LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Simulasi Relay, Button & LED, Sensor Jarak, dan** **Pembuatan API   
Menggunakan Laravel 11 dan Ngrok**



*Dina Hanifa*

*Fakultas Vokasi, Universitas Brawijaya*

*Email :* [*dinahanifa@student.ub.ac.id*](mailto:dinahanifa@student.ub.ac.id)

**PRAKTIK SIMULASI RELAY, BUTTON & LED**

**Abstrak**

Pada eksperimen ini, dilakukan pengendalian perangkat listrik menggunakan ESP32, sebuah modul relay, dan tombol push button. Saat tombol ditekan, relay dan LED akan menyala, menandakan aktivasi perangkat terhubung. Hasil percobaan menunjukkan sistem bekerja sesuai harapan, dengan respons cepat dan stabil. Kesimpulannya, ESP32 dapat digunakan untuk mengendalikan perangkat listrik secara efektif dalam proyek berbasis IoT.

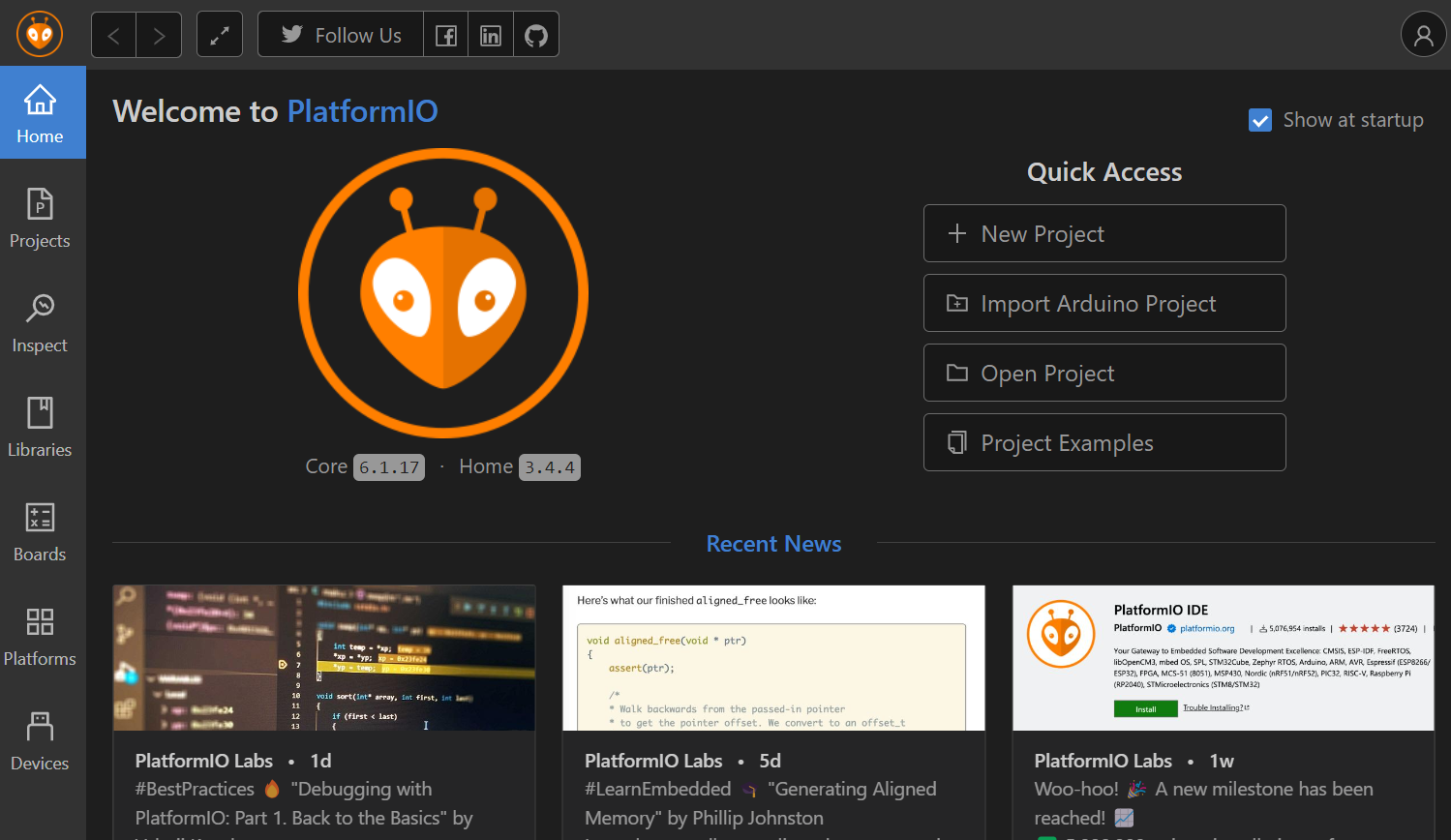
*Kata kunci: ESP32, Relay, Push Button, IoT, Otomatisasi*

1. **Pendahuluan** 
   1. **Latar Belakang**

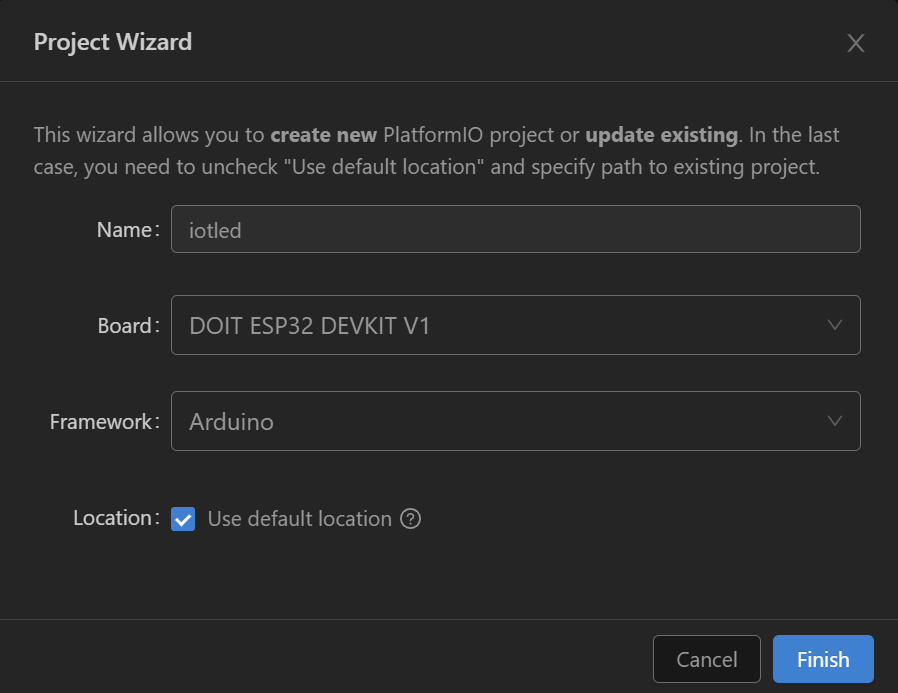
Internet of Things (IoT) semakin berkembang dalam berbagai aspek kehidupan, termasuk dalam otomatisasi rumah dan industri. Dengan penggunaan mikrokontroler seperti ESP32, perangkat listrik dapat dikendalikan secara efisien dan otomatis. Dalam praktikum ini, dibuat sistem sederhana yang mengontrol relay menggunakan tombol push button dan LED sebagai indikator. Modul relay digunakan sebagai perantara antara ESP32 dan perangkat listrik. Relay adalah komponen elektronik yang berfungsi sebagai saklar elektronik yang dapat diaktifkan atau dinonaktifkan oleh mikrokontroler. Untuk memberikan kontrol manual atas relay, *push button* digunakan sebagai input, dan LED sebagai indikator visual yang menunjukkan status relay. Relay ini memungkinkan kontrol atas perangkat dengan daya yang lebih besar, seperti lampu atau motor listrik, sehingga memberikan lebih banyak fleksibilitas dalam pengembangan aplikasi otomasi

