**LAPORAN PRAKTIKUM INTERNET OF THINGS**

**PRAKTIK PEMBUATAN LAMPU LALU LINTAS**

Fakultas Vokasi, Universitas Brawijaya

**Autors**

Fadhilah Azkia

Fakultas Vokasi, Universitas Brawijaya

Email : [fadhilahazkia95@gmail.com](mailto:fadhilahazkia95@gmail.com)

**Abstract**

Percoobaan ini bertujuan untuk merancang dan mengimplementasikan sistem lampu lalu lintas menggunakan mikrokontroler ESP32. Sistem ini terdiri dari tiga lampu LED yang mewakili lampu merah, kuning, dan lampu hijau. Mikrokontroler diprogram untuk mengatur waktu dari setiap lampu yang ada sesuai dengan kinerja atau sistematis lampu lalu lintas. Praktik ini sangat efektif untuk mengetahui bagaimana lampu lalu lintas itu bekerja secara efektif dengan waktu yang terstruktur.

1. **Introduction**  
   1. **Latar Belakang**  
        
      Lampu lalu lintas merupakan sistem yang sangat berguna untuk mengatur aliran kendaraan di persimpangan jalan. Sistem ini dirancang dengan baik dan efektif sesuai dengan waktu yang mempengaruhi kapan lampu menyala, dari lampu satu ke lampu lainnya. Eksperimen ini mensimulasikan bagaimana sistematis dari lampu lalulintas menggunakan mikrokontroler ESP32.
   2. **Tujuan Eksperimen**

* Membuat lampu lalu lintas menggunakan mikrokontroler.
* Mampu mengetahui sistematis dari lampu lalu lintas.

1. **Metodologi**
   1. **Alat dan Bahan**

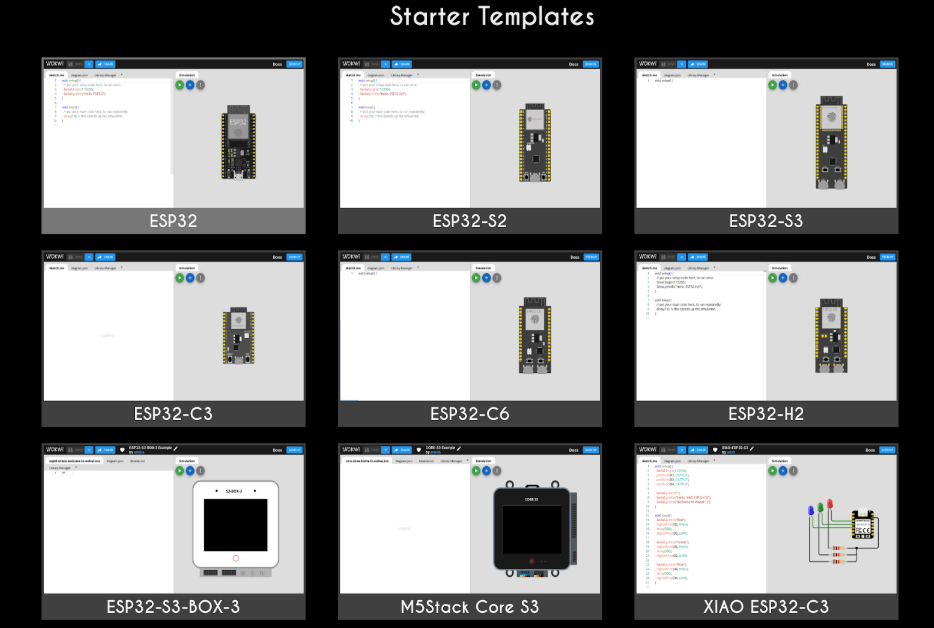
* Hardware : Laptop
* Wokwi online : <https://wokwi.com/>
* Visual Studio Code
  1. **Langkah Implementasi**

