LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Pembuatan Traffic Lights pada Wokwi Simulator**

*Eunike Febianti*

*Fakultas Vokasi, Universitas Brawijaya*

[*eunikefebianti93@gmail.com*](mailto:eunikefebianti93@gmail.com)

**Abstract (Abstrak)**

Proyek ini bertujuan untuk mensimulasikan sistem lampu lalu lintas menggunakan Wokwi Simulator di Visual Studio Code (VS Code). Simulasi ini menggunakan mikrokontroler Arduino untuk mengontrol LED sebagai lampu merah, kuning, dan hijau, dengan logika waktu dan urutan yang sesuai dengan sistem lalu lintas nyata. Wokwi memungkinkan visualisasi real-time, memudahkan pengujian dan penyempurnaan logika sebelum implementasi pada perangkat fisik. Proyek ini menjadi dasar pembelajaran sistem tertanam dan manajemen lalu lintas, serta dapat dikembangkan lebih lanjut dengan sensor, algoritma adaptif, atau integrasi IoT untuk aplikasi kota pintar.

1. **Introduction (Pendahuluan)**

**1.1 Latar Belakang**

Lampu lalu lintas berperan penting dalam mengatur arus kendaraan dan pejalan kaki untuk menjaga keselamatan serta kelancaran lalu lintas. Simulasi sistem ini menjadi langkah awal dalam memahami pengaturan lalu lintas berbasis mikrokontroler.

Wokwi Simulator di VS Code memungkinkan simulasi tanpa perangkat fisik, memudahkan eksperimen dan pengujian logika lampu lalu lintas menggunakan Arduino. Selain itu, fitur interaktif seperti tombol penyeberangan pejalan kaki dapat ditambahkan. Proyek ini dapat dikembangkan lebih lanjut dengan sensor atau IoT untuk mendukung sistem transportasi cerdas.

**1.2 Tujuan Eksperimen**

1. Menyimulasikan Sistem Lampu Lalu Lintas

Mengembangkan simulasi lampu lalu lintas menggunakan Wokwi Simulator di Visual Studio Code (VS Code) untuk memahami cara kerja sistem pengaturan lalu lintas berbasis mikrokontroler.

1. Menguji Logika dan Urutan Lampu

Memprogram Arduino untuk mengontrol LED sebagai lampu merah, kuning, dan hijau dengan urutan dan waktu yang sesuai dengan sistem lalu lintas nyata.

1. Mengoptimalkan Konfigurasi Lampu Lalu Lintas

Melakukan pengujian dan penyempurnaan urutan serta durasi nyala lampu untuk meningkatkan efisiensi dan kesesuaian dengan kondisi lalu lintas nyata.

1. Mengembangkan Pemahaman tentang Sistem Tertanam

Memanfaatkan Wokwi sebagai alat pembelajaran dalam pemrograman mikrokontroler, khususnya dalam implementasi sistem kontrol lalu lintas berbasis Arduino.

1. Menjadi Dasar Pengembangan Lebih Lanjut

Memberikan landasan untuk pengembangan sistem lampu lalu lintas yang lebih canggih, seperti penggunaan sensor, algoritma adaptif, atau integrasi IoT untuk mendukung konsep kota pintar.

