LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Pembuatan API   
Menggunakan Laravel dan Ngrok**

**Tugas BAB 12**

****

*Afi Kristiani*

*Fakultas Vokasi, Universitas Brawijaya*

*Email :* [*afikristianii@gmail.com*](mailto:afikristianii@gmail.com)

**Abstrak**

Laporan praktikum ini membahas proses pengujian akses API melalui simulasi perangkat IoT menggunakan platform WOKWI. WOKWI merupakan simulator berbasis web yang memungkinkan pengembangan dan pengujian kode untuk mikrokontroler tanpa memerlukan perangkat keras fisik. Praktikum ini bertujuan untuk mempelajari integrasi antara perangkat IoT yang disimulasikan dengan API berbasis REST, sehingga dapat mengirim dan menerima data secara real-time. Proses simulasi dilakukan dengan memanfaatkan board virtual (misalnya Arduino atau ESP32) untuk melakukan request ke API, menguji respons, serta menganalisis alur komunikasi data. Hasil praktikum menunjukkan bahwa simulasi menggunakan WOKWI merupakan alternatif yang efektif untuk menguji integrasi sistem IoT sebelum implementasi di perangkat nyata.

*Keyword : WOKWI, API, REST API, IoT, Simulasi*

**Pendahuluan**

* 1. **Latar Belakang**

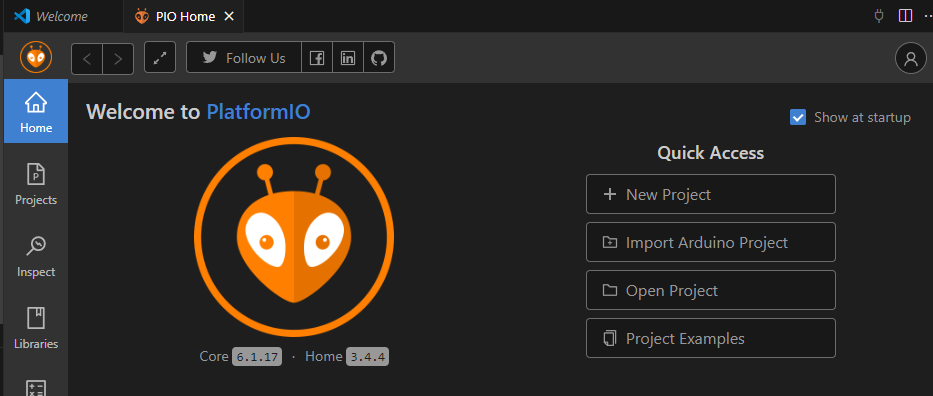
Perkembangan teknologi Internet of Things (IoT) mengharuskan adanya integrasi yang efisien antara perangkat keras dan perangkat lunak, terutama dalam komunikasi data antar perangkat melalui API. Pembuatan API berbasis REST menjadi solusi populer untuk menjembatani pertukaran informasi antara server dan perangkat IoT. Namun, pengujian langsung pada perangkat keras memerlukan waktu dan sumber daya yang tidak sedikit.

