**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)** Fakultas Vokasi, Universitas Brawijaya

### **BAB 8 Praktik Penggunaan Platform Wokwi**



*Cantika Kelana*  
 Fakultas Vokasi, Universitas Brawijaya  
 Email: [cantikakln@student.ub.ac.id](mailto:cantikakln@student.ub.ac.id) / [cantikakln@student.ub.ac.id](mailto:cantikakln@student.ub.ac.id)

## **Abstract (Abstrak)**

Praktikum ini bertujuan untuk memahami dasar penggunaan platform Wokwi dan GitHub dalam mendukung pengembangan proyek berbasis IoT. Wokwi berperan sebagai simulator perangkat keras yang memungkinkan pengguna untuk mendesain dan menguji sistem IoT tanpa perlu menggunakan perangkat fisik. Sementara itu, GitHub difungsikan sebagai alat manajemen versi yang memudahkan penyimpanan, kolaborasi, serta pengelolaan kode sumber proyek IoT secara efisien.

Dalam eksperimen ini, dilakukan serangkaian langkah mulai dari pembuatan akun Wokwi dan GitHub, pembuatan simulasi mikrokontroler di Wokwi, hingga proses pengunggahan dan pengelolaan kode proyek ke repositori GitHub. Hasil percobaan menunjukkan bahwa Wokwi memberikan kemudahan dalam simulasi perangkat keras dengan antarmuka yang ramah pengguna, sementara GitHub memfasilitasi pengelolaan kode dengan sistem version control serta fitur kolaborasi tim.

Kesimpulan dari praktikum ini adalah bahwa pemanfaatan Wokwi dan GitHub sangat membantu dalam pengembangan proyek IoT, terutama bagi mahasiswa dan pengembang yang ingin bereksperimen tanpa harus memiliki perangkat keras secara langsung. Kombinasi kedua platform ini membuat proses pembelajaran dan pengembangan sistem berbasis IoT menjadi lebih efisien, terstruktur, dan mudah diakses.

***Keywords****—Internet of Things, Wokwi, GitHub, Simulasi, Mikrokontroler*

## **1. Introduction (Pendahuluan)**

### **1.1 Latar Belakang**

Perkembangan teknologi Internet of Things (IoT) semakin pesat dan telah diimplementasikan dalam berbagai sektor, mulai dari rumah pintar, industri manufaktur, hingga layanan kesehatan. Untuk mengembangkan sistem berbasis IoT, pemahaman tidak hanya terbatas pada perangkat keras, tetapi juga mencakup keterampilan dalam pemrograman dan pengelolaan sistem berbasis cloud. Oleh karena itu, mahasiswa teknologi informasi perlu menguasai perangkat dan platform yang mendukung pengembangan IoT.

Salah satu kendala dalam pembelajaran IoT adalah keterbatasan akses terhadap perangkat keras seperti mikrokontroler, sensor, dan aktuator. Untuk mengatasi hal ini, Wokwi menyediakan solusi berupa platform simulasi berbasis web yang memungkinkan pengguna untuk mengembangkan dan menguji kode IoT tanpa memerlukan perangkat fisik. Dengan fitur interaktif dan antarmuka yang intuitif, Wokwi membantu mahasiswa dalam memahami konsep pemrograman mikrokontroler seperti Arduino dan ESP32.

Di sisi lain, pengelolaan kode sumber dan kolaborasi tim menjadi aspek penting dalam pengembangan proyek IoT. GitHub sebagai layanan version control memungkinkan pengguna untuk menyimpan, mengelola, dan berkolaborasi dalam pengembangan kode program secara daring. Dengan GitHub, mahasiswa dapat bekerja dalam tim, melacak perubahan kode, serta mengelola berbagai versi proyek dengan lebih efektif.

Dengan menguasai penggunaan Wokwi dan GitHub, mahasiswa dapat lebih siap dalam mengembangkan proyek berbasis IoT, baik dalam skala kecil maupun proyek kolaboratif yang lebih kompleks. Oleh karena itu, praktikum ini bertujuan untuk memberikan pemahaman yang lebih mendalam mengenai pemanfaatan kedua platform tersebut dalam pengembangan sistem IoT.

### **1.2 Tujuan**

1. Memahami cara membuat dan menggunakan akun Wokwi untuk simulasi perangkat keras.
2. Memahami cara membuat dan menggunakan akun Github untuk menyimpan dan mengelola kode proyek IoT.
3. Mengimplementasikan integrasi antara Wokwi dan Github dalam pengembangan proyek IoT.

## **2. Methodology (Metodologi)**

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

1. **Mikrokontroler:** ESP32 (simulasi di Wokwi)
2. **Komponen elektronik:** LED merah, kuning, hijau, resistor 220Ω
3. **Software:** Wokwi ([https://wokwi.com](https://wokwi.com/)), GitHub ([https://github.com](https://github.com/))

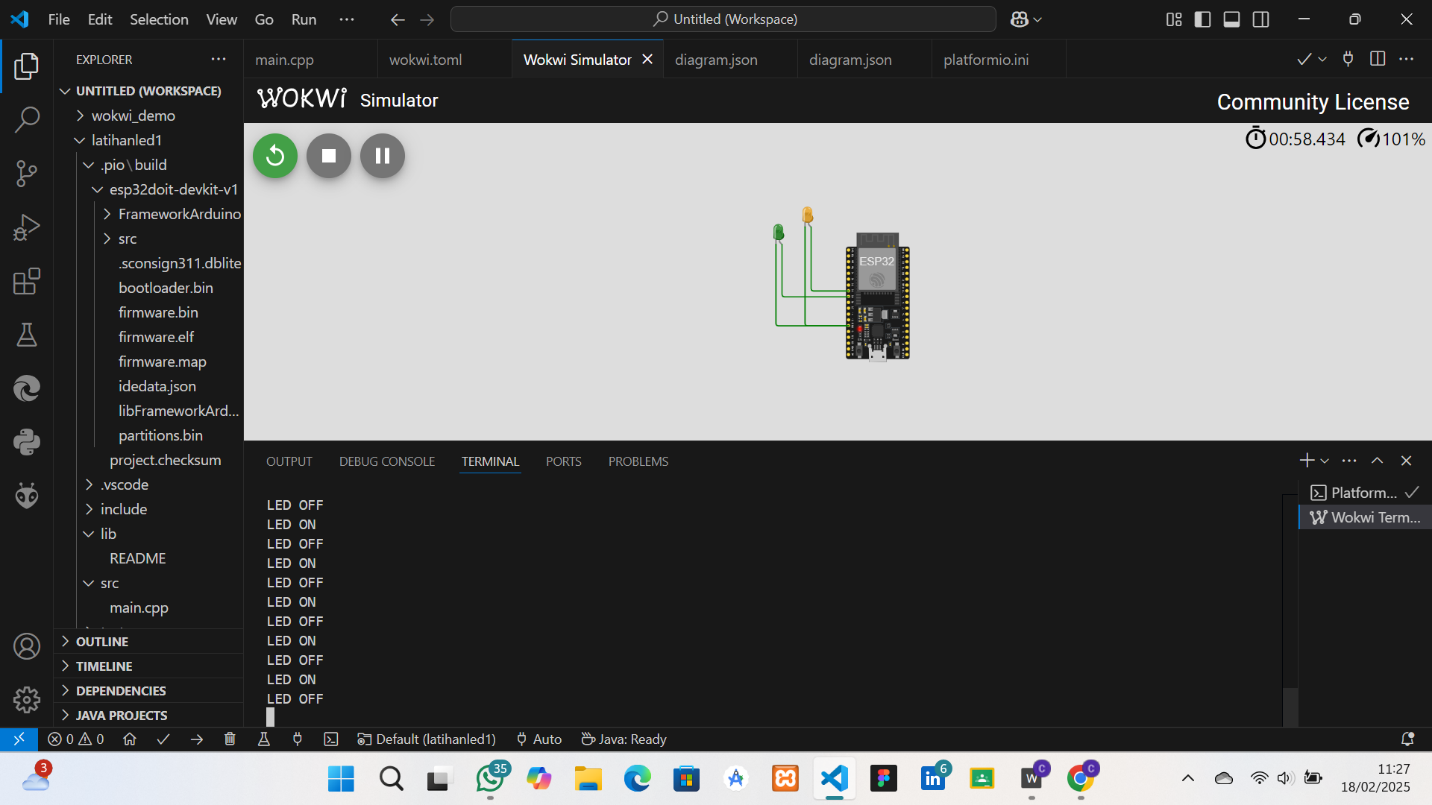
### **2.2 Implementation Steps (Langkah Implementasi)**

1. **Pembuatan Akun Wokwi dan GitHub**
   1. Mendaftar di platform Wokwi dan GitHub.
2. **Membangun Rangkaian Lampu Lalu Lintas di Wokwi**
   1. Menyusun skematik rangkaian LED untuk lampu merah, kuning, dan hijau.
   2. Menulis kode program ESP32 untuk mengontrol lampu lalu lintas.
3. **Mengunggah Kode ke GitHub**
   1. Membuat repository baru di GitHub.
   2. Mengunggah file kode program dan dokumentasi.

