**Laporan minggu 2 praktikum wokwi**

*Gunawan Danyarto*

*University of Brawijaya Vocation Faculty*

*Email : [gunawandany11@gmail.com](mailto:gunawandany11@g,ail.com)*

**Abstract**

This experiment aims to analyze the implementation of the MQTT protocol in a smart home IoT system. The system consists of temperature sensors and actuators controlled via an ESP8266 microcontroller. The results show that MQTT-based communication is more efficient in terms of latency and power consumption compared to HTTP-based communication. This practice typically involves learning how to use Wokwi (an online IoT and electronics simulator) and GitHub (a platform for version control and collaboration) for IoT projects. Below is an example structure for such a topic:

**1. Introduction**

* 1. **Background of the IoT Experiment**

Wokwi is an online platform for simulating IoT and electronics projects, while GitHub is a widely used platform for version control and collaborative coding. Combining these tools allows developers to design, simulate, and share IoT projects efficiently.

* 1. **Objective of the Experiment**

The goal of this practice is to learn how to create and simulate IoT projects using Wokwi and manage the project code using GitHub.

**2. Methodology (Metodologi)**

**2.1 Tools & Materials**

* Microcontrollers: ESP8266,
* Arduino, Raspberry Pi, etc.
* Sensors: DHT11, PIR, etc.
* Software: Arduino IDE, MQTT Broker, etc.

**2.2 Implementation Steps**

2.1 Tools & Materials

* Wokwi (https://wokwi.com)
* GitHub (https://github.com)
* ESP8266 or other microcontrollers (simulated in Wokwi)
* Sensors and actuators (simulated in Wokwi)

**2.2 Implementation Steps**

**1. Creating a Wokwi Project :**

* Open Wokwi and select a microcontroller (e.g., ESP8266).
* Add components (e.g., sensors, LEDs) and write the code for the project. Simulate the project to ensure it works as expected.

