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

Fakultas Vokasi, Universitas Brawijaya

**Praktik Pembuatan Akun Wokwi dan GitHub serta Implementasi Rangkaian Lampu Lalu Lintas di Wokwi**



*Novita Lunar Pratiwi*

*Fakultas Vokasi, Universitas Brawijaya*

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

**Abstract** (Abstrak)

Praktikum ini bertujuan untuk memperkenalkan Wokwi sebagai simulator proyek IoT serta pemanfaatan GitHub dalam pengelolaan kode dan kerja sama tim. Dalam eksperimen ini, dilakukan pembuatan akun pada kedua platform tersebut serta implementasi sistem lampu lalu lintas menggunakan Wokwi. Hasil yang diperoleh menunjukkan bahwa Wokwi mampu mensimulasikan proyek IoT dengan praktis tanpa memerlukan perangkat keras nyata. Selain itu, penggunaan GitHub memberikan kemudahan dalam mengelola proyek secara lebih terorganisir dan mendukung kolaborasi antar pengguna.

*Keywords— Internet of Things, Wokwi, GitHub, Traffic Light, Simulation*

**1. Introduction** (Pendahuluan)

**1.1 Latar belakang** praktikum IoT yang dilakukan

Dalam era digital saat ini, **Internet of Things (IoT)** menjadi salah satu teknologi yang semakin berkembang pesat. IoT memungkinkan berbagai perangkat fisik terhubung ke internet, sehingga dapat beroperasi secara otomatis dan lebih efisien. Dalam proses pembelajaran IoT, penggunaan simulator seperti **Wokwi** menjadi solusi yang efektif bagi mahasiswa untuk memahami konsep dasar tanpa harus memiliki perangkat keras secara langsung. Dengan Wokwi, berbagai proyek berbasis IoT dapat diuji dan dijalankan secara virtual.

Selain itu, **GitHub** berperan penting dalam manajemen kode dan kolaborasi tim. Melalui platform ini, mahasiswa dapat menyimpan, berbagi, dan mengelola proyek secara lebih terstruktur. Dengan kombinasi Wokwi dan GitHub, proses pembelajaran IoT menjadi lebih fleksibel, efisien, dan dapat diakses oleh siapa saja tanpa keterbatasan perangkat fisik.

**1.2 Tujuan eksperimen**

⦁ Membuat akun Wokwi dan GitHub.

⦁ Mengimplementasikan sistem lampu lalu lintas di Wokwi.

⦁ Mengunggah dan mengelola kode program melalui GitHub.

**2. Methodology (Metodologi)**

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

⦁ Mikrokontroler: ESP32 (simulasi di Wokwi)

⦁ Komponen elektronik: LED merah, kuning, hijau, resistor 220Ω

⦁ Software: Wokwi (https://wokwi.com), GitHub (https://github.com)

**2.2 Implementation Steps (Langkah Implementasi)**

**⦁Pembuatan Akun Wokwi dan GitHub**

Mendaftar di platform Wokwi dan GitHub.

**⦁Membangun Rangkaian Lampu Lalu Lintas di Wokwi**

Menyusun skematik rangkaian LED untuk lampu merah, kuning, dan hijau.

Menulis kode program ESP32 untuk mengontrol lampu lalu lintas.

**⦁Mengunggah Kode ke GitHub**

Membuat repository baru di GitHub.

