LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Pembuatan Traffic Lights pada Wokwi Simulator**



*Amanda aurelia*

*233140701111049/T4F*

*Fakultas Vokasi, Universitas Brawijaya*

[*sky.lee1505@gmail.com*](mailto:sky.lee1505@gmail.com)

**Abstract (Abstrak)**

Proyek ini bertujuan untuk mensimulasikan sistem lampu lalu lintas menggunakan Wokwi Simulator di Visual Studio Code (VS Code). Dengan simulasi ini, mikrokontroler Arduino digunakan untuk mengontrol tiga buah LED yang berfungsi sebagai lampu merah, kuning, dan hijau. Lampu-lampu ini akan menyala sesuai dengan logika waktu dan urutan yang telah disesuaikan dengan sistem lalu lintas nyata, sehingga dapat mempresentasikan cara kerja lampu lalu lintas di persimpangan jalan secara lebih realistis.

Dengan menggunakan Wokwi Simulator, pengguna dapat melihat visualisasi real-time dari sistem yang dibuat, memungkinkan pengujian serta penyempurnaan logika sebelum diimplementasikan pada perangkat fisik. Hal ini tidak hanya mempermudah proses debugging, tetapi juga dapat menghemat biaya dan waktu dibandingkan dengan langsung menguji pada perangkat keras yang sebenarnya.

**Introduction (Pendahuluan)**

**1.1 Latar Belakang**

Lampu lalu lintas berperan penting dalam mengatur arus kendaraan dan juga pejalan kaki untuk menjaga keselamatan serta kelancaran lalu lintas di jalan raya. Memahami cara kerja dan pengaturan sistem ini menjadi langkah awal yang penting, terutama dalam penerapan teknologi berbasis mikrokontroler untuk otomatisasi lalu lintas.

Dengan menggunakan Wokwi Simulator di Visual Studio Code (VS Code), simulasi sistem lalu lintas dapat dilakukan tanpamemerlukan perangkat fisik. Hal ini memungkinkan pengguna untuk bereksperimen dan menguji logika kendali lampu lalu lintas secara lebih fleksibel menggunakan mikrokontroler Arduino. Selain itu, fitur interaktif seperti tombol penyeberangan bagi pejalan kaki dapat ditambahkan untuk meningkatkan realisme simulasi.

**1.2 Tujuan Eksperimen**

1. Menyimulasikan Sistem Lampu Lalu Lintas

Mengembangkan simulasi sistem lampu lalu lintas menggunakan Wokwi Simulator di Visual Studio Code (VS Code) untuk memahami prinsip kerja sistem pengaturan lalu lintas berbasis mikrokontroler. Simulasi ini memungkinkan pengujian tanpa perangkat fisik, sehingga lebih fleksibel dalam eksplorasi konsep dan logika sistem.

1. Menguji Logika dan Urutan Lampu

Memprogram Arduino untuk mengontrol LED sebagai representasi lampu merah, kuning, dan hijau dengan menerapkan urutan nyala serta durasi yang sesuai dengan sistem lalu lintas nyata. Hal ini bertujuan untuk memastikan bahwa simulasi mencerminkan kondisi sebenarnya di jalan raya.

1. Mengoptimalkan Konfigurasi Lampu Lalu Lintas

Melakukan berbagai pengujian guna menyempurnakan urutan serta durasi nyala lampu agar lebih efisien dan sesuai dengan kondisi lalu lintas yang sesungguhnya. Penyesuain ini dapat mencakup faktor seperti waktu siklus lampu yang optimal dan penerapan kondisi tertentu, misalnya mode darurat/penyesuain berbasis volume kendaraan.

1. Mengembangkan Pemahaman tentang Sistem Tertanam

Memanfaatkan Wokwi sebagai alat pembelajaran dalam pemrograman mikrokontroler, khususnya dalam implementasi sistem kontrol lalu lintas berbasis Arduino. Proyek ini membantu dalam memahami bagaimana perangkat tertanam dapat digunakan untuk mengotomatisasi pengaturan lalu lintas secara efektif.

1. Menjadi Dasar Pengembangan Lebih Lanjut

Menyediakan landasan bagi pengembangan sistem lampu lalu lintas yang lebih canggih di masa depan, seperti penerapa sensor untuk mendeteksi kendaraan dan pejalan kaki, penggunaan algoritma adaptif untuk menyesuaikan durasi lampu berdasarkan kondisi lalu lintas, serta integrasidengan Internet Of Things (IoT) guna mendukung konsep kota pintar yang lebih efisien dan responsif.