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

### **3.1 Experimental Results (Hasil Eksperimen)**

1. Berhasil membuat akun Wokwi dan GitHub.
2. Simulasi sistem lampu lalu lintas berjalan dengan baik di Wokwi.
3. Kode program berhasil diunggah ke GitHub.



## **4. Appendix (Lampiran)**

### **4.1 Kode Program**

#include <Arduino.h>  // Wajib untuk PlatformIO + ESP32

// Deklarasi pin LED

int lampu = 26;

int lampu2 = 33;

void setup() {

    Serial.begin(115200);  // Inisialisasi komunikasi Serial

    Serial.println("ESP32 Blinking LED");

    // Atur pin sebagai OUTPUT

    pinMode(lampu, OUTPUT);

    pinMode(lampu2, OUTPUT);

}

void loop() {

    // Nyalakan kedua LED

    digitalWrite(lampu, HIGH);

    digitalWrite(lampu2, HIGH);

    Serial.println("LED ON");

    delay(1000); // Tunggu 1 detik

    // Matikan kedua LED

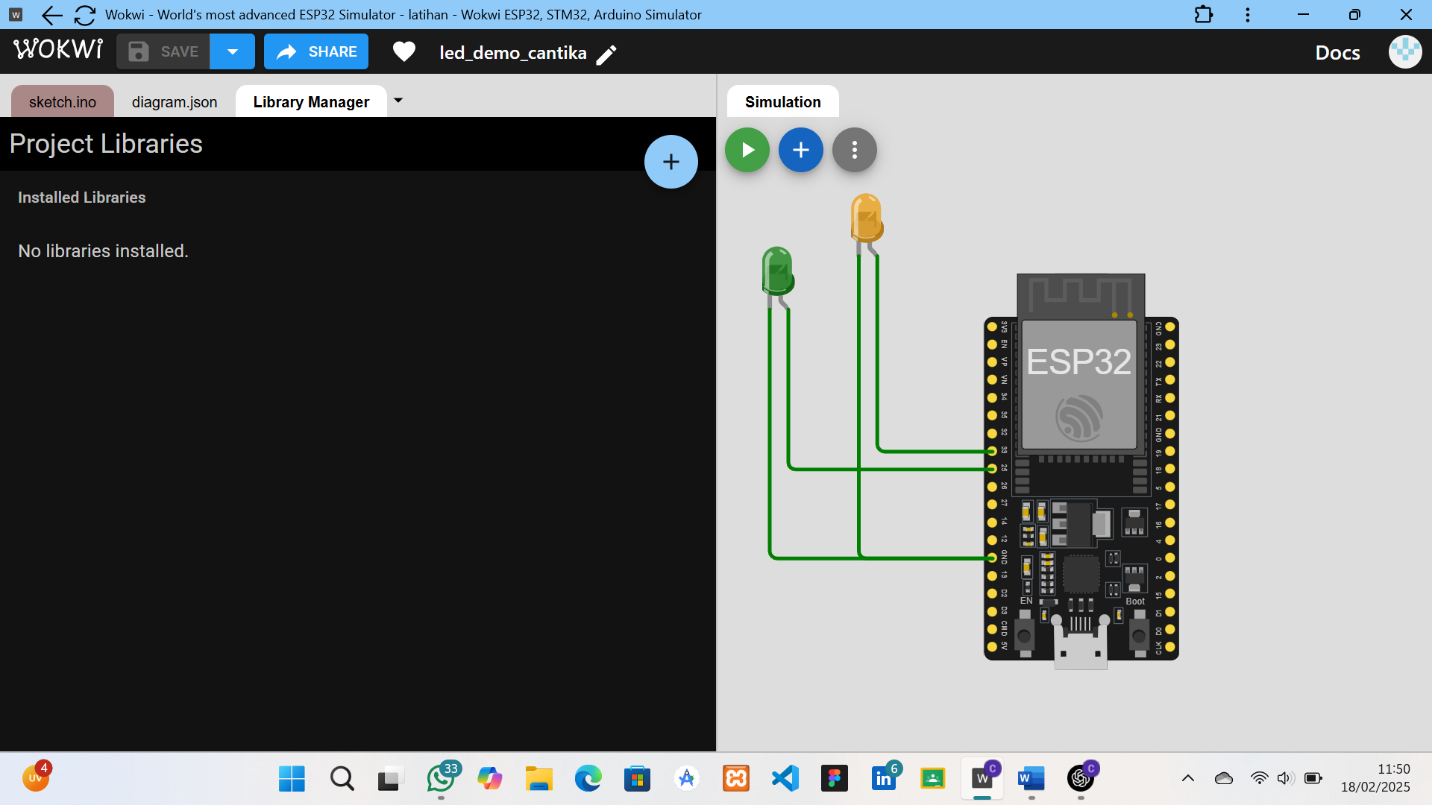
    digitalWrite(lampu, LOW);