1. Membuka wokwi online.
2. Pilih simulasi mengggunakan ‘ESP32’



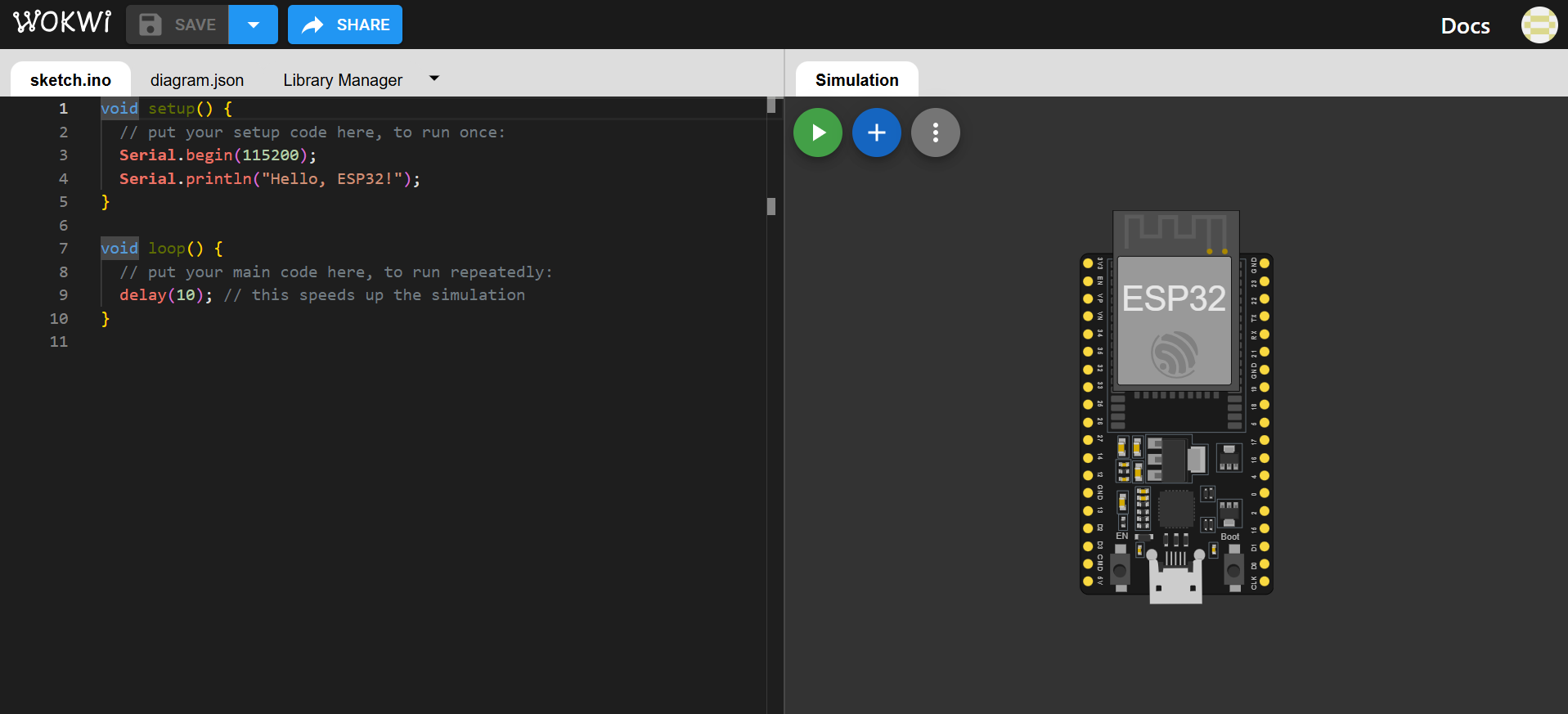
**Gambar 2.2.1** Home page Wokwi online ( wokwi.com ).

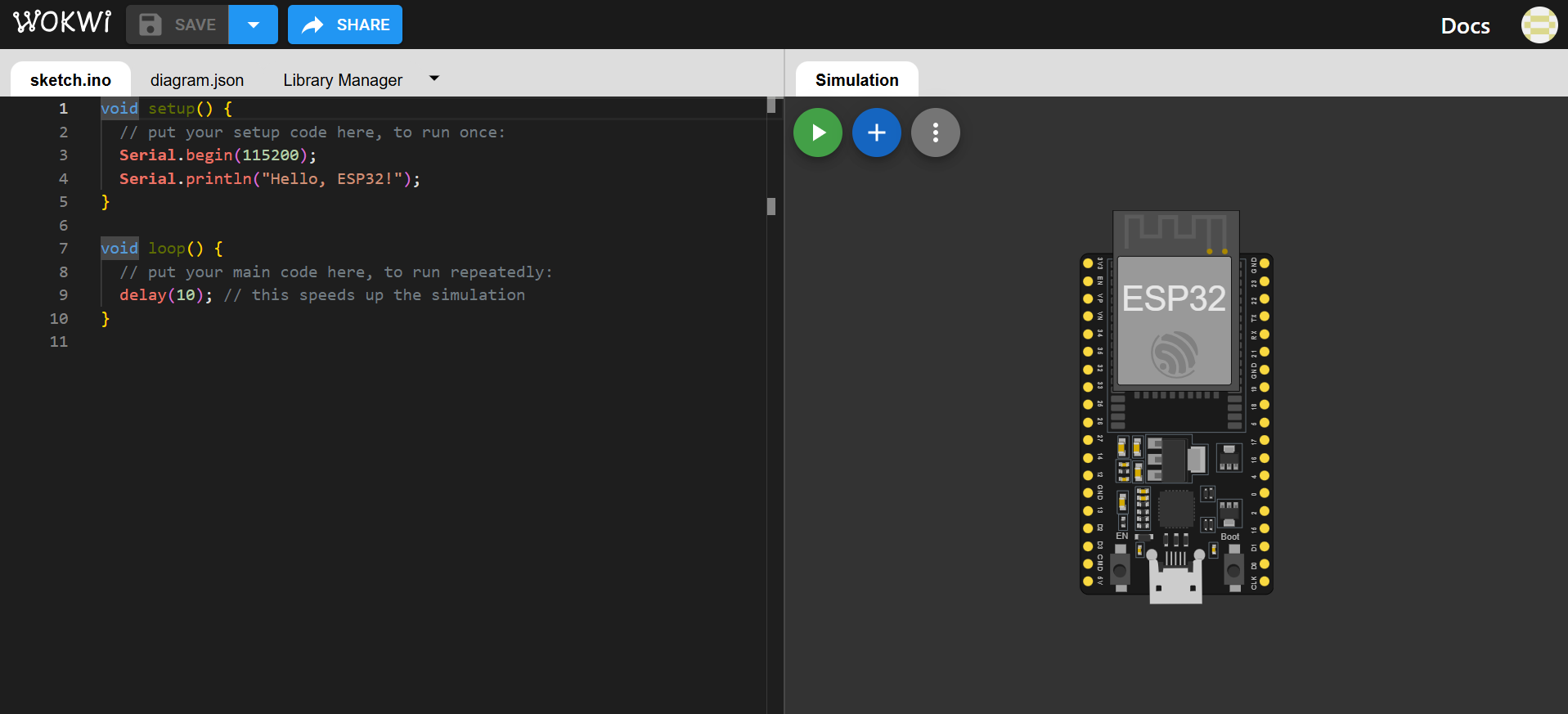
1. Selanjutnya memilih starter templates ‘ESP32’



