

**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)  
PRAKTIK PENGENDALIAN LAMPU LED**



*Nadia Aulia Zahra*  
*Fakultas Vokasi, Universitas Brawijaya*  
*Email: [nadiaaulia@student.ub.ac.id](mailto:nadiaaulia@student.ub.ac.id)*

**Fakultas Vokasi  
Universitas Brawijaya  
Tahun Ajaran 2025**

## **ABSTRAK**

Eksperimen ini bertujuan untuk memahami prinsip dasar pengendalian lampu LED menggunakan rangkaian elektronik sederhana. LED (Light Emitting Diode) merupakan komponen elektronik yang banyak digunakan sebagai indikator atau penerangan. Dalam percobaan ini, LED dikendalikan melalui sumber tegangan yang diatur menggunakan komponen seperti resistor dan saklar. Hasil eksperimen menunjukkan bahwa LED hanya akan menyala apabila mendapatkan arus dalam polaritas dan besaran yang sesuai.

**Kata Kunci:** *LED, resistor, arus listrik, pengendalian, rangkaian elektronik, saklar.*

## **1. Pendahuluan**

### **1.1. Latar Belakang**

LED merupakan salah satu komponen penting dalam dunia elektronika. Dibandingkan lampu pijar konvensional, LED lebih hemat energi dan tahan lama. Dalam dunia praktikum atau proyek, LED juga sering digunakan untuk menunjukkan status sistem (ON/OFF, aktif/non-aktif). Oleh karena itu, penting bagi pelajar dan praktisi untuk memahami cara kerja dan metode pengendalian LED, baik secara manual maupun otomatis. Eksperimen ini memberikan gambaran dasar tentang bagaimana LED dapat dikendalikan menggunakan komponen sederhana dan prinsip dasar arus listrik.

### **1.2. Tujuan Eksperimen**

1. Mengetahui cara kerja dasar LED dalam rangkaian elektronik.
2. Mempelajari pengaruh resistor terhadap arus dan tegangan pada LED.
3. Memahami konsep pengendalian ON/OFF lampu LED secara manual.

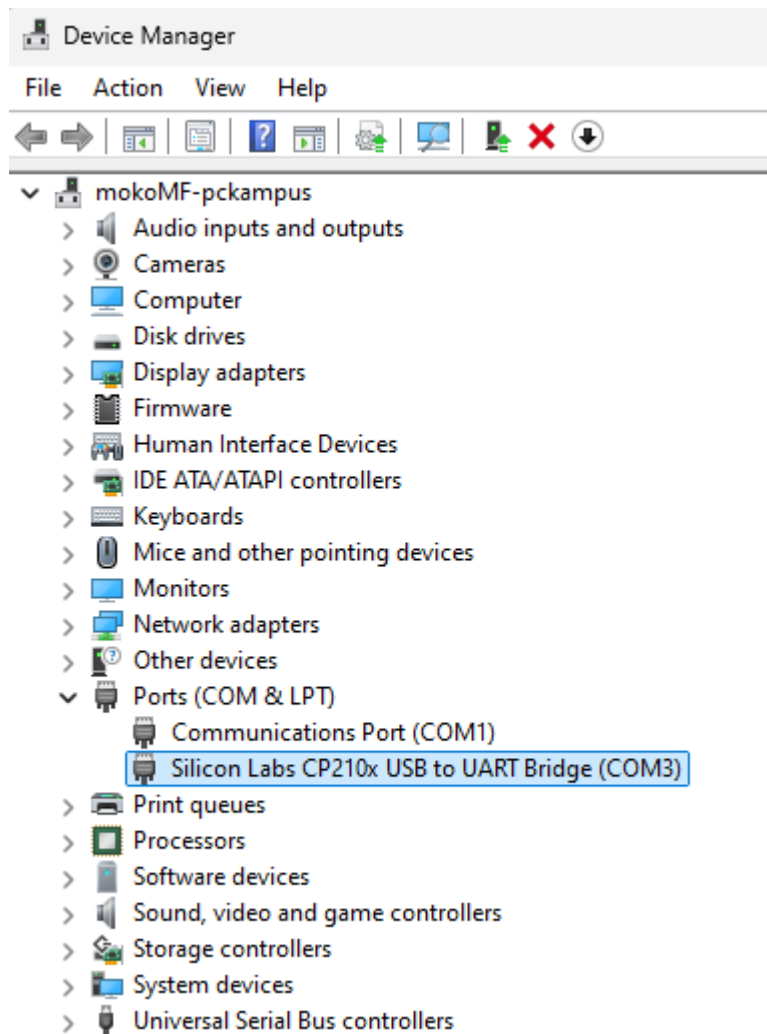
## **2. Metodologi**

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

1. Breadboard
2. 1 buah LED (merah/hijau/biru)
3. 1 resistor 220 ohm
4. Kabel jumper
5. Power supply 5V (atau baterai 9V + voltage regulator)
6. Saklar (opsional)
7. Multimeter (untuk pengukuran)

### **2.2. Langkah Implementasi**

Pastikan Hardware ESP32 dikenali oleh komputer



Perhatikan bagian Ports (COM & LPT) harus muncul **Silicon Labs**..... Jika belum muncul harus dilakukan proses instalasi driver secara manual mengikuti tutorial sebagai berikut :

Download Driver Silicon Labs CP210x di alamat :

<https://www.silabs.com/developer-tools/usb-to-uart-bridge0-vcp-drivers?tab=downloads>

[OVERVIEW](#) [DOWNLOADS](#) [TECH DOCS](#) [COMMUNITY & SUPPORT](#)

## Download and Install VCP Drivers

Downloads for Windows, Macintosh, Linux and Android below.