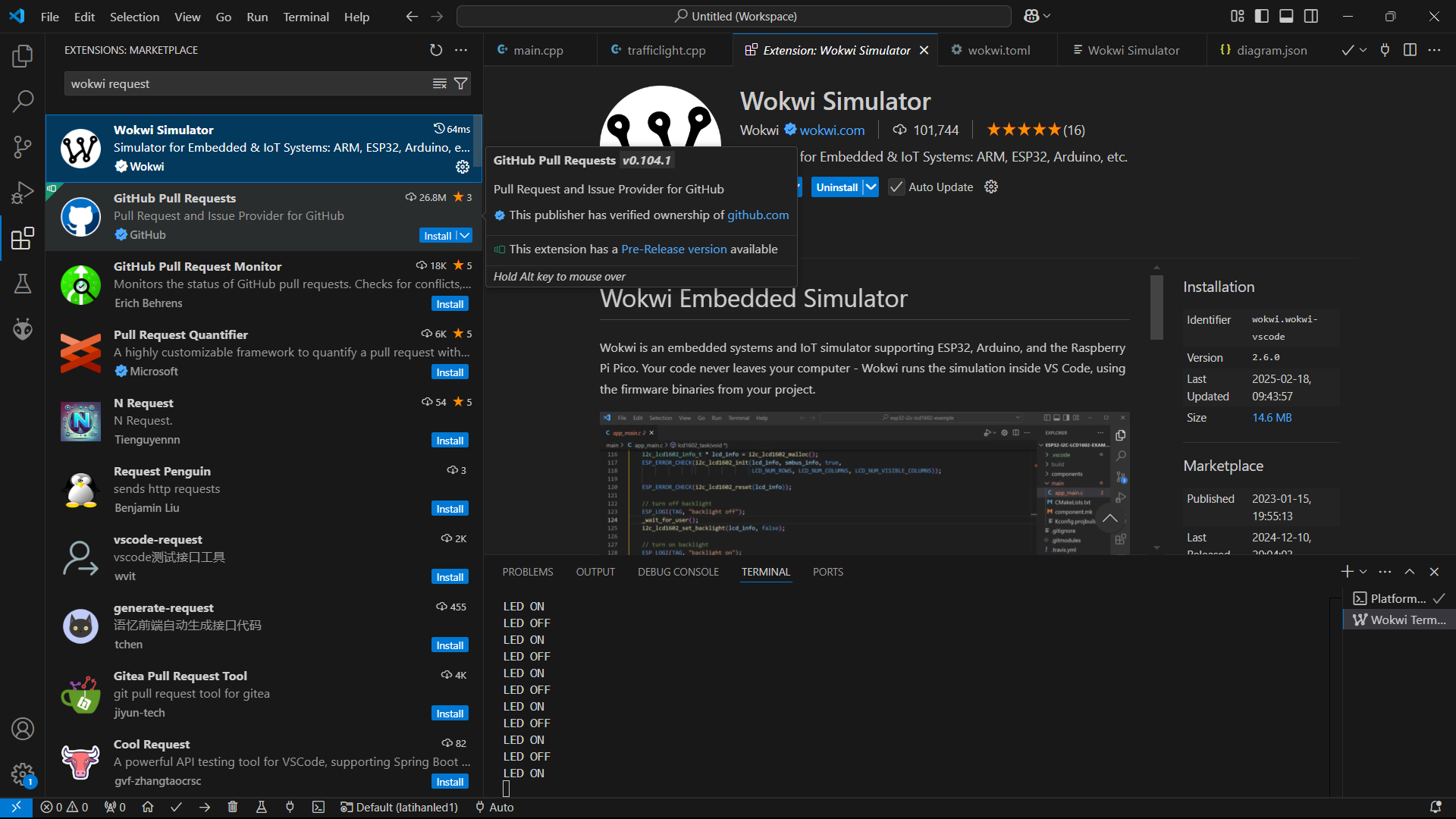
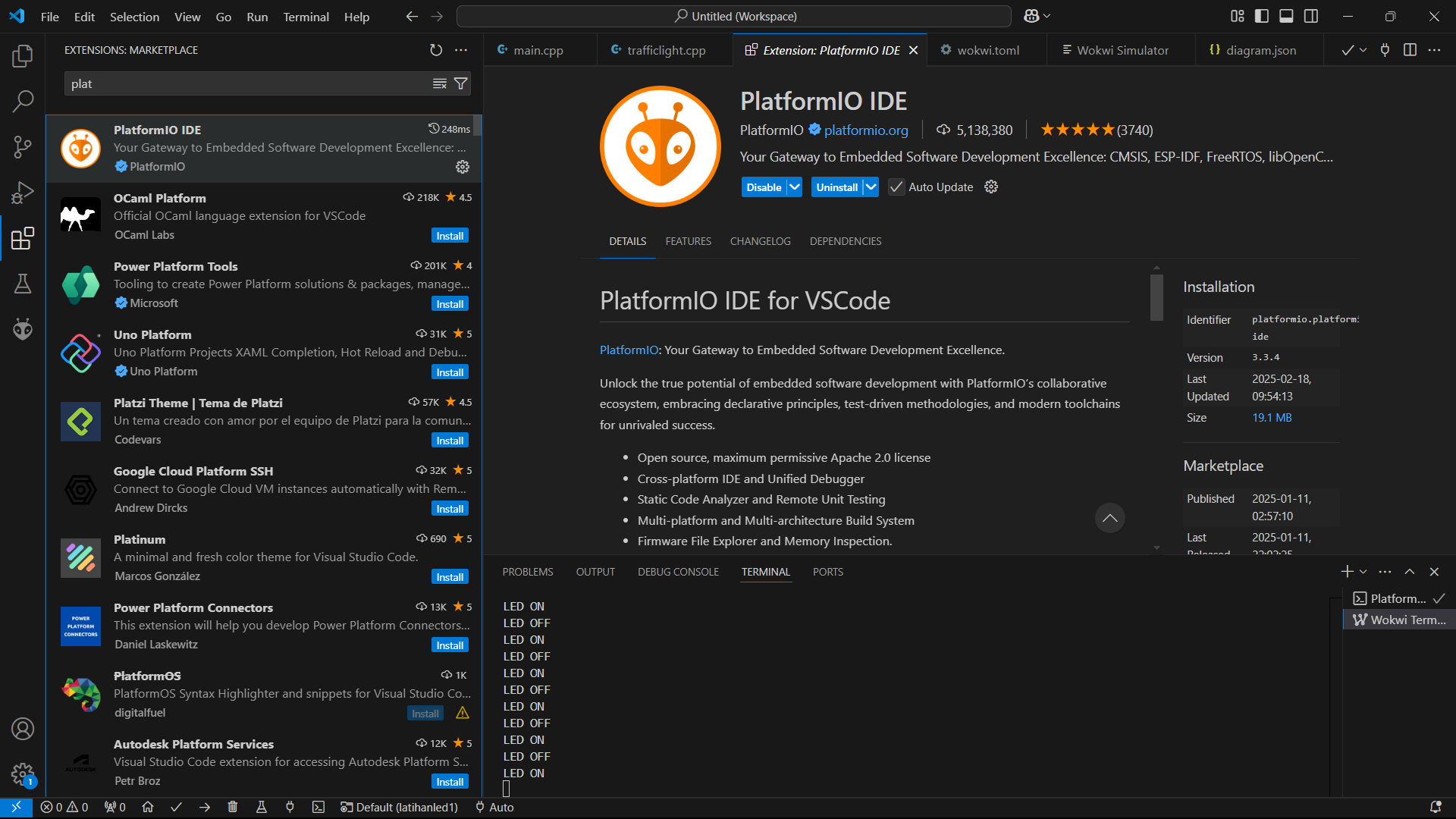
Simulate the project directly on the Wokwi platform to ensure it works correctly.