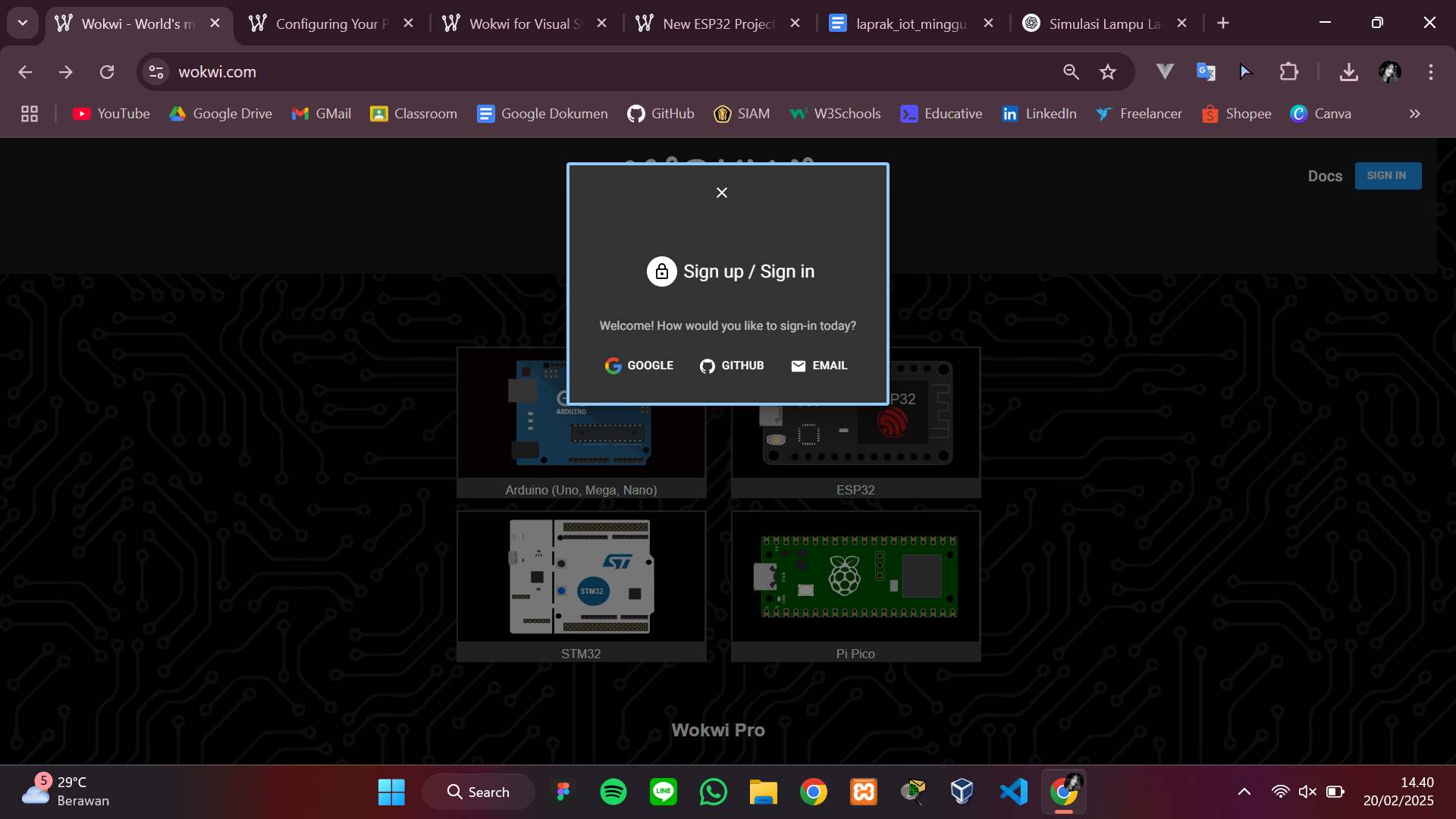
1. **Methodology (Metodologi)**

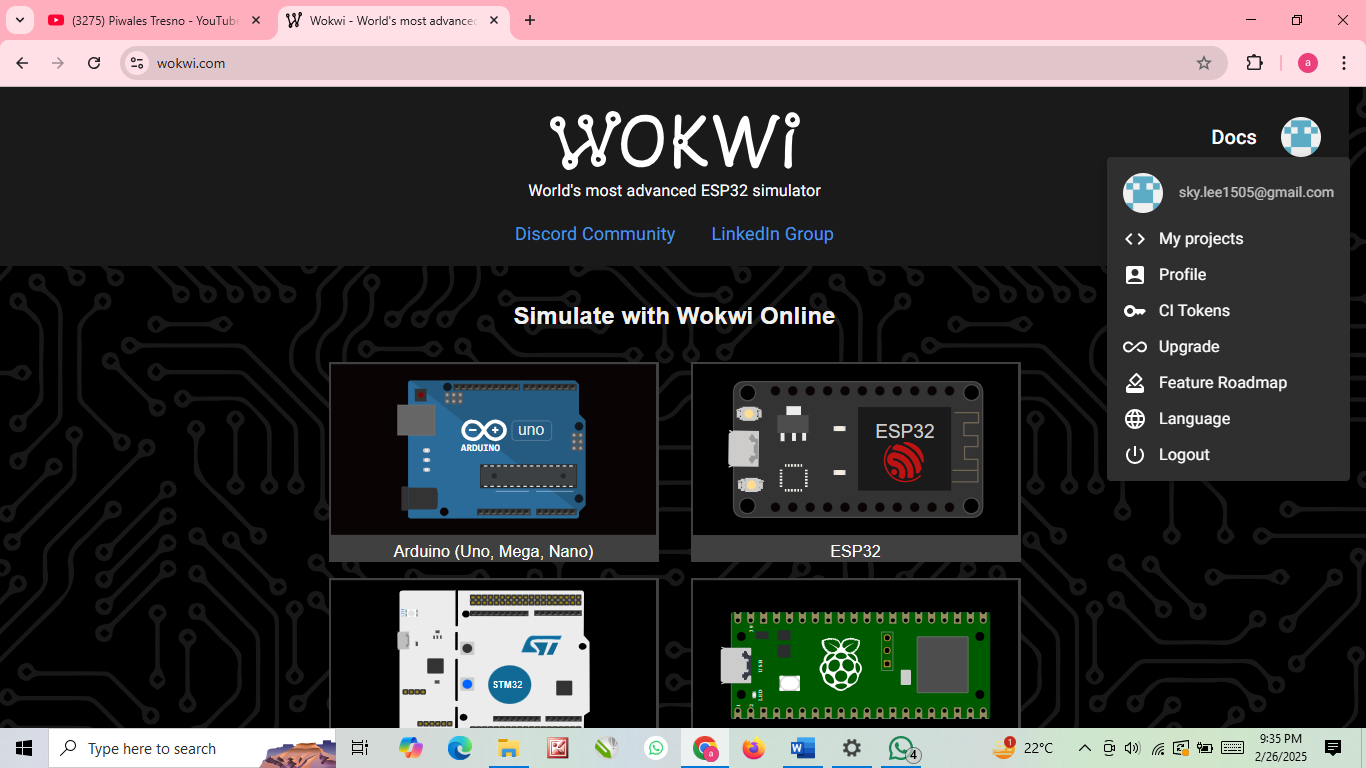
**2.1 Tools & Materials (Alat dan Bahan)**

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)**

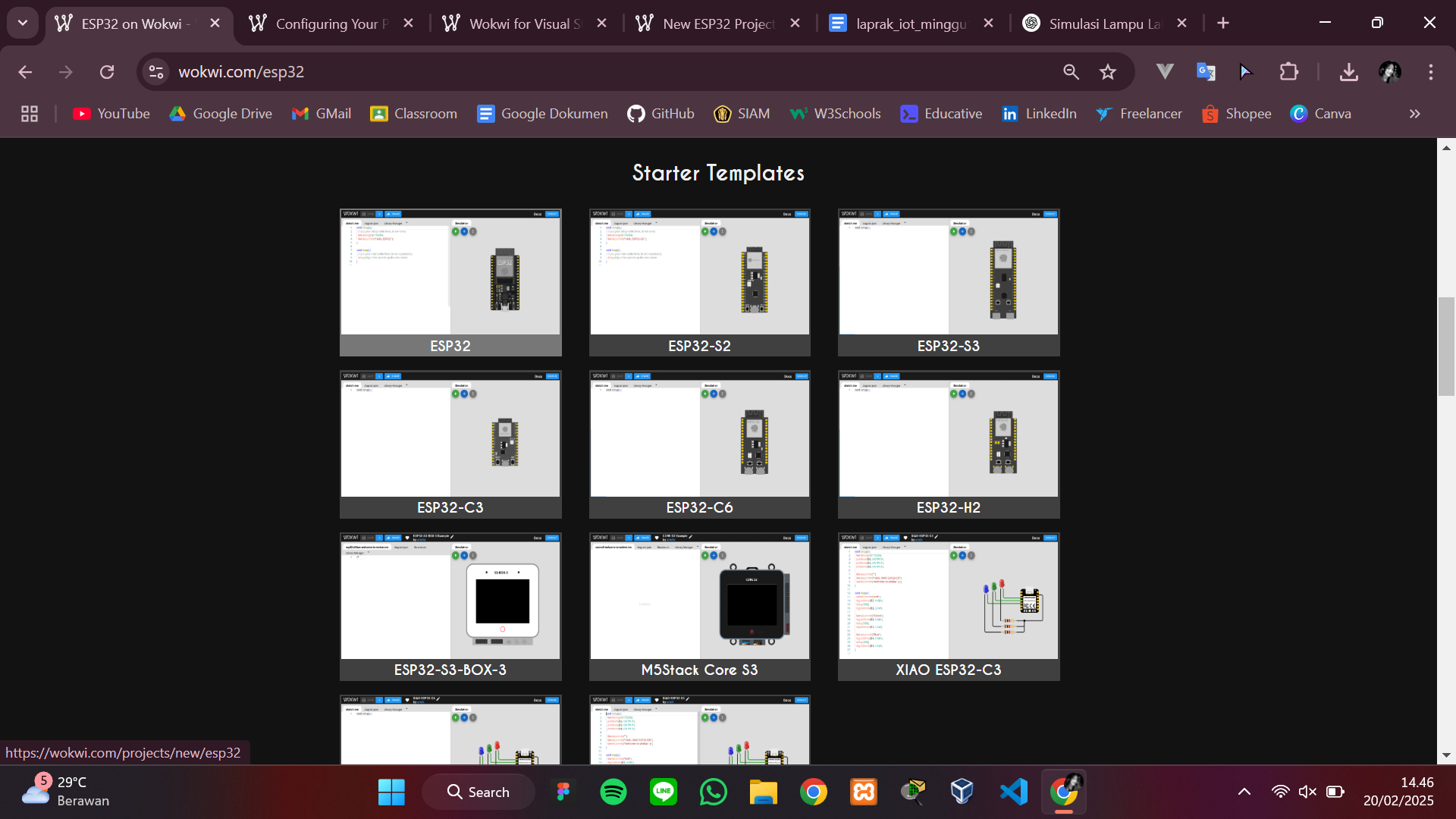
1. Membuat akun Wokwi (<https://wokwi.com/> ) atau Sign In menggunakan Github.