\*Note: The Linux 3.x.x and 4.x.x version of the driver is maintained in the current Linux 3.x.x and 4.x.x tree at [www.kernel.org](http://www.kernel.org).

## Software Downloads

[Software](#) (11)

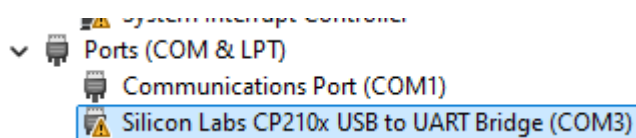
Software • 11

[CP210x Universal Windows Driver](#)


v11.4.0

12/18/2024

Klik kanan Update Driver



Pilih Browse my computer for drivers

←  Update Drivers - Silicon Labs CP210x USB to UART Bridge (COM3)

How do you want to search for drivers?

→ Search automatically for drivers


Windows will search your computer for the best available driver and install it on your device.

→ Browse my computer for drivers

Locate and install a driver manually.

Cancel

Pilih ***Let me pick from a list of available drivers on my computer***

←  Update Drivers - Silicon Labs CP210x USB to UART Bridge (COM3)

Browse for drivers on your computer

Search for drivers in this location:

D:\driveresp32

Browse...

☒ Include subfolders

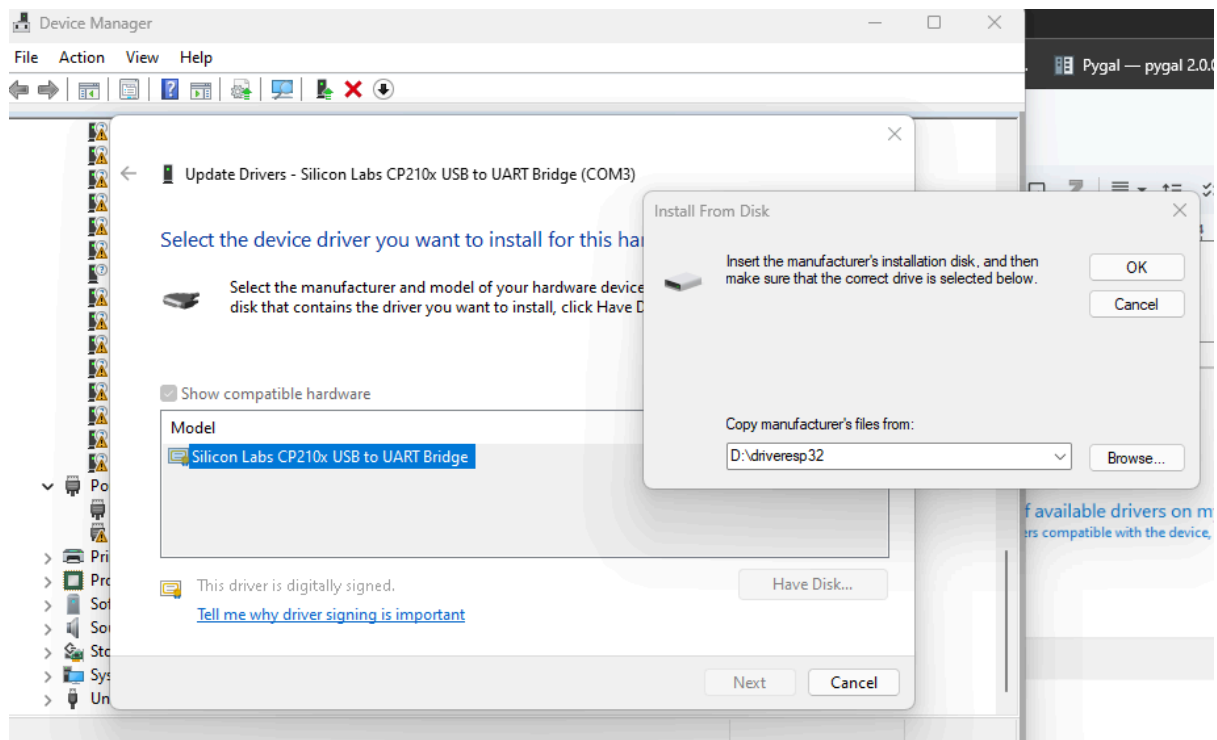
→ Let me pick from a list of available drivers on my computer