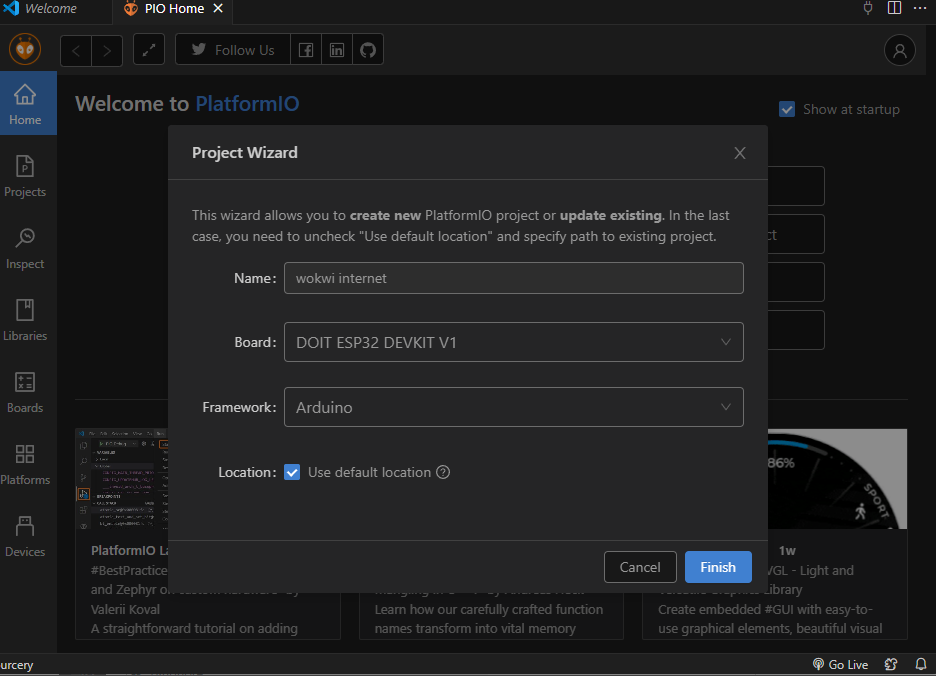
WOKWI hadir sebagai platform simulasi berbasis web yang memungkinkan pengembang untuk menguji kode dan interaksi perangkat IoT tanpa harus menggunakan perangkat fisik. Dengan menggunakan WOKWI, praktikum ini mengintegrasikan simulasi perangkat dengan API sehingga memudahkan dalam menguji komunikasi data secara real-time, mengurangi biaya, serta mempercepat siklus pengembangan.

1. **Tujuan Eksperimen**

* **Memahami Konsep Integrasi IoT dan API**: Mengetahui bagaimana perangkat IoT dapat berkomunikasi dengan server melalui API berbasis REST.
* **Menerapkan Simulasi dengan WOKWI**: Menggunakan platform WOKWI untuk mensimulasikan perangkat IoT (misalnya Arduino/ESP32) yang melakukan request ke API.
* **Menguji Respons API**: Memastikan bahwa API memberikan respons yang benar dan sesuai dengan pengujian yang dilakukan melalui simulasi.

1. **Metodologi**
   1. **Alat dan Bahan**
2. Laptop
3. Internet
4. Web Server Lokal: XAMPP, WAMP, atau LAMP.
5. Composer: Manajer paket untuk PHP.
6. Laravel: Framework PHP yang digunakan untuk membangun API.
7. Ngrok: Tool untuk membuat tunnel ke server lokal.
8. Text Editor/IDE: Visual Studio Code, Sublime Text, atau sejenisnya.
9. Postman: Aplikasi untuk menguji API
   1. **Langkah Implementasi**
10. Buat new project pada platformio IDE





1. Setelah itu masukan script main.cpp dan lakukan compiling

#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://a771-125-166-12-93.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");

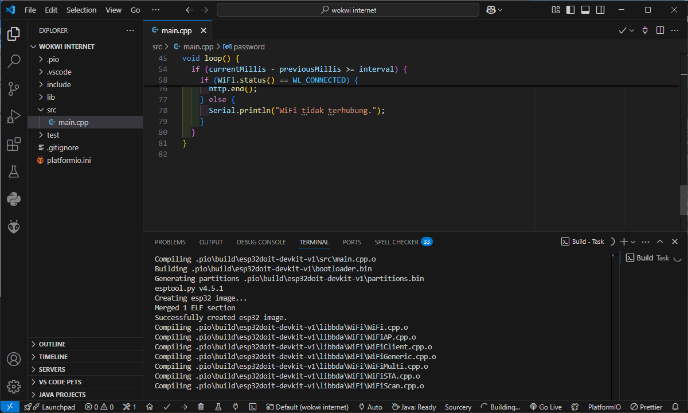
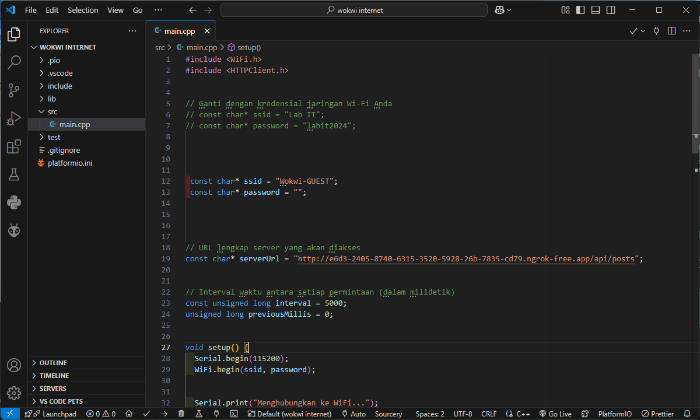
    }