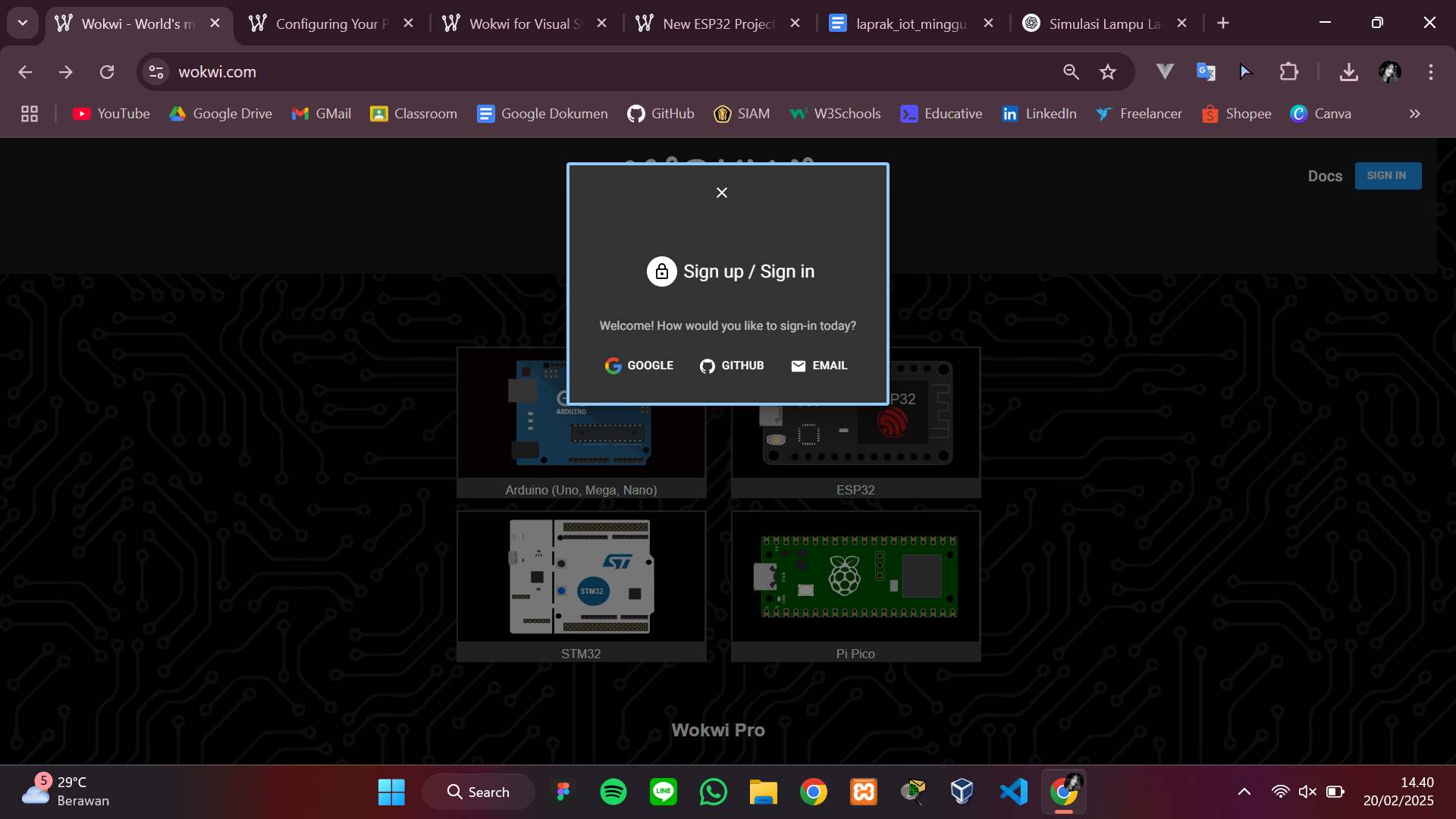
1. **Methodology (Metodologi)**

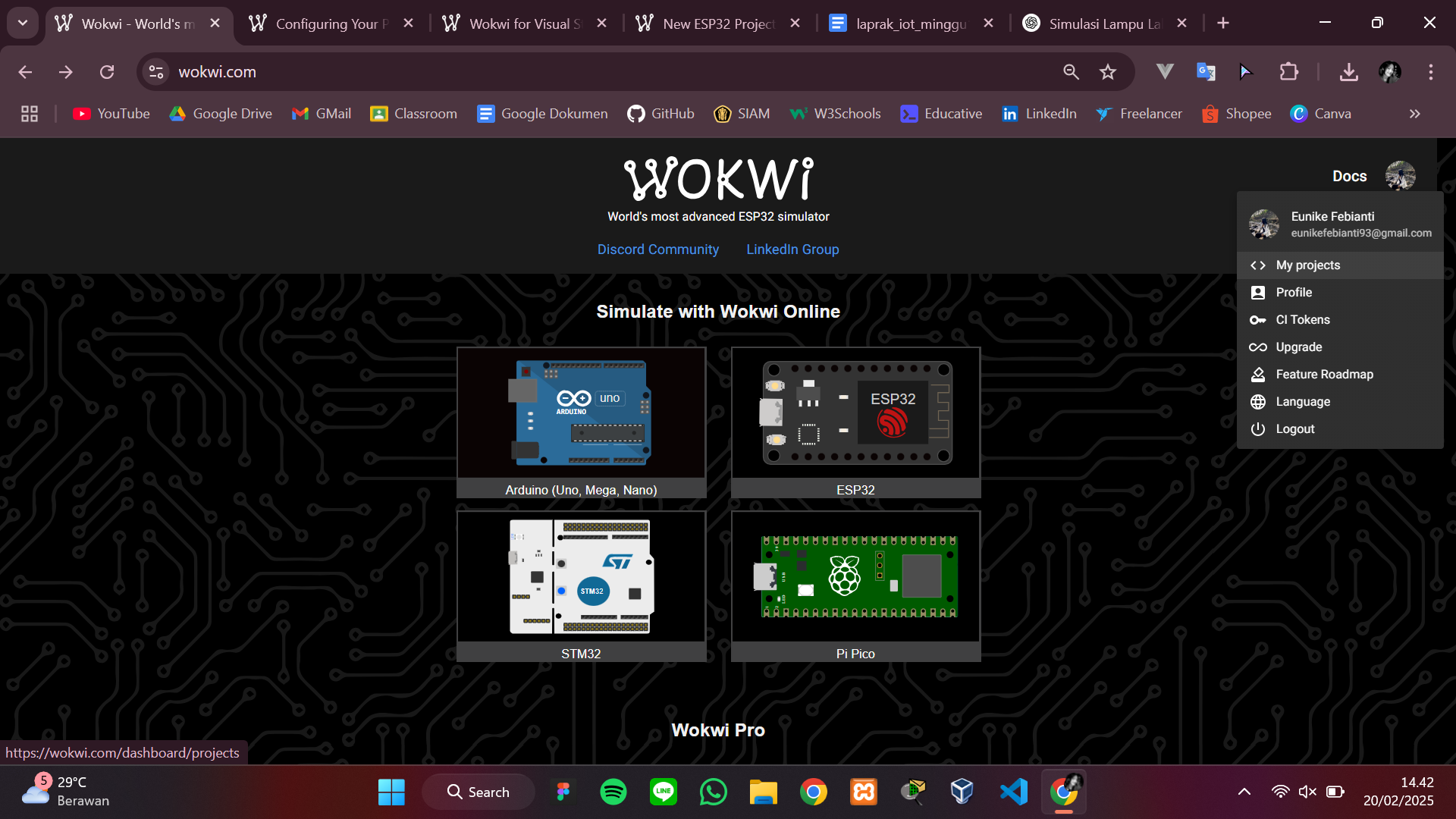
**2.1 Tools & Materials (Alat dan Bahan)**

1. Laptop
2. Internet
3. Visual Studio Code (VS Code)
4. Wokwi Simulator Extension
5. Arduino IDE (opsional untuk pengujian kode)

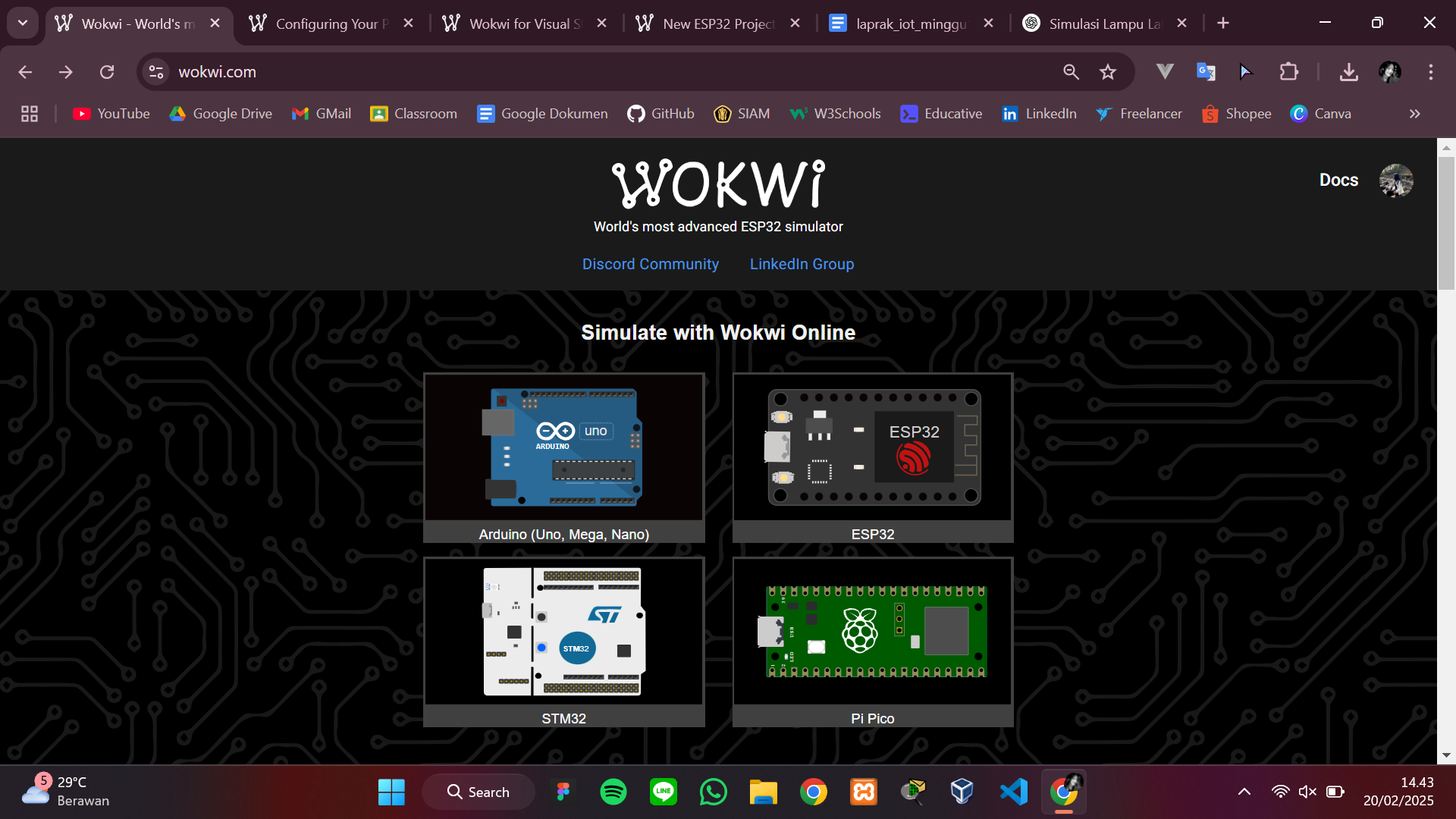
**2.2 Implementation Steps (Langkah Implementasi)**

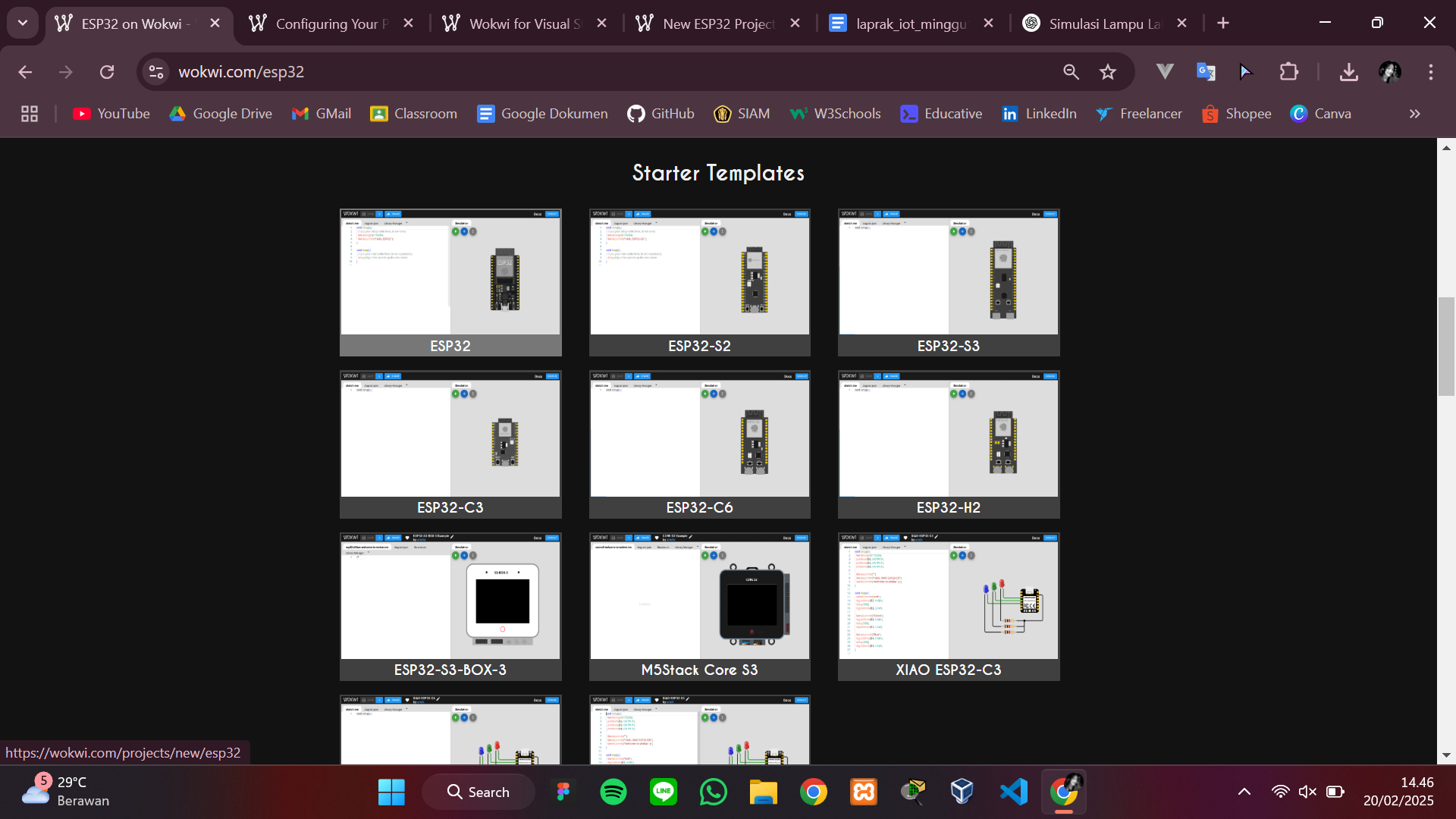
1. Membuat akun Wokwi (<https://wokwi.com/> ) atau Sign In menggunakan Github