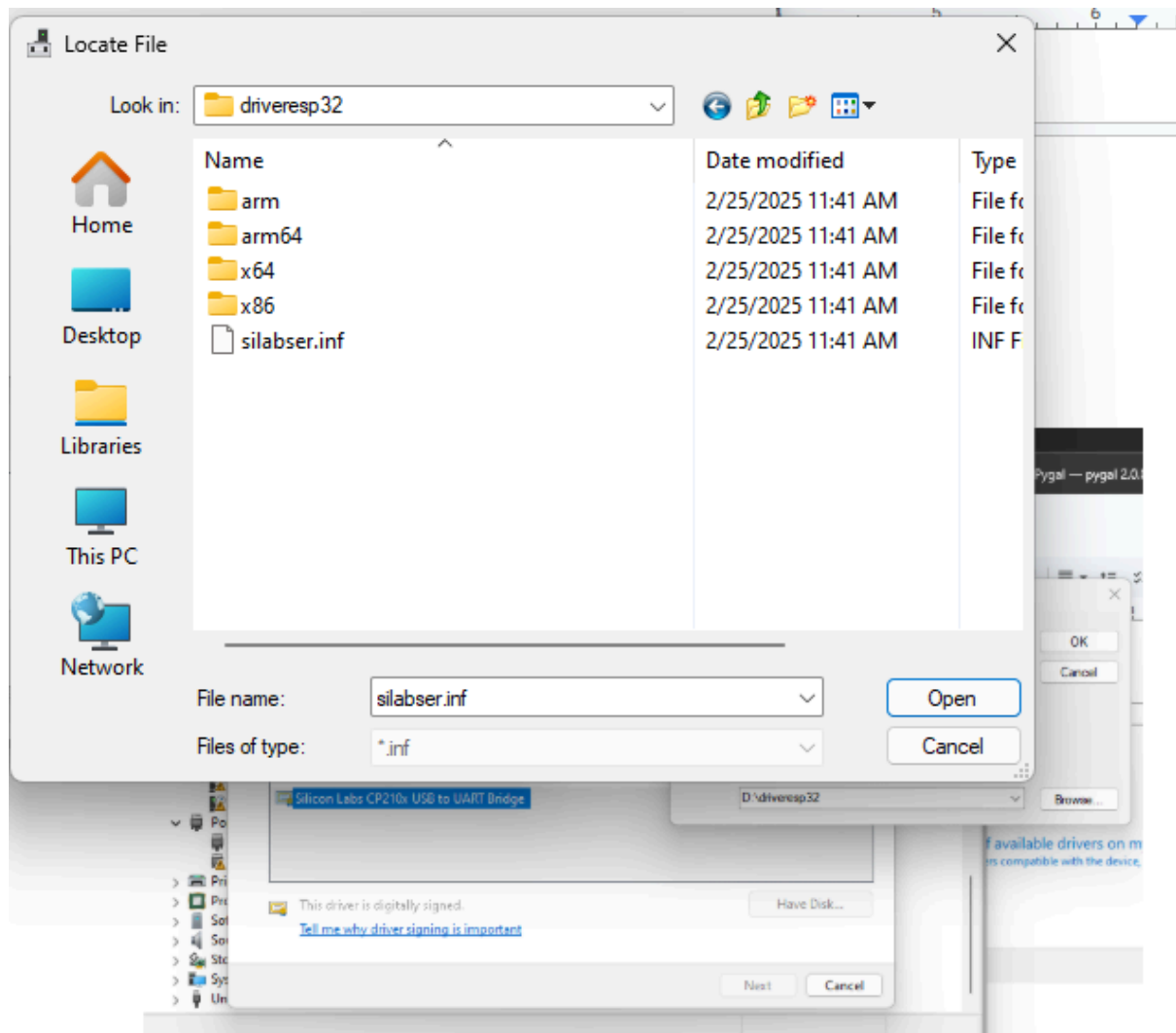
This list will show available drivers compatible with the device, and all drivers in the same category as the device.

Next

Cancel

Klik menu **Browse**, arahkan ke folder driver yang telah di download (harus di extract folder)





Setelah dipilih, klik next dan proses instalasi akan berjalan. Pastikan tampilan device manager tidak menunjukkan masalah dan hardware ESP32 terkenal dengan baik seperti tampilan berikut :

- ▼ Ports (COM & LPT)
  - Communications Port (COM1)
  - Silicon Labs CP210x USB to UART Bridge (COM3)