**Gambar 2.2.2** Halaman starter templates.

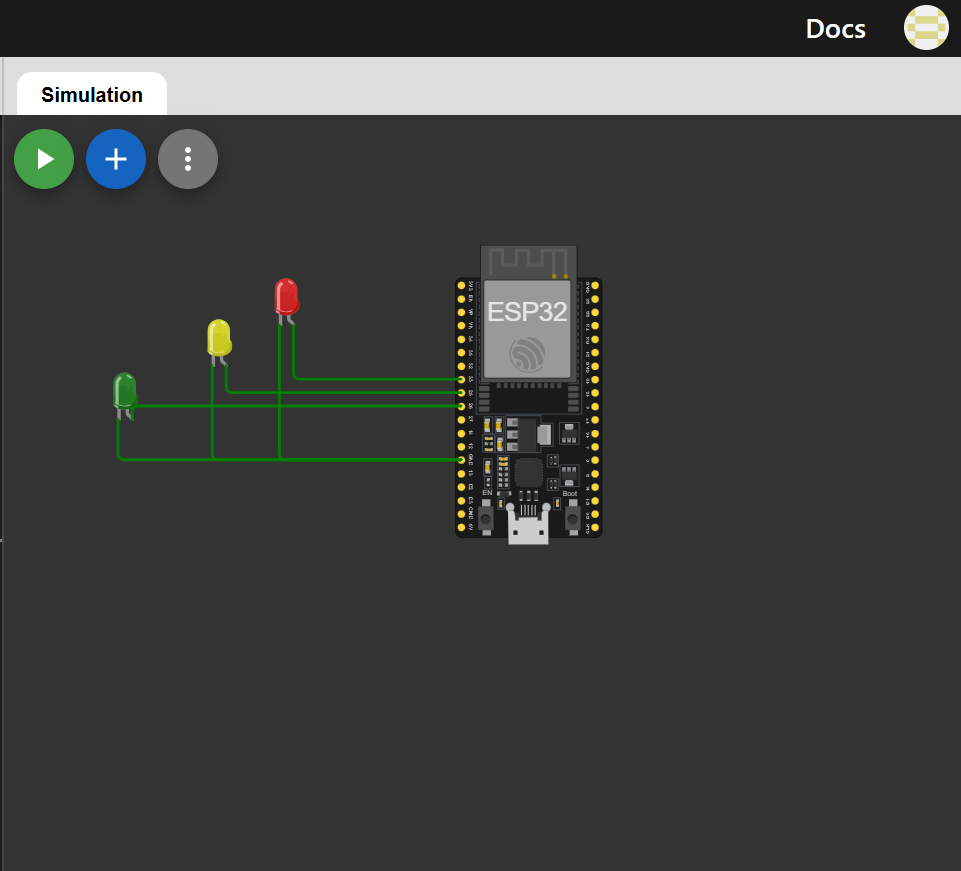
1. Tampilannya akan seperti yang ada di bawah ini.





**Gambar 2.2.3** Halaman project wokwi online.

1. Menambahkan Lampu LED sesuai dengan yang kita butuhkan ( merah, kuning, hijau ) dengan menekan ikon ‘+’ berwarna biru pada halaman.



**Gambar 2.2.4** Gambar device setelah di pasangkan.

1. Memasukkan source code sesuai dengan yang dibutukan pada main.cpp.

#include <Arduino.h>

int lampu = 33;

int lampu2 = 25;

int lampu3 = 26;

void setup() {

**Serial**.begin(115200);

**Serial**.println("ESP32 Blinking LED");

    pinMode(lampu, OUTPUT);

    pinMode(lampu2, OUTPUT);

    pinMode(lampu3, OUTPUT);

}

void loop() {

    digitalWrite(lampu, HIGH);

    digitalWrite(lampu2, LOW);

    digitalWrite(lampu3, LOW);

**Serial**.println("LED 1 ON");

    delay(3000);

    digitalWrite(lampu, LOW);

    digitalWrite(lampu2, HIGH);

    digitalWrite(lampu3, LOW);

**Serial**.println("LED 2 ON");

    delay(1000);

    digitalWrite(lampu, LOW);

    digitalWrite(lampu2, LOW);