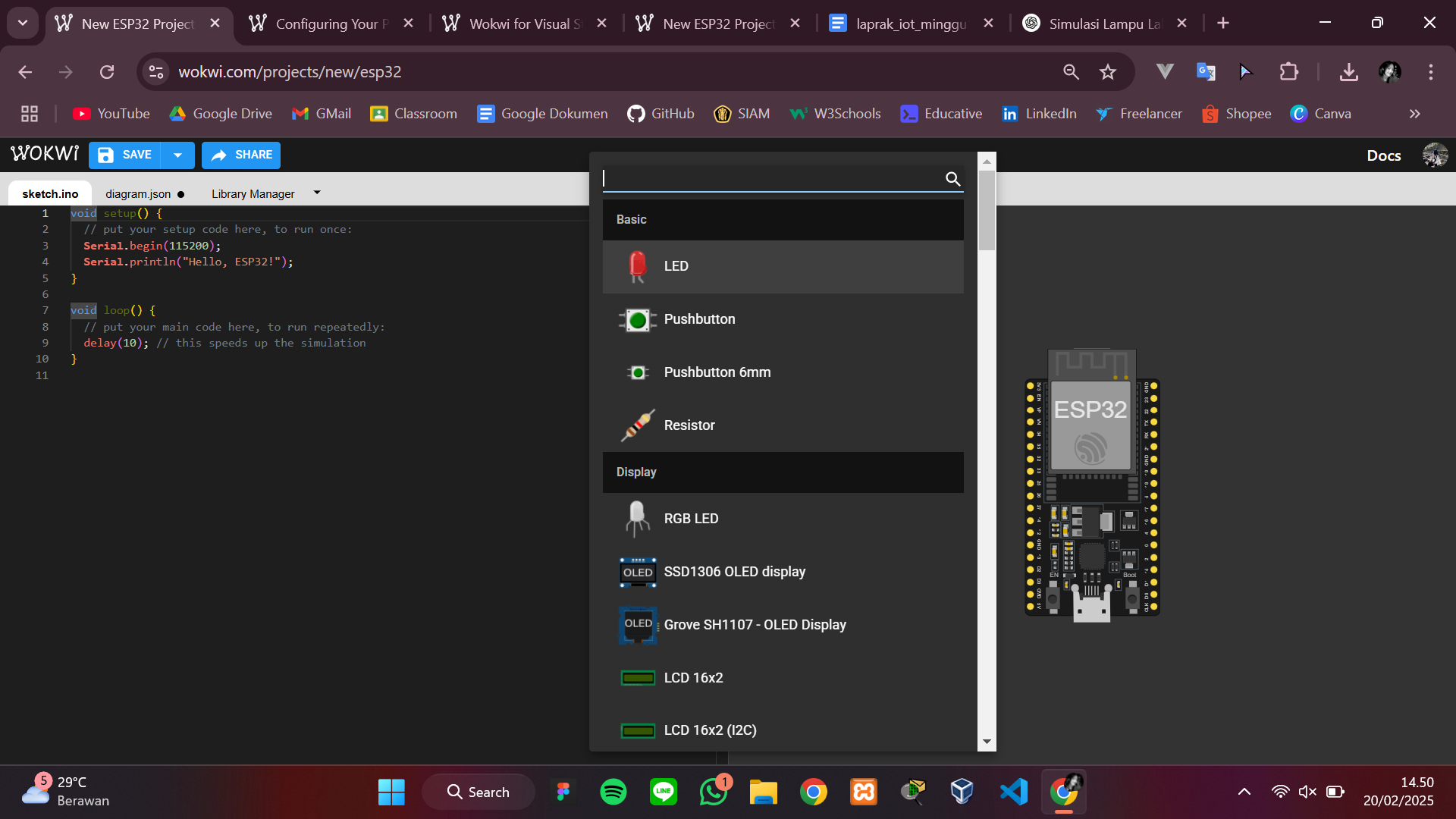


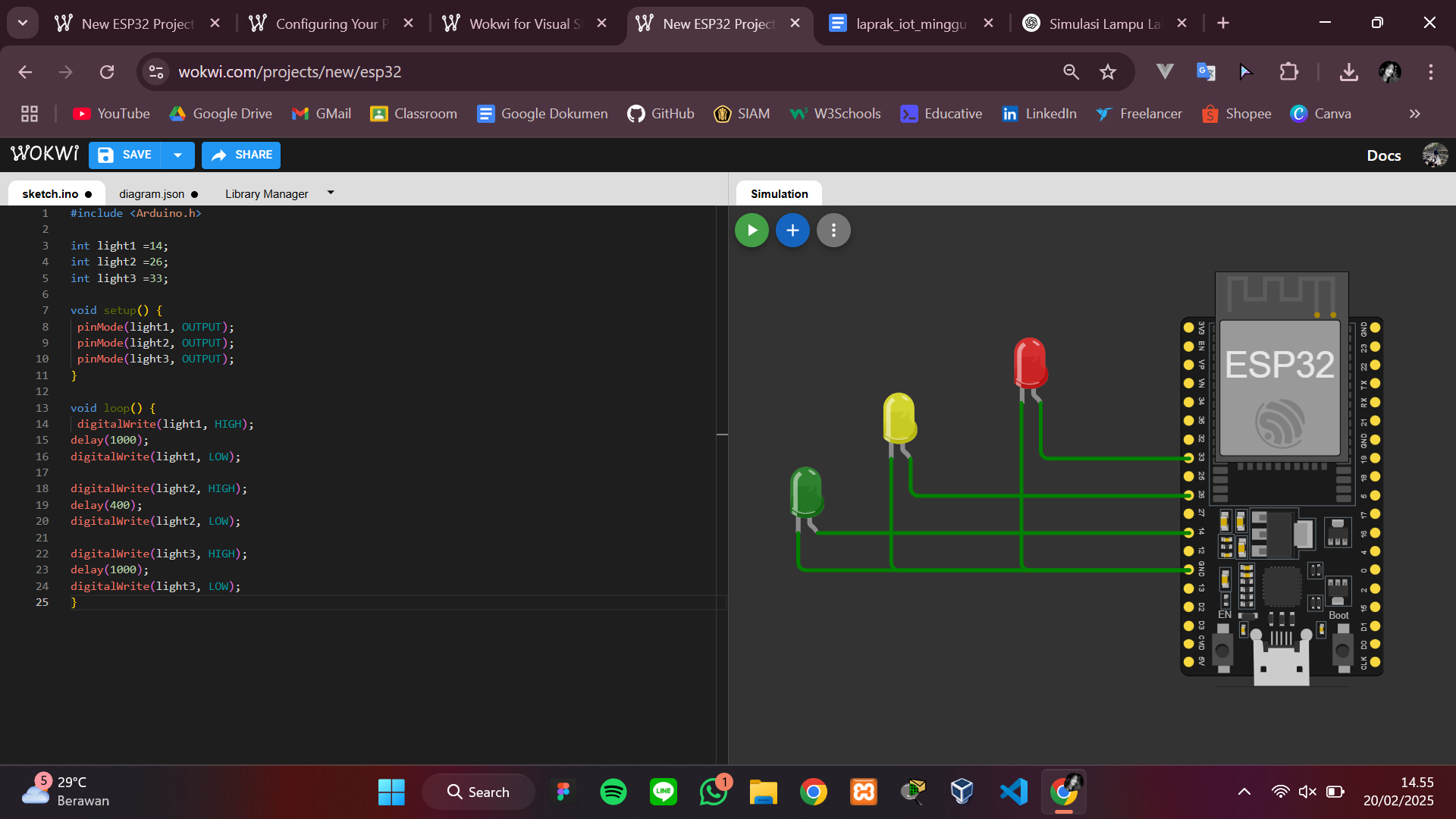
1. Pilih ESP32, kemudian pilih starter templates “ESP32”.