* 1. **Tujuan Eksperimen**

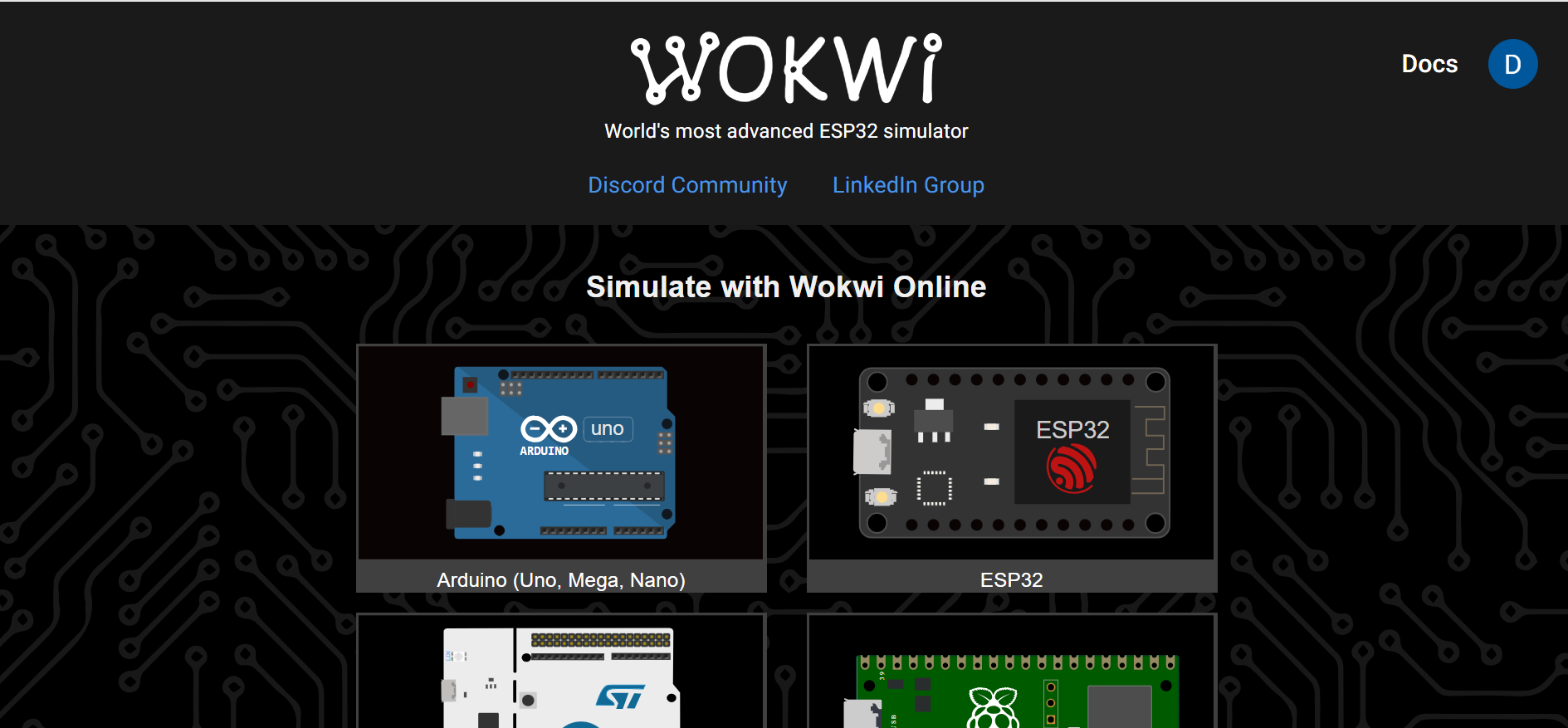
1. Mempelajari cara kerja relay dan interaksinya dengan ESP32.
2. Mengembangkan sistem kontrol sederhana berbasis tombol push button.
3. Menguji kestabilan dan kecepatan respons sistem.
4. Mengidentifikasi potensi pengembangan lebih lanjut dengan integrasi IoT berbasis jaringan.
5. **Metodologi**
   1. **Alat dan Bahan**
6. Perangkat komputer dengan koneksi internet
7. ESP32 sebagai mikrokontroler utama
8. Modul relay 5V
9. Push button
10. LED dan resistor 220Ω
11. Software Visual Studio Code (VS Code)
12. Akses ke situs Wokwi (<https://wokwi.com>)
    1. **Langkah Implementasi**
13. Buka PlatformIO di VSCode lalu buat project baru