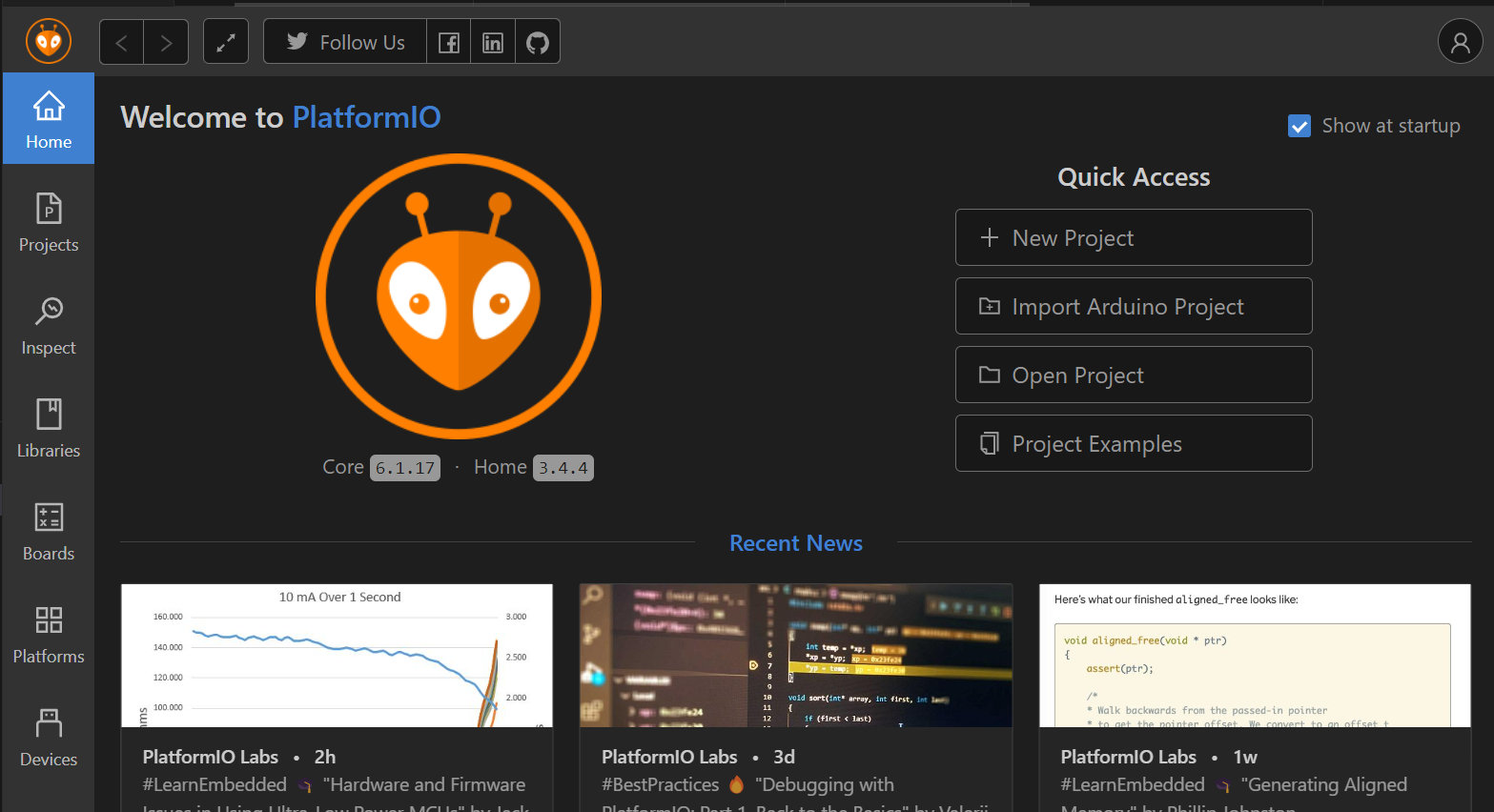
    digitalWrite(lampu3, HIGH);

**Serial**.println("LED 3 ON");

    delay(3000);

