# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

# Fakultas Vokasi, Universitas Brawijaya

**Praktik Pembuatan Akun Wokwi, Github, dan Simulasi 3 LED ESP32**

**Author(s)** Muhammad Hafizh Al Furqon

*Fakultas Vokasi, Universitas Brawijaya*

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

Simulasi demo ESP32 dengan 3 buah LED sebagai representasi traffic light menggunakan platform Wokwi dan PlatformIO. Praktikum ini dilakukan untuk memahami dasar-dasar pemrograman mikrokontroler ESP32, penggunaan extension PlatformIO di VSCode, serta integrasi antara pembuatan akun Wokwi dan Github. Hasil praktikum menunjukkan bahwa simulasi traffic light dapat dijalankan dengan lancar dan memberikan gambaran konsep pengaturan LED secara bergantian dengan delay tertentu.

**Keywords**—*Internet of Things, Wokwi, PlatformIO, LED, ESP32*

**1. Introduction**

**1.1 Latar belakang**

Perkembangan teknologi Internet of Things (IoT) telah memungkinkan integrasi antara dunia nyata dengan dunia digital melalui penggunaan mikrokontroler seperti ESP32. Simulasi traffic light merupakan salah satu aplikasi dasar yang menggambarkan bagaimana mikrokontroler dapat mengendalikan LED untuk mensimulasikan sistem pengaturan lampu lalu lintas. Praktikum ini dilakukan sebagai dasar pemahaman tentang pemrograman ESP32, pengaturan GPIO, dan penerapan simulasi menggunakan platform berbasis cloud seperti Wokwi serta proses kompilasi menggunakan PlatformIO di VSCode.

**1.2 Tujuan eksperimen**

* Mempelajari cara penggunaan ESP32 dalam mengendalikan LED.
* Mempelajari pembuatan akun dan penggunaan Wokwi untuk simulasi IoT.
* Menerapkan konsep dasar pemrograman mikrokontroler melalui simulasi traffic light.
* Mengintegrasikan proyek dengan Github untuk dokumentasi dan versioning.
* Menggunakan extension PlatformIO di VSCode untuk proses kompilasi dan upload program.

**2. Methodology (Metodologi)**

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

Mikrokontroler: ESP32

LED: Tiga LED (merah, kuning, hijau) untuk simulasi traffic light

Software:

* Visual Studio Code (VSCode)
* Extension PlatformIO (untuk proses compile)
* Extension Wokwi (untuk simulasi dan pembuatan diagram)
* Github (untuk repository dan version control)

Diagram: File diagram.json yang diambil dari website Wokwi sebagai acuan skematik

Kabel **dan Breadboard** (opsional, jika menggunakan perangkat fisik)

**2.2 Implementation Steps (Langkah Implementasi)**

Pembuatan Akun dan Setup Tools

* Mendaftar akun di Wokwi dan Github.
* Menginstall VSCode beserta extension PlatformIO dan Wokwi.

Penyusunan Proyek di VSCode dengan PlatformIO

* Membuat proyek baru dengan PlatformIO dan memilih board ESP32.
* Menyalin file diagram.json yang telah dibuat di website Wokwi ke dalam direktori proyek (misalnya di folder docs/ atau resources/) sebagai dokumentasi skematik.

Pengkodean Program

* Menentukan pin untuk masing-masing LED (misalnya: LED Merah pada GPIO13, LED Kuning pada GPIO33, dan LED Hijau pada GPIO34 sesuai kode; namun catatan: gunakan GPIO output-capable seperti GPIO32 untuk LED Hijau pada aplikasi nyata).
* Menulis kode untuk simulasi traffic light dengan urutan:
  1. LED Merah menyala (Stop).
  2. LED Kuning menyala (Persiapan).
  3. LED Hijau menyala (Jalan).
* Menambahkan delay antara pergantian LED.

Pengujian dan Kompilasi Menggunakan PlatformIO

* Menggunakan PlatformIO di VSCode untuk melakukan kompilasi dan upload kode ke board ESP32.
* Jika menggunakan simulasi, file diagram.json dari Wokwi dapat digunakan sebagai referensi visual.

Integrasi dengan Github

* Melakukan commit dan push kode beserta dokumentasi (termasuk diagram.json) ke repository Github.
* Menyertakan dokumentasi dan laporan praktikum.

