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

**Fakultas Vokasi, Universitas Brawijaya**

**Praktik Akses API Melalui Simulasi WOKWI**

*Author Bayu Maha Resi*

*Email: bayumaharesii@gmail,com*

**Abstract**

Akses API menjadi elemen penting dalam pengembangan aplikasi berbasis web dan IoT, memungkinkan komunikasi antar sistem secara efisien. Eksperimen ini bertujuan untuk mensimulasikan akses API menggunakan **WOKWI**, sebuah platform simulasi berbasis web untuk pengembangan perangkat IoT. Dalam eksperimen ini, digunakan mikrokontroler virtual yang berinteraksi dengan API melalui protokol HTTP untuk mengambil dan mengirim data secara real-time.Konfigurasi dilakukan dengan pemrograman mikrokontroler pada WOKWI untuk mengakses API, memproses respons, serta menampilkan hasilnya dalam antarmuka simulasi. Hasil simulasi menunjukkan bahwa sistem dapat mengakses dan berkomunikasi dengan API secara real-time, memungkinkan integrasi dengan berbagai layanan berbasis cloud. Eksperimen ini memberikan wawasan tentang cara mengakses API dalam lingkungan simulasi, serta bagaimana WOKWI dapat digunakan sebagai alat pengujian sebelum implementasi pada perangkat fisik.

**Kata kunci**: API, WOKWI, simulasi IoT, mikrokontroler virtual, HTTP

**1. Introduction**

Dalam pengembangan sistem berbasis IoT, Application Programming Interface (API) berperan penting dalam memungkinkan komunikasi antara perangkat dan layanan cloud. Salah satu cara untuk menguji akses API tanpa perangkat fisik adalah melalui WOKWI, sebuah platform simulasi berbasis web yang mendukung pengembangan mikrokontroler virtual. Pada praktik ini, dilakukan simulasi akses API menggunakan WOKWI, di mana mikrokontroler virtual dikonfigurasi untuk mengirim dan menerima data melalui protokol HTTP. Simulasi ini bertujuan untuk memahami cara kerja API dalam lingkungan IoT, termasuk proses permintaan data, pengolahan respons, serta integrasi dengan layanan cloud. Dengan menggunakan WOKWI, pengujian API dapat dilakukan secara efisien sebelum implementasi pada perangkat keras sebenarnya, memungkinkan pengembang untuk mengoptimalkan fungsionalitas sistem dengan lebih fleksibel dan tanpa risiko kerusakan perangkat.

* 1. **Latar belakang**

Application Programming Interface (API) menjadi elemen penting dalam memungkinkan komunikasi antara perangkat dan layanan cloud secara efisien. Untuk menguji akses API tanpa perangkat fisik, digunakan WOKWI, sebuah platform simulasi berbasis web yang mendukung pengembangan dan pengujian mikrokontroler virtual. Dalam praktik ini, dilakukan simulasi akses API melalui WOKWI, di mana mikrokontroler virtual dikonfigurasi untuk mengirim permintaan HTTP dan mengolah respons dari server. Simulasi ini memungkinkan pengembang memahami cara kerja API dalam sistem IoT, termasuk integrasi dengan layanan cloud dan pengujian konektivitas secara real-time. Dengan WOKWI, pengujian API dapat dilakukan dengan lebih fleksibel, tanpa perlu konfigurasi perangkat keras yang kompleks, sehingga mempercepat proses pengembangan dan mengurangi risiko kesalahan sebelum implementasi pada perangkat fisik.

**1.2 Tujuan eksperimen**

Eksperimen ini bertujuan untuk membangun dan menguji API menggunakan Laravel 11 serta mengekspose-nya secara publik dengan Ngrok. Laravel 11 digunakan untuk membuat endpoint API, mengelola routing, dan menerapkan autentikasi, sementara Ngrok mempermudah akses tanpa konfigurasi server yang kompleks.Pengujian dilakukan menggunakan Postman atau cURL untuk memastikan API berfungsi dengan baik. Eksperimen ini memberikan wawasan tentang pengembangan API yang efisien, aman, dan scalable, serta integrasinya dalam sistem berbasis cloud.