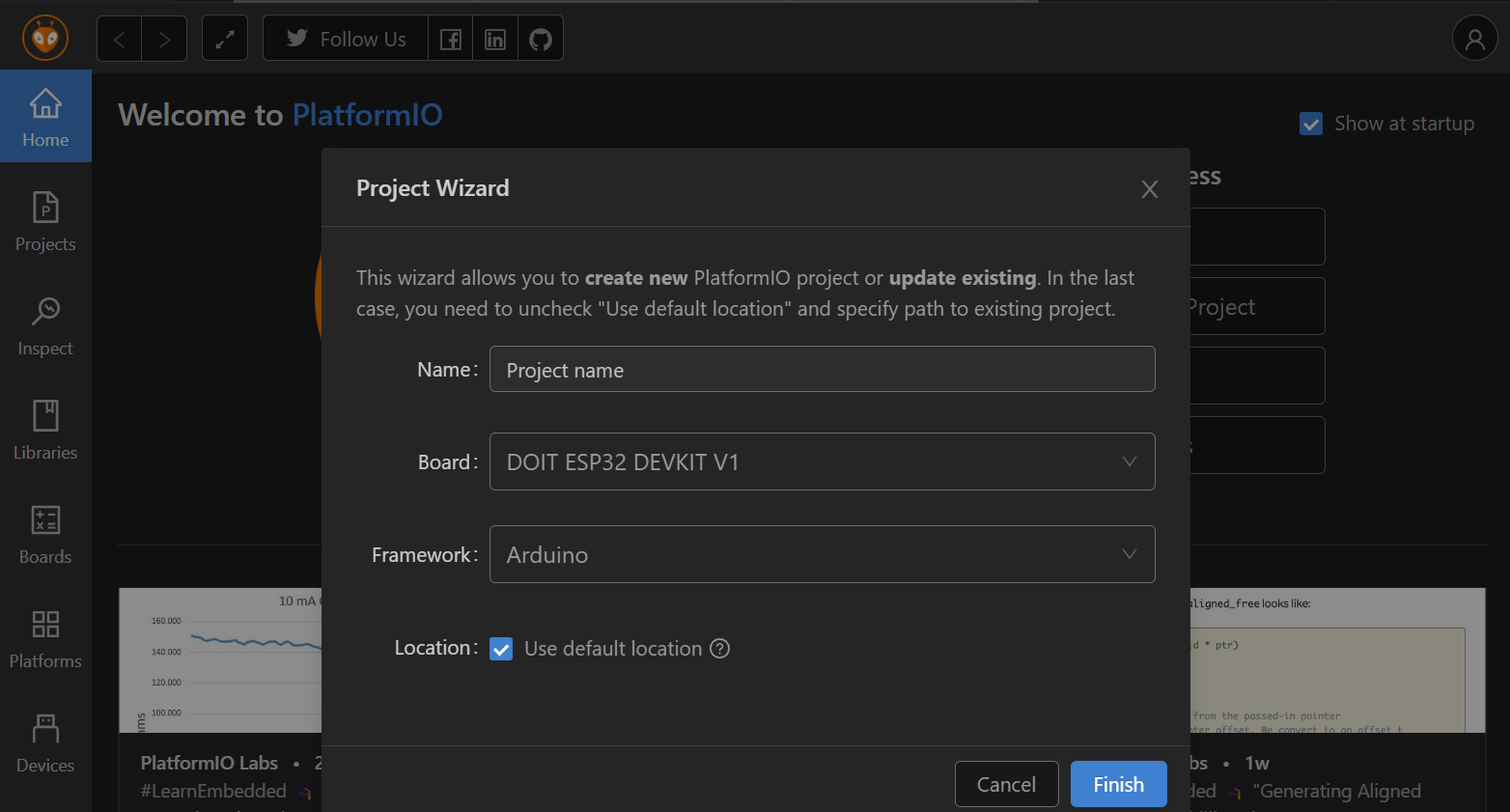
}

1. Selanjutnya masuk ke aplikasi Visual Studio Code, dikarenakan akun wokwi gratis memiliki traffic yang sangat besar sehingga untuk menjalankan project kita membutuhkan waktu yang sangat lama.
2. Membuat project baru di Visual Studio Code dengan extensi PlatformIO.



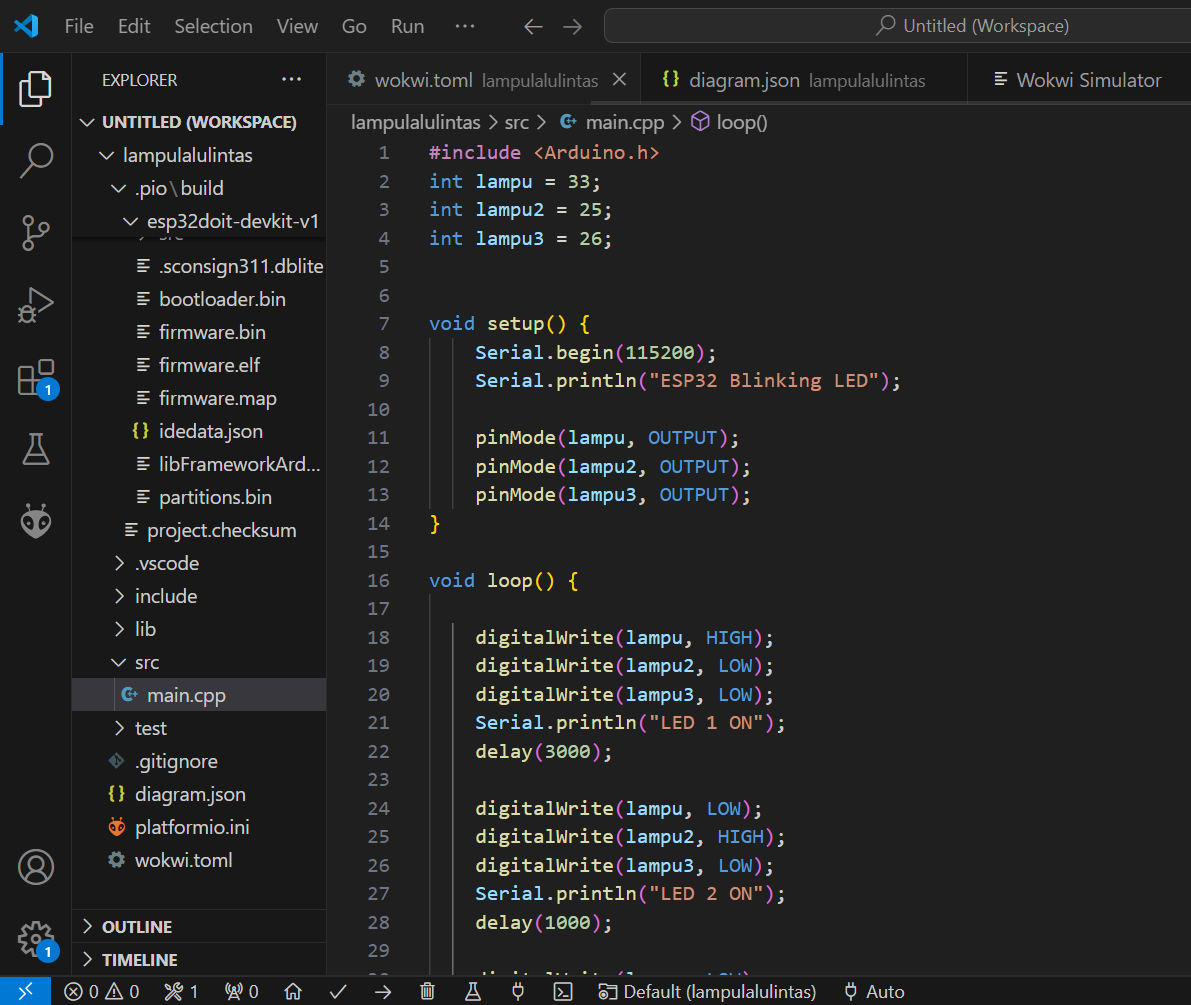
**Gambar 2.2.5** Halaman utama PlatformIO.