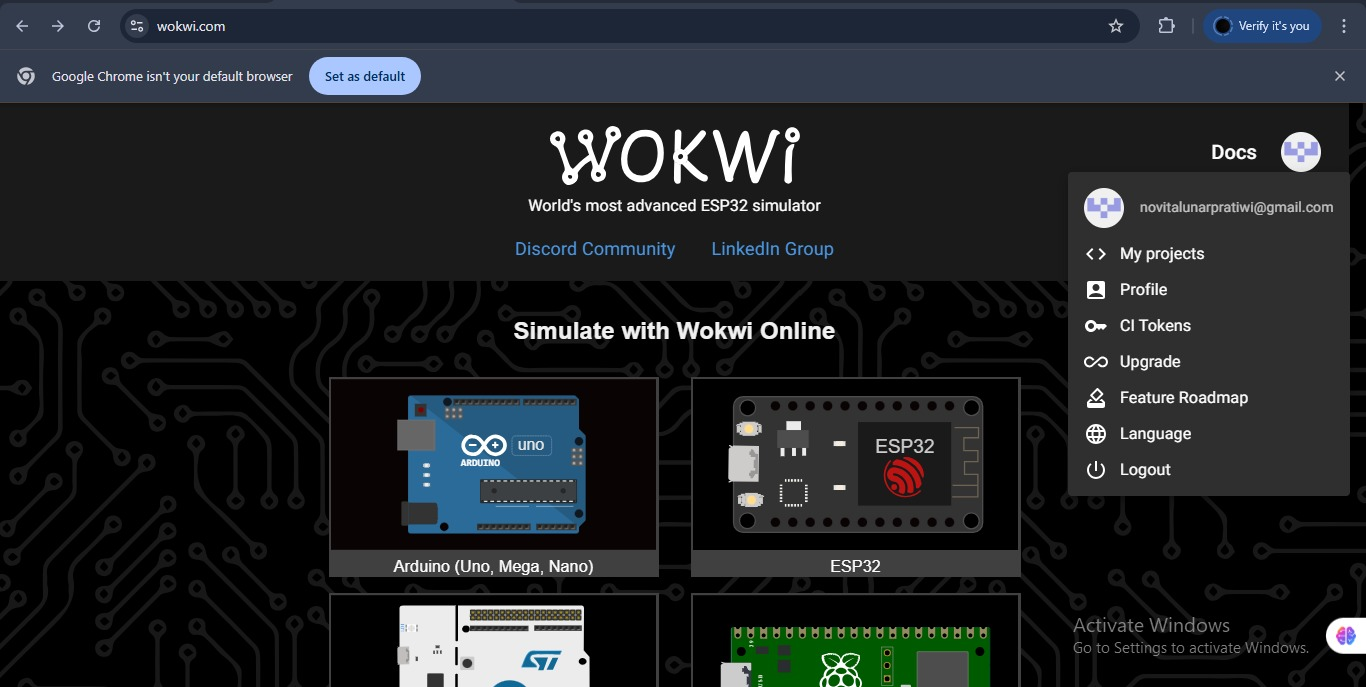
Mengunggah file kode program dan dokumentasi.

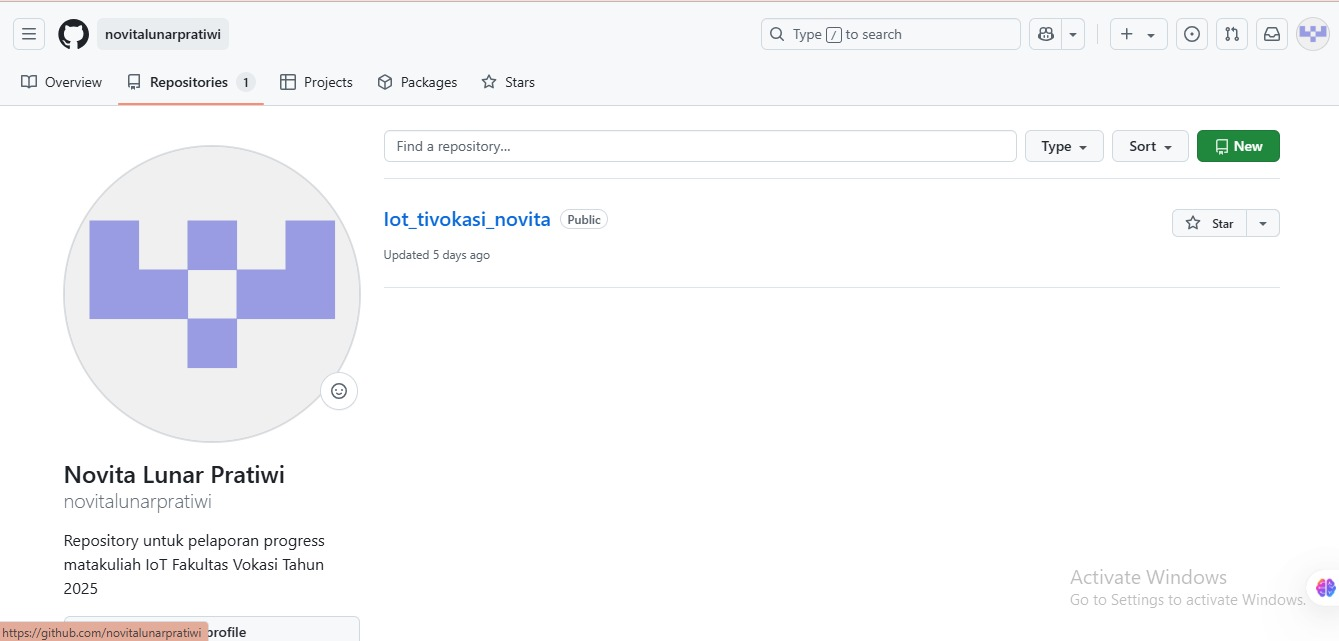
**3. Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

⦁Berhasil membuat akun Wokwi dan GitHub.