    // Tutup koneksi HTTP

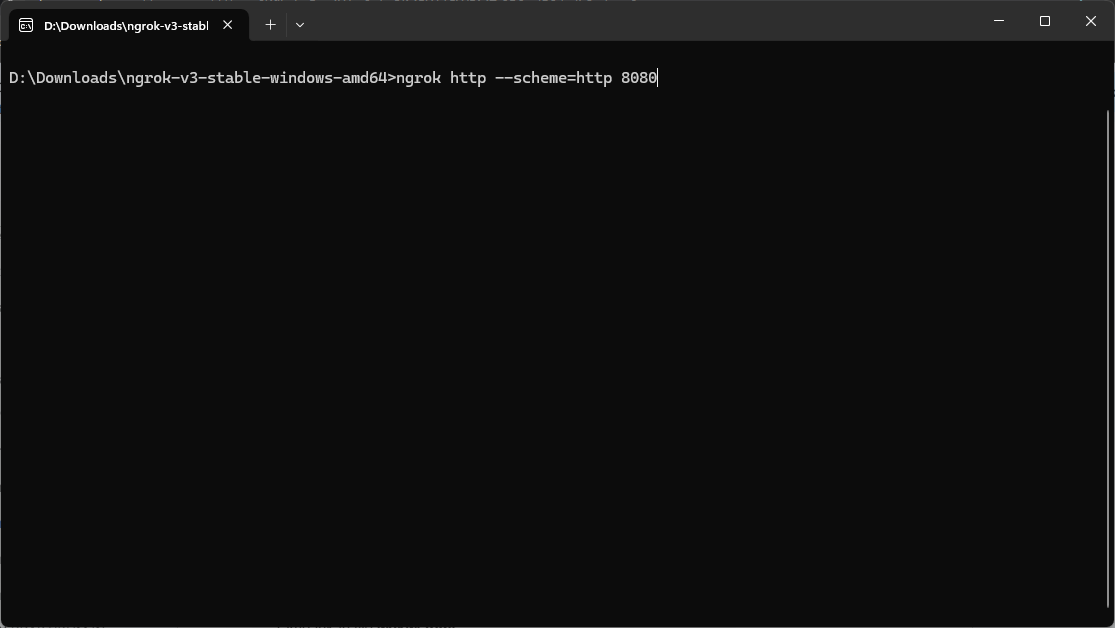
    http.end();