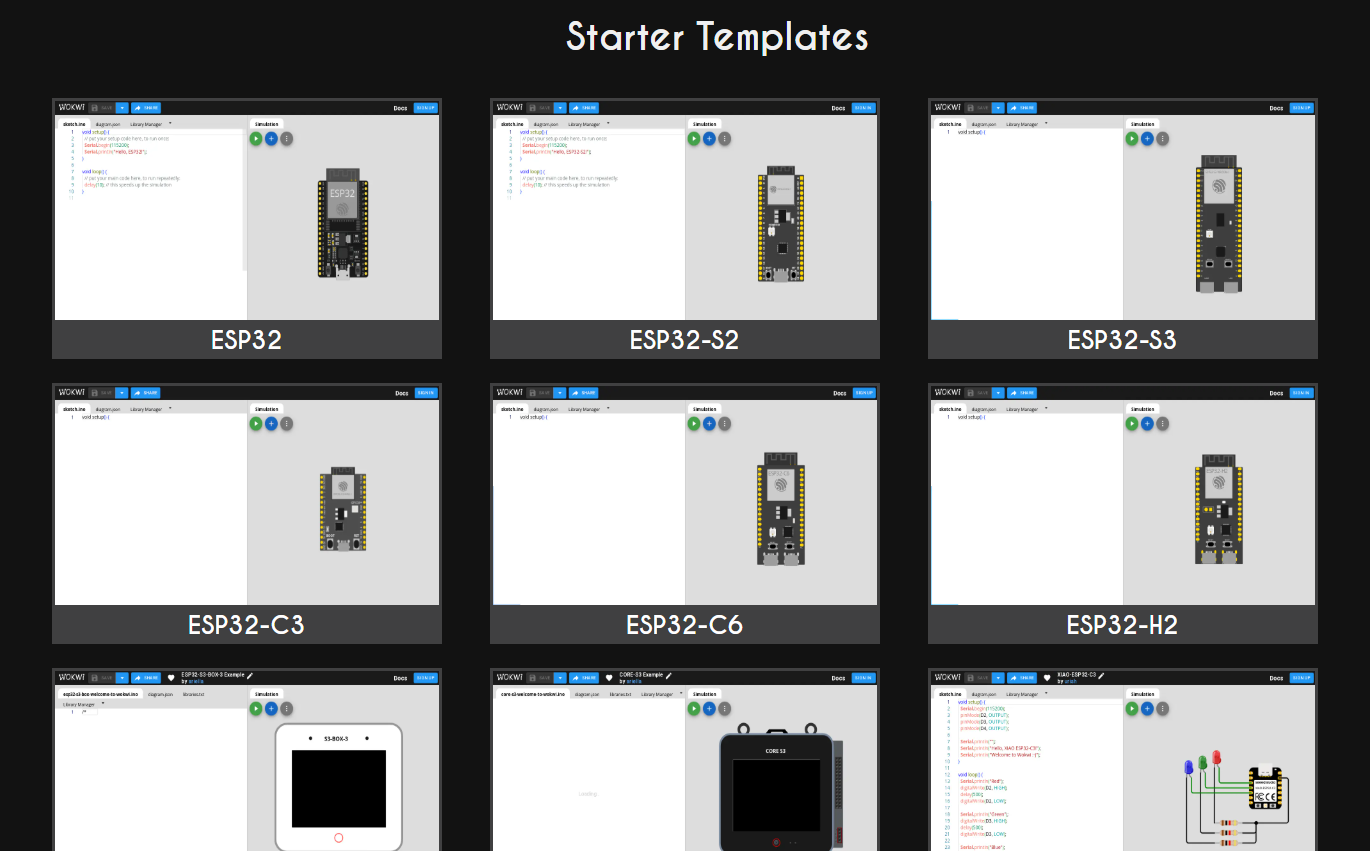
Sesuaikan nama project yang ingin dibuat, pilih Board DOIT ESP32 DEVKIT V1 dan gunakan framework Arduino, setelah itu tekan finish



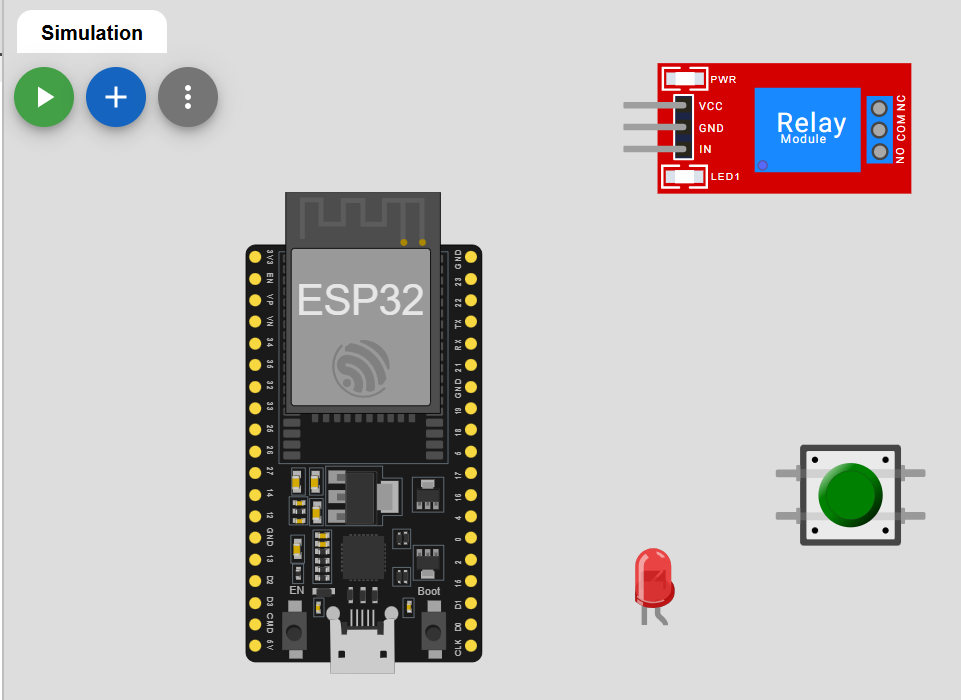
1. Buka situs wokwi dan pilih mikrokontroler ESP32



Pilih ESP32 di Starter Templates



1. Tambahkan modul relay, *push button* dan LED di bagian simulation



1. Menghubungkan kabel sesuai intruksi berikut:

Relay Module:

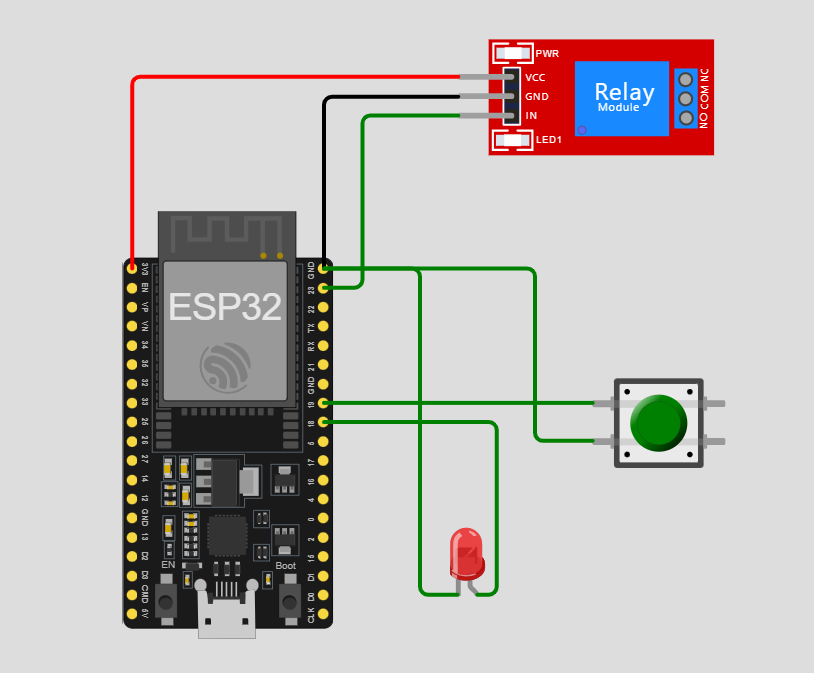
* VCC relay terhubung ke 3.3V ESP32 untuk memberikan daya ke modul.
* GND relay terhubung ke GND ESP32 sebagai referensi ground.
* IN relay terhubung ke GPIO23 ESP32 untuk mengontrol status relay.

Push Button:

* Satu kaki push button terhubung ke GPIO19 ESP32 sebagai input.
* Kaki lainnya terhubung ke GND untuk memberikan sinyal LOW saat ditekan.

LED:

* Anoda (kaki panjang) LED terhubung ke GPIO18 ESP32 melalui resistor 220Ω untuk membatasi arus.
* Katoda (kaki pendek) LED terhubung ke GND ESP32.



1. Mendeklarasikan Variasi Integer

Deklarasikan variabel integer dengan menetapkan pin yang digunakan.

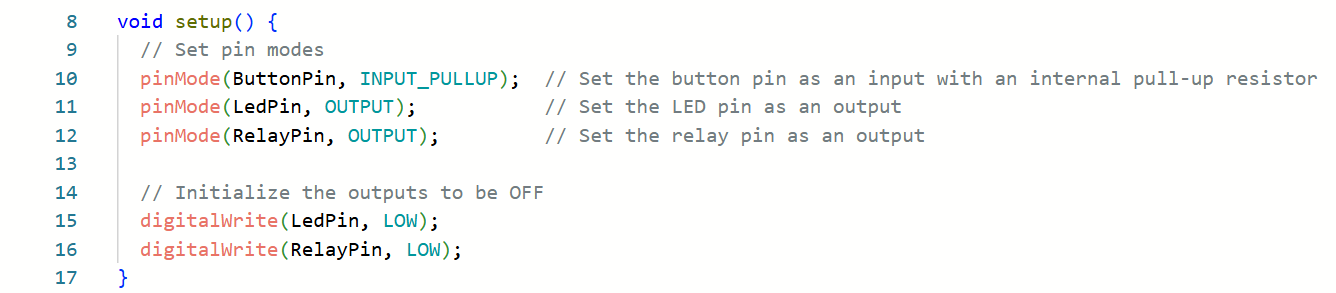
* Tombol push button ditempatkan pada GPIO19.
* LED ditempatkan pada GPIO18.
* Relay ditempatkan pada GPIO23.



1. Menambahkan fungsi setup

Tambahkan fungsi setup dengan mengatur semua pin yang digunakan sebagai input atau output.

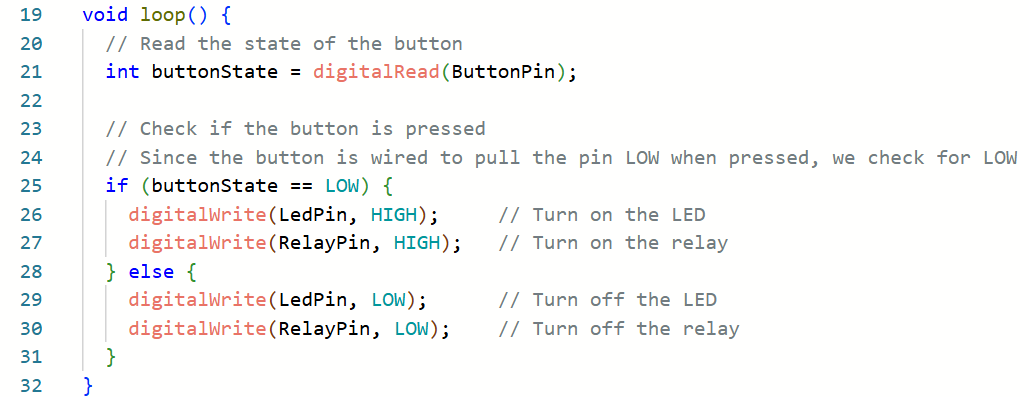
* Tombol push button sebagai INPUT\_PULLUP.
* LED dan relay sebagai OUTPUT.
* Inisialisasi awal LED dan relay dalam keadaan LOW.