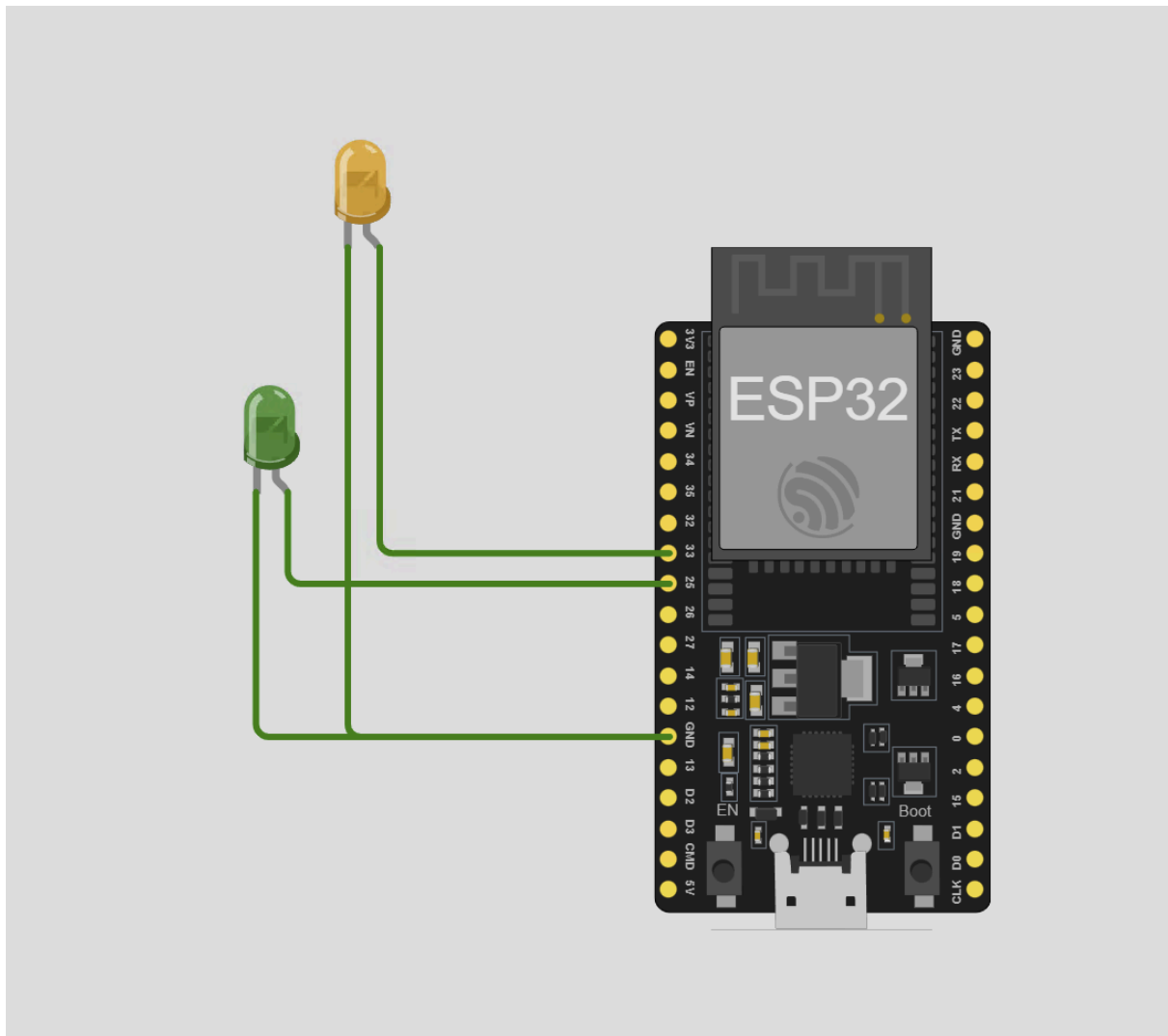
**Video Instalasi dapat dilihat di**  
[https://www.youtube.com/watch?v=r\\_eMEXvt0v0](https://www.youtube.com/watch?v=r_eMEXvt0v0)

Setelah memastikan hardware ESP32 terkenal, berikutnya adalah mulai melakukan koding dan upload ke hardware. Untuk melakukan upload digunakan library platform.io yang telah digunakan pada bab sebelumnya.