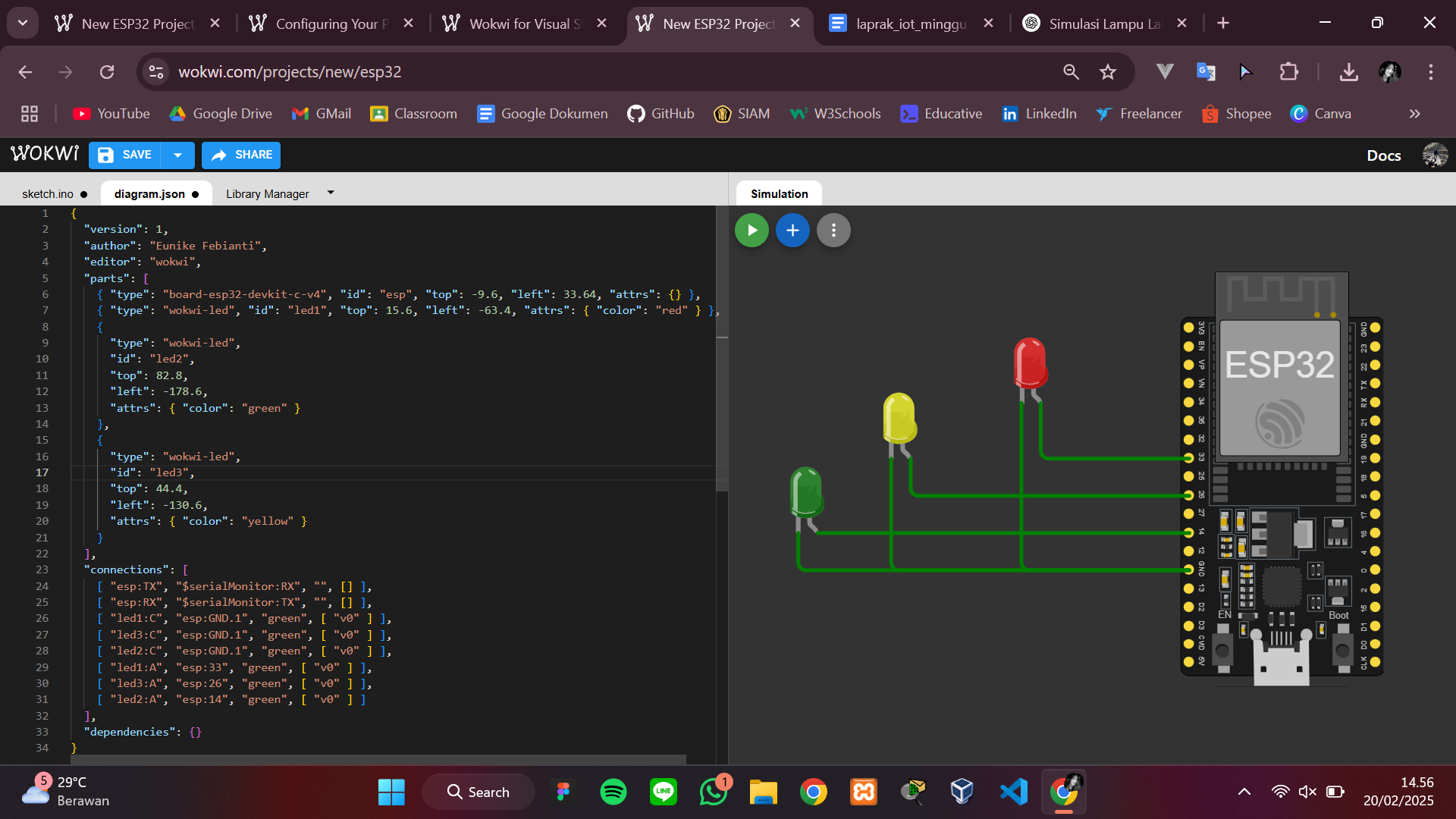


1. Tambahkan LED berwarna merah, kuning, dan hijau kemudian hubungkan LED merah dengan esp 33 & esp GND, LED kuning dengan esp 26 & esp GND, LED hijau dengan esp 14 & esp GND.