1. Membuat fungsi loop

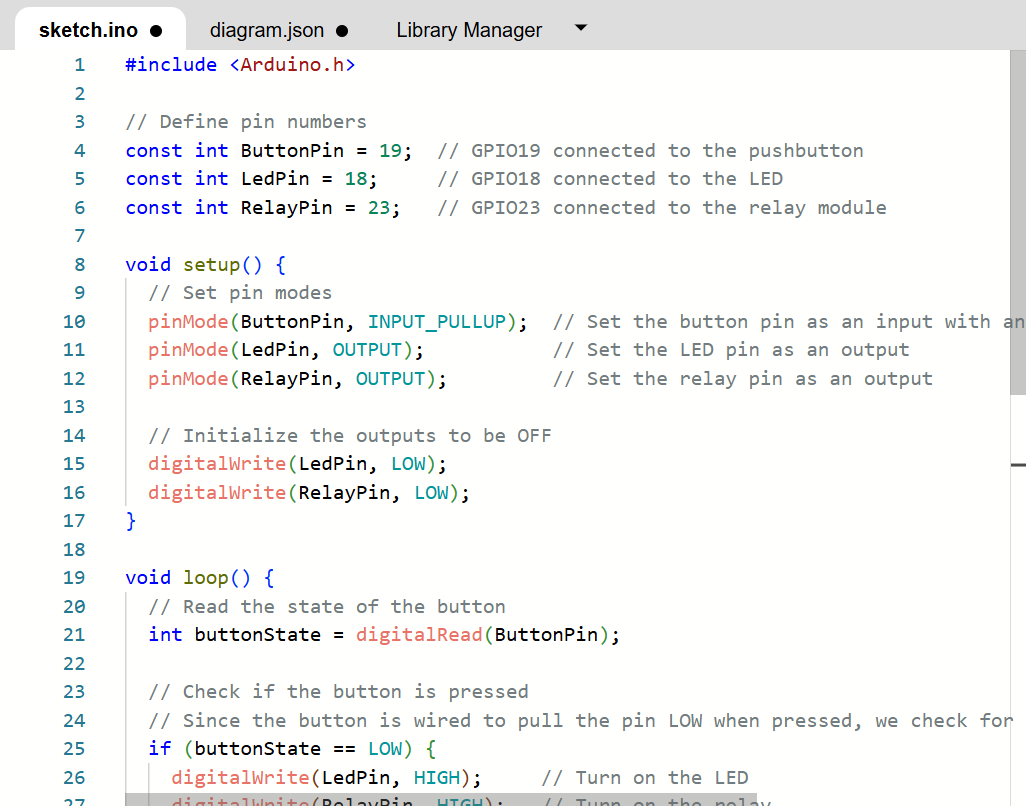
* Membaca status tombol push button.
* Jika tombol ditekan (LOW), LED dan relay menyala.
* Jika tombol dilepas (HIGH), LED dan relay mati.

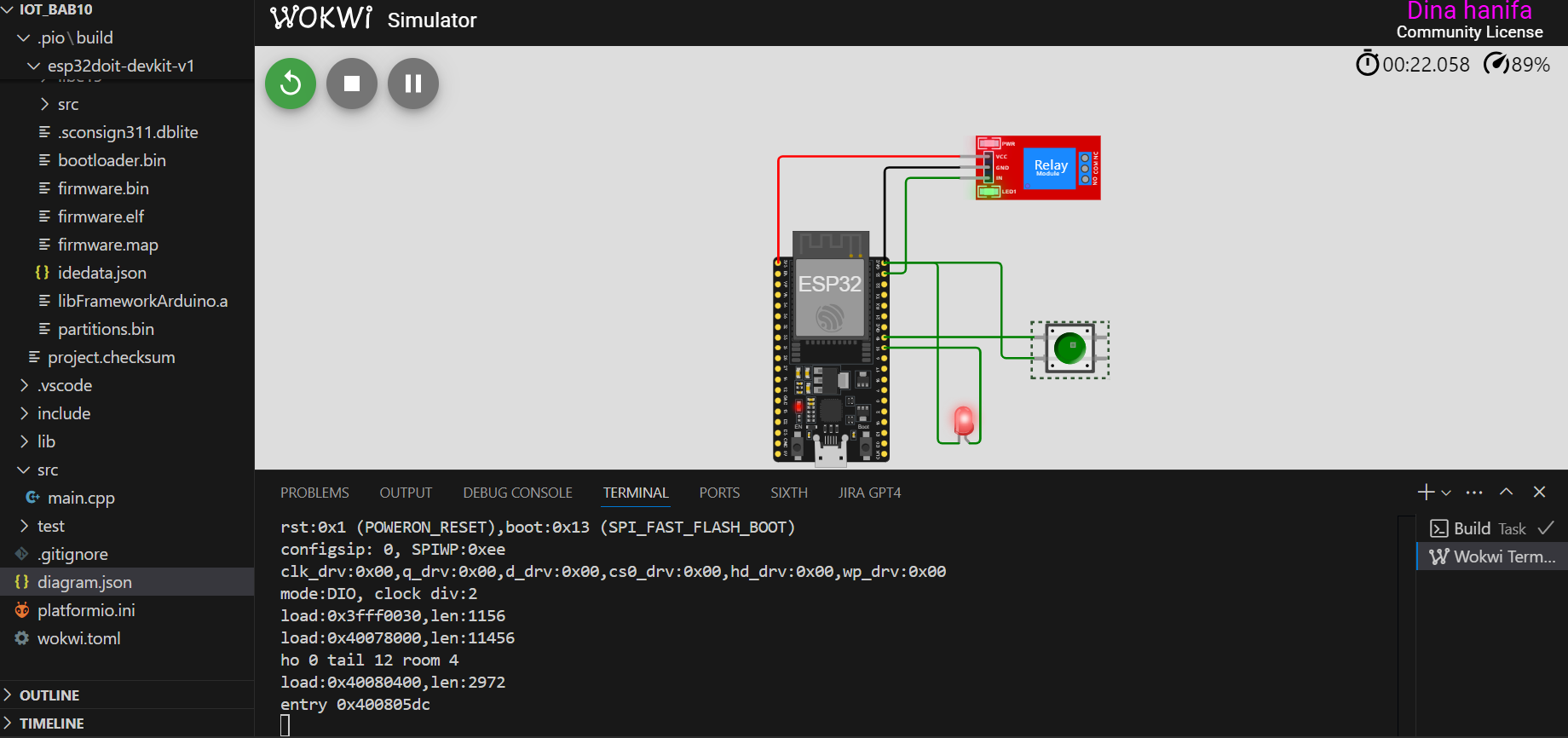
Proses ini berulang secara terus-menerus.