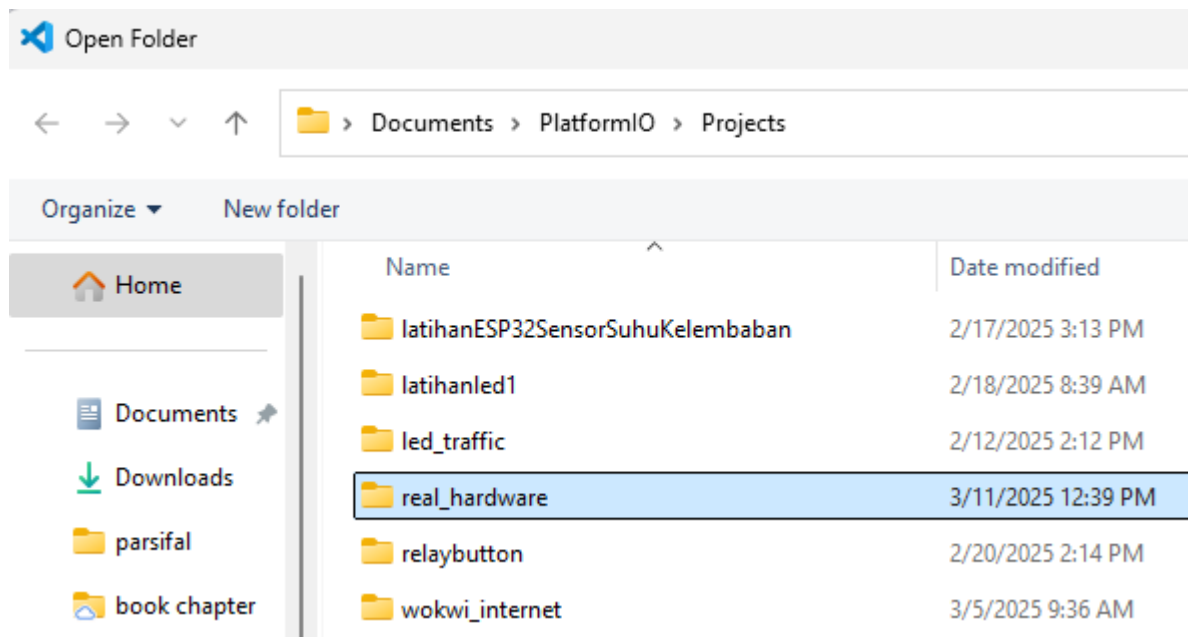
Pada praktik ini akan dilakukan percobaan yang pernah dilakukan pada Bab 8 yaitu pengendalian lampu LED.

Lakukan prosedur wiring kabel dan sensor sesuai dengan diagram yang telah dibuat berikut ini :

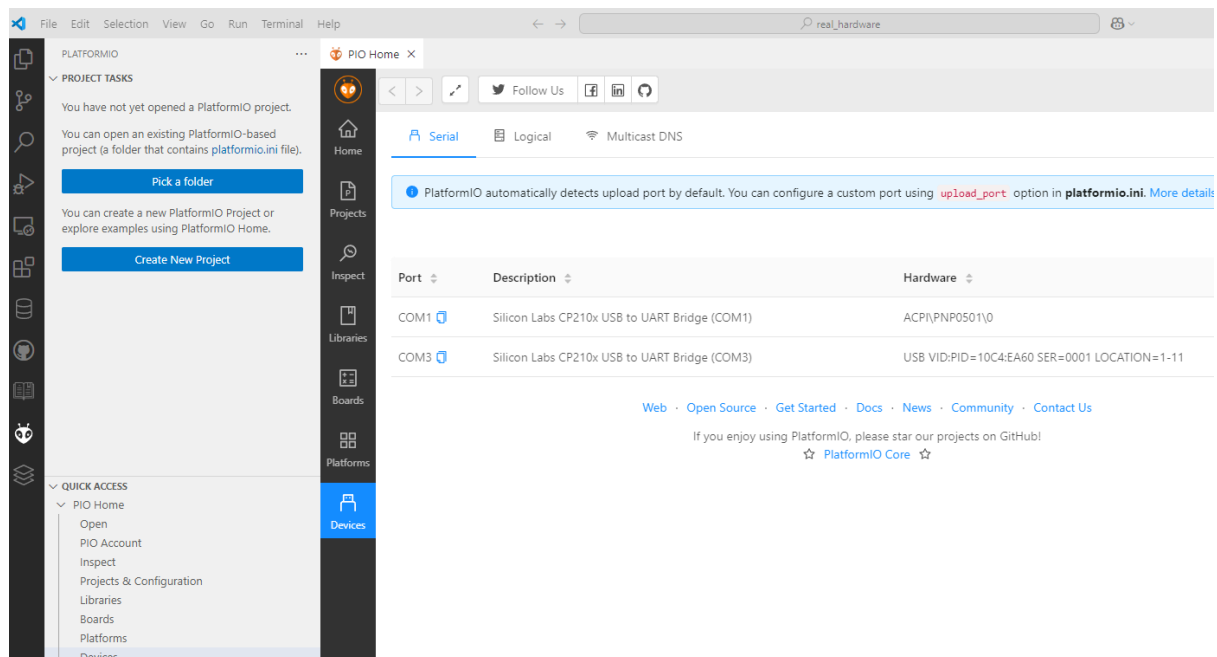


Kemudian lakukan koding di platform io seperti yang pernah dilakukan pada bab 8 dengan tahapan :