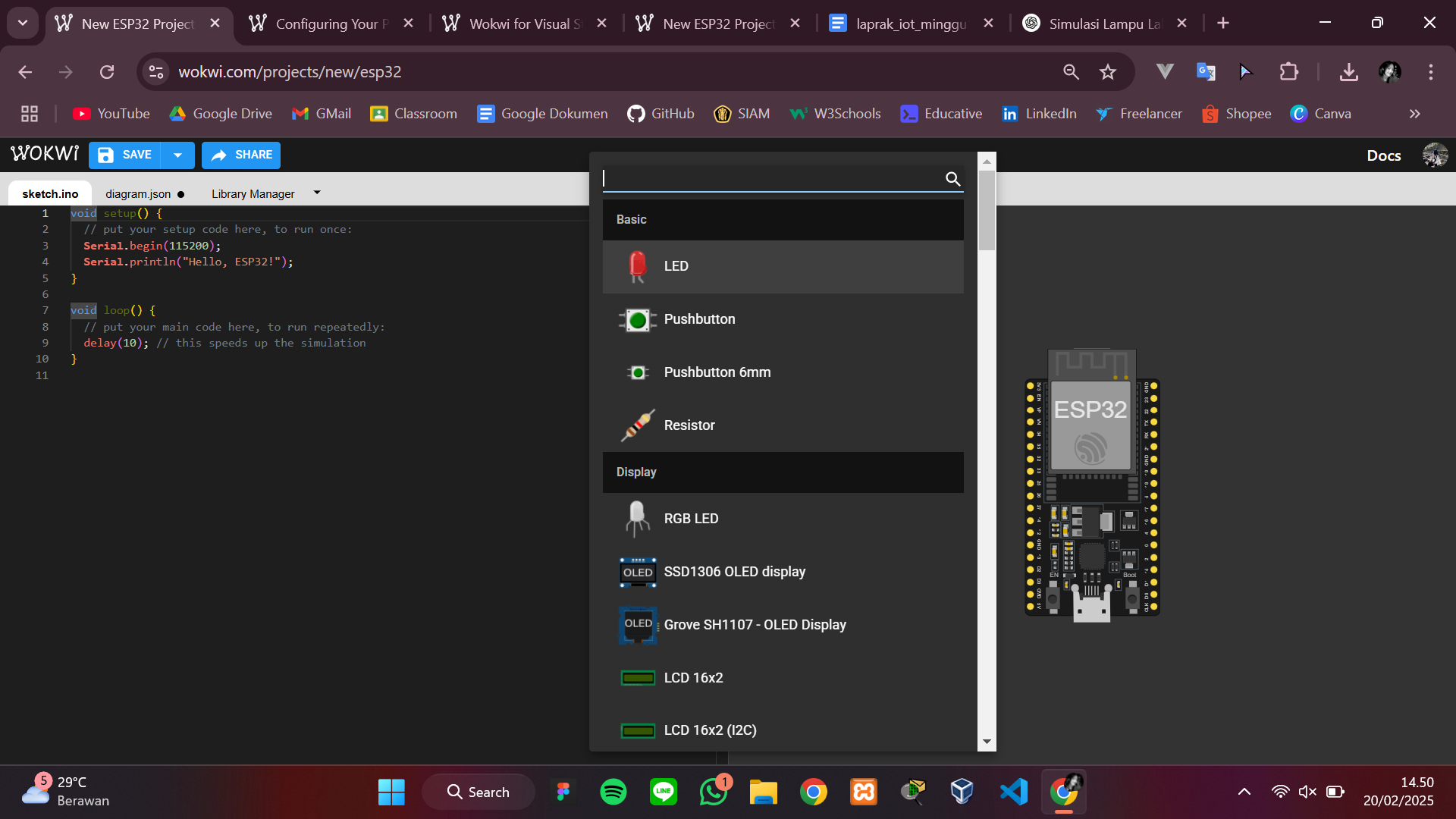


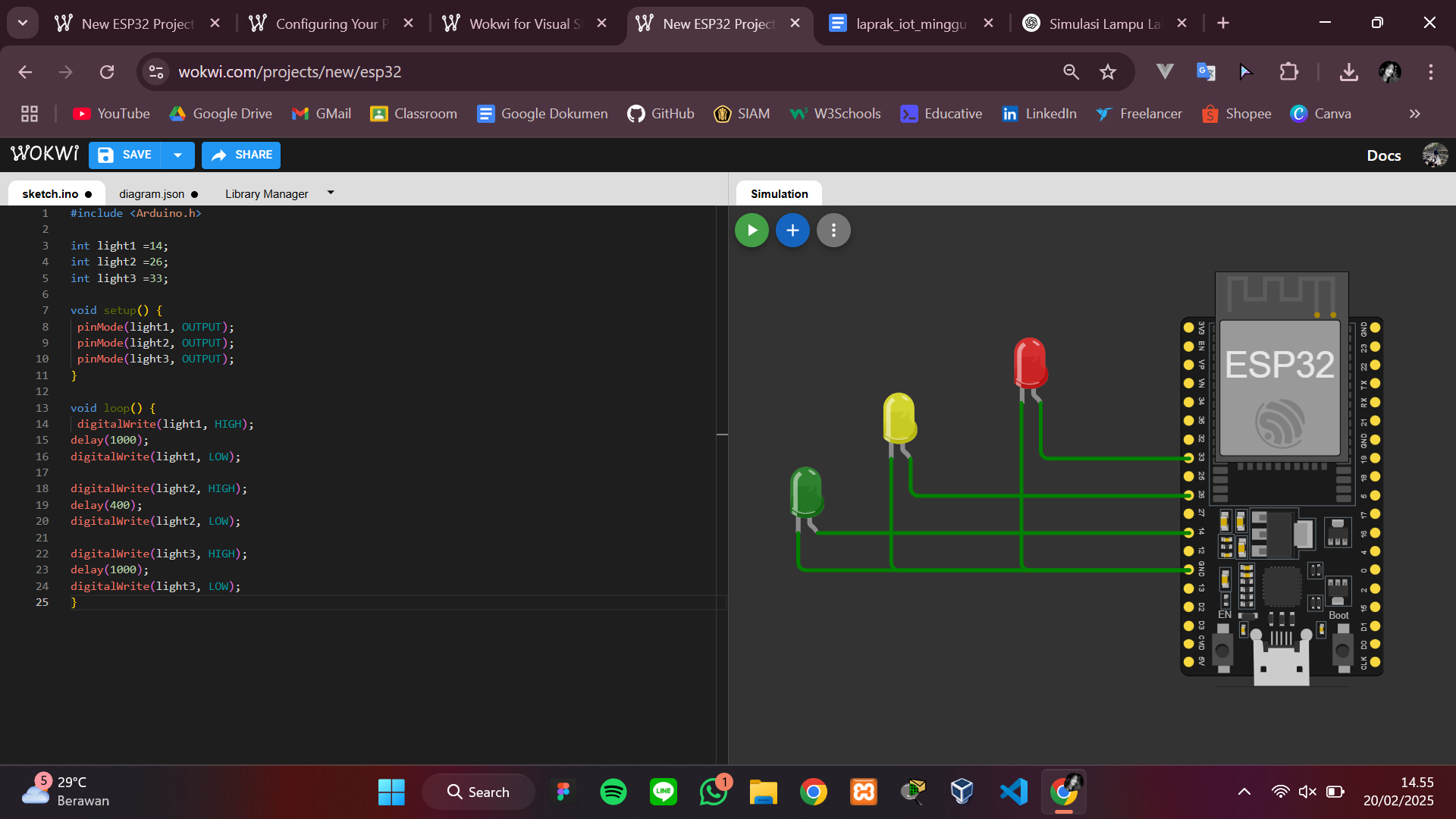
1. Pilih ESP32, kemudian pilih starter templates “ESP32”