**Screenshot hasil simulasi:**

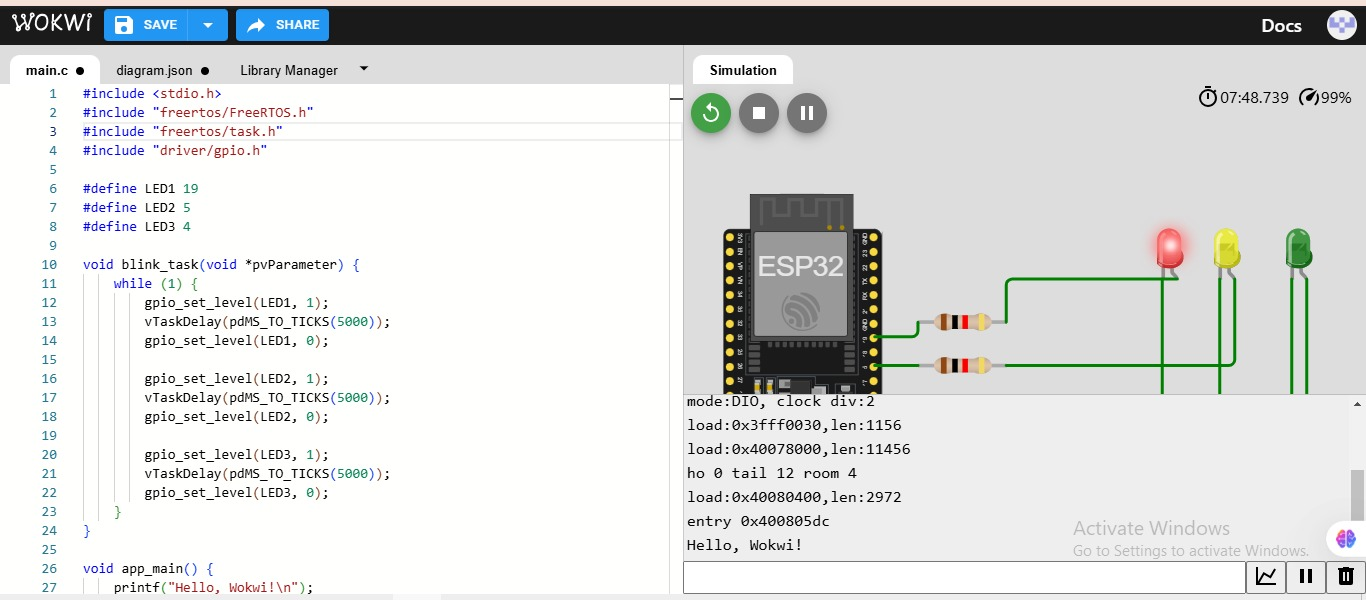


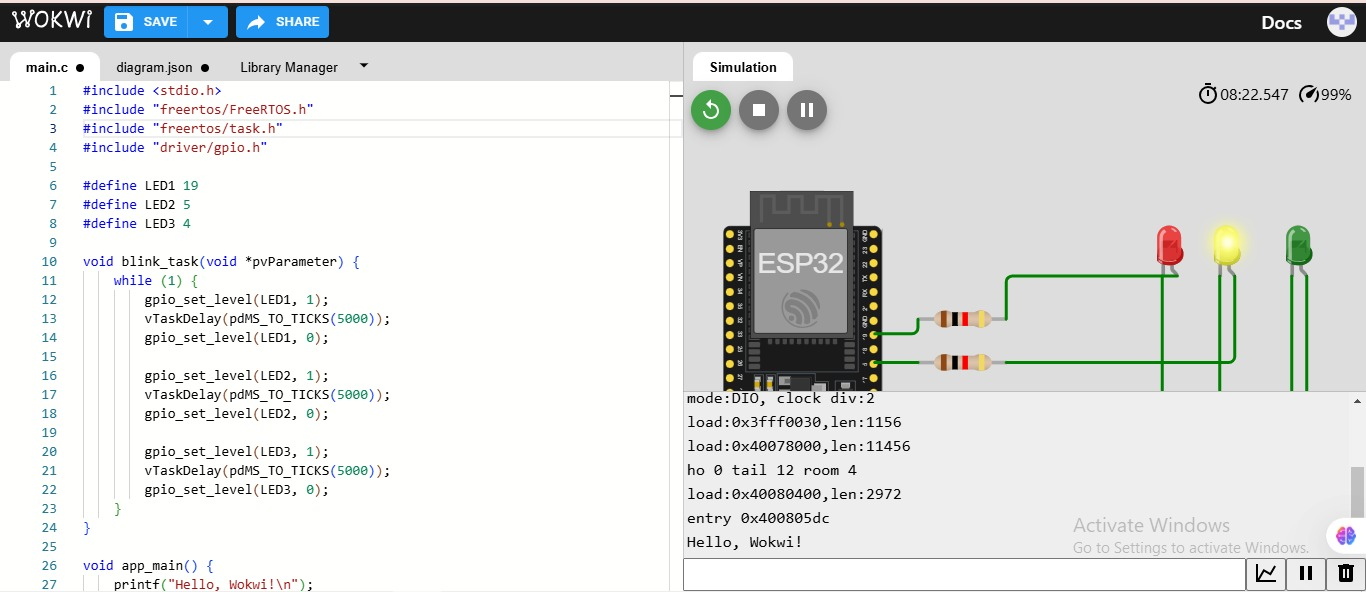


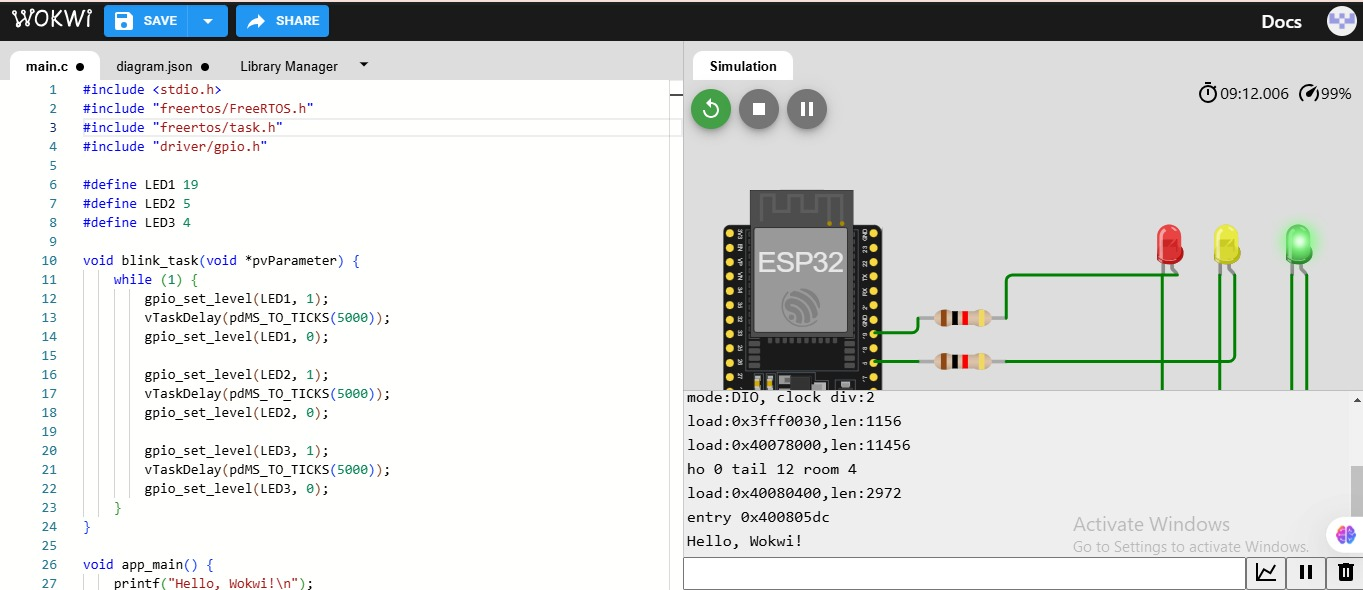
⦁Simulasi sistem lampu lalu lintas berjalan dengan baik di Wokwi.

⦁Kode program berhasil diunggah ke GitHub.

**Screenshot hasil simulasi:**







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

**4.1 Kode program**

#include <stdio.h>

#include "freertos/FreeRTOS.h"

#include "freertos/task.h"

#include "driver/gpio.h"

#define LED1 19

#define LED2 5

#define LED3 4

void blink\_task(void \*pvParameter) {

while (1) {

gpio\_set\_level(LED1, 1);

vTaskDelay(pdMS\_TO\_TICKS(5000));

gpio\_set\_level(LED1, 0);

gpio\_set\_level(LED2, 1);

vTaskDelay(pdMS\_TO\_TICKS(5000));

gpio\_set\_level(LED2, 0);

gpio\_set\_level(LED3, 1);

vTaskDelay(pdMS\_TO\_TICKS(5000));

gpio\_set\_level(LED3, 0);

}

}

void app\_main() {

printf("Hello, Wokwi!\n");

// Konfigurasi GPIO sebagai output

gpio\_reset\_pin(LED1);

gpio\_set\_direction(LED1, GPIO\_MODE\_OUTPUT);

gpio\_reset\_pin(LED2);

gpio\_set\_direction(LED2, GPIO\_MODE\_OUTPUT);

gpio\_reset\_pin(LED3);

gpio\_set\_direction(LED3, GPIO\_MODE\_OUTPUT);

// Membuat task untuk blinking LED

xTaskCreate(&blink\_task, "blink\_task", 2048, NULL, 5, NULL);

}