Buat folder baru dan buka di vscode

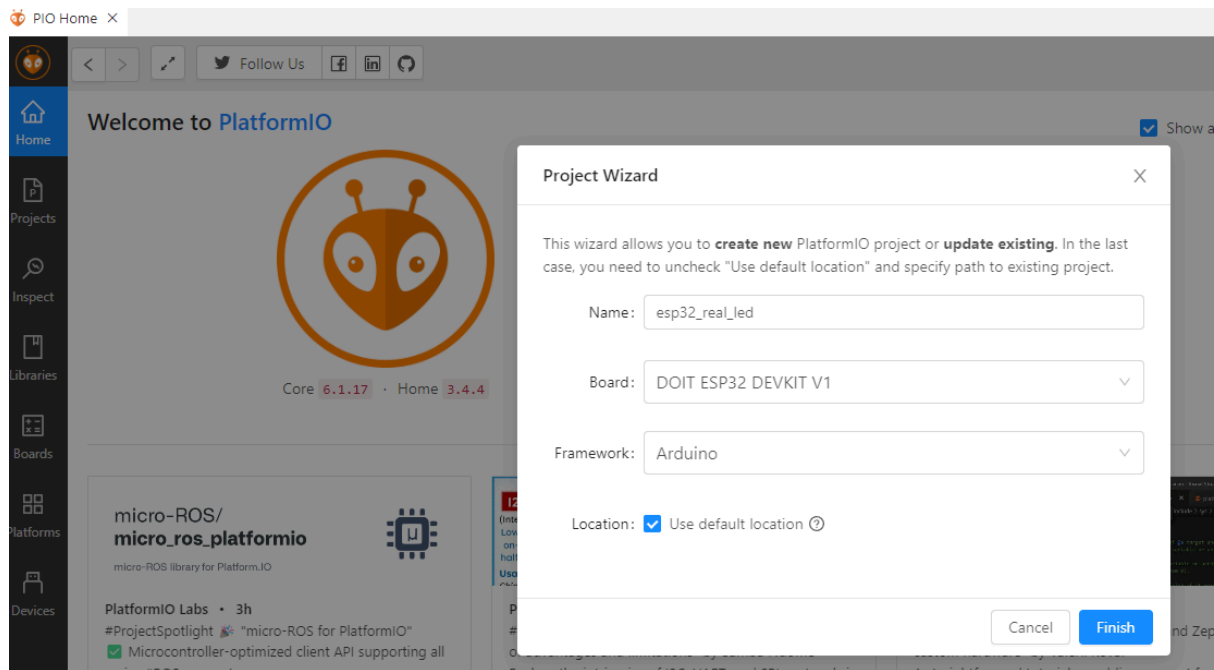


Pastikan device ESP32 muncul pada platform.io sebagai berikut



COM3 adalah alamat device ESP32 yang terkoneksi ke laptop/komputer.