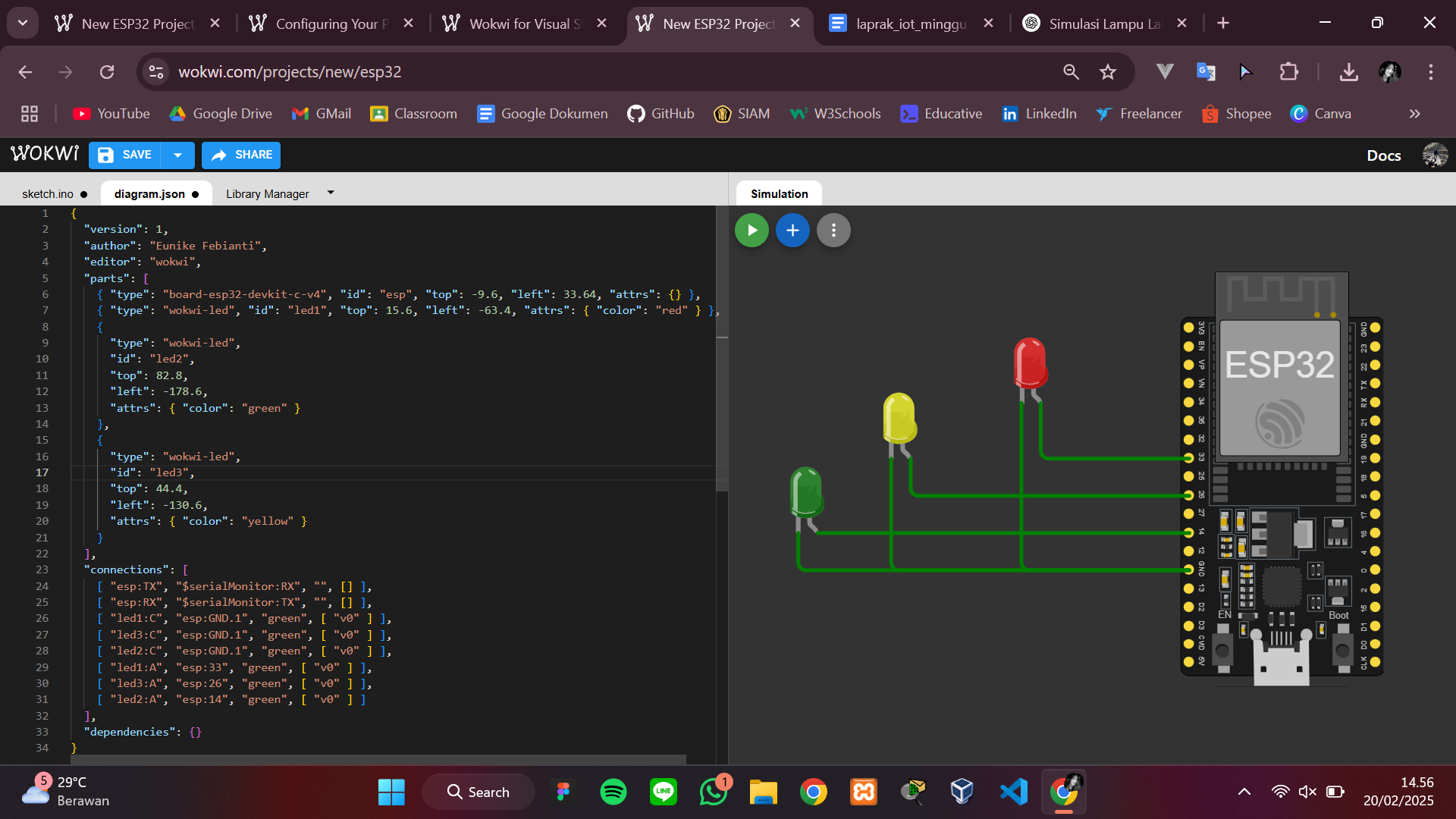


1. Tambahkan LED berwarna merah, kuning, dan hijau kemudian hubungkan LED merah dengan esp 33 & esp GND, LED kuning dengan esp 26 & esp GND, LED hijau dengan esp 14 & esp GND