2.2 Using Wokwi Extension in VS Code

1. Install the Wokwi extension in Visual Studio Code (VS Code):

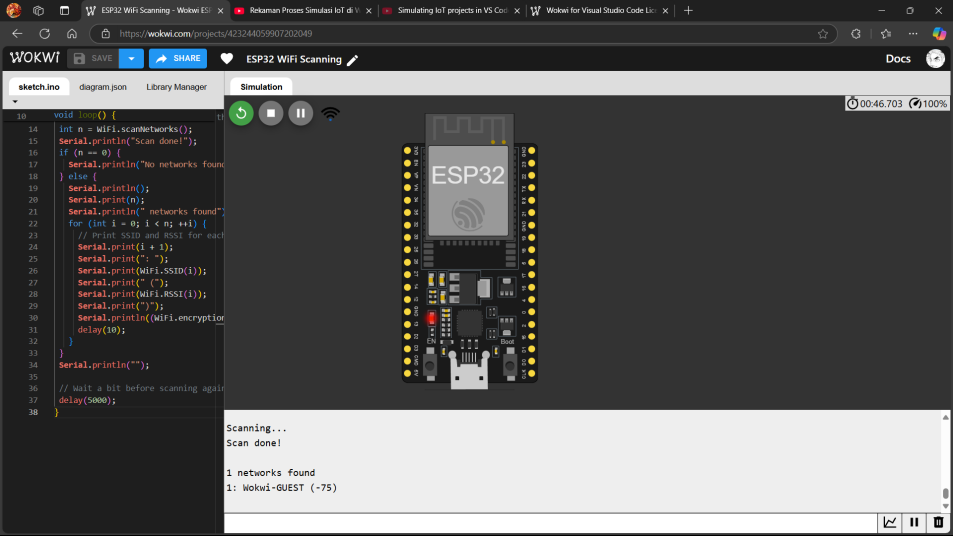
* Open VS Code and go to the Extensions Marketplace.
* Search for "Wokwi" and install the extension.