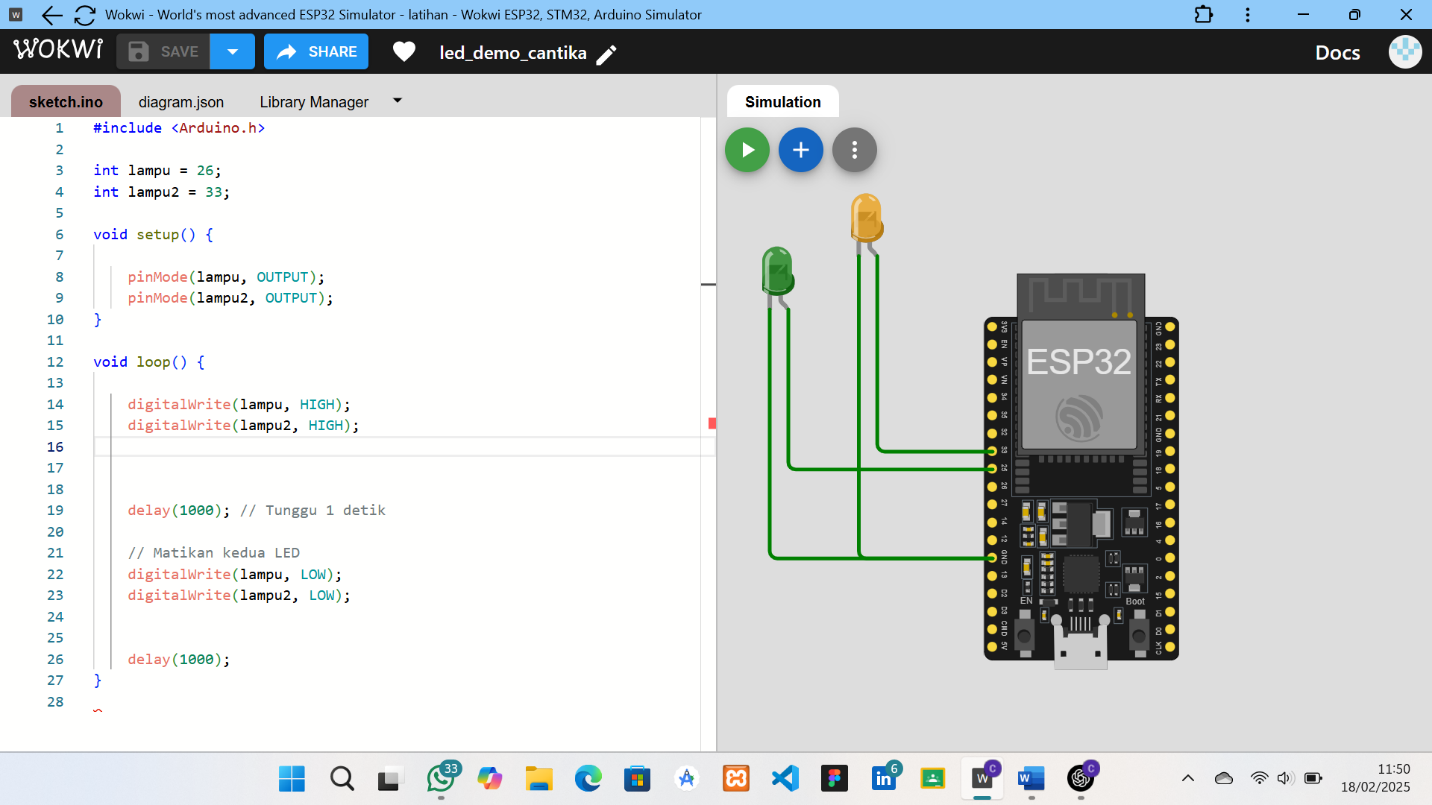
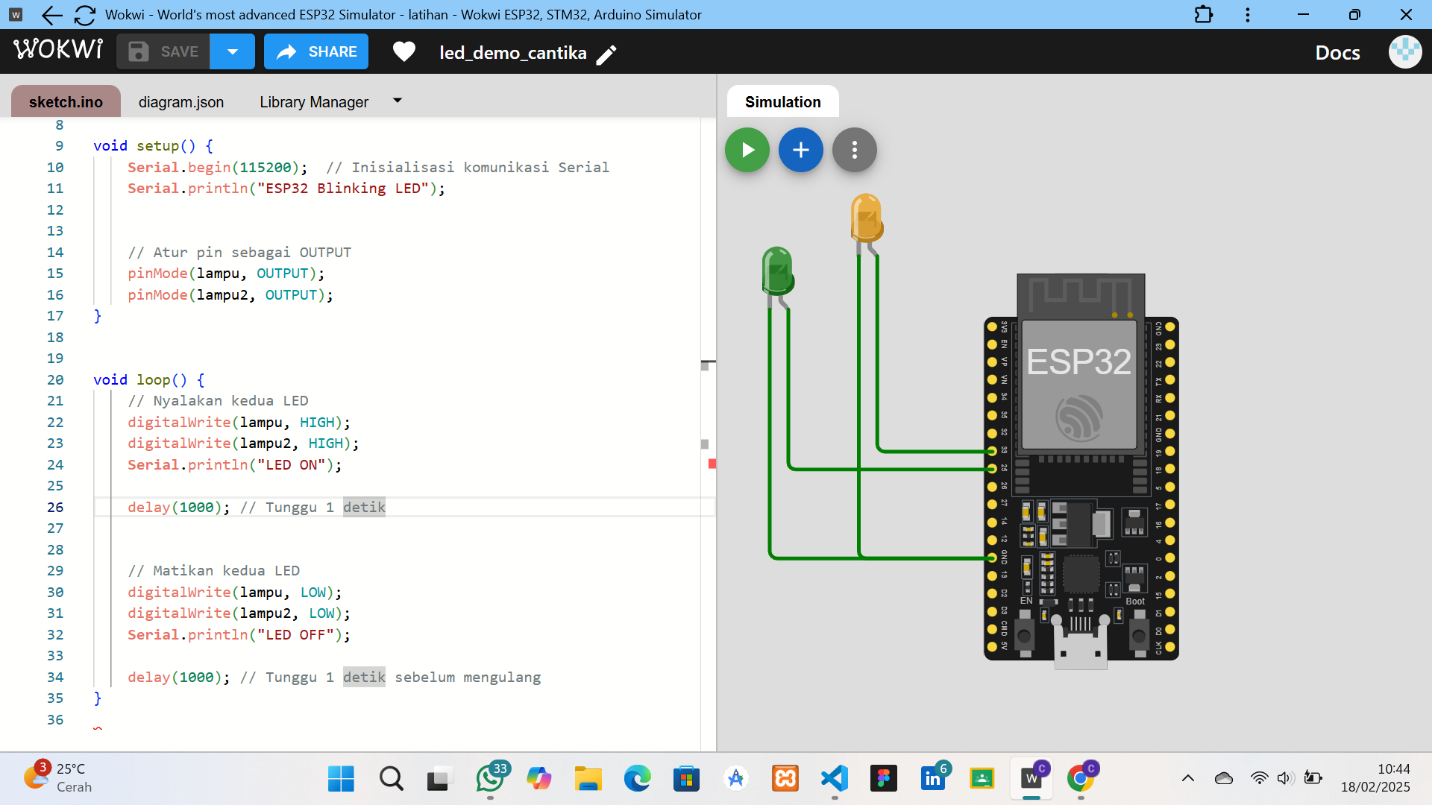
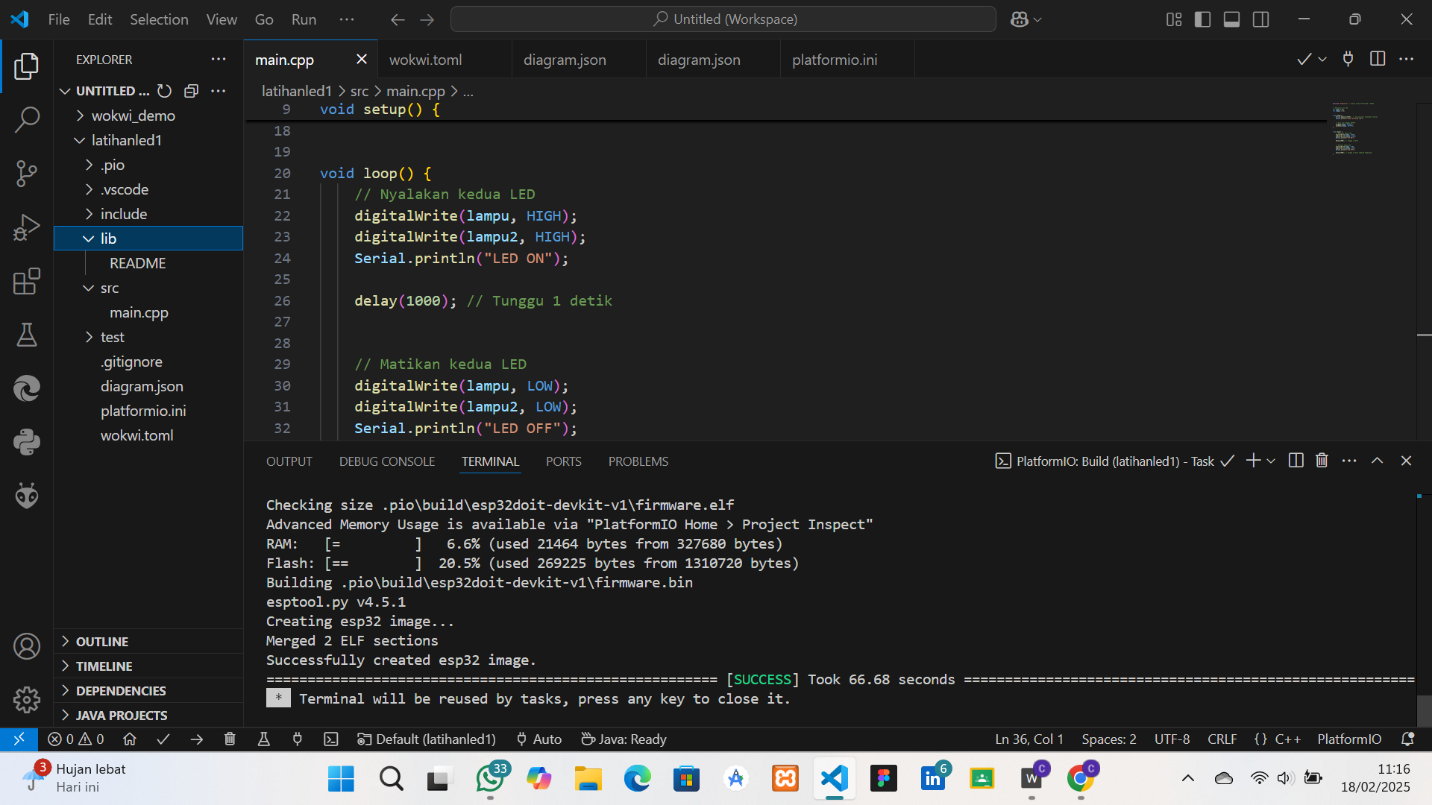