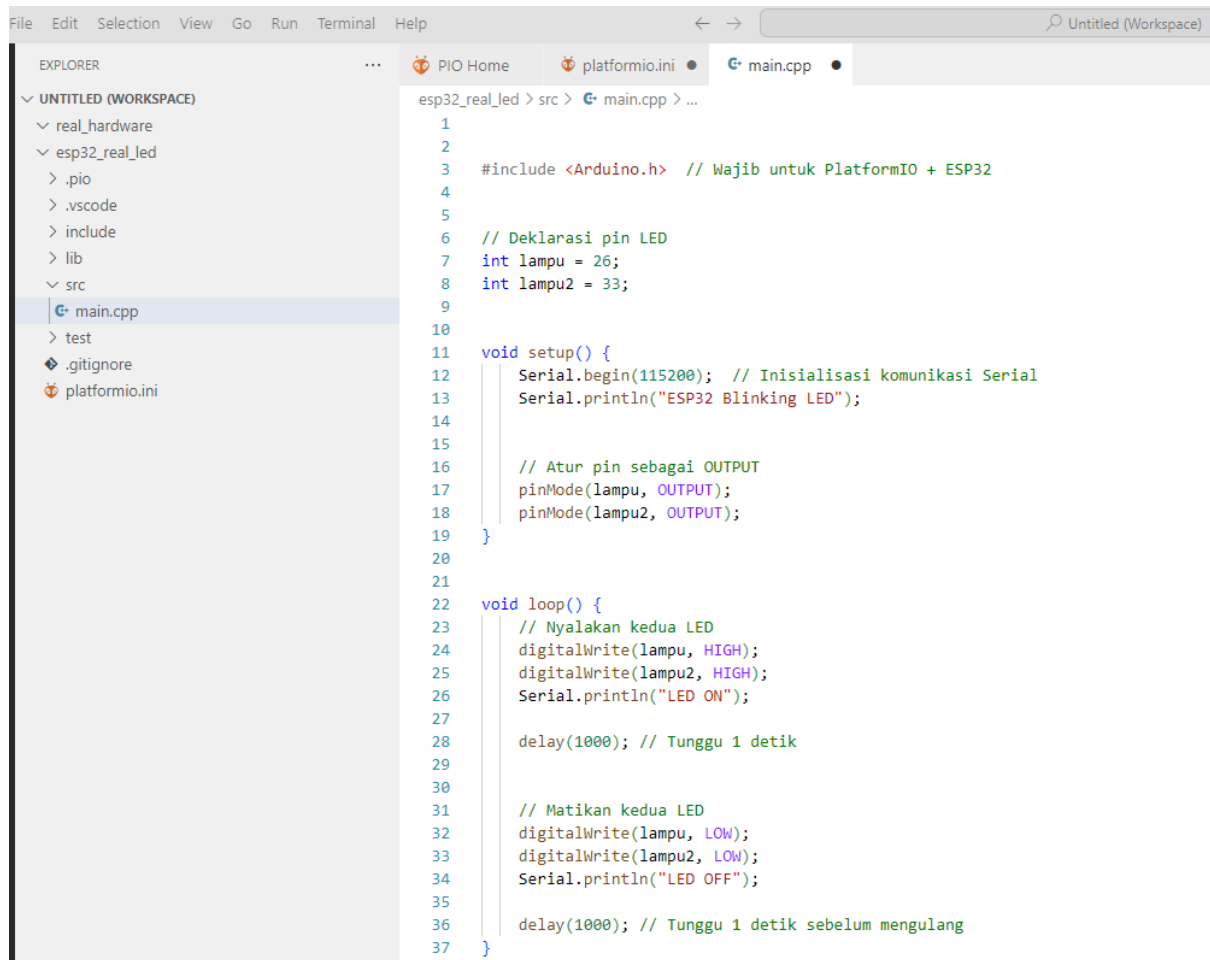
Buat project baru di platformio



Ubah file platformio.ini dan modifikasi sebagai berikut

```
[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
upload_port = COM3
monitor_port = COM3
```

Kemudian pada file main.cpp masukkan koding lampu LED yang telah dibuat di bab 8



```
1
2
3 #include <Arduino.h> // Wajib untuk PlatformIO + ESP32
4
5
6 // Deklarasi pin LED
7 int lampu = 26;
8 int lampu2 = 33;
9
10
11 void setup() {
12     Serial.begin(115200); // Inisialisasi komunikasi Serial
13     Serial.println("ESP32 Blinking LED");
14
15
16     // Atur pin sebagai OUTPUT
17     pinMode(lampu, OUTPUT);
18     pinMode(lampu2, OUTPUT);
19 }
20
21
22 void loop() {
23     // Nyalakan kedua LED
24     digitalWrite(lampu, HIGH);
25     digitalWrite(lampu2, HIGH);
26     Serial.println("LED ON");
27
28     delay(1000); // Tunggu 1 detik
29
30
31     // Matikan kedua LED
32     digitalWrite(lampu, LOW);
33     digitalWrite(lampu2, LOW);
34     Serial.println("LED OFF");
35
36     delay(1000); // Tunggu 1 detik sebelum mengulang
37 }
```

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

// Deklarasi pin LED
int lampu = 25;
int lampu2 = 26;

void setup() {
    Serial.begin(115200); // Inisialisasi komunikasi Serial
    Serial.println("ESP32 Blinking LED");

    // Atur pin sebagai OUTPUT
    pinMode(lampu, OUTPUT);
    pinMode(lampu2, OUTPUT);
}
```

```

void loop() {
    // Nyalakan kedua LED
    digitalWrite(lampu, HIGH);
    digitalWrite(lampu2, HIGH);
    Serial.println("LED ON");

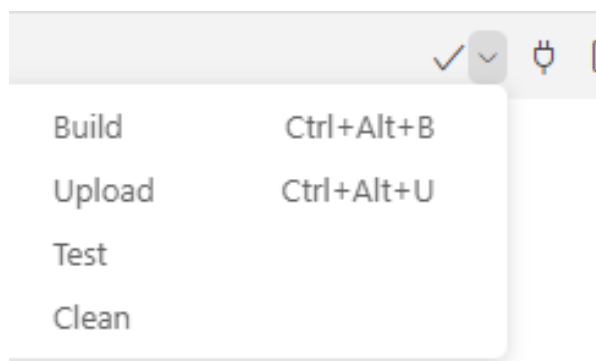
    delay(1000); // Tunggu 1 detik

    // Matikan kedua LED
    digitalWrite(lampu, LOW);
    digitalWrite(lampu2, LOW);
    Serial.println("LED OFF");

    delay(1000); // Tunggu 1 detik sebelum mengulang
}

```

Kemudian lakukan Upload pada menu **Upload**



Proses compiling dan upload akan berjalan dan pastikan berhasil seperti tampilan berikut

PROBLEMS    DEBUG CONSOLE    TERMINAL    PORTS

✓ **TERMINAL**