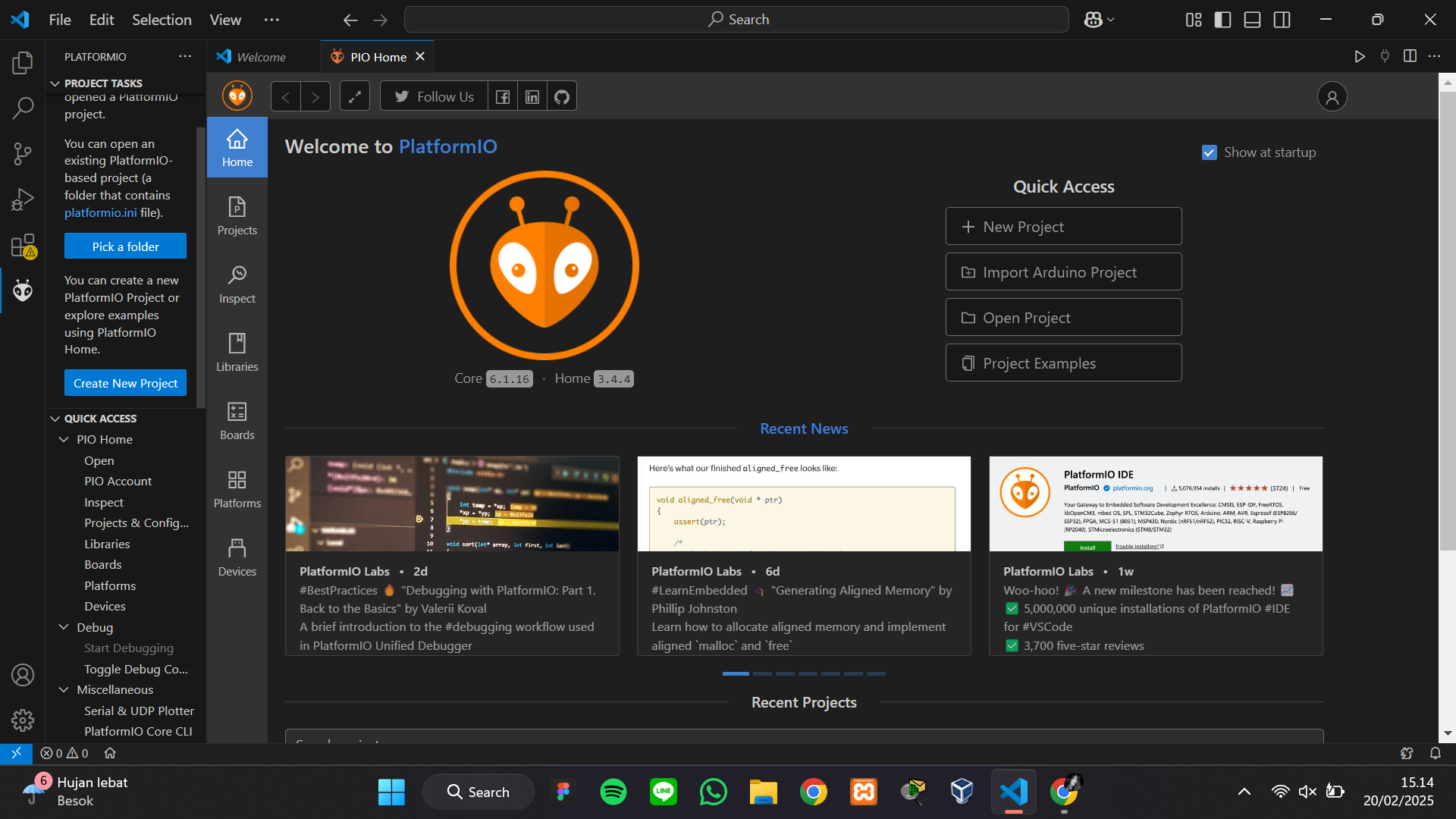


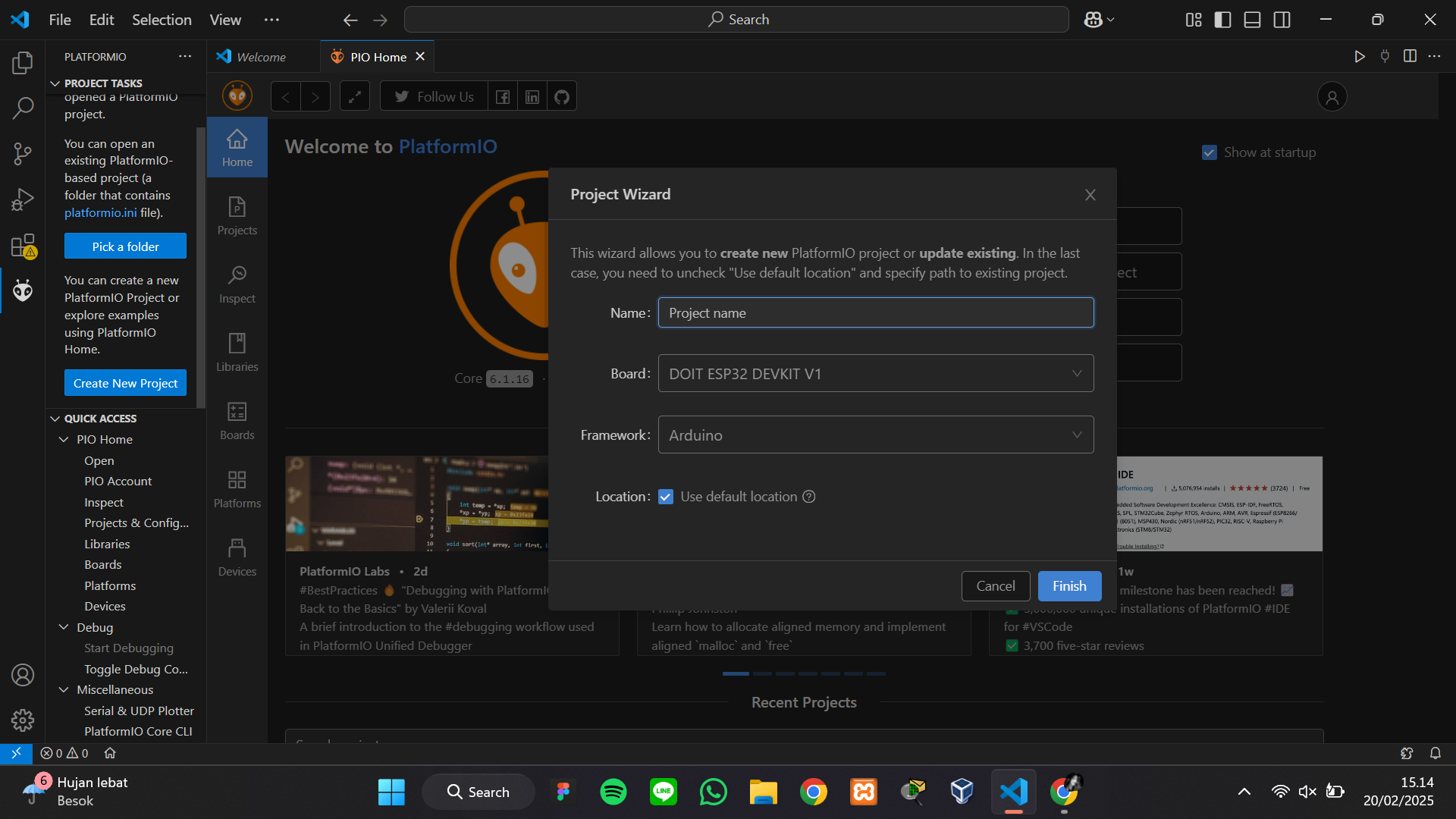


1. Buka file diagram.json lalu copy ke dalam folder di VS Code

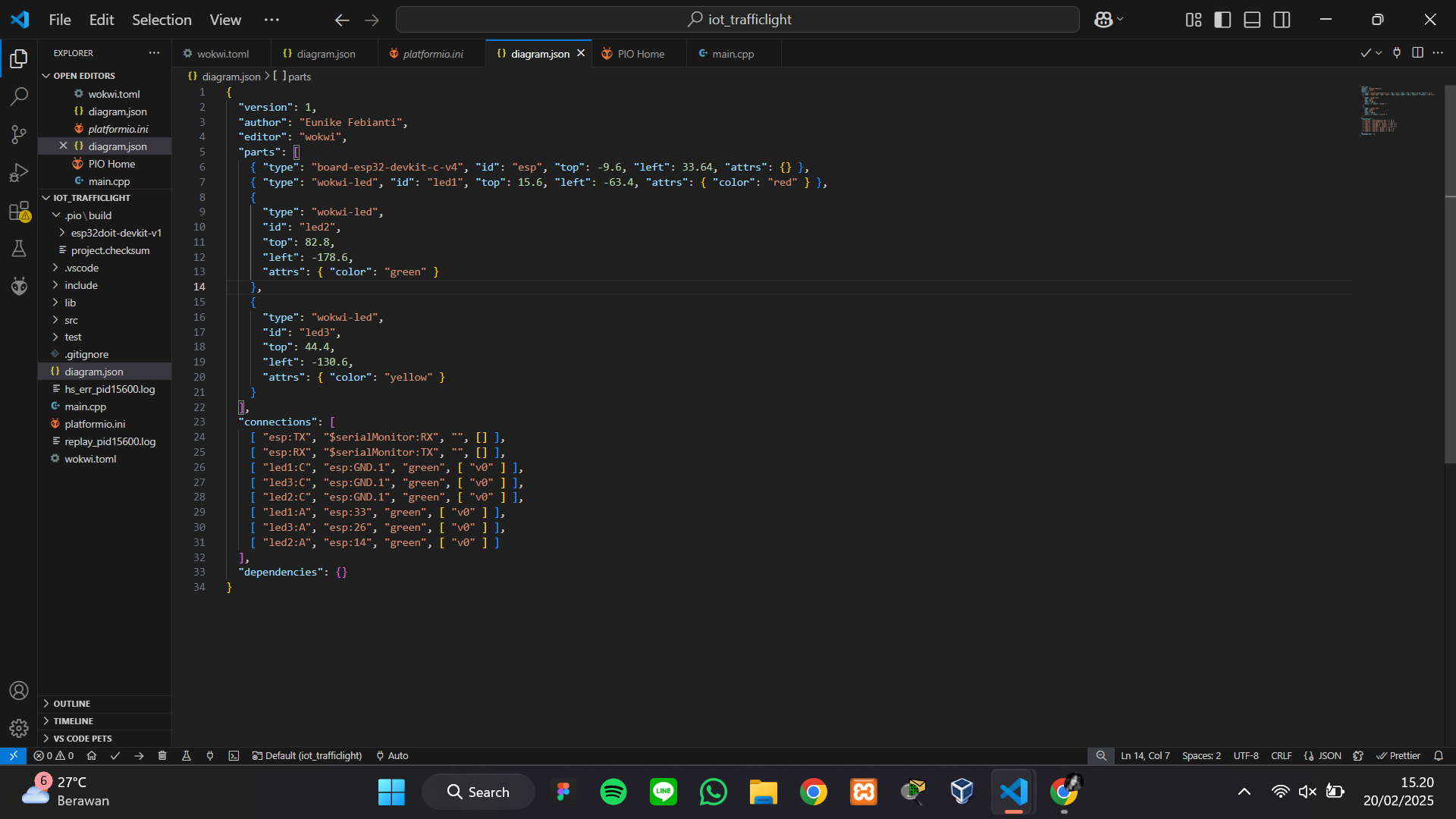


1. Buka aplikasi VS Code kemudian createnew project pada PlatformIO