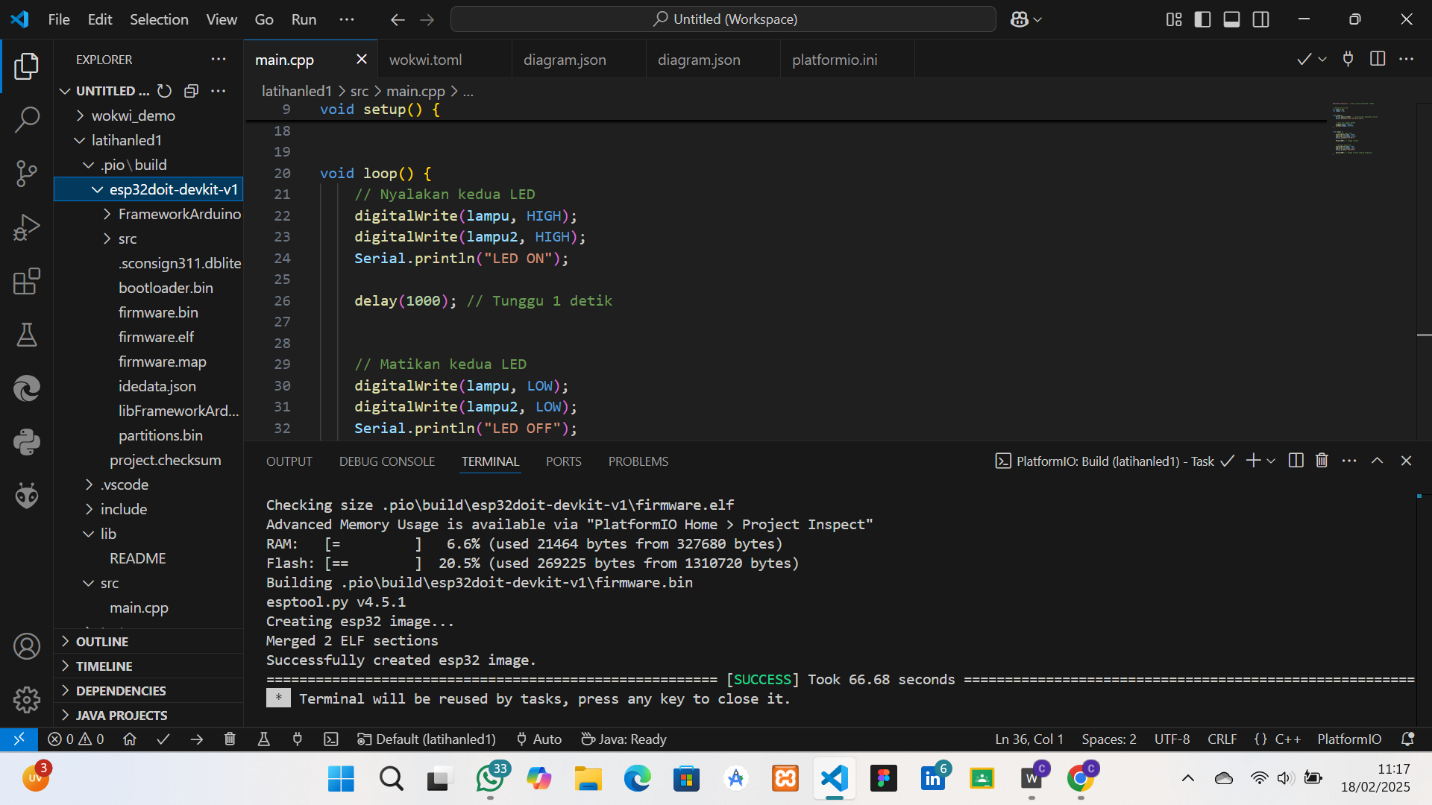
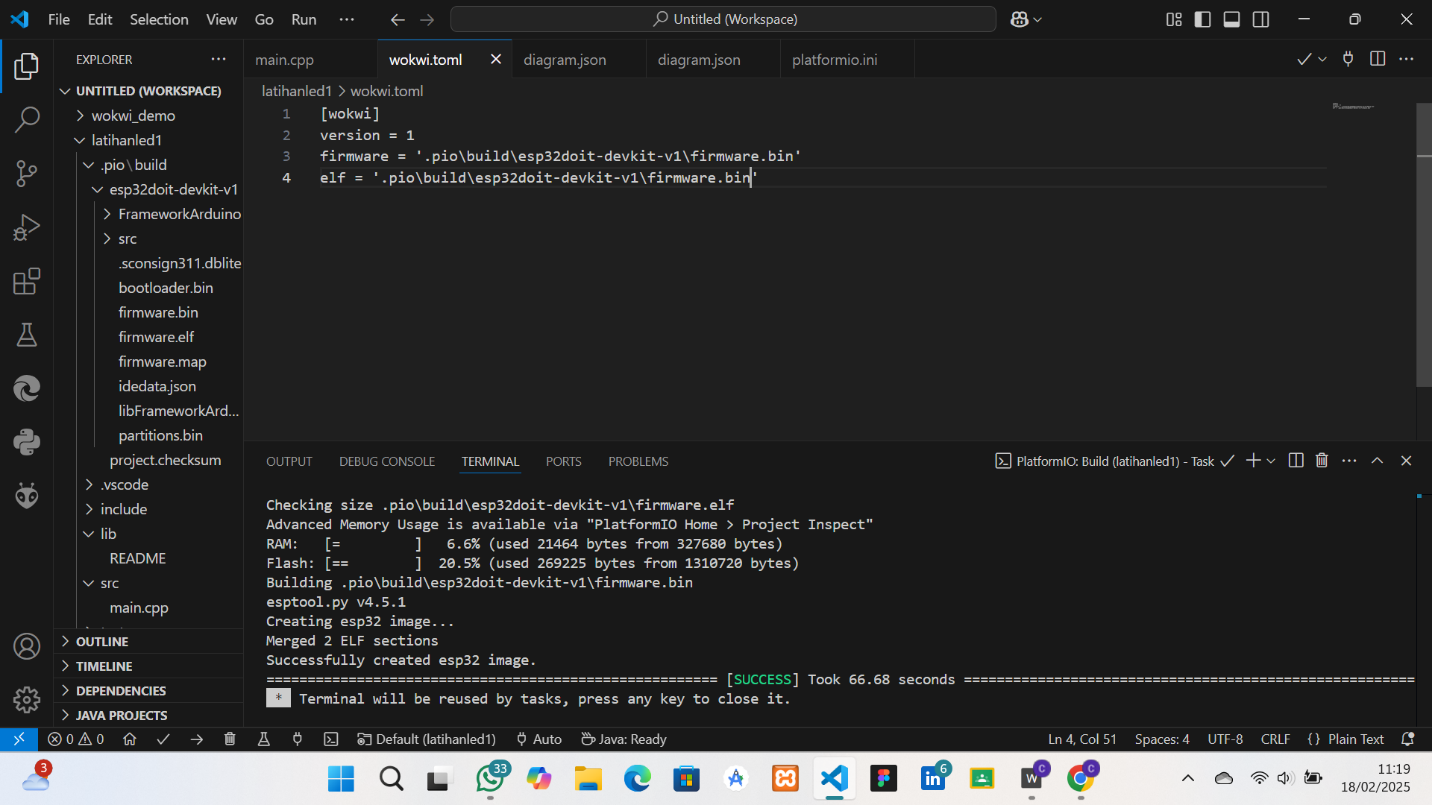