1. Memasukkan nama project, memilih board ‘DOIT ESP32 DEVKIT V1’ lalu finish.

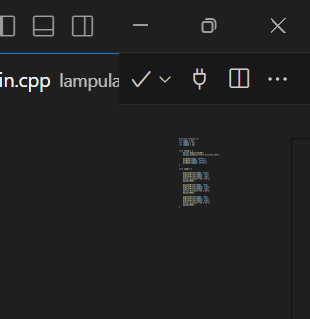


**Gambar 2.2.6** Opsi memilih board dan framework.

1. Pada folder yang sudah dibuat, cari folder src->file main.cpp kemudian salin source code yang tadi sudah dibuat pada wokwi online.



**Gambar 2.2.7** Main.cpp



Vscode akan melakukan proses kompilasi seperti tampilan berikut.tunggu hingga succes.

1. Setelah proses compiling berhasil. anda mendaparkan 2 file penting yang akan dipakai pada proses simulasi. Perhatikan ada file **firmware.bin** dan **firmware.elf**

File ini akan muncul setelah dibuild.

1. Buatlah file wokwi.toml

Isikan dengan source code ini.

[wokwi]

version = 1

firmware = 'path-to-your-firmware.hex'

elf = 'path-to-your-firmware.elf'

1. Gantikan firmware dan elf sesuai Relative Path.
2. Membuat diagram.json kemudian salin dari wokwi online juga.

{

    "version": 1,

    "author": "Anonymous maker",

    "editor": "wokwi",

    "parts": [

      {

        "type": "board-esp32-devkit-c-v4",

        "id": "esp",

        "top": 0,

        "left": 0,

        "attrs": { "builder": "esp-idf" }

      },

      {

        "type": "wokwi-led",

        "id": "led1",

        "top": 15.6,

        "left": -140.2,

        "attrs": { "color": "red" }

      },

      {

        "type": "wokwi-led",

        "id": "led2",

        "top": 44.4,

        "left": -188.2,

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

      },

      {

        "type": "wokwi-led",

        "id": "led3",

        "top": 82.8,

        "left": -255.4,

        "attrs": { "color": "green" }

      }

    ],

    "connections": [

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

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

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

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

      [ "led2:A", "esp:25", "green", [ "v0" ] ],

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

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

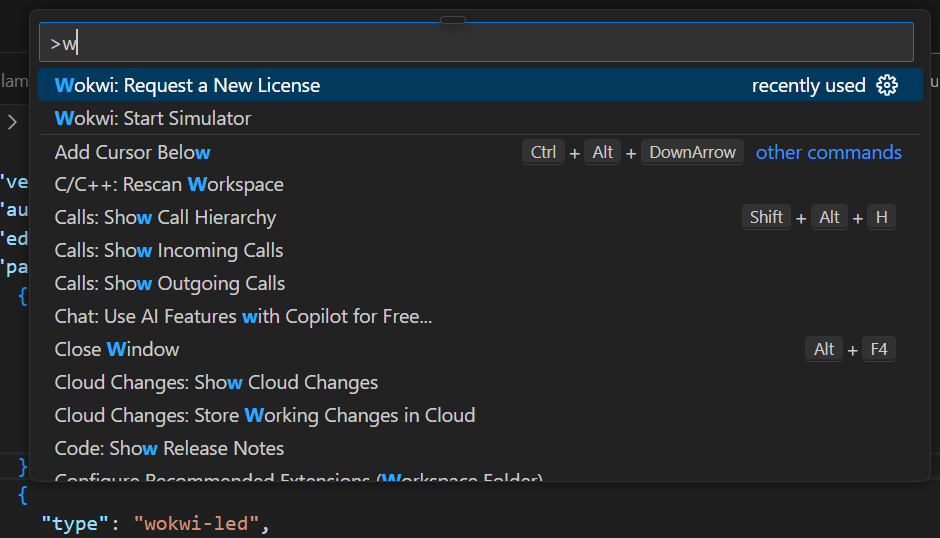
      [ "led3:C", "esp:GND.1", "green", [ "v0" ] ]