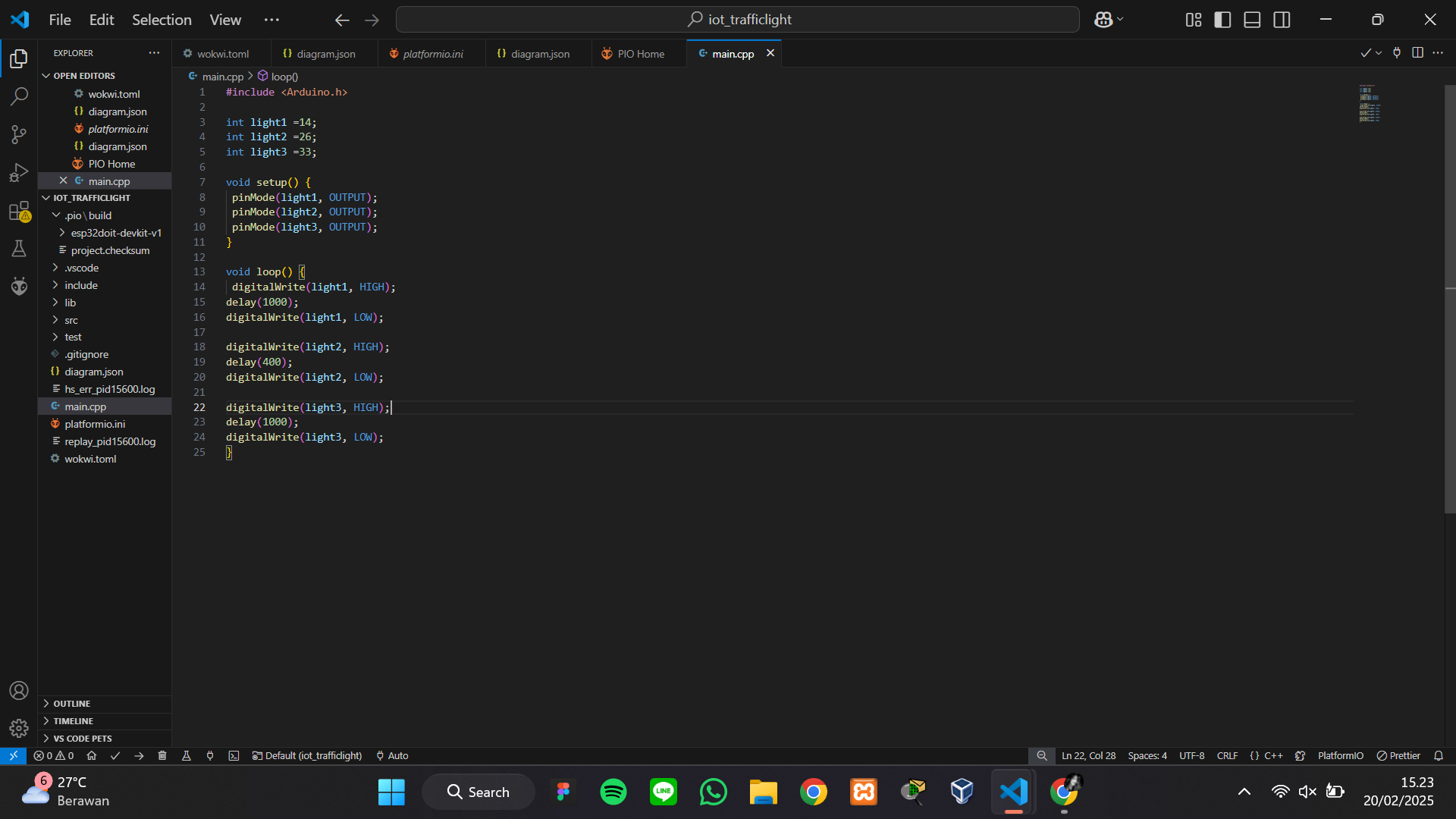




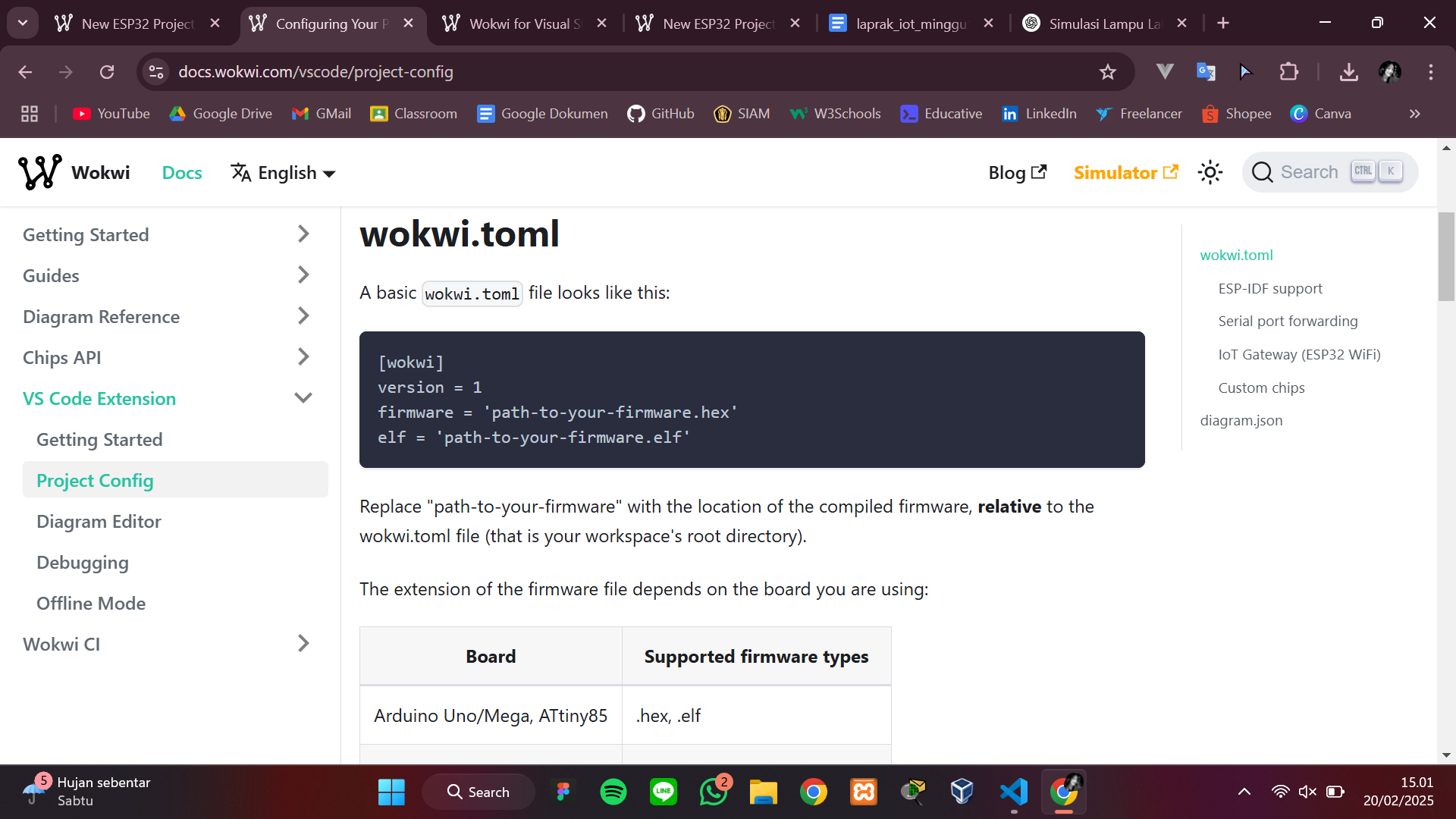
1. Buat new file bernama “diagram.json”, “wokwi.toml”. Paste kan diagram.json pada web wokwi simulator ke dalam file diagram.json di vscode yang baru saja dibuat