**3. Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

Setelah melakukan kompilasi dan simulasi menggunakan PlatformIO serta memverifikasi skematik melalui file diagram.json dari Wokwi, sistem traffic light berhasil dijalankan dengan urutan pergantian LED sesuai dengan pengaturan:

* LED1 (GPIO13) menyala selama 1 detik dengan output Serial yang menampilkan status.
* LED2 (GPIO33) menyala selama 1 detik dengan output Serial yang menampilkan status.
* LED3 (GPIO34) menyala selama 1 detik dengan output Seri**al yang menampilkan status.**

Tabel berikut menunjukkan urutan dan waktu delay setiap LED:

| **LED** | **Pin** | **Durasi Menyala(Detik)** | **Fungsi** |
| --- | --- | --- | --- |
| LED1 | GPIO13 | 1 | Menyala (misal, Stop) |
| LED2 | GPIO33 | 1 | Menyala (misal, Ready) |
| LED3 | GPIO34 | 1 | Menyala (misal, Go) |

Simulasi berjalan sesuai dengan urutan program. Keluaran serial memberikan informasi status LED yang aktif. Meskipun kode menggunakan GPIO34 untuk LED3, sebaiknya dalam implementasi nyata gunakan GPIO yang mendukung output (contoh: GPIO32) karena GPIO34 bersifat input-only pada ESP32. Penggunaan PlatformIO mempermudah proses kompilasi dan upload kode, sedangkan file diagram.json yang diimpor dari Wokwi berfungsi sebagai dokumentasi visual skematik.

**4. Appendix (Lampiran, jika diperlukan)**

**4.1 Kode Program (src/main.cpp)**

| #include <Arduino.h>  #define LED1 13 #define LED2 33 #define LED3 34   void setup() {  // Inisialisasi Serial Monitor  Serial.begin(115200);  // Inisialisasi pin sebagai output  pinMode(LED1, OUTPUT);  pinMode(LED2, OUTPUT);  pinMode(LED3, OUTPUT); }  void loop() {  // Nyalakan LED1, matikan LED2 dan LED3  digitalWrite(LED1, HIGH);  digitalWrite(LED2, LOW);  digitalWrite(LED3, LOW);  Serial.println("LED2 is off");  Serial.println("LED3 is off");  Serial.println("LED1 is on");  Serial.println("----------");  delay(1000);   // Nyalakan LED2, matikan LED1 dan LED3  digitalWrite(LED1, LOW);  digitalWrite(LED2, HIGH);  digitalWrite(LED3, LOW);  Serial.println("LED1 is off");  Serial.println("LED2 is on");  Serial.println("----------");  delay(1000);   // Nyalakan LED3, matikan LED1 dan LED2  digitalWrite(LED1, LOW);  digitalWrite(LED2, LOW);  digitalWrite(LED3, HIGH);  Serial.print("LED3 is on");  Serial.println(" | LED2 is off");  Serial.println("----------");  delay(1000); } |
| --- |

**4.2 Diagram Skematik (diagram.json)**

File diagram.json yang dihasilkan dari website Wokwi disimpan dalam folder proyek (misalnya, docs/diagram.json) dan berisi informasi visual mengenai koneksi antara LED dan ESP32. Contoh struktur file:

| {  "version": 1,  "author": "Naxumi",  "editor": "wokwi",  "parts": [  { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": -38.4, "left": -14.36, "attrs": {} },  {  "type": "wokwi-led",  "id": "led1",  "top": -30,  "left": -200.2,  "rotate": 270,  "attrs": { "color": "green", "flip": "1" }  },  {  "type": "wokwi-resistor",  "id": "r1",  "top": -15.25,  "left": -115.2,  "attrs": { "value": "100" }  },  {  "type": "wokwi-led",  "id": "led2",  "top": 94.8,  "left": -200.2,  "rotate": 270,  "attrs": { "color": "red", "flip": "1" }  },  {  "type": "wokwi-resistor",  "id": "r2",  "top": 109.55,  "left": -134.4,  "attrs": { "value": "100" }  },  {  "type": "wokwi-led",  "id": "led3",  "top": 27.6,  "left": -200.2,  "rotate": 270,  "attrs": { "color": "yellow", "flip": "1" }  },  {  "type": "wokwi-resistor",  "id": "r3",  "top": 42.65,  "left": -126.2,  "rotate": 180,  "attrs": { "value": "1000" }  }  ],  "connections": [  [ "esp:TX", "$serialMonitor:RX", "", [] ],  [ "esp:RX", "$serialMonitor:TX", "", [] ],  [ "r2:1", "led2:C", "green", [ "v0" ] ],  [ "led2:A", "esp:13", "green", [ "h0" ] ],  [ "r2:2", "esp:GND.1", "green", [ "v0" ] ],  [ "led3:C", "r3:2", "green", [ "h0" ] ],  [ "r3:1", "esp:GND.3", "green", [ "v0" ] ],  [ "led3:A", "esp:33", "green", [ "h0" ] ],  [ "led1:C", "r1:1", "green", [ "h0" ] ],  [ "r1:2", "esp:GND.2", "green", [ "v0" ] ],  [ "led1:A", "esp:23", "green", [ "h0" ] ]  ],  "dependencies": {} } |
| --- |

**4.3 Screenshot**