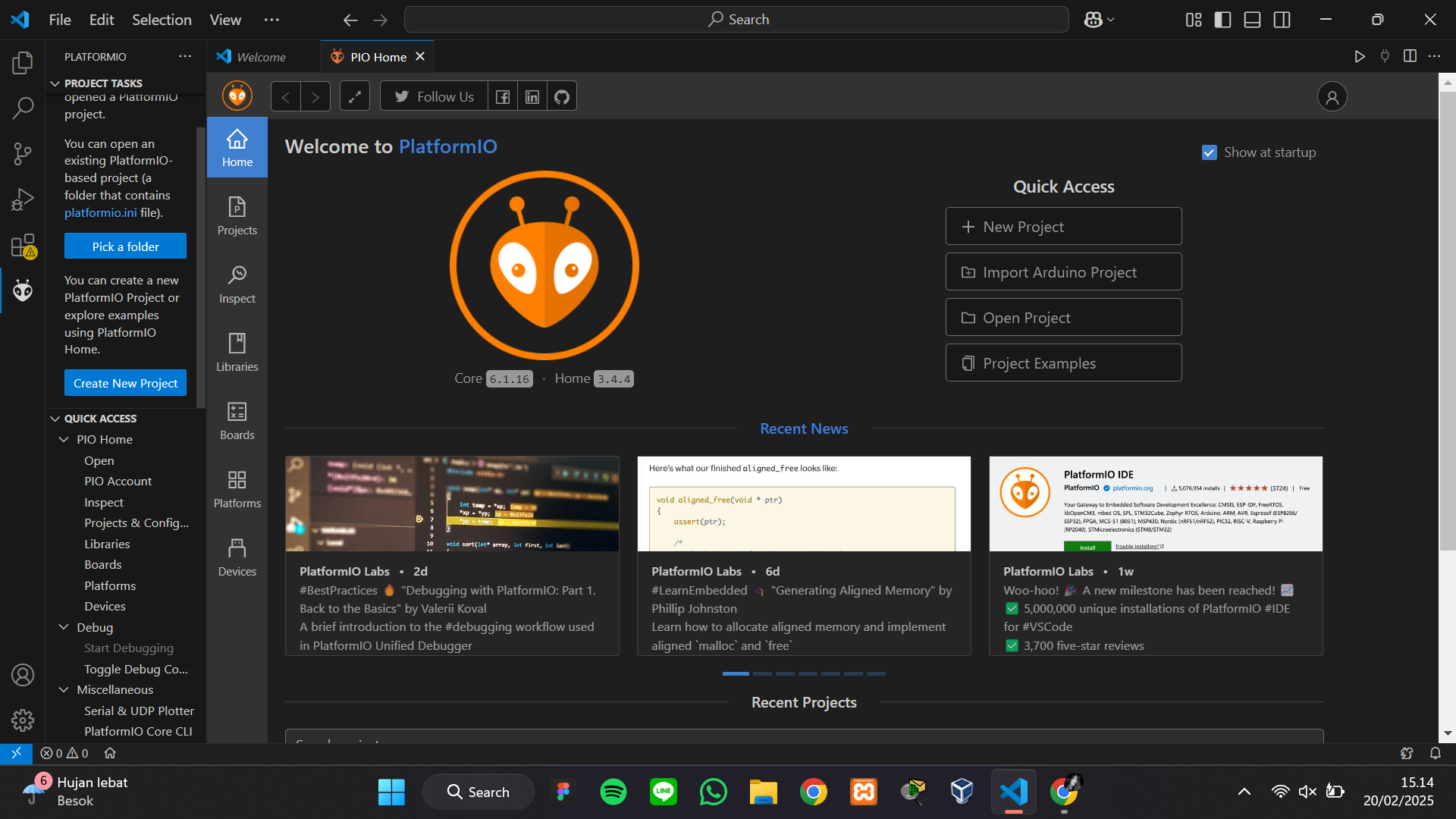


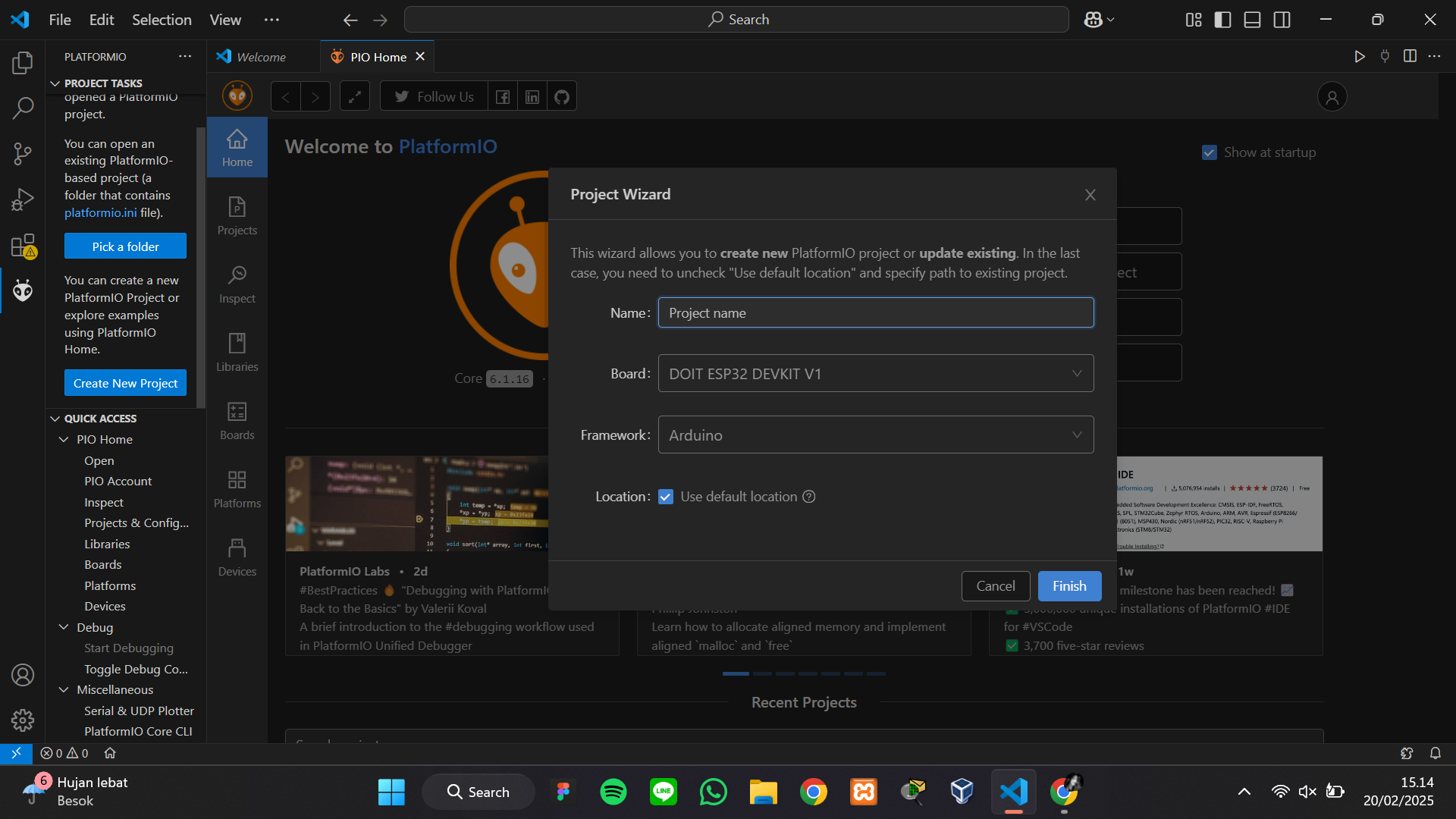


1. Buka file diagram.json lalu copy ke dalam folder di VS Code

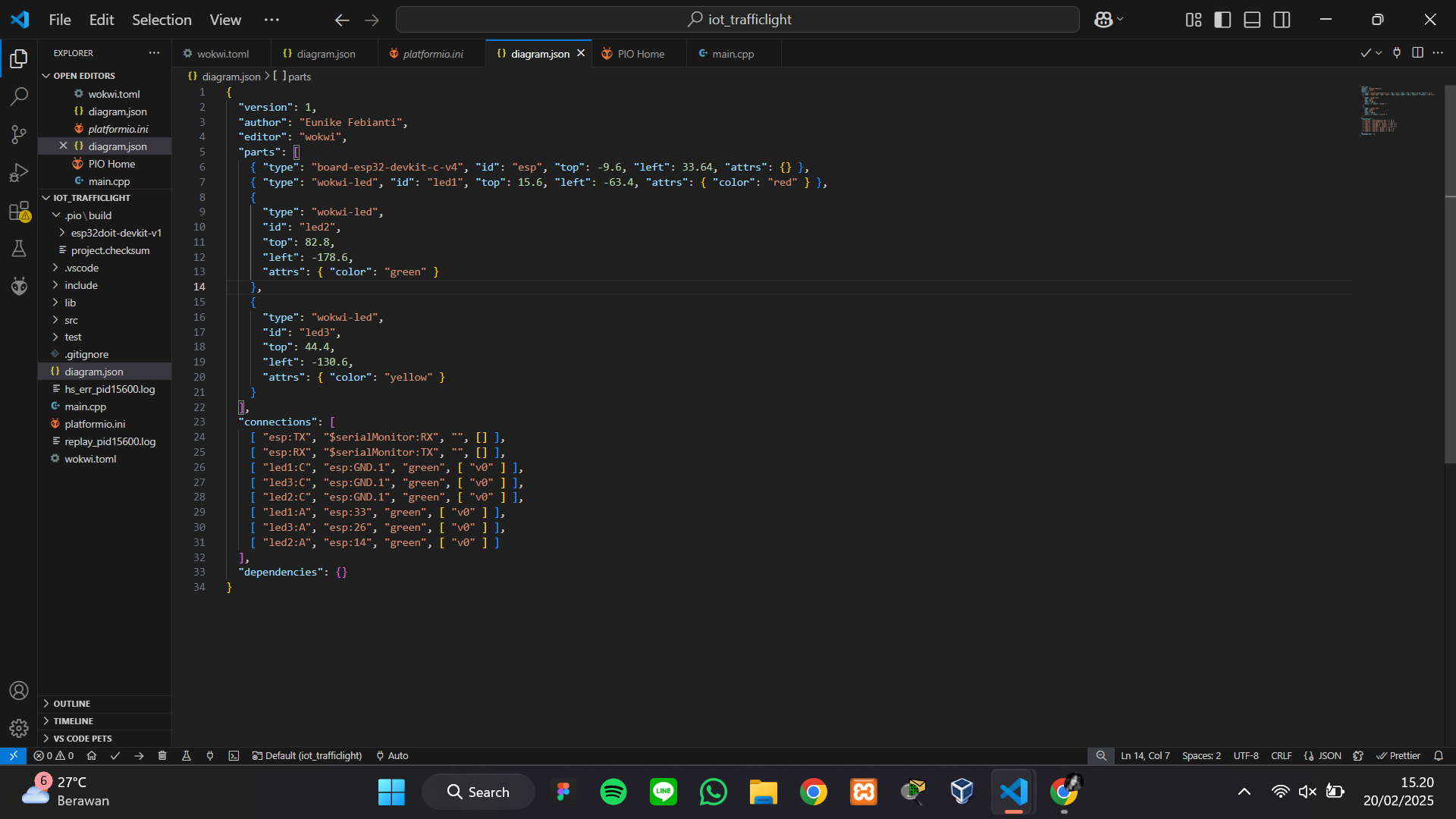


1. Buka aplikasi VS Code kemudian createnew project pada PlatformIO