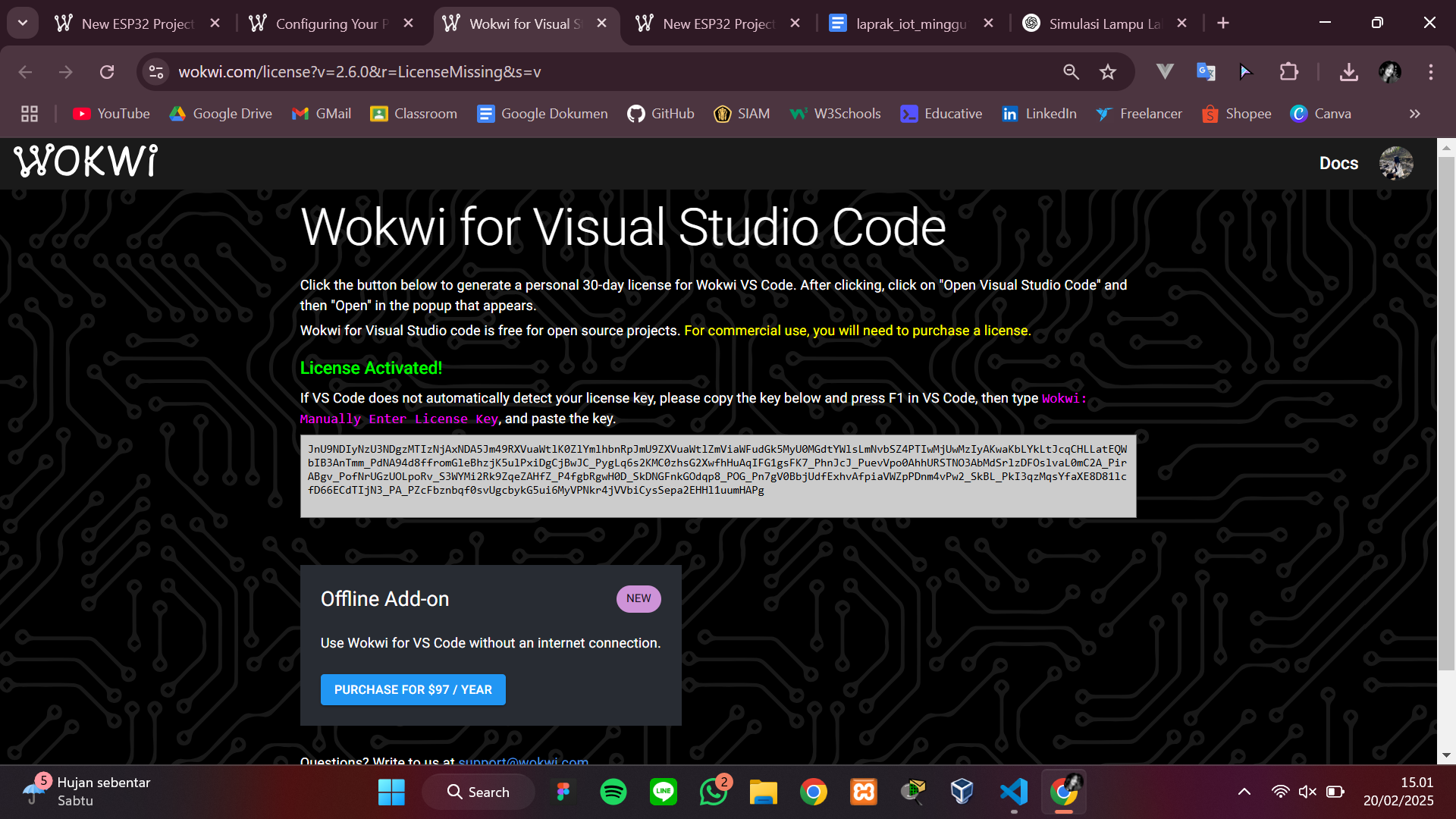
1. Isikan code berikut ke dalam file src/main.cpp



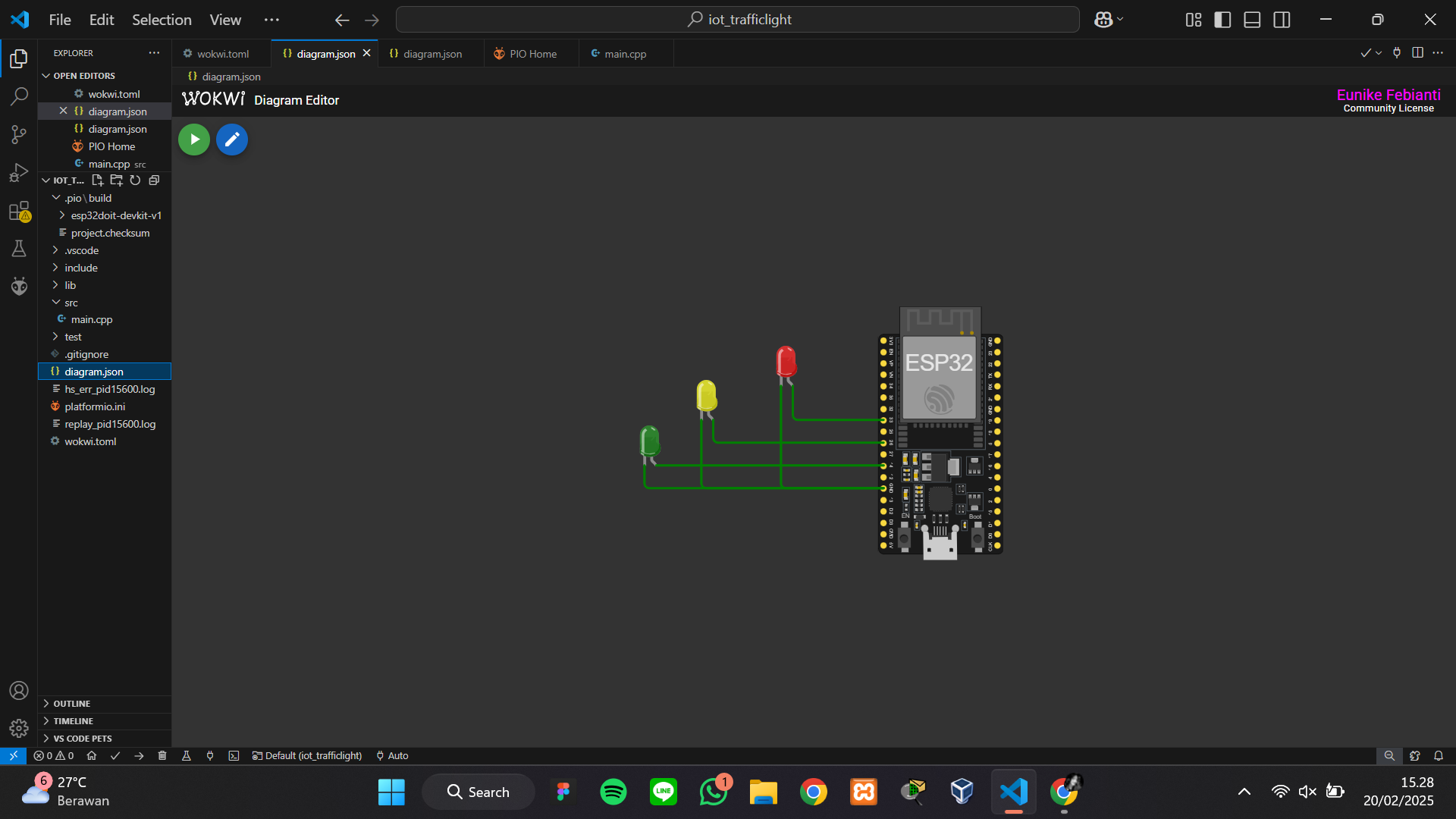
1. Isikan kode berikut ke dalam file wokwi.toml