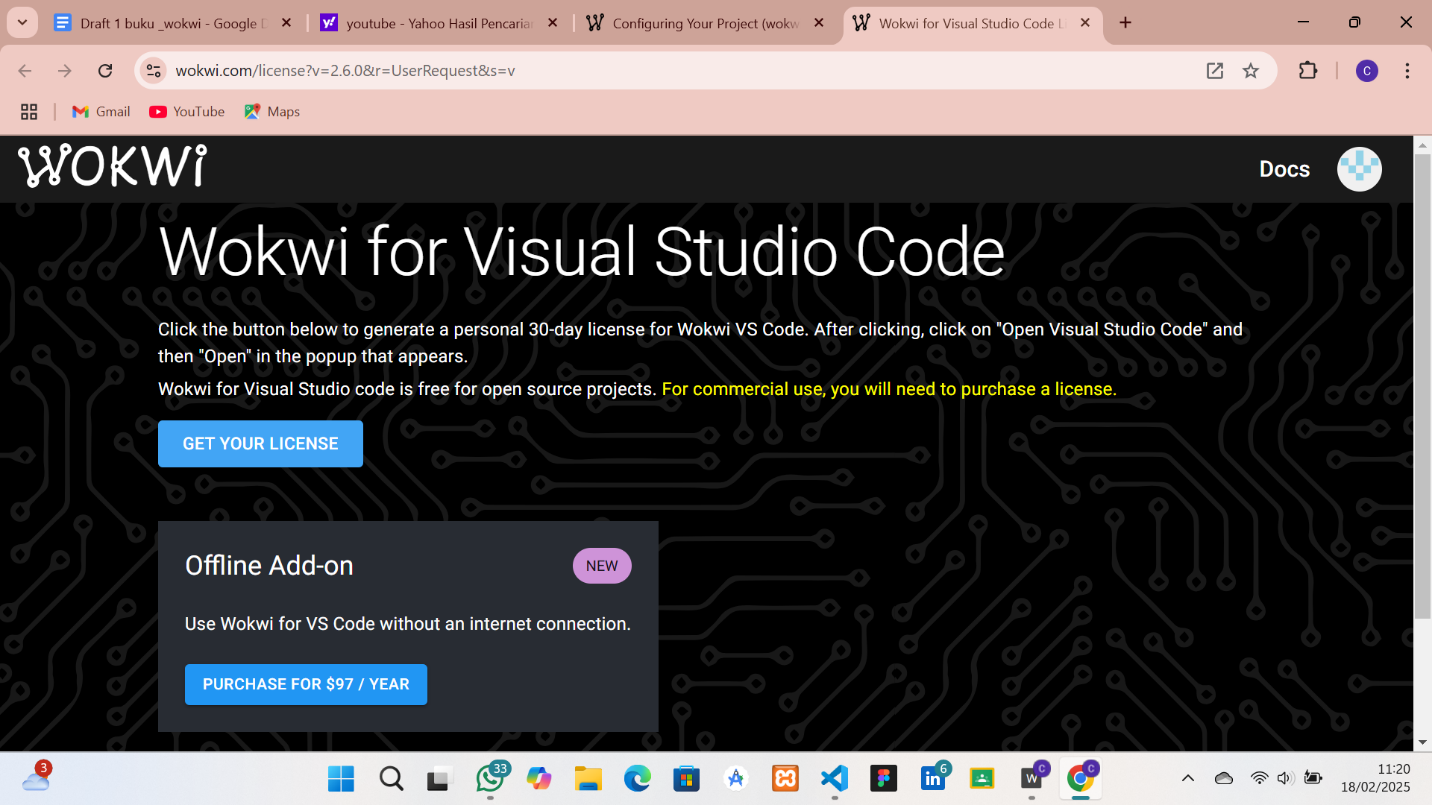
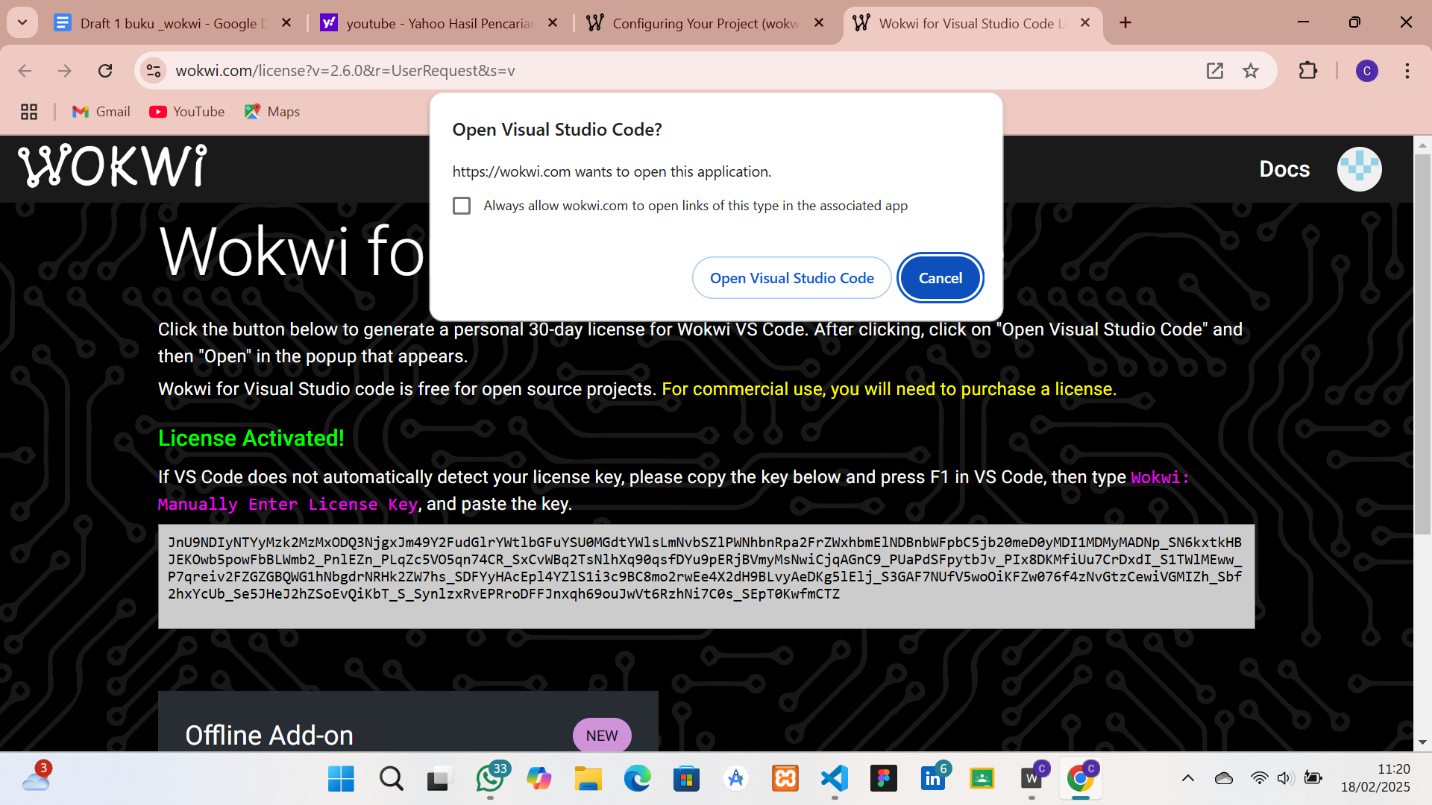
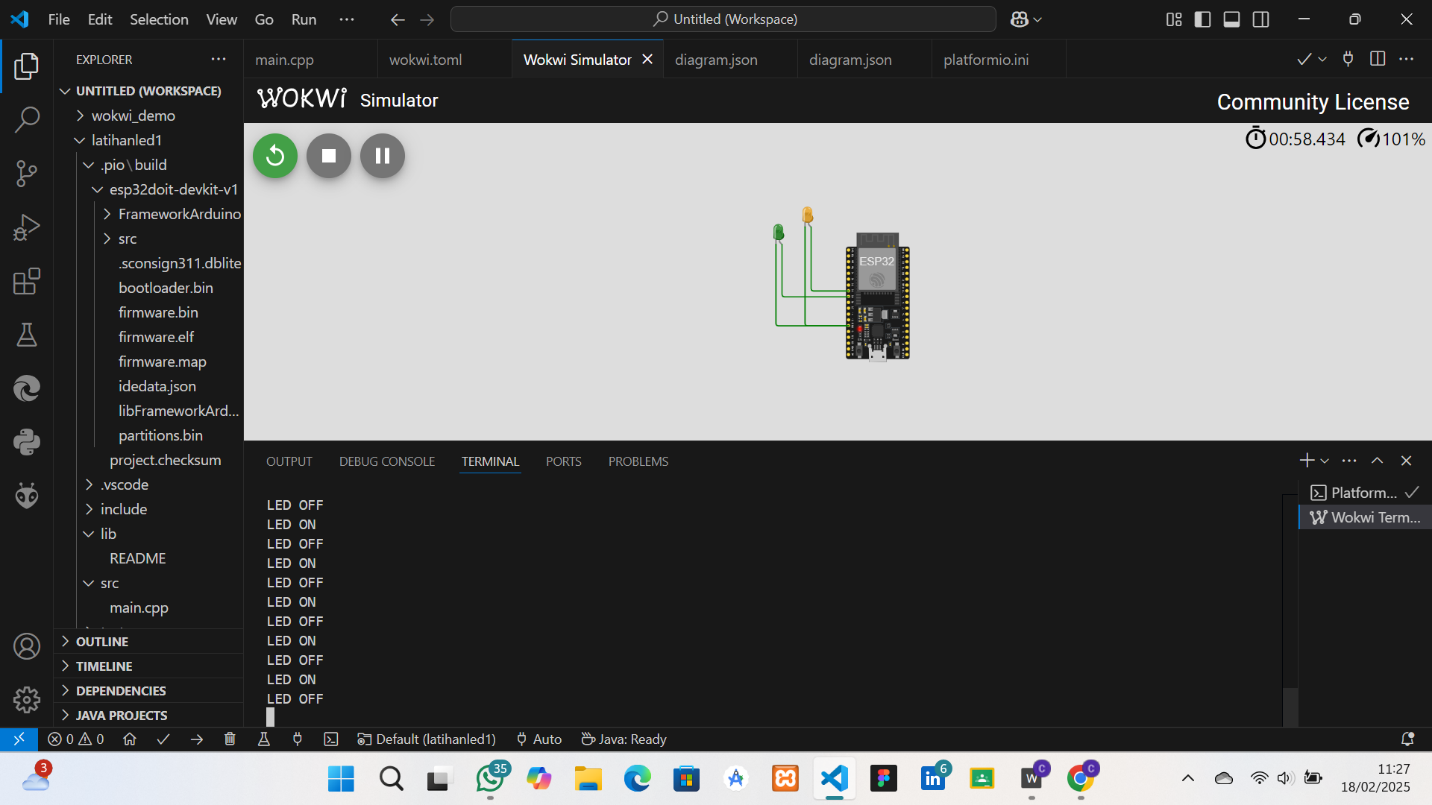