    ],

    "dependencies": {}

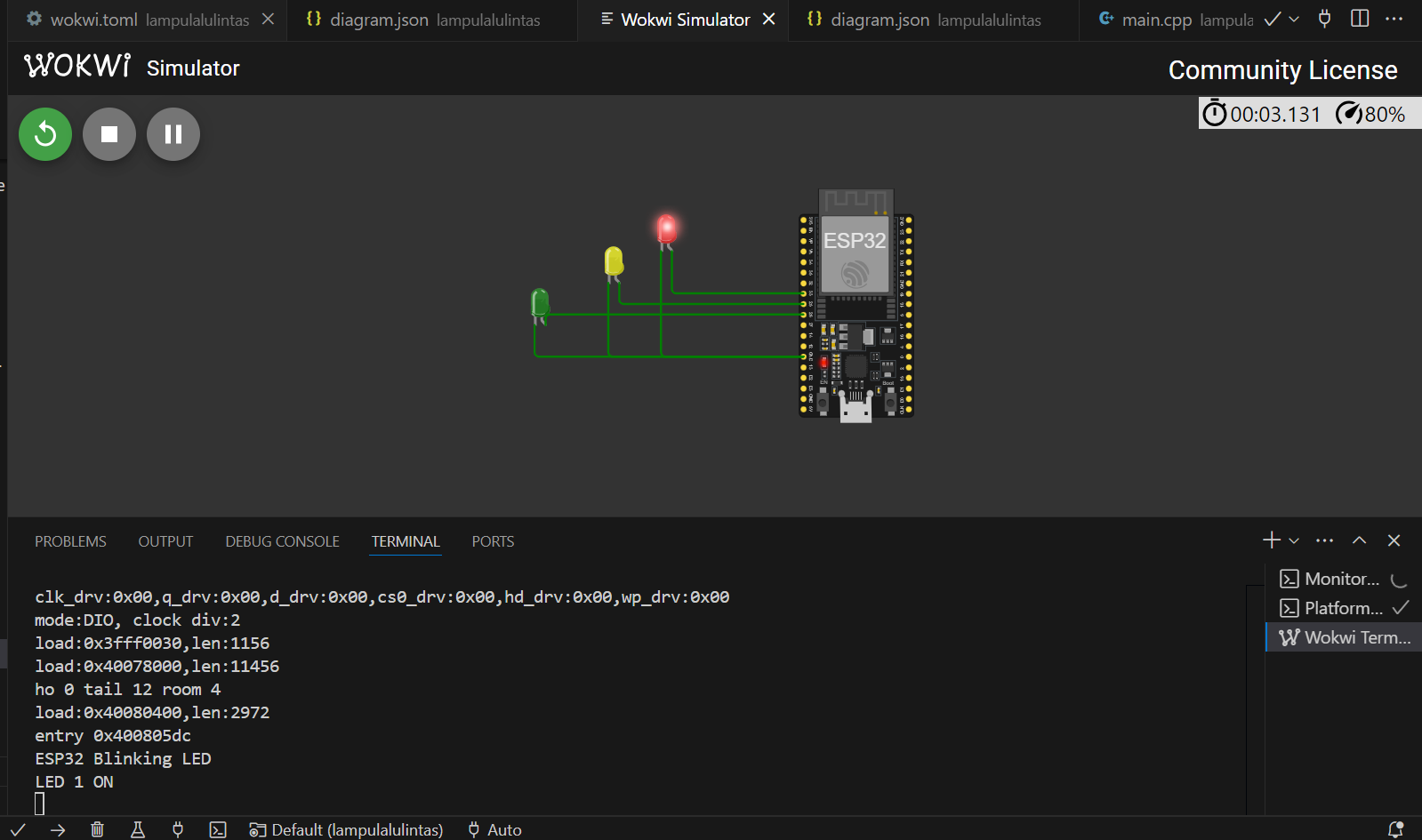
  }

Sebelum memulai simulasi, pastikan telah malakukan request a new license dengan menjalankan perintah **> Wokwi: Request a New License**