1. Dapatkan license key pada website wokwi



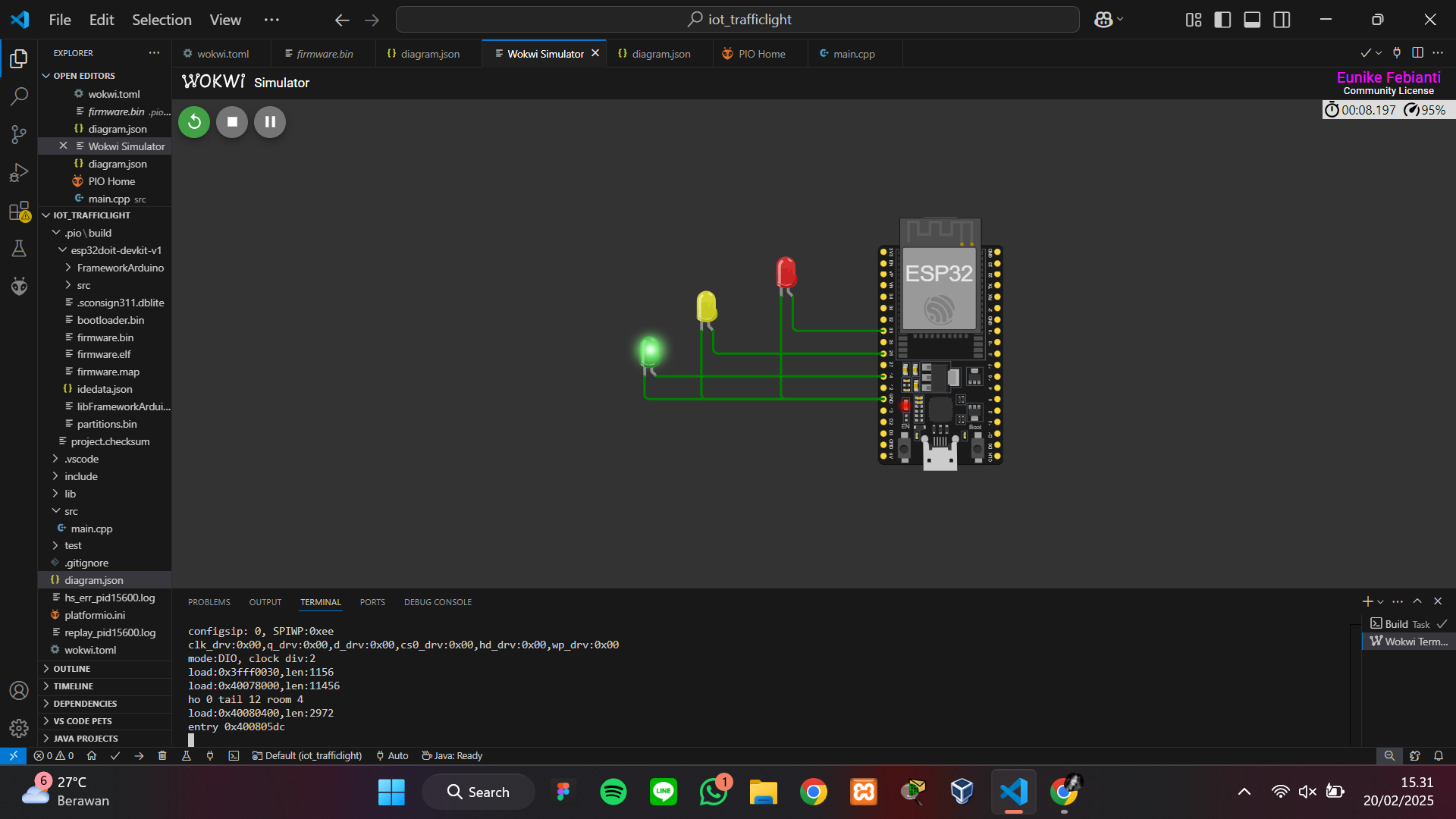
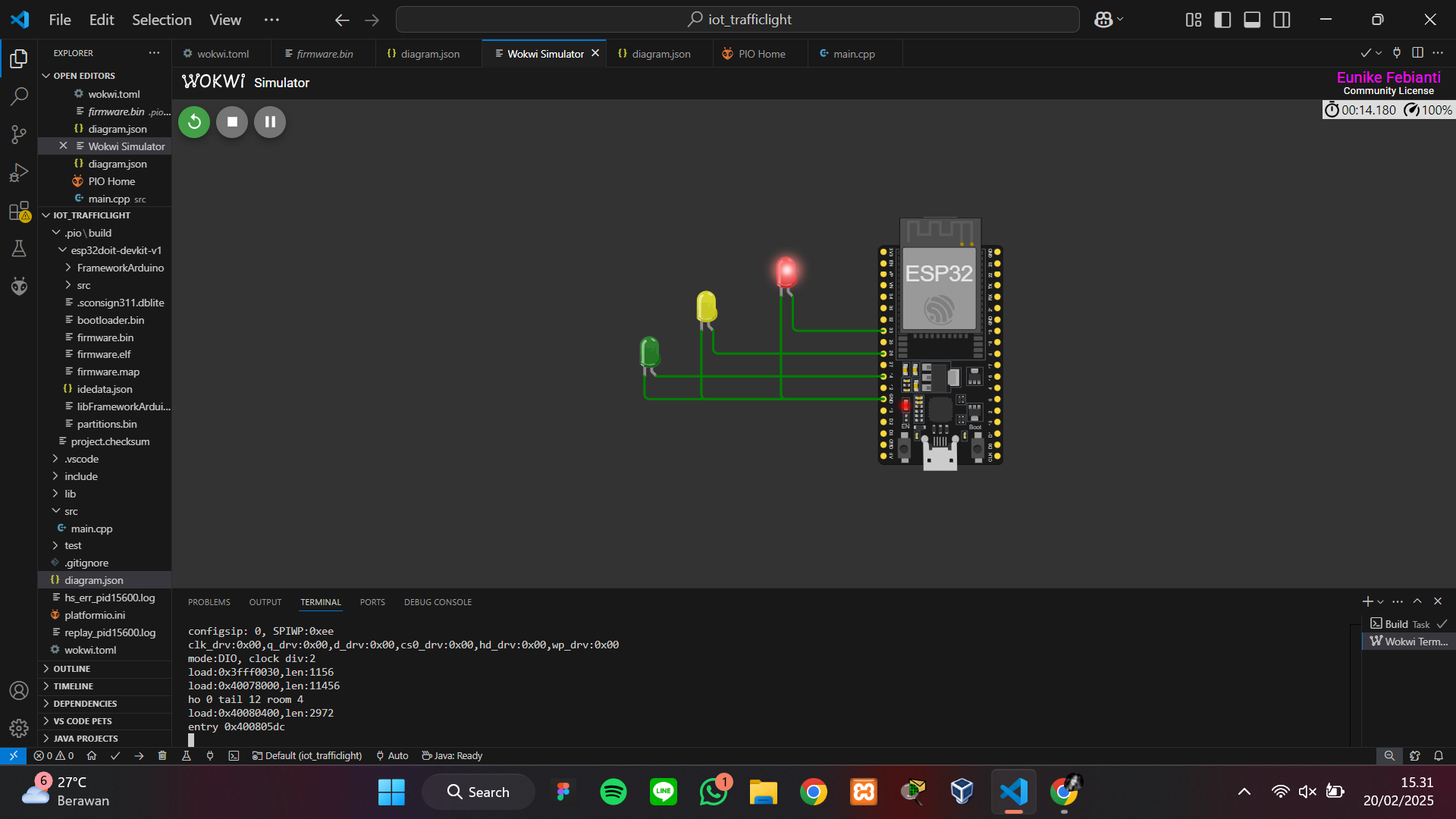
1. Jika sudah melakukan semua step coba compile dan run simulator



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

**3.1 Experimental Results (Hasil Eksperimen)**

Hasil running simulator wokwi



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

Kode program akhir :

main.cpp

#include <Arduino.h>

int light1 =14;

int light2 =26;

int light3 =33;

void setup() {

pinMode(light1, OUTPUT);

pinMode(light2, OUTPUT);

pinMode(light3, OUTPUT);

}

void loop() {

digitalWrite(light1, HIGH);

delay(1000);

digitalWrite(light1, LOW);

digitalWrite(light2, HIGH);

delay(400);

digitalWrite(light2, LOW);

digitalWrite(light3, HIGH);

delay(1000);

digitalWrite(light3, LOW);

}

**diagram.json**

{

"version": 1,

"author": "Eunike Febianti",

"editor": "wokwi",

"parts": [

{ "type": "board-esp32-devkit-c-v4", "id": "esp", "top": -9.6, "left": 33.64, "attrs": {} },

{ "type": "wokwi-led", "id": "led1", "top": 15.6, "left": -63.4, "attrs": { "color": "red" } },

{

"type": "wokwi-led",

"id": "led2",

"top": 82.8,

"left": -178.6,

"attrs": { "color": "green" }

},

{

"type": "wokwi-led",

"id": "led3",

"top": 44.4,

"left": -130.6,

"attrs": { "color": "yellow" }

}

],

"connections": [

[ "esp:TX", "$serialMonitor:RX", "", [] ],

[ "esp:RX", "$serialMonitor:TX", "", [] ],

[ "led1:C", "esp:GND.1", "green", [ "v0" ] ],

[ "led3:C", "esp:GND.1", "green", [ "v0" ] ],

[ "led2:C", "esp:GND.1", "green", [ "v0" ] ],

[ "led1:A", "esp:33", "green", [ "v0" ] ],

[ "led3:A", "esp:26", "green", [ "v0" ] ],

[ "led2:A", "esp:14", "green", [ "v0" ] ]

],

"dependencies": {}

}

**wokwi.toml**

[wokwi]

version = 1

firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'

elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'