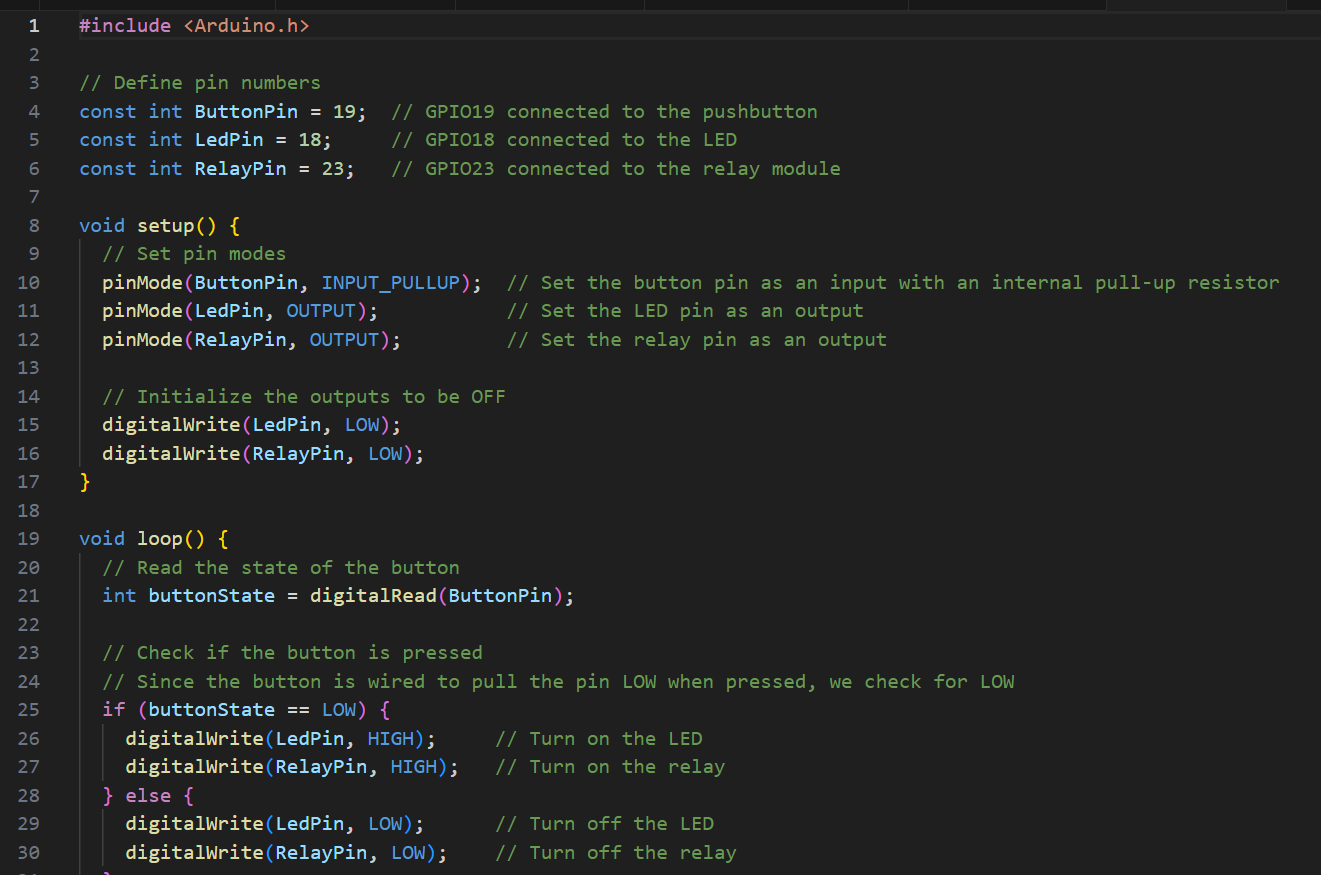
1. Pengujian dan Verifikasi

* Mengunggah kode program ke ESP32.
* Menekan tombol untuk memastikan sistem bekerja dengan benar.
* Memastikan LED dan relay merespons sesuai dengan perintah.



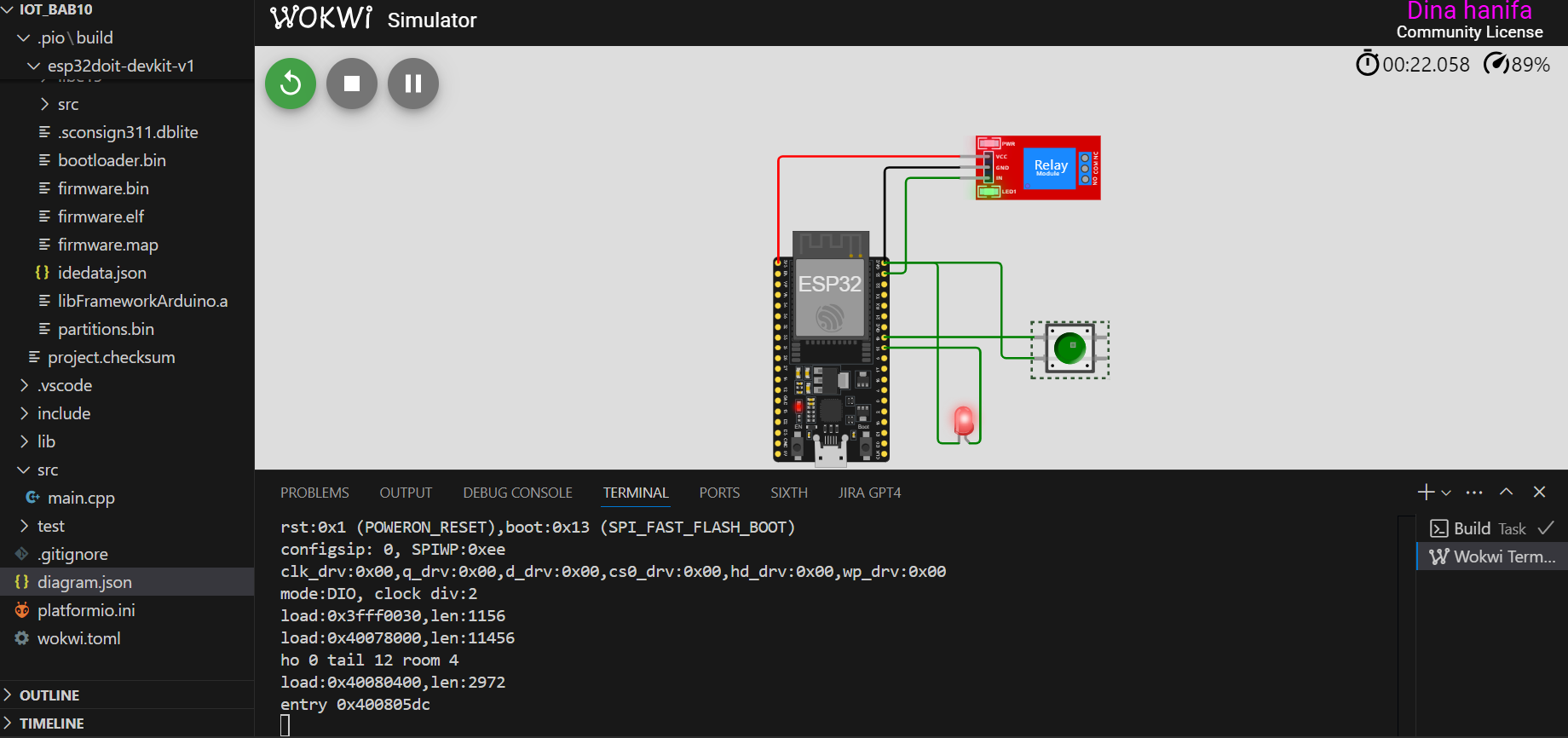


1. Salin kode program yang sudah dibuat di wokwi dan tempelkan ke dalam file main.cpp



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

Hasil eksperimen menunjukkan bahwa saat tombol ditekan, LED dan relay menyala, dan saat tombol dilepas, keduanya mati. Hal ini membuktikan bahwa ESP32 berhasil membaca input tombol dan mengontrol output dengan benar. Selain itu, dilakukan pengukuran tegangan pada pin output relay untuk memastikan kestabilan sistem dalam mengendalikan perangkat eksternal.