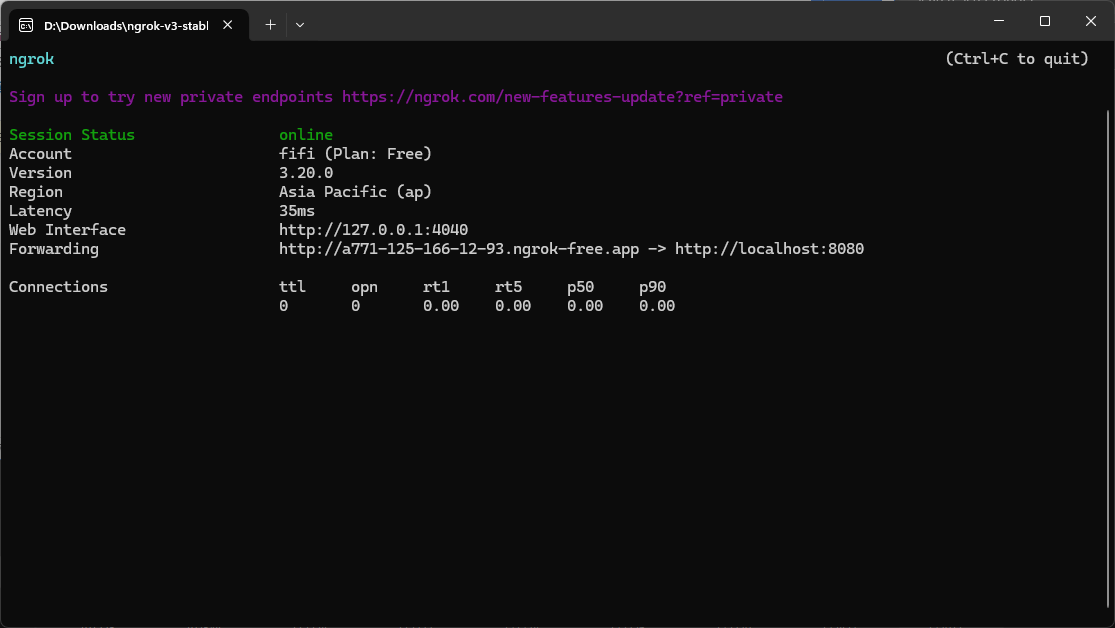
  }

}



1. Menambahkan URL pada bagian **const char\* serverUrl = "#/api/posts";** yang diberikan oleh ngrok dengan cara menulis scema berikut pda cmd: **ngrok http --scheme=http 8080**





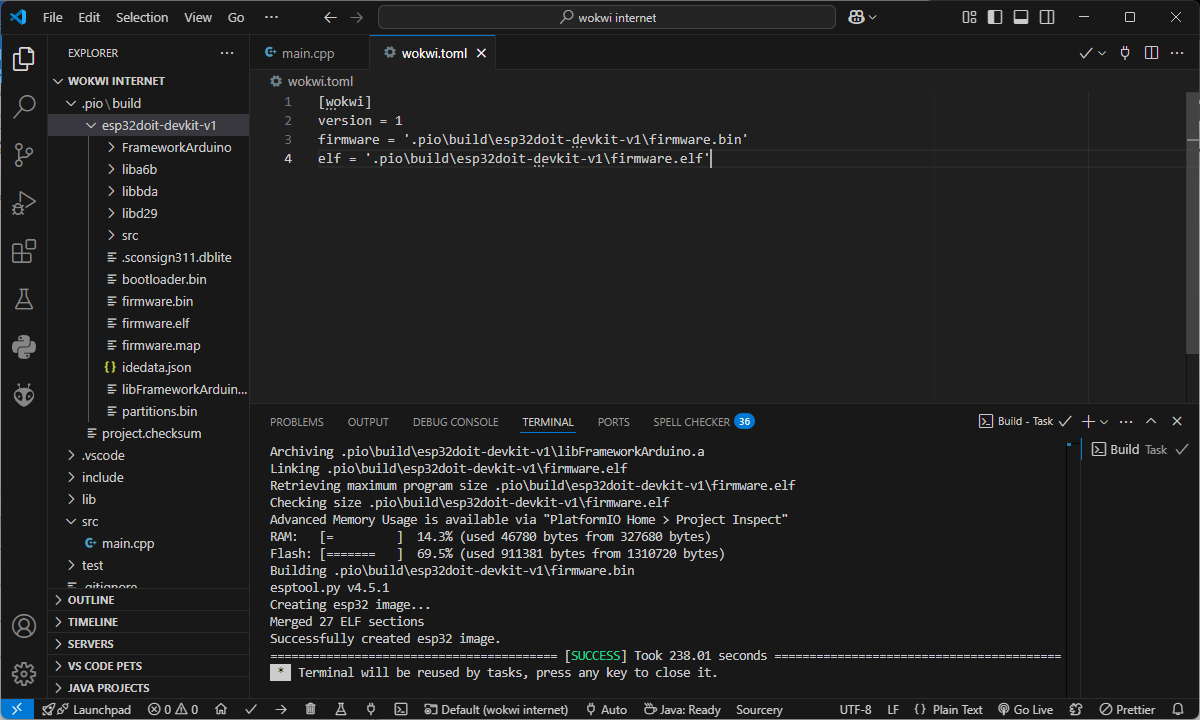
1. Kemudian tambahkan file wokwi.toml.dan diagram.json
2. Wokwi.toml

