_deps =

adafruit/DHT sensor library@^1.4.4

adafruit/Adafruit Unified [Sensor@^1.1.14](mailto:Sensor@%5e1.1.14)

**APPENDIX**

****