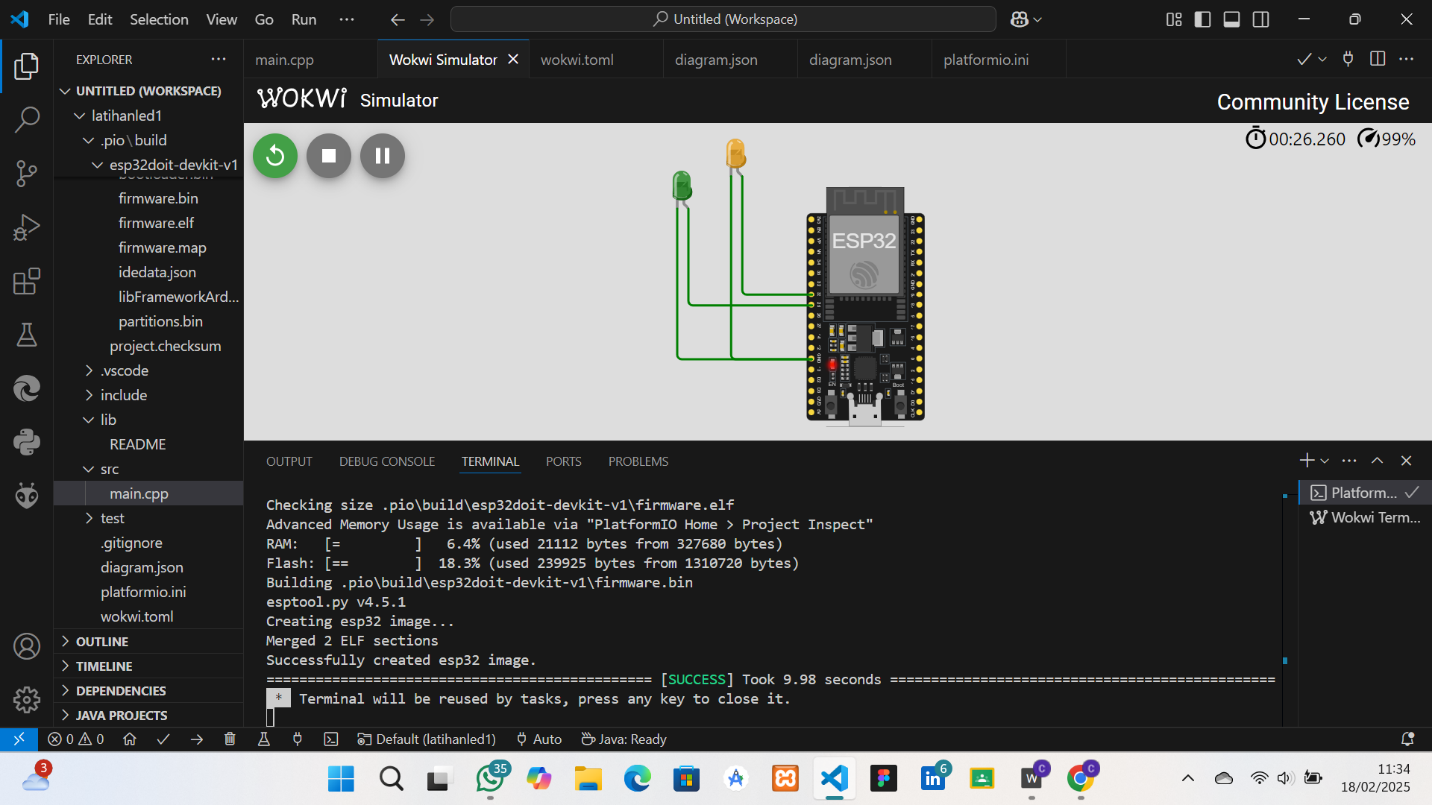
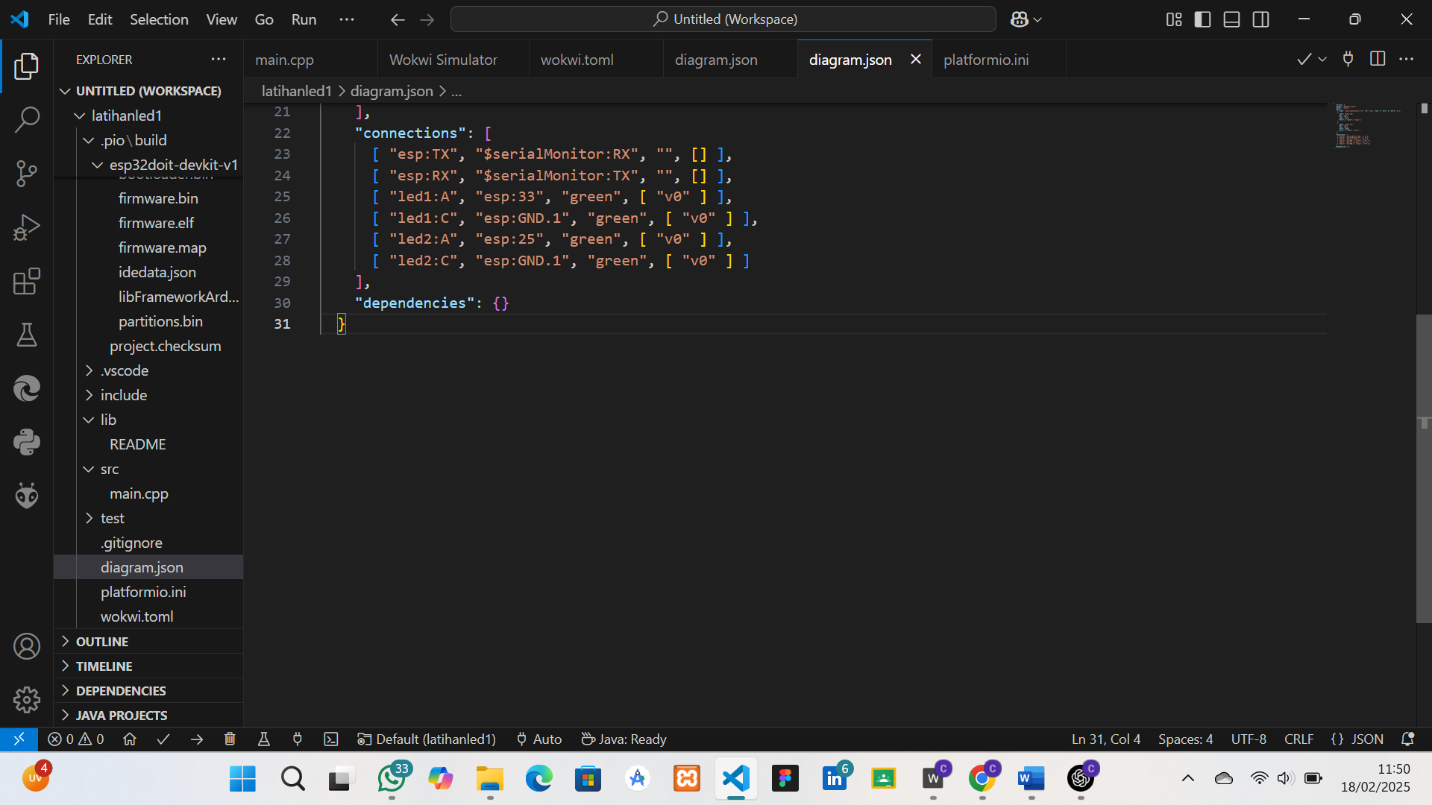