```
Hash of data verified.
Compressed 269520 bytes to 149510...
Writing at 0x00010000... (10 %)
Writing at 0x0001c467... (20 %)
Writing at 0x00024a7c... (30 %)
Writing at 0x00029cf6... (40 %)
Writing at 0x0002f25f... (50 %)
Writing at 0x000347d9... (60 %)
Writing at 0x0003d2a6... (70 %)
Writing at 0x00046281... (80 %)
Writing at 0x0004b817... (90 %)
Writing at 0x000510a7... (100 %)
Wrote 269520 bytes (149510 compressed) at 0x00010000 in 3.5 seconds (effective 613.2 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 13.38 seconds =====
* Terminal will be reused by tasks, press any key to close it.
```

Setelah langkah ini, jika wiring kabel dilakukan dengan benar seharusnya lampu LED menyala sesuai logika program.

### **3. Hasil dan Pembahasan**

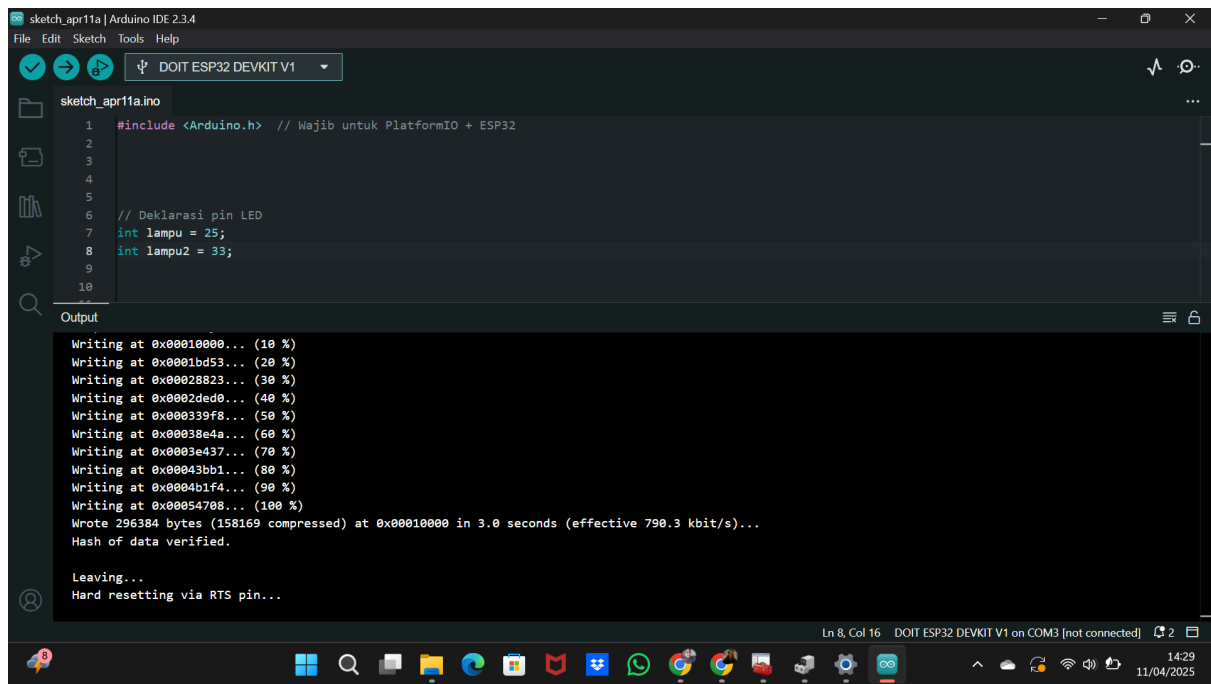
#### **3.1 Hasil Eksperimen**

- A. LED berhasil menyala saat dihubungkan dengan sumber tegangan 5V dan resistor 220 ohm sebagai pembatas arus.
- B. Saat polaritas dibalik (anoda ke negatif, katoda ke positif), LED tidak menyala, sesuai dengan sifat dioda yang hanya mengalirkan arus satu arah.
- C. Penambahan saklar memungkinkan kontrol manual terhadap nyala LED. Saat saklar ditekan (tertutup), LED menyala; saat dilepas (terbuka), LED mati.
- D. Tanpa resistor, LED menyala terlalu terang dan akhirnya rusak, membuktikan pentingnya pembatas arus dalam rangkaian LED.

#### **3.2 Kesimpulan**

Eksperimen menunjukkan bahwa pengendalian LED sangat bergantung pada arah arus dan kebutuhan arus maksimum LED itu sendiri. Resistor memainkan peran penting untuk menjaga arus tetap aman bagi LED. Saklar memungkinkan kontrol manual atas kondisi ON/OFF LED. Pengetahuan ini sangat penting sebagai dasar untuk merancang sistem otomatis berbasis mikrokontroler seperti Arduino atau Raspberry Pi.

#### 4. Lampiran Jika diperlukan



```
sketch_apr11a.ino
1 #include <Arduino.h> // Wajib untuk PlatformIO + ESP32
2
3
4
5
6 // Deklarasi pin LED
7 int lampu = 25;
8 int lampu2 = 33;
9
10

Output
Writing at 0x00010000... (10 %)
Writing at 0x0001bd53... (20 %)
Writing at 0x00028823... (30 %)
Writing at 0x0002ded0... (40 %)
Writing at 0x000339f8... (50 %)
Writing at 0x00038e4a... (60 %)
Writing at 0x0003e437... (70 %)
Writing at 0x00043bb1... (80 %)
Writing at 0x0004b1f4... (90 %)
Writing at 0x00054708... (100 %)
Wrote 296384 bytes (158169 compressed) at 0x00010000 in 3.0 seconds (effective 790.3 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

Ln 8, Col 16 DOIT ESP32 DEVKIT V1 on COM3 [not connected] 14:29 11/04/2025
```