**2. Methodology**

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

* Website Wokwi
* Laptop
* Github
* VScode

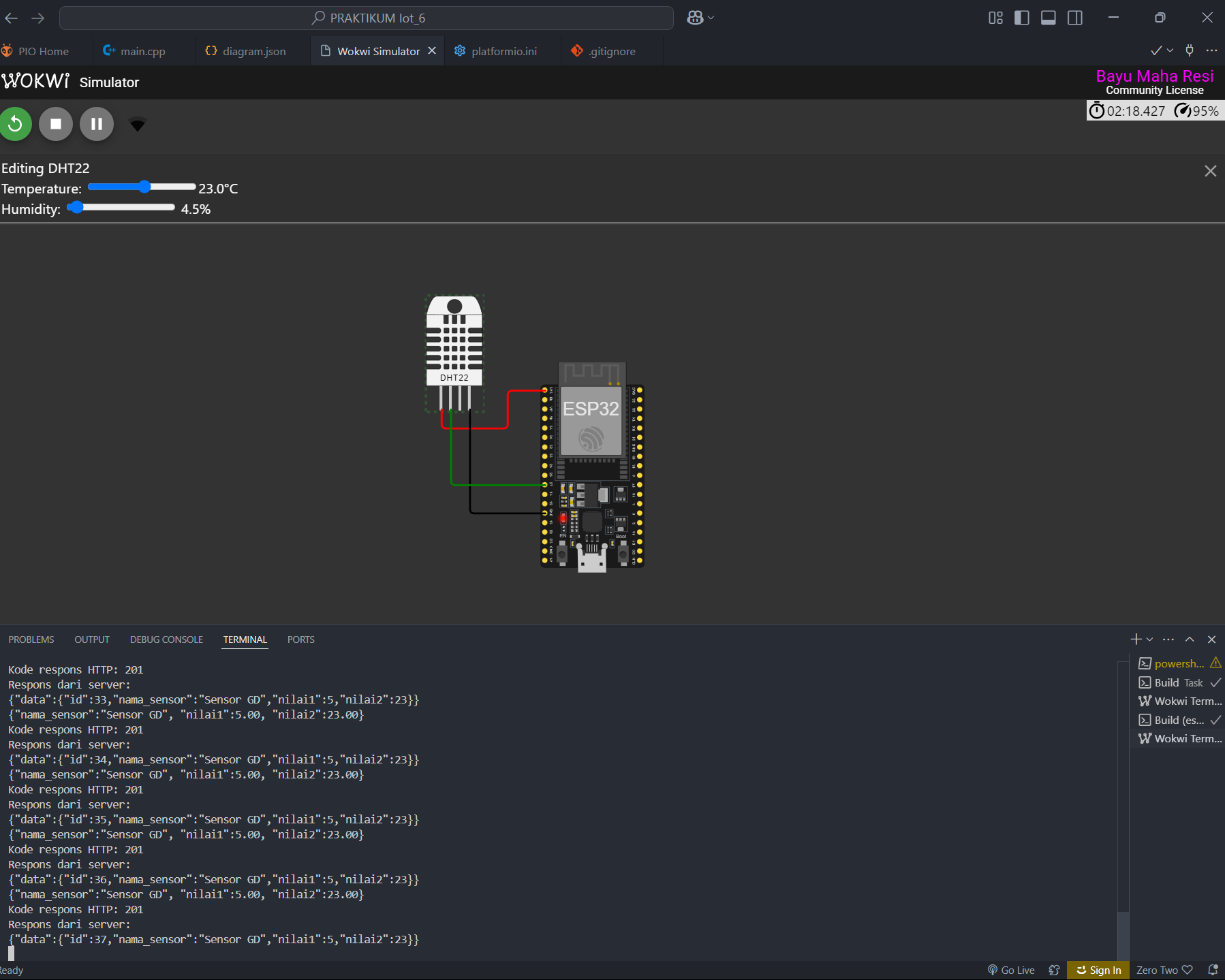
**2.2 Implementation Steps (Langkah Implementasi)**