1. Create a new project or open an existing one in VS Code.
2. Use the Wokwi extension to configure your microcontroller (e.g., ESP32) and add components like LEDs.
3. Write and debug your code directly in VS Code.
4. Simulate the project using the Wokwi extension by clicking the "Start Simulation" button.
5. Observe the simulation results in the Wokwi visualizer within VS Code.



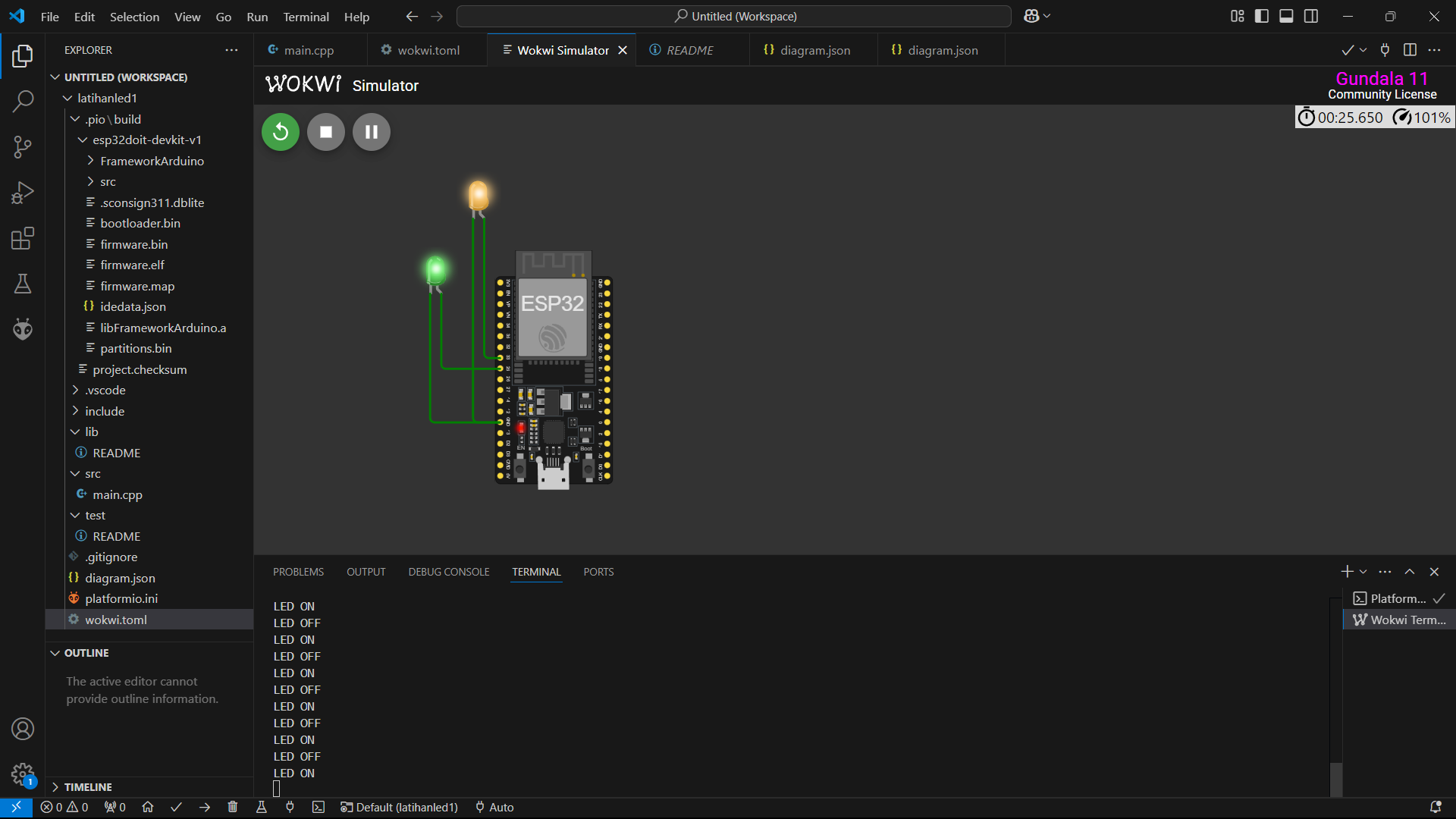
**2.3 Experiment 1: WiFi Scanning**

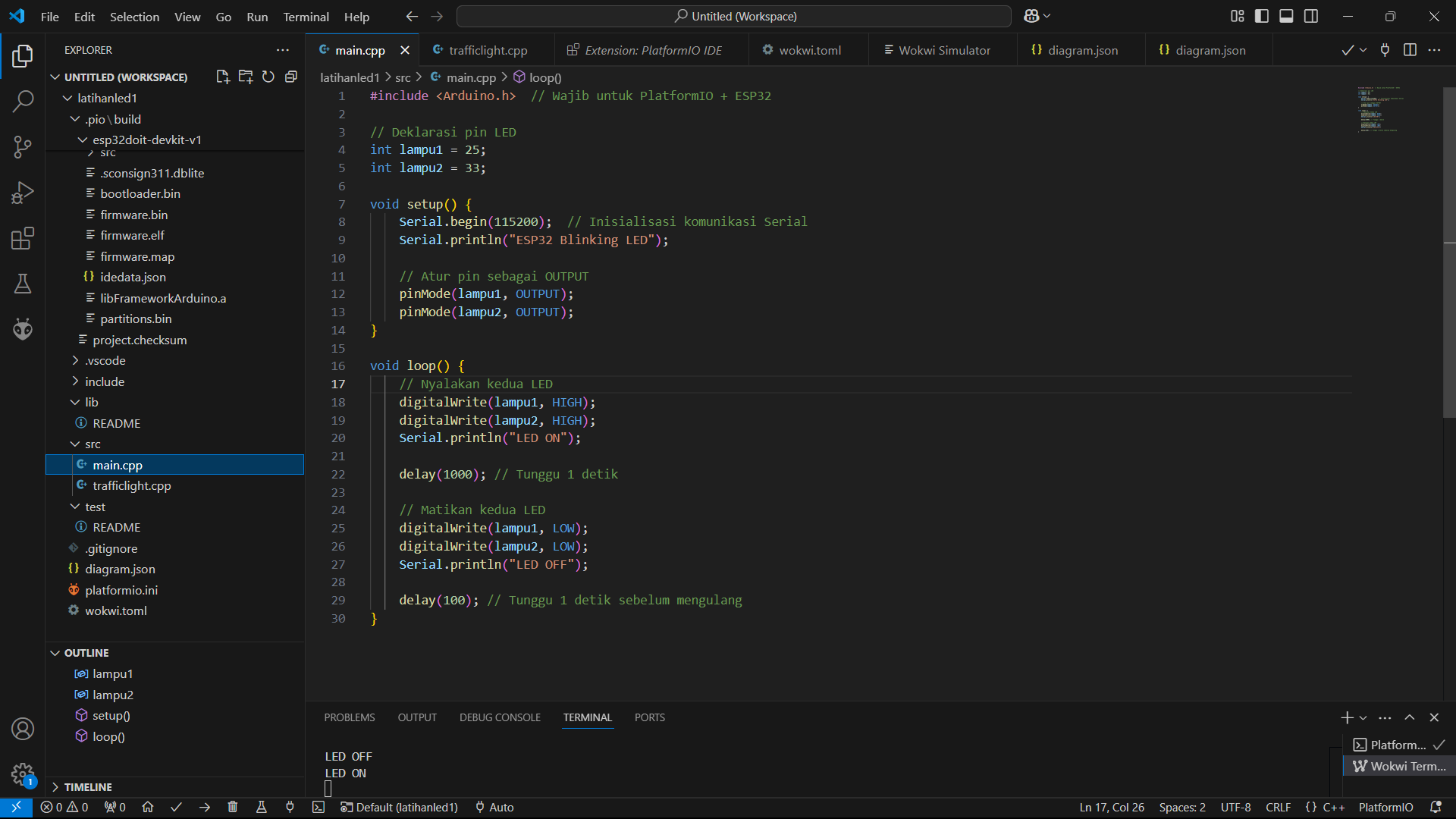
* Program the ESP32 to scan available WiFi networks.
* Display the scanned networks on the Serial Monitor.
* Simulate and observe the results using the Wokwi platform or VS Code extension.