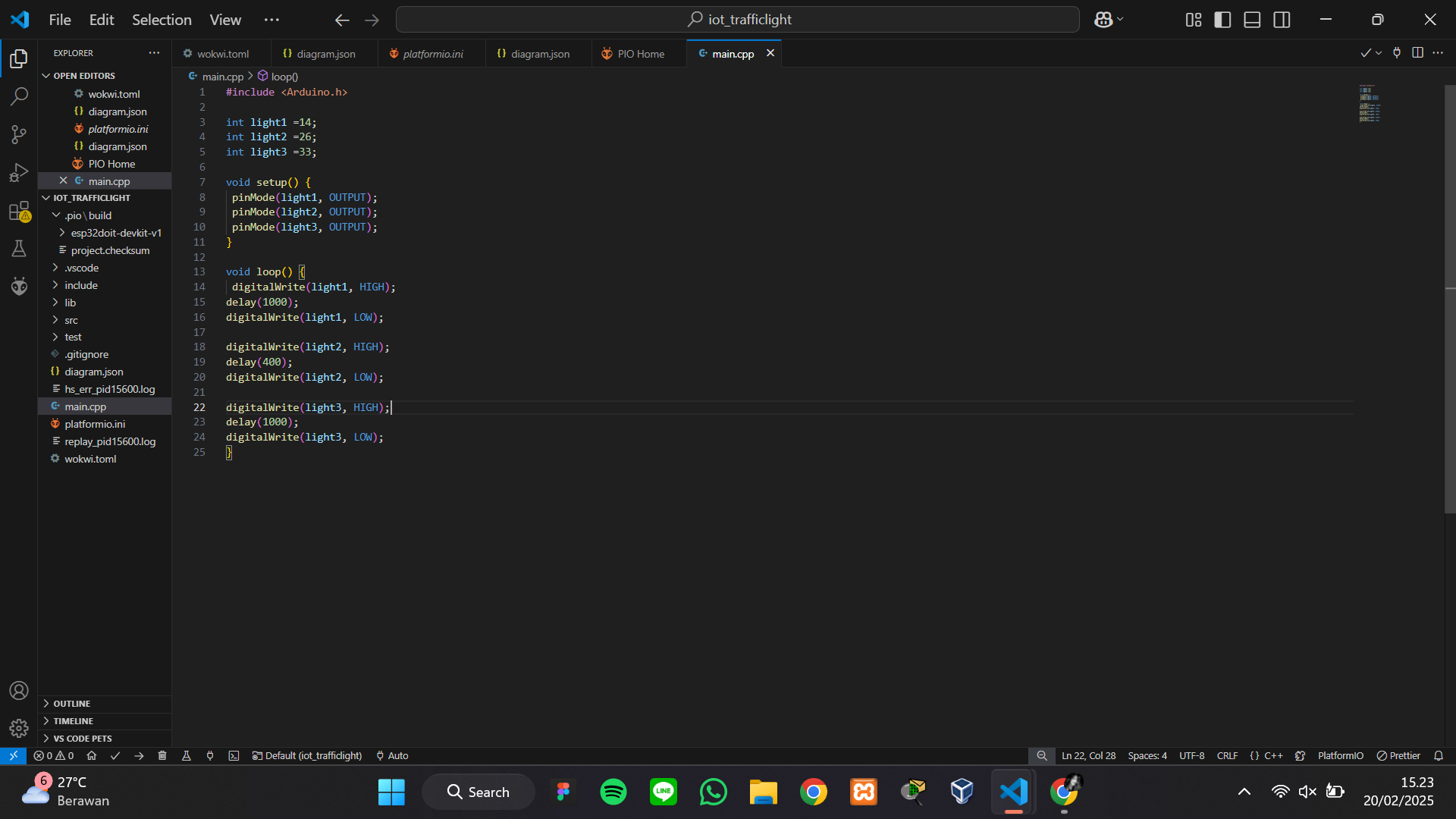




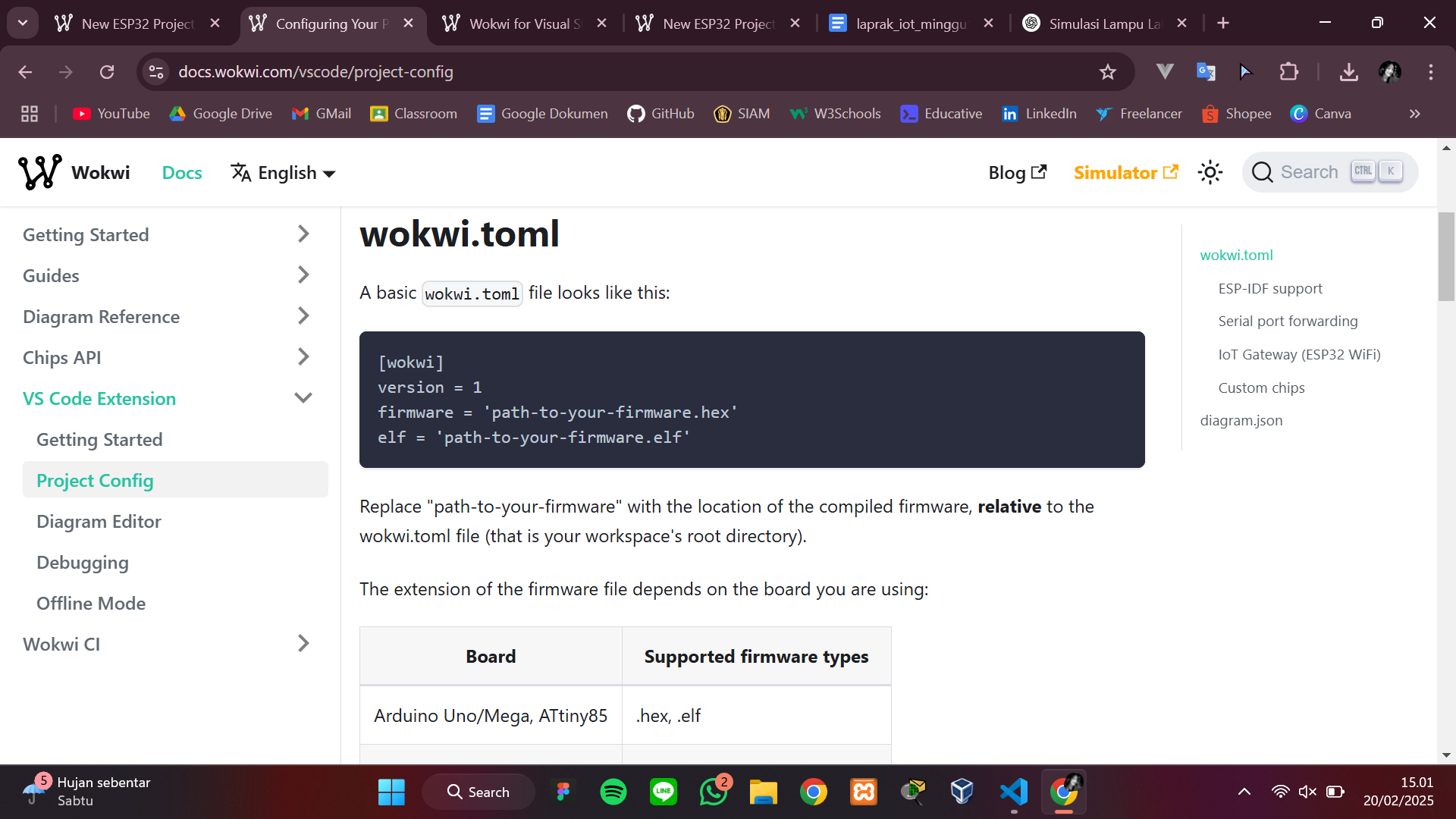
1. Buat new file bernama “diagram.json”, “wokwi.toml”. Paste kan diagram.json pada web wokwi simulator ke dalam file diagram.json di vscode yang baru saja dibuat