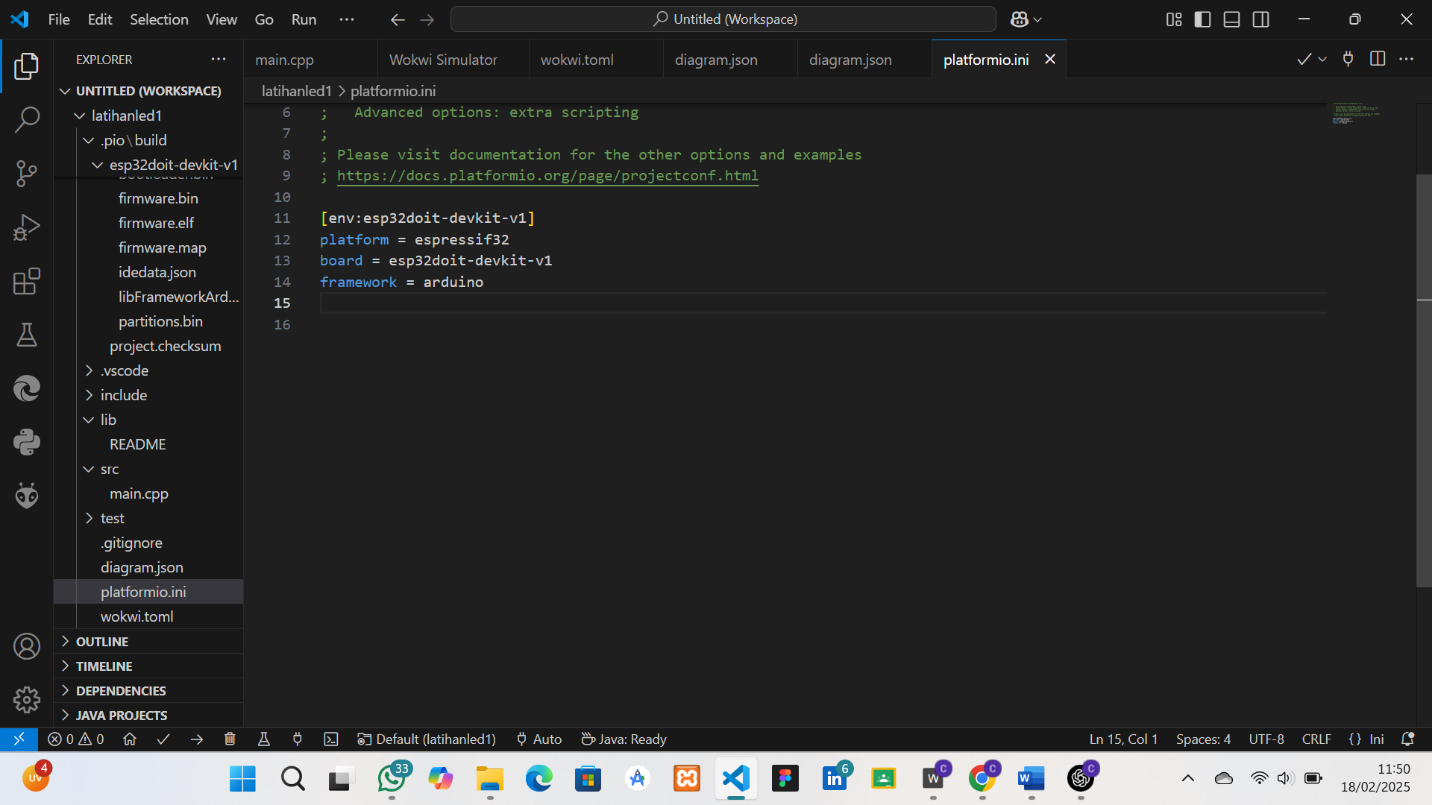
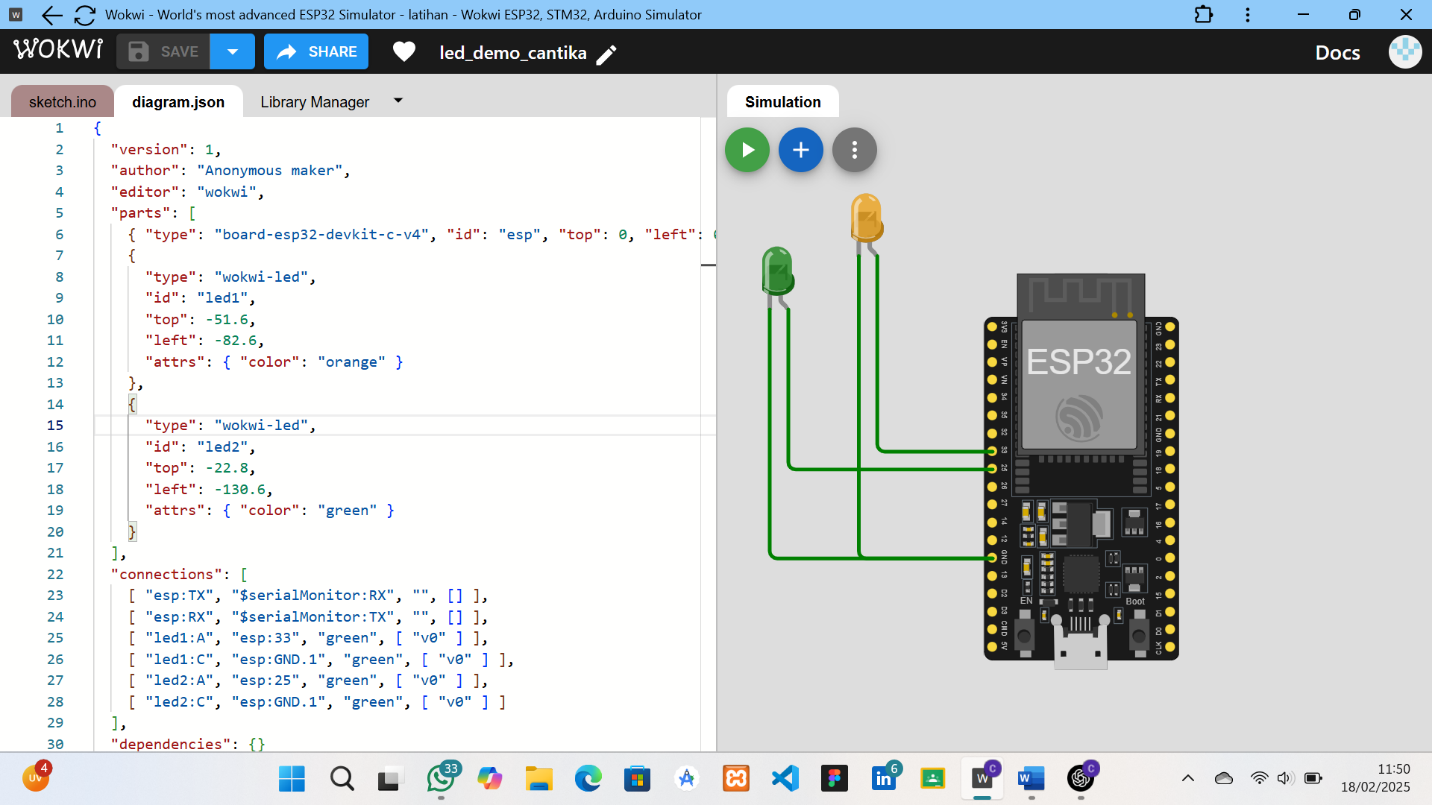
    digitalWrite(lampu2, LOW);