2.2.4 Experiment 2: LED Control

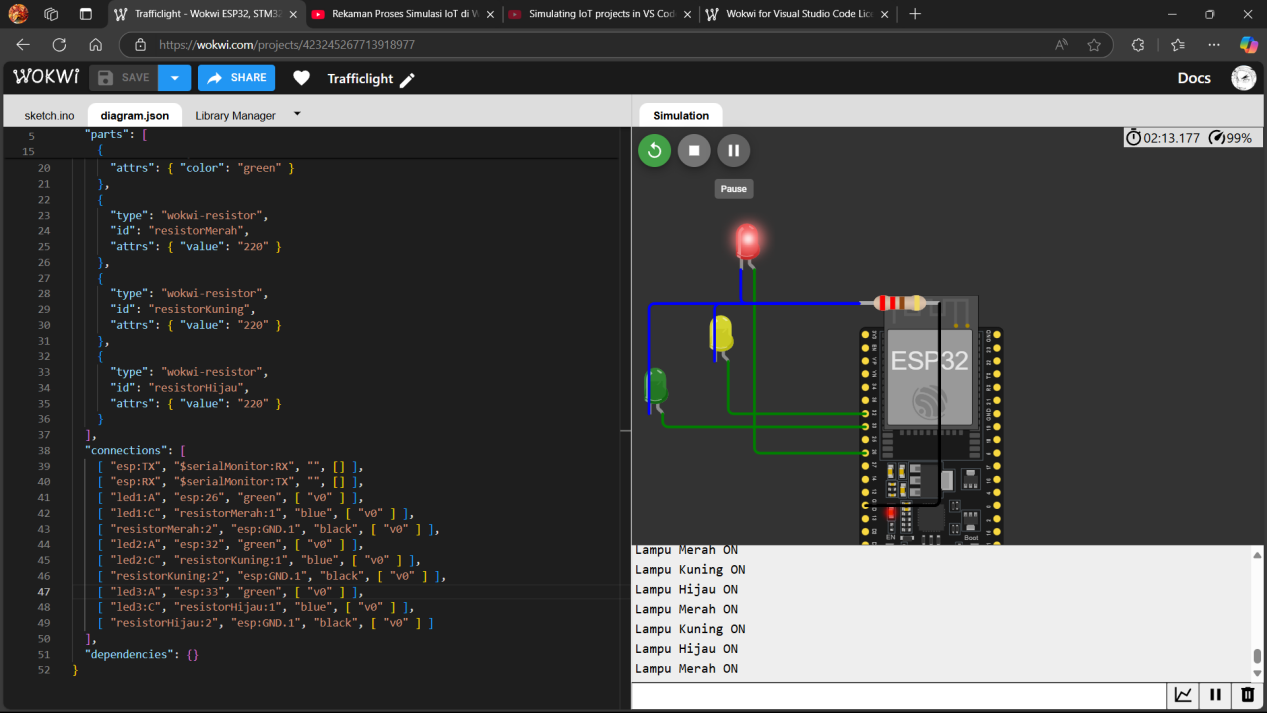
* Connect an LED to the ESP32 in the Wokwi simulator or VS Code extension.
* Write code to turn the LED on and off with a delay.
* Simulate the circuit and verify functionality.