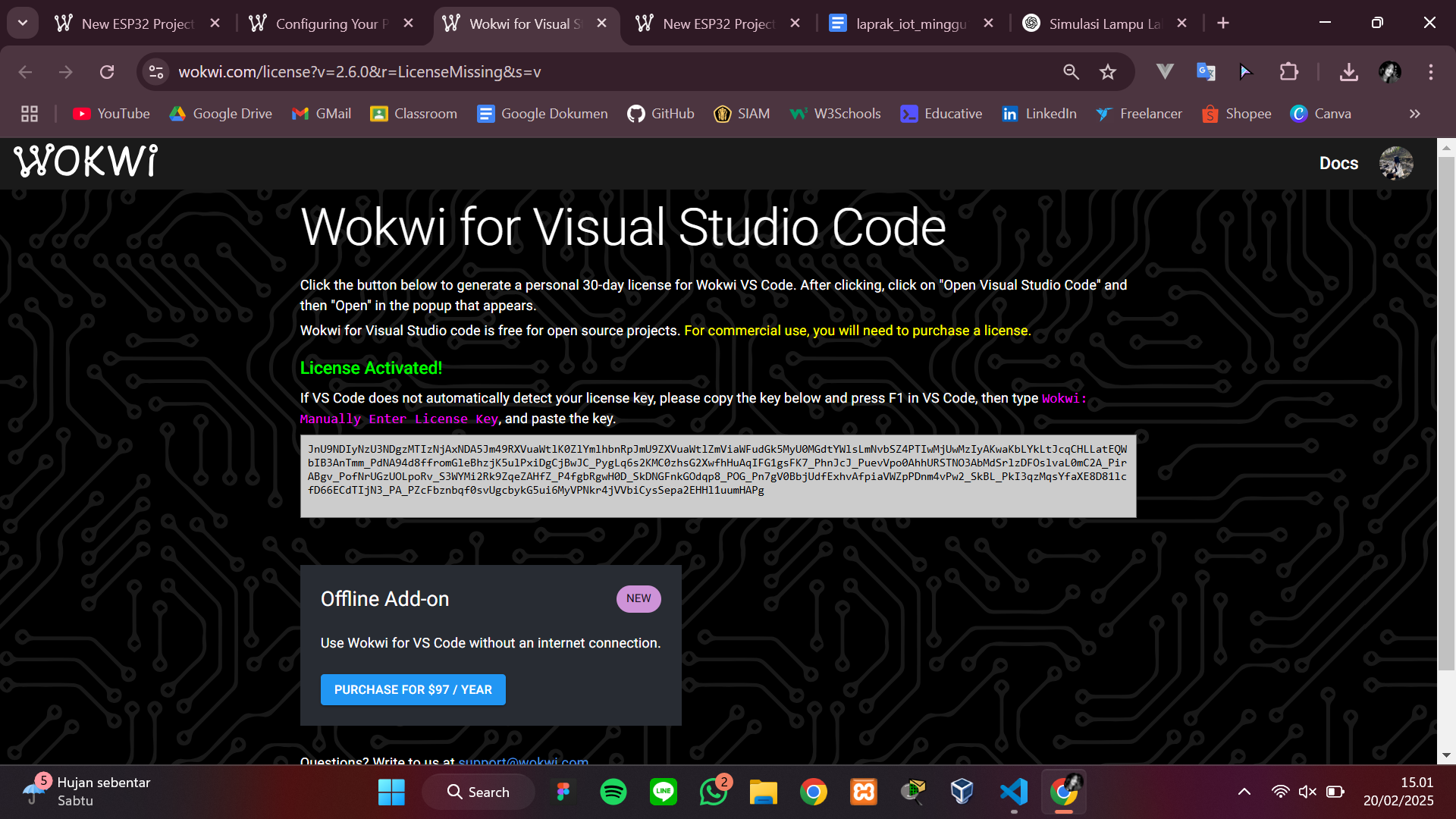
1. Isikan code berikut ke dalam file src/main.cpp



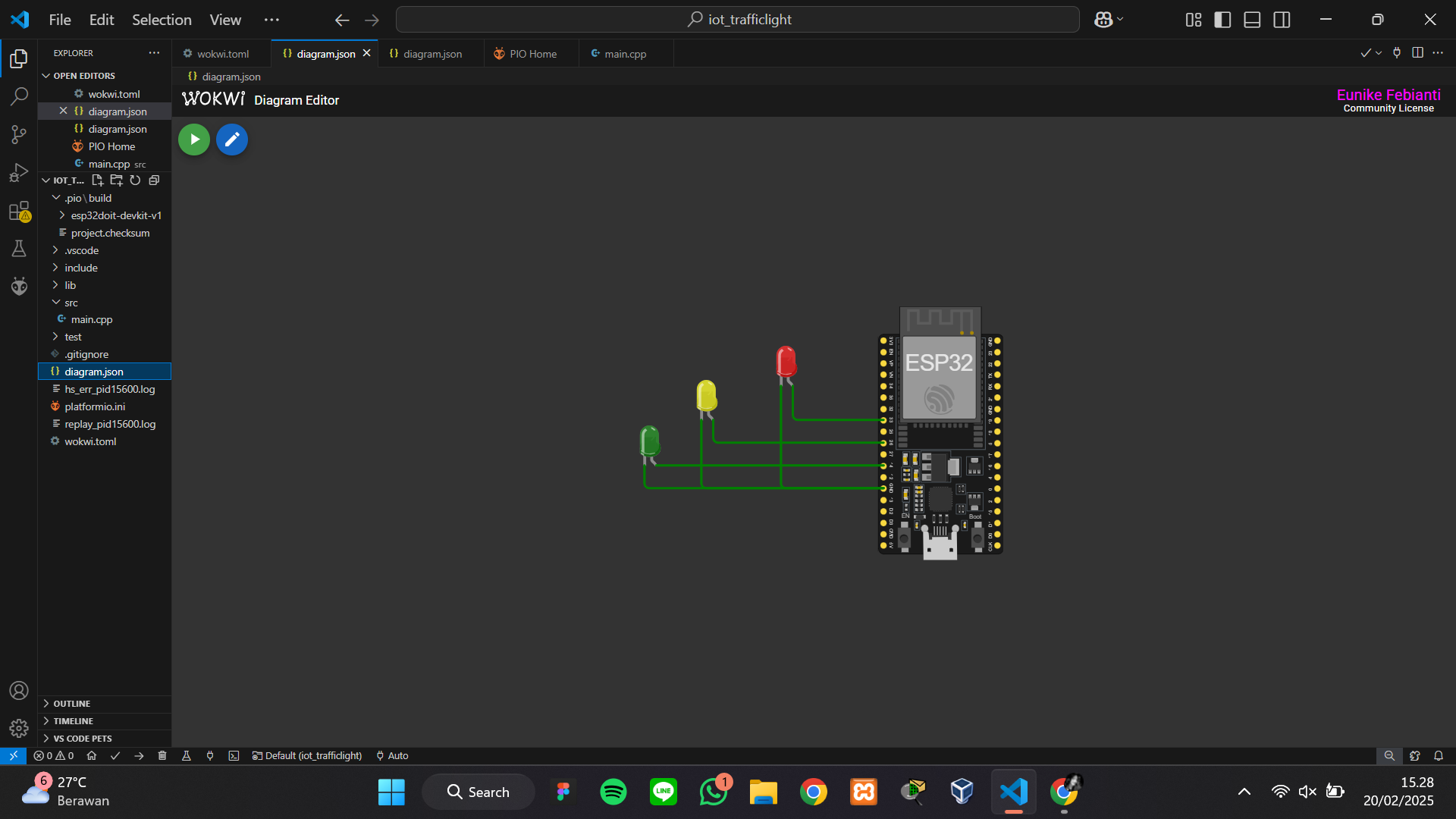
1. Isikan kode berikut ke dalam file wokwi.toml