[wokwi]

version = 1

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

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



1. Diagram.json

{

"version": 1,

"author": "fifi",

"editor": "wokwi",

"parts": [

{ "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },

{ "type": "wokwi-dht22", "id": "dht1", "top": -95.7, "left": -168.6, "attrs": {} }

],

"connections": [

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

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

[ "dht1:VCC", "esp:3V3", "red", [ "v0" ] ],

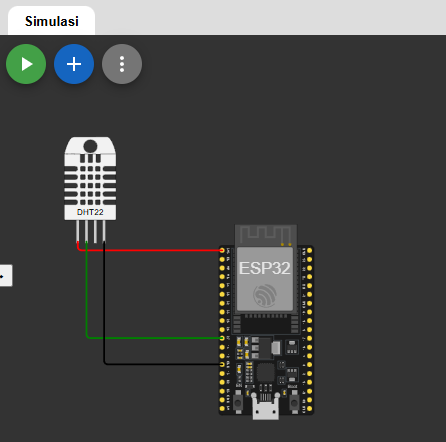
[ "dht1:SDA", "esp:27", "green", [ "v0" ] ],

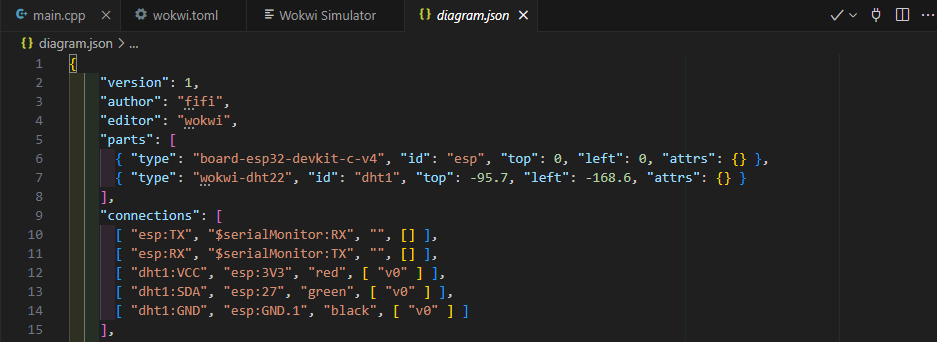
[ "dht1:GND", "esp:GND.1", "black", [ "v0" ] ]

],

"dependencies": {}

}





1. Kemudian lakukan pengeditan dengan menambahkan library dan monitor speed pada file platformIO IDE dan lakukan compiling kembali.

[env:esp32doit-devkit-v1]

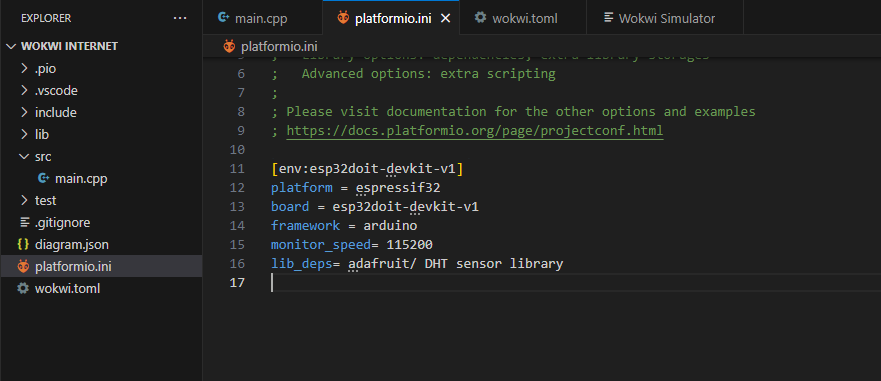
platform = espressif32

board = esp32doit-devkit-v1

framework = arduino

monitor\_speed= 115200

lib\_deps= adafruit/ DHT sensor library



1. Setelah itu