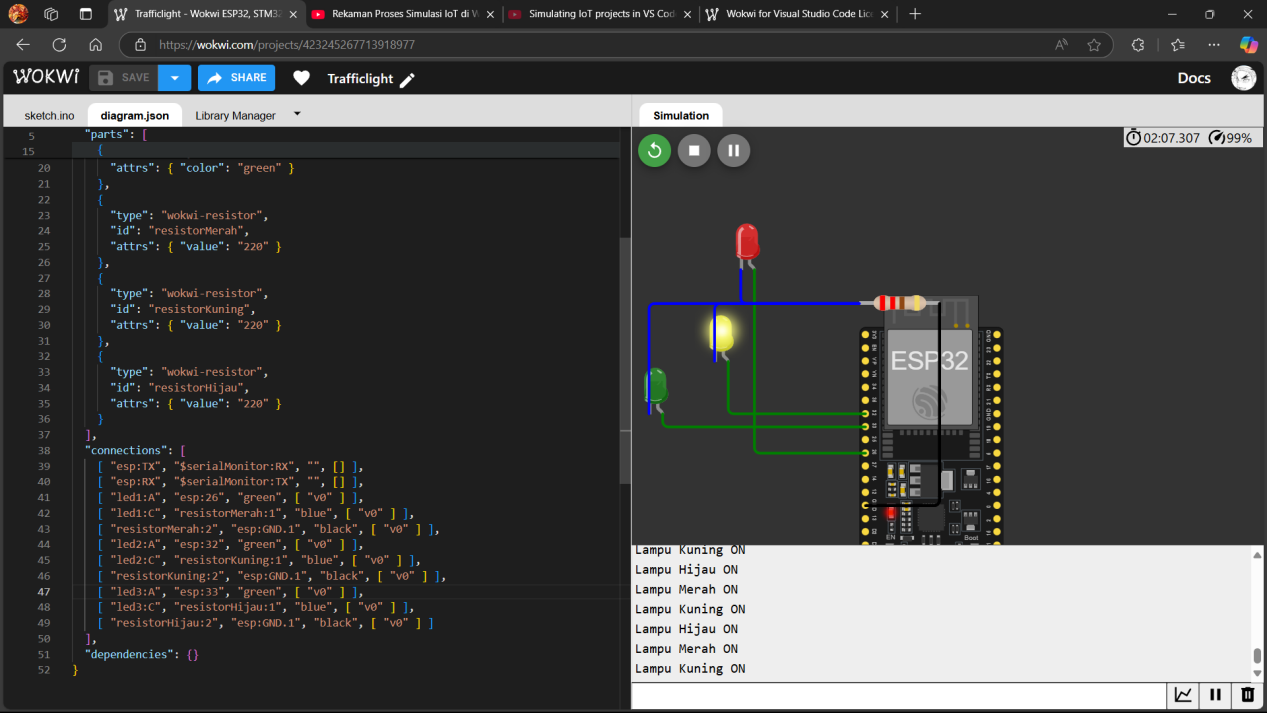


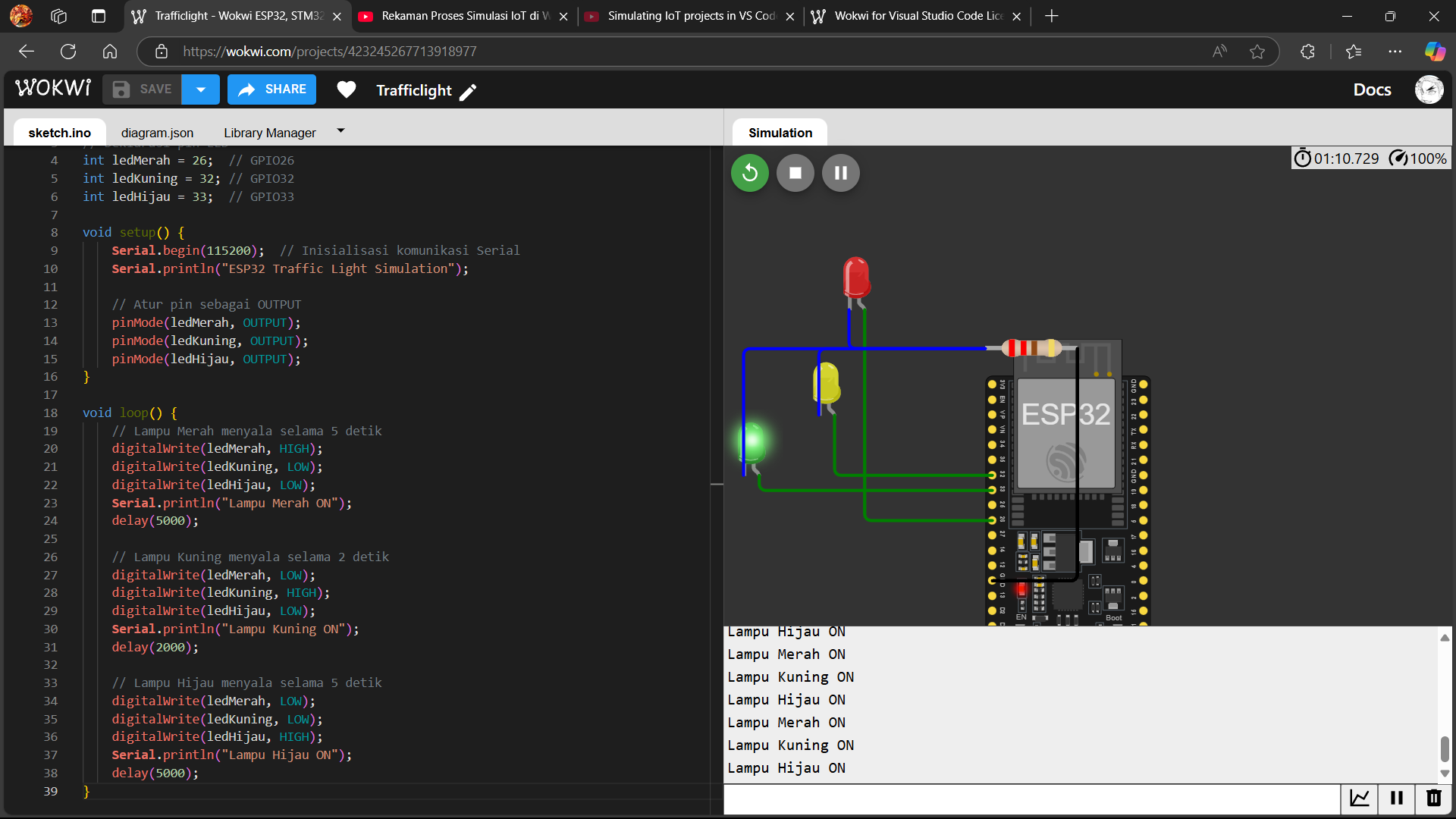


2.2.5 Experiment 3: Traffic Light Simulation

* Connect three LEDs (red, yellow, green) to the ESP32.
* Write a program to turn them on and off sequentially, mimicking a traffic light.
* Simulate and analyze timing behavior using Wokwi or the VS Code extension.







**3. Results and Discussion**

**3.1 Experimental Results**

WiFi Scanning: Successfully detected and listed available WiFi networks. LED Control: The LED blinked at the expected intervals. Traffic Light Simulation: LEDs changed states sequentially, simulating real traffic lights.Successful simulation of an IoT project in Wokwi.

**3.2 Discussion**

1. Benefits of using Wokwi for simulation and prototyping.
2. Advantages of using GitHub for version control and collaboration.