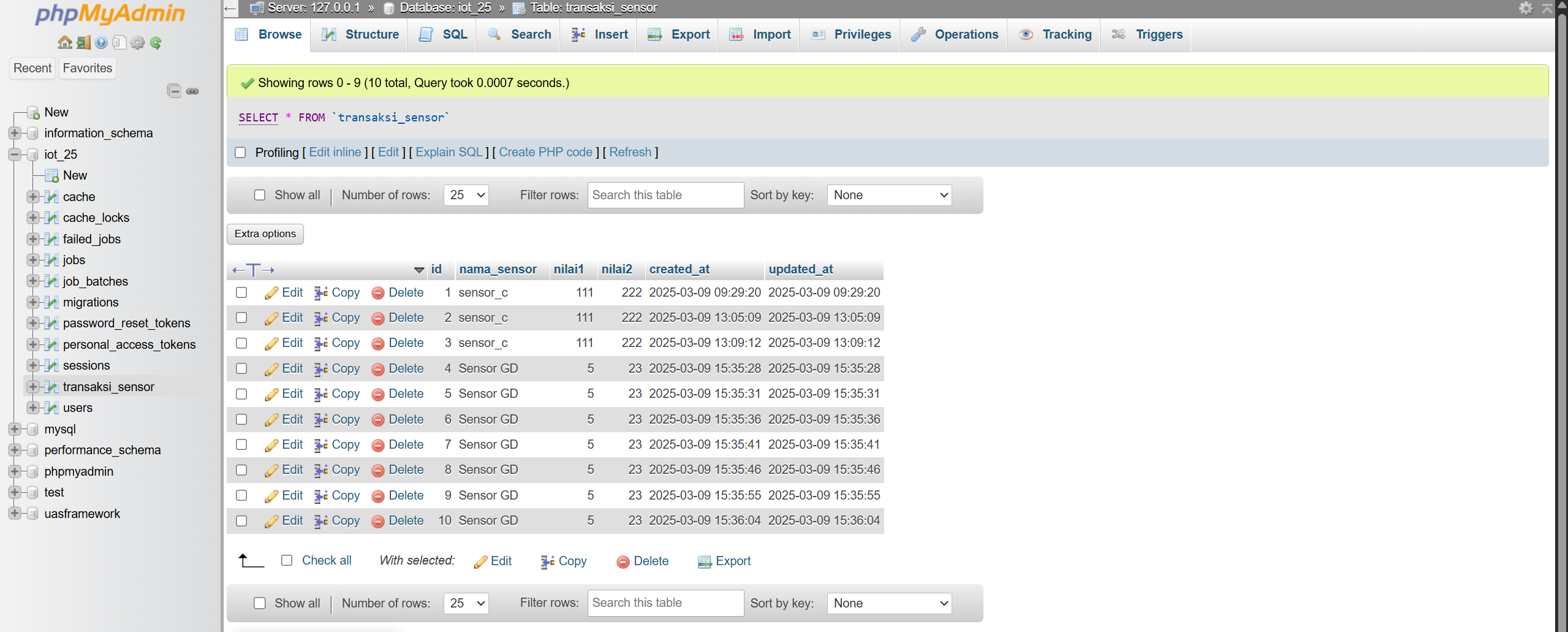
Buat akun wokwi dengan GitHub

* Membuat topology
* Memindah diagram.json di Wokwi kedalam Vscode
* Memindahkan code dari wokwi ke Vs Code
* Lakukan configurasi ulang
* Tambahkan code pada file platform io ini

**3. Results and Discussion**

**3.1 Experimental Results**





**4. Appendix**

**Code Program:**

**Main.cpp**

#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://e85a-114-10-46-166.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();

  }

}