**PRAKTIK SIMULASI SENSOR JARAK**

**Abstrak**

Eksperimen ini bertujuan untuk mengukur jarak menggunakan sensor ultrasonik HC-SR04 yang dikontrol oleh mikrokontroler ESP32. Sensor bekerja dengan mengirimkan gelombang ultrasonik dan mengukur waktu yang dibutuhkan untuk memantulkannya kembali. Jarak dihitung berdasarkan waktu tempuh gelombang suara dengan konversi ke cm dan inci. Hasil percobaan menunjukkan bahwa sistem dapat mendeteksi jarak secara akurat dan menampilkan hasilnya pada Serial Monitor. Kesimpulannya, kombinasi ESP32 dan HC-SR04 dapat digunakan dalam berbagai aplikasi IoT, seperti sistem keamanan, robotika, dan otomatisasi rumah pintar..

*Kata kunci: ESP32, HRC-SR04, Sensor Ultrasonik, IoT, Pengukuran Jarak*

* + - 1. **Pendahuluan**
  1. **Latar Belakang**

*Internet of Things* (IoT) semakin berkembang dalam berbagai aspek kehidupan, termasuk dalam otomatisasi rumah dan industri. Dengan penggunaan mikrokontroler seperti ESP32, perangkat listrik dapat dikendalikan secara efisien dan otomatis. Dalam praktikum ini, dibuat sistem sederhana yang mengontrol relay menggunakan tombol push button dan LED sebagai indikator. Modul relay digunakan sebagai perantara antara ESP32 dan perangkat listrik. Relay adalah komponen elektronik yang berfungsi sebagai saklar elektronik yang dapat diaktifkan atau dinonaktifkan oleh mikrokontroler. Untuk memberikan kontrol manual atas relay, *push button* digunakan sebagai input, dan LED sebagai indikator visual yang menunjukkan status relay. Relay ini memungkinkan kontrol atas perangkat dengan daya yang lebih besar, seperti lampu atau motor listrik, sehingga memberikan lebih banyak fleksibilitas dalam pengembangan aplikasi otomasi.

* 1. **Tujuan Eksperimen**

Menghubungkan Sensor Ultrasonik dengan Arduino Uno

Menentukan jarak objek dalam nilai tertentu pada kode pemrograman Arduino

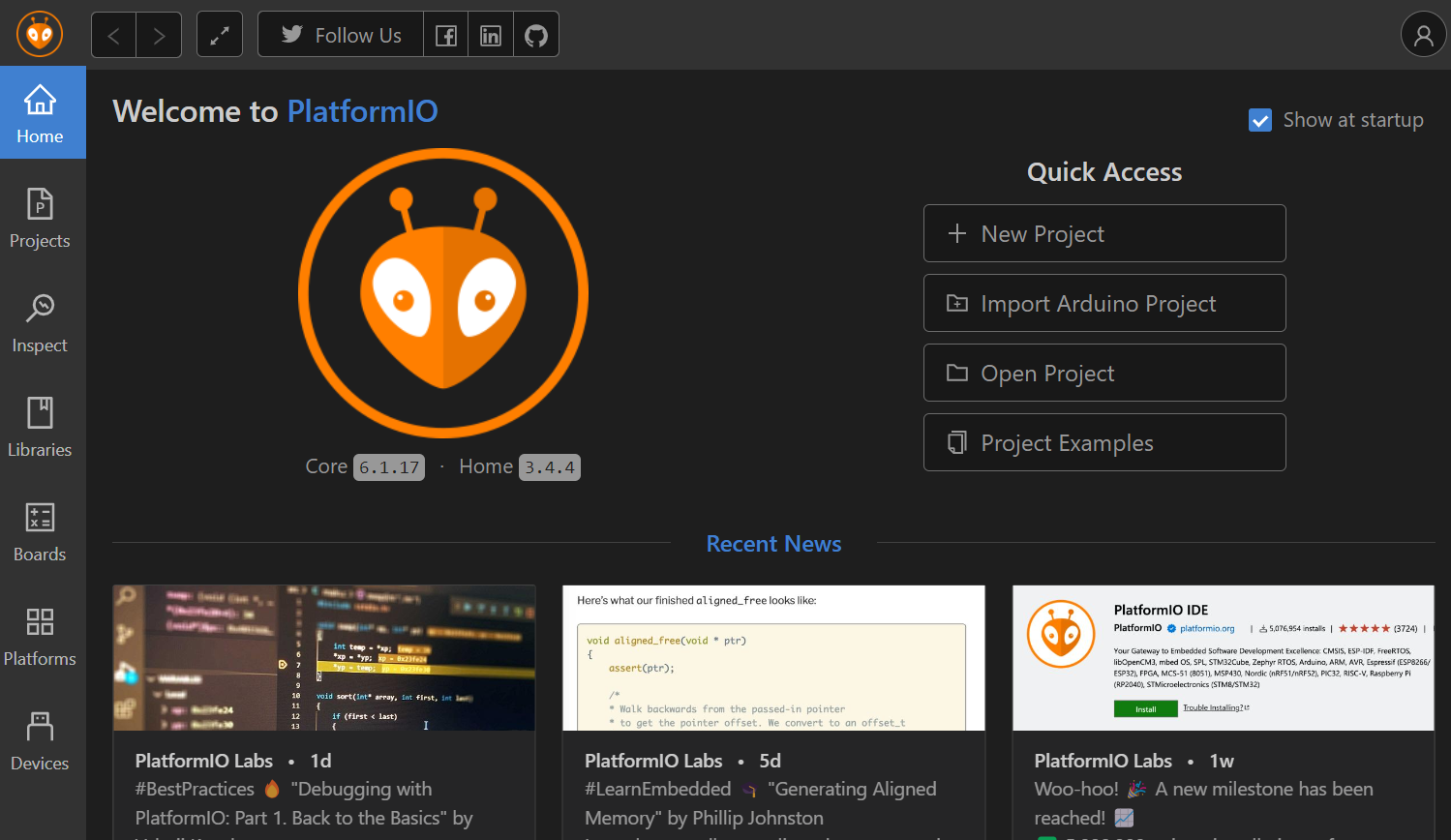
Merangkai rangkaian ultrasonic dengan output

