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

**Fakultas Vokasi, Universitas Brawijaya**



**Praktik Akses API Melalui   
Simulasi WOKWI**

*Angger Abed Nego*

Fakultas Vokasi, Universitas Brawijaya

Email : [bedvyy7@student.ub.ac.id](mailto:bedvyy7@student.ub.ac.id)

* **Abstrack**

Eksperimen ini bertujuan untuk mempelajari dan mengimplementasikan akses API menggunakan ESP32 melalui simulasi Wokwi. Dalam praktik ini, ESP32 mengirimkan permintaan HTTP ke server API dan menerima respons dalam format JSON. Respons tersebut kemudian diproses dan ditampilkan melalui Serial Monitor. Hasil eksperimen menunjukkan bahwa ESP32 mampu melakukan komunikasi HTTP dengan baik melalui simulasi Wokwi, sehingga dapat dijadikan dasar untuk pengembangan sistem IoT berbasis web API.

Keywords: Internet of Things, ESP32, Wokwi, API, HTTP Request

1. **Introduction**

1.1 **Latar Belakang**

Dalam era Internet of Things (IoT), kemampuan perangkat untuk berkomunikasi dengan layanan web melalui Application Programming Interface (API) menjadi sangat penting. Salah satu mikrokontroler populer yang mendukung komunikasi internet adalah ESP32. Dengan bantuan simulator Wokwi, proses pengembangan dan pengujian program menjadi lebih mudah tanpa memerlukan perangkat keras fisik.

1.2 **Tujuan**

Tujuan dari eksperimen ini adalah:

1. Mempelajari cara ESP32 mengakses dan memproses data dari API.
2. Menguji kemampuan simulasi Wokwi dalam menjalankan permintaan HTTP.
3. Menampilkan dan menganalisis hasil respons dari API secara real-time.
4. **Metodhlogy**

2.1 **Alat dan Bahan**

* Mikrokontroler ESP32 (simulasi Wokwi)
* Koneksi internet
* Wokwi Simulator
* API publik (misalnya, API cuaca, quotes, atau lainnya)
* Software Arduino IDE

2.2 **Langkah Implementasi**

1. Membuat proyek baru di simulator Wokwi.
2. Menambahkan library WiFi dan HTTPClient di Arduino IDE.
3. Menulis program untuk menghubungkan ESP32 ke jaringan WiFi dan mengakses API.
4. Mengatur URL API dan memproses data JSON dari respons.
5. Menampilkan hasil data pada Serial Monitor.
6. **Result and Disscusion**

3.1 **Hasil Eksperimen**

|  |
| --- |
|  |

1. **Appendix**

**4.1 Kode Program**

|  |
| --- |
| #include <WiFi.h>  #include <HTTPClient.h>  // Ganti dengan kredensial jaringan Wi-Fi Anda  // const char\* ssid = "Lab IT";  // const char\* password = "labit2024";    const char\* ssid = "Wokwi-GUEST";  const char\* password = "";    // URL lengkap server yang akan diakses  const char\* serverUrl = "http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app/api/posts";  // Interval waktu antara setiap permintaan (dalam milidetik)  const unsigned long interval = 5000;  unsigned long previousMillis = 0;  void setup() {  Serial.begin(115200);  WiFi.begin(ssid, password);  Serial.print("Menghubungkan ke WiFi...");  while (WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.println(" Terhubung!");  }    void loop() {    unsigned long currentMillis = millis();  // Periksa apakah interval waktu telah berlalu  if (currentMillis - previousMillis >= interval) {  previousMillis = currentMillis;  if (WiFi.status() == WL\_CONNECTED) {  HTTPClient http;  // Inisialisasi HTTPClient dengan URL server  http.begin(serverUrl);  // Mengirim permintaan HTTP GET  int httpResponseCode = http.GET();  // Menampilkan kode status HTTP  Serial.print("Kode status HTTP: ");  Serial.println(httpResponseCode);  // Menutup koneksi  http.end();  } else {  Serial.println("WiFi tidak terhubung.");  }  }  } |

4.2 Dokumentasi Tambahan

|  |
| --- |
|  |