    Serial.println("LED OFF");

    delay(1000); // Tunggu 1 detik sebelum mengulang

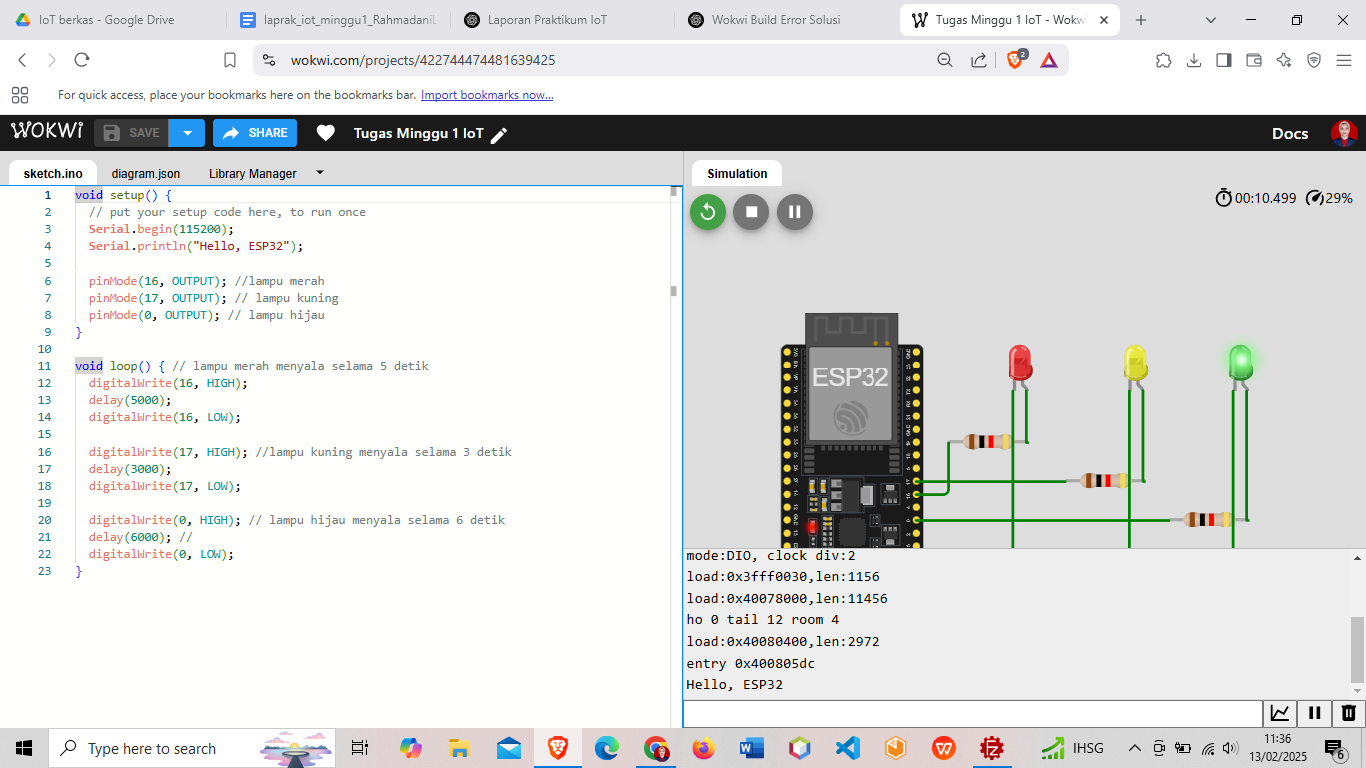
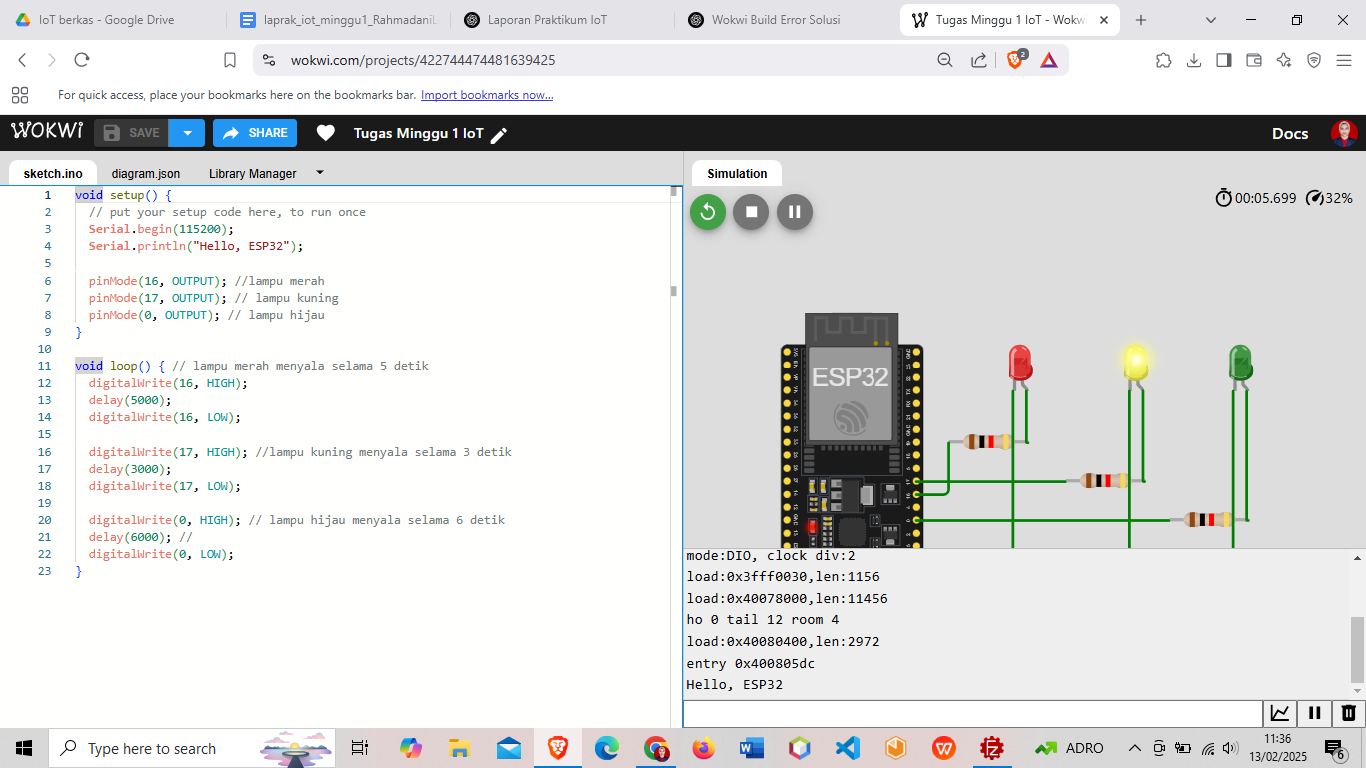
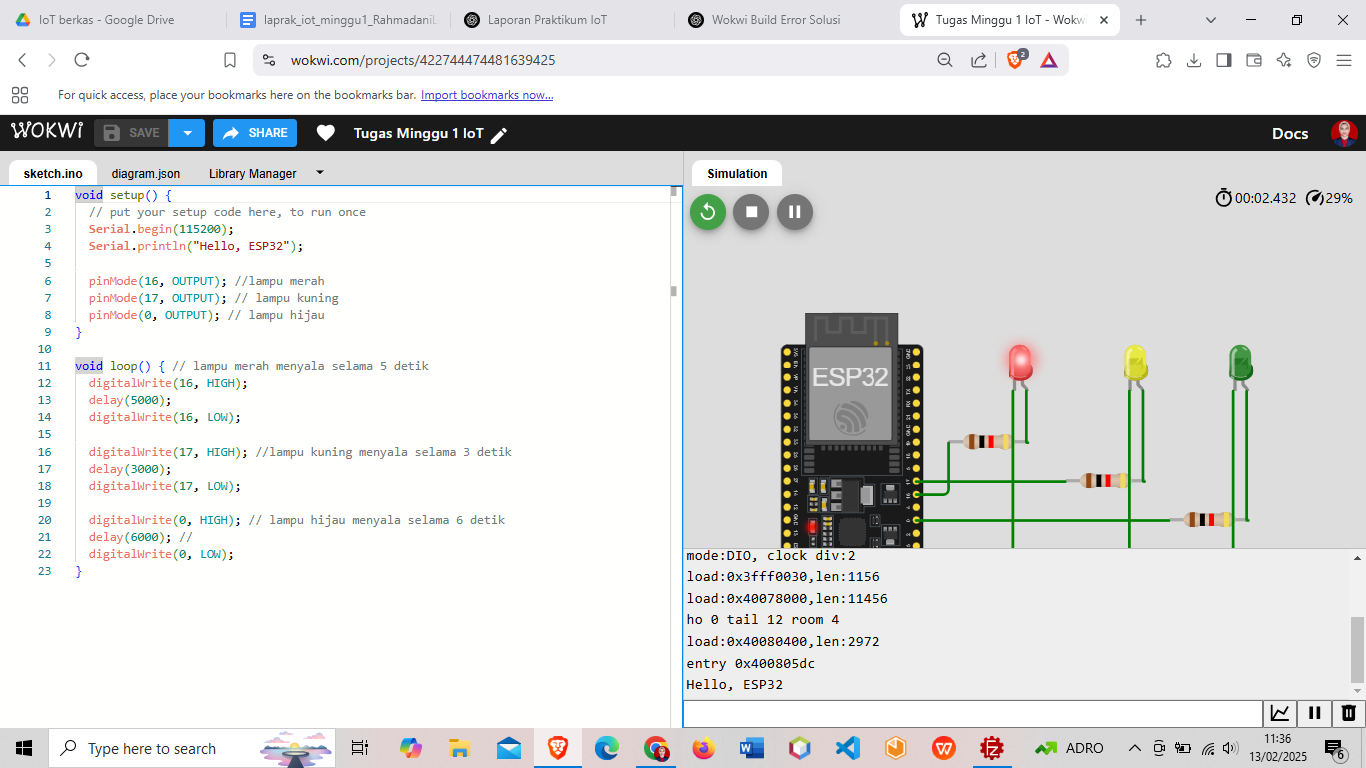
}





**Tugas 1**

Dari latihan 2, kembangkan untuk membuat rangkaian Lampu Lalu Lintas (Traffic Light), Terdiri dari 3 LED berwarna hijau, merah, dan kuning. Menyala secara bergantian.

**Screenshot hasil simulasi:**

### **Kode Program**

void setup() {

Serial.begin(115200);

Serial.println("Hello, ESP32");

pinMode(16, OUTPUT); // lampu merah

pinMode(17, OUTPUT); // lampu kuning

pinMode(0, OUTPUT); // lampu hijau

}

void loop() {

digitalWrite(16, HIGH);

delay(5000);

digitalWrite(16, LOW);

digitalWrite(17, HIGH);

delay(3000);

digitalWrite(17, LOW);

digitalWrite(0, HIGH);

delay(6000);

digitalWrite(0, LOW);

}