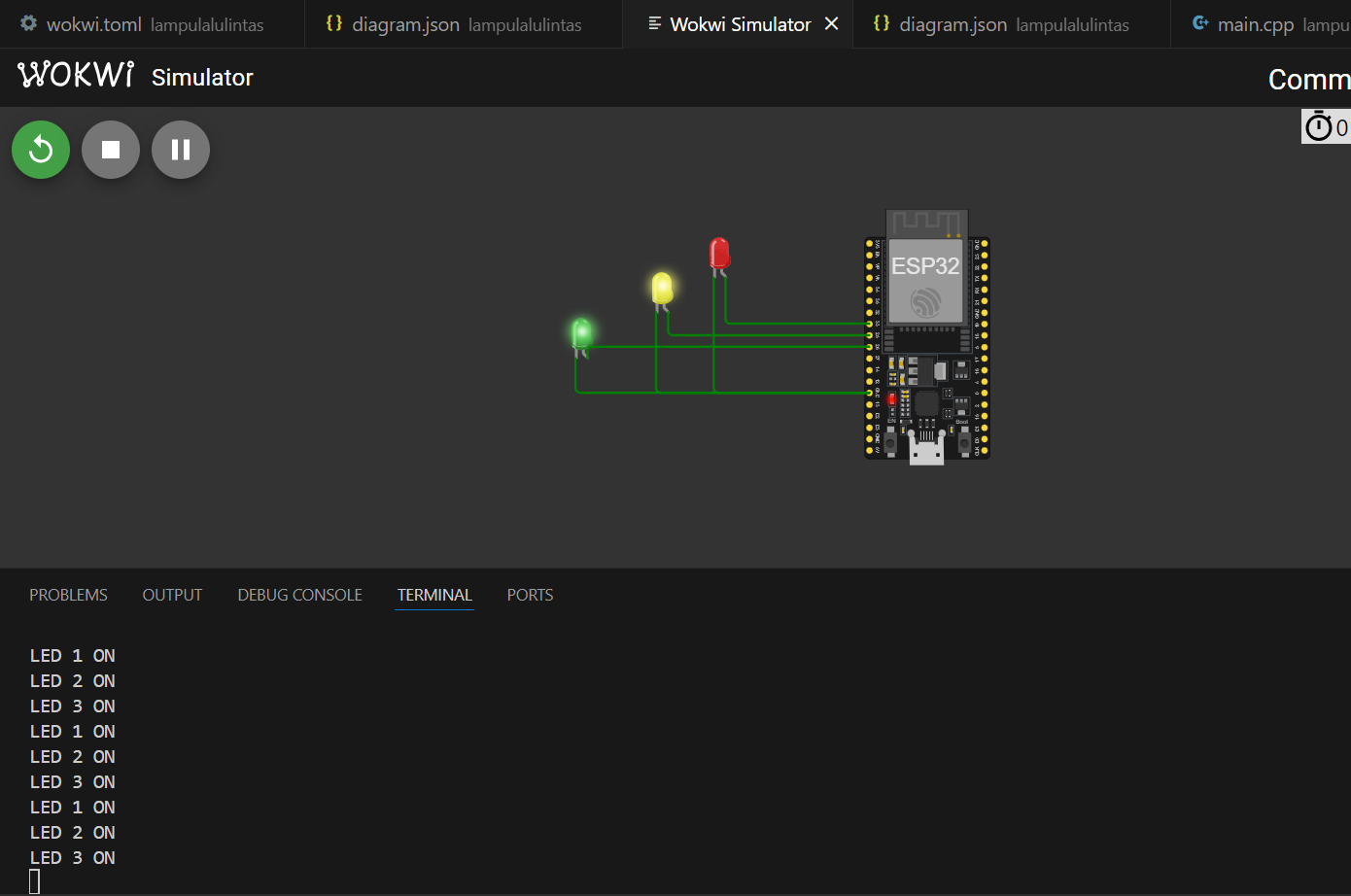
**Gambar 2.2.7** Request License.

1. Jalankan simulasinya dengan memilih opsi ‘Start Simulator’



**Gambar 2.2.8** Gambar hasil simulasi.

1. **Hasil dan Pembahasan**
   1. **Hasil Eksperimen**

****

1. **Appendix**

Main.cpp

#include <Arduino.h>

int lampu = 33;

int lampu2 = 25;

int lampu3 = 26;

void setup() {

    Serial.begin(115200);

    Serial.println("ESP32 Blinking LED");

    pinMode(lampu, OUTPUT);

    pinMode(lampu2, OUTPUT);

    pinMode(lampu3, OUTPUT);

}

void loop() {

    digitalWrite(lampu, HIGH);

    digitalWrite(lampu2, LOW);

    digitalWrite(lampu3, LOW);

    Serial.println("LED 1 ON");

    delay(3000);

    digitalWrite(lampu, LOW);

    digitalWrite(lampu2, HIGH);

    digitalWrite(lampu3, LOW);

    Serial.println("LED 2 ON");

    delay(1000);

    digitalWrite(lampu, LOW);

    digitalWrite(lampu2, LOW);

    digitalWrite(lampu3, HIGH);

    Serial.println("LED 3 ON");

    delay(3000);

}

Diagram.json

{

    "version": 1,

    "author": "Anonymous maker",

    "editor": "wokwi",

    "parts": [

      {

        "type": "board-esp32-devkit-c-v4",

        "id": "esp",

        "top": 0,

        "left": 0,

        "attrs": { "builder": "esp-idf" }

      },

      {

        "type": "wokwi-led",

        "id": "led1",

        "top": 15.6,

        "left": -140.2,

        "attrs": { "color": "red" }

      },

      {

        "type": "wokwi-led",

        "id": "led2",

        "top": 44.4,

        "left": -188.2,

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

      },

      {

        "type": "wokwi-led",

        "id": "led3",

        "top": 82.8,

        "left": -255.4,

        "attrs": { "color": "green" }

      }

    ],

    "connections": [

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

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

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

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

      [ "led2:A", "esp:25", "green", [ "v0" ] ],

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

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

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

    ],

    "dependencies": {}

  }

Wokwi.toml

[wokwi]

version = 1

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

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