1. Dapatkan license key pada website wokwi



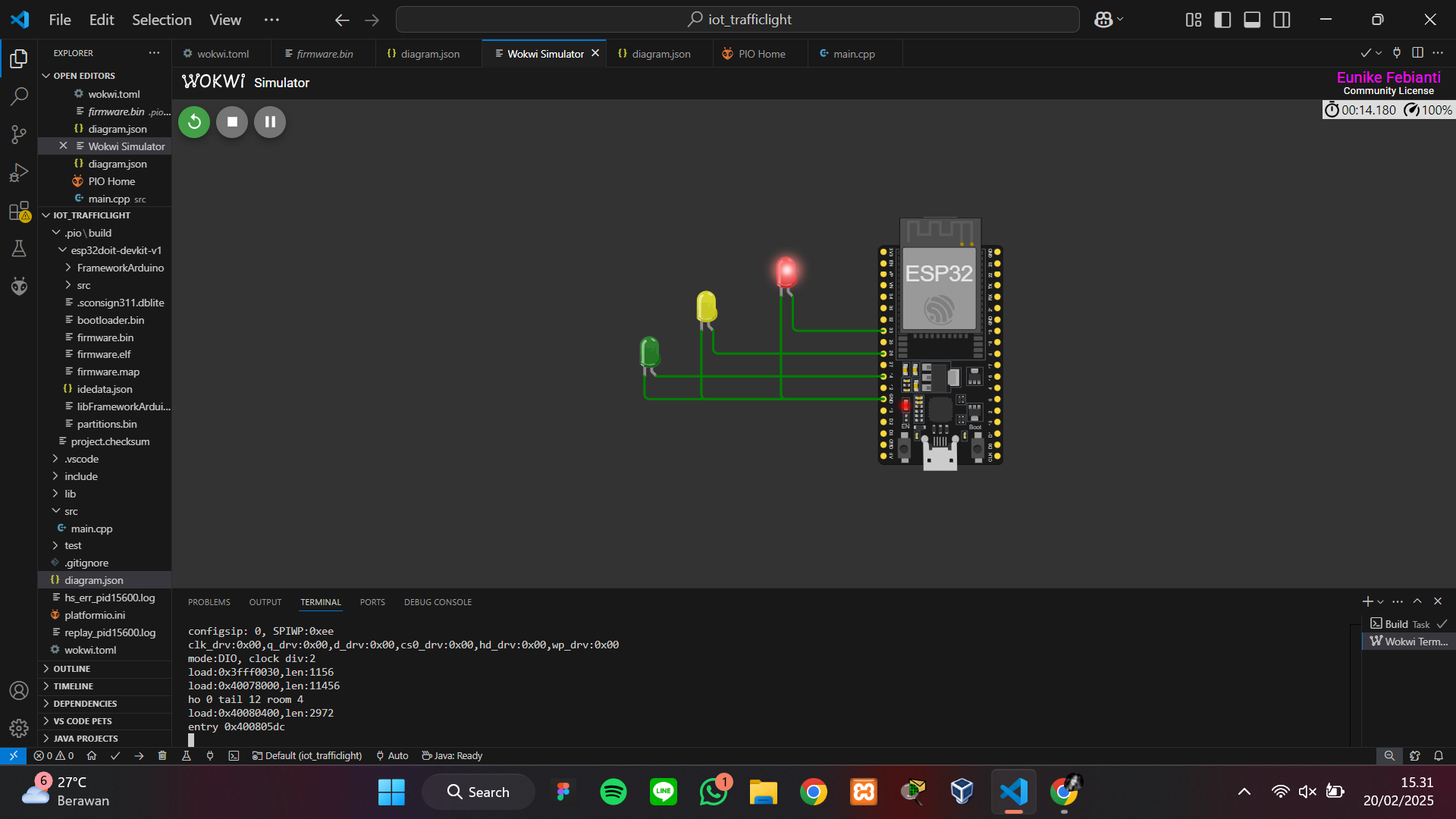
1. Jika sudah melakukan semua step coba compile dan run simulator



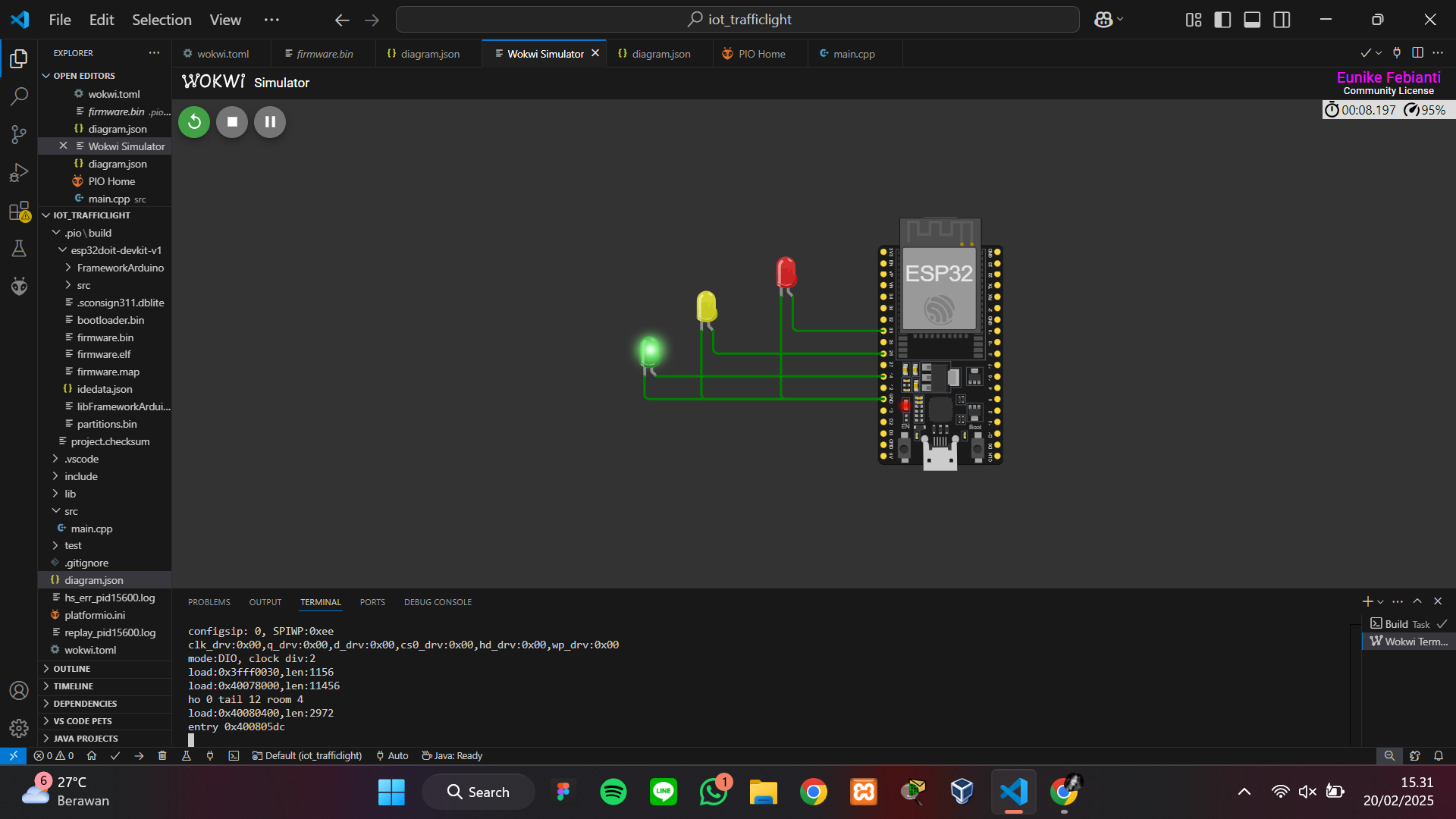
1. **Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

Hasil running simulator wokwi







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

Kode program akhir :

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

// Deklarasi pin LED

int lampu1 = 25;

int lampu2 = 33;

int lampu3 = 32;

void setup() {

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

    Serial.println("ESP32 Blinking LED");

    // Atur pin sebagai OUTPUT

    pinMode(lampu1, OUTPUT);

    pinMode(lampu2, OUTPUT);

    pinMode(lampu3, OUTPUT);

}

void loop() {

  digitalWrite(lampu1, HIGH);

  digitalWrite(lampu2, LOW);

  digitalWrite(lampu3, LOW);

  delay(1000);

  digitalWrite(lampu1, LOW);

  digitalWrite(lampu2, HIGH);

  digitalWrite(lampu3, LOW);

  delay(1000);

  digitalWrite(lampu1, LOW);

  digitalWrite(lampu2, LOW);

  digitalWrite(lampu3, HIGH);

  delay(1000);

}

**diagram.json**

{

    "version": 1,

    "author": "Anonymous maker",

    "editor": "wokwi",

    "parts": [

      { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },

      {

        "type": "wokwi-led",

        "id": "led1",

        "top": -22.8,

        "left": -92.2,

        "attrs": { "color": "red" }

      },

      {

        "type": "wokwi-led",

        "id": "led2",

        "top": 25.2,

        "left": -140.2,

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

      },

      {

        "type": "wokwi-led",

        "id": "led3",

        "top": 63.6,

        "left": -169,

        "attrs": { "color": "limegreen" }

      }

    ],

    "connections": [

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

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

      [ "led1:C", "esp:GND.1", "green", [ "v134.4", "h77.2" ] ],

      [ "led2:C", "esp:GND.1", "green", [ "v86.4", "h125.2" ] ],

      [ "led3:C", "esp:GND.1", "green", [ "v48", "h154" ] ],

      [ "led1:A", "esp:33", "green", [ "v76.8", "h67.2" ] ],

      [ "led2:A", "esp:26", "green", [ "v0" ] ],

      [ "led3:A", "esp:14", "green", [ "v28.8", "h144" ] ]

    ],

    "dependencies": {}

  }

**wokwi.toml**

[wokwi]

version = 1

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

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

**output**