* + - 1. **Metodologi**

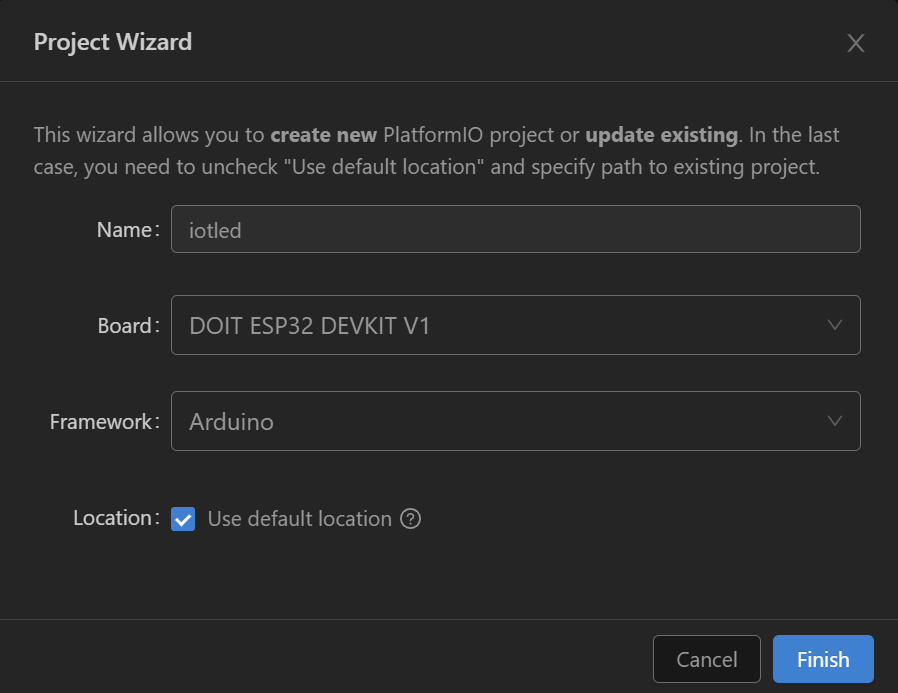
**2.1 Alat dan Bahan**

* + - 1. Perangkat komputer dengan koneksi internet
      2. ESP32 sebagai mikrokontroler utama
      3. Sensor ultrasonik HC-SR04
      4. Arduino IDE
      5. Software Visual Studio Code (VS Code)
      6. Akses ke situs Wokwi (<https://wokwi.com>)
  1. **Langkah Implementasi**

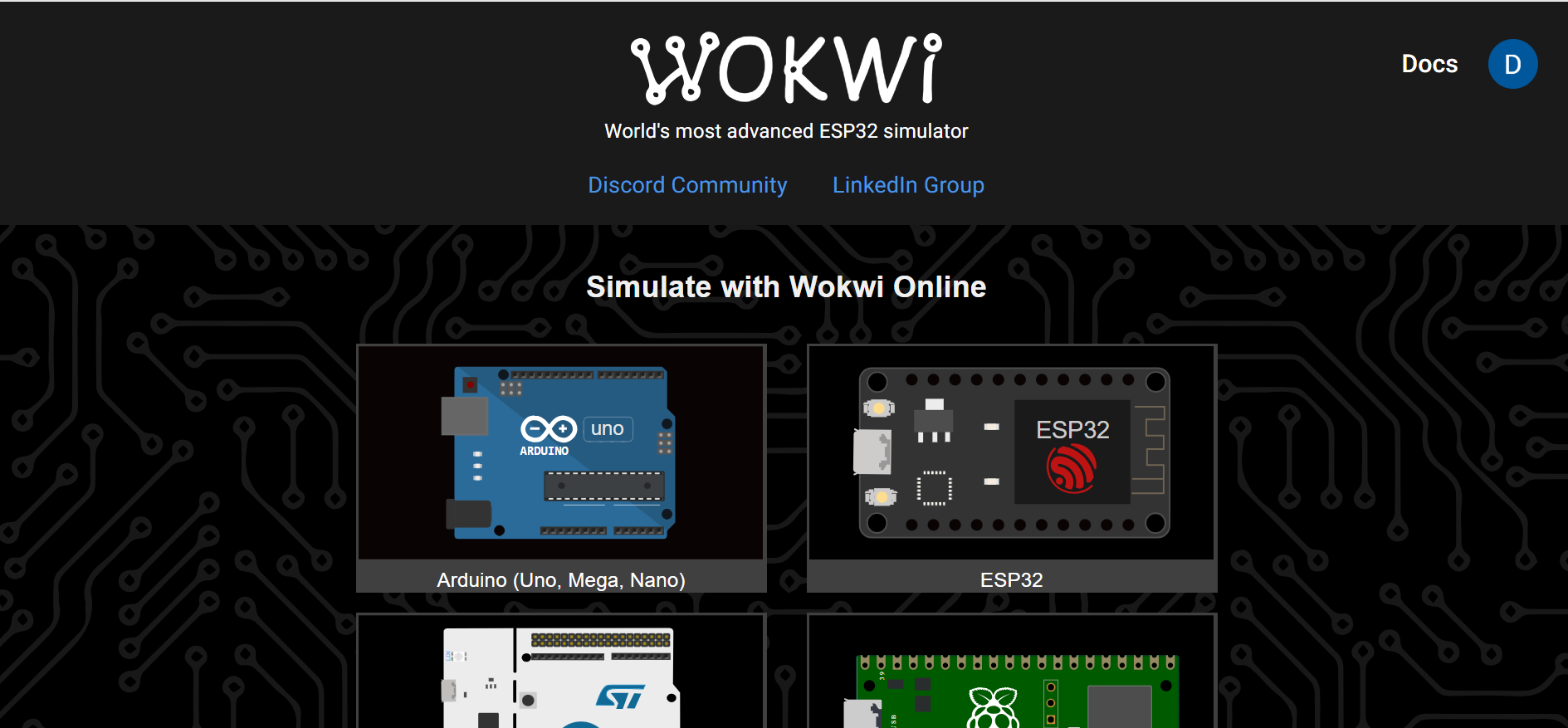
Buka PlatformIO di VSCode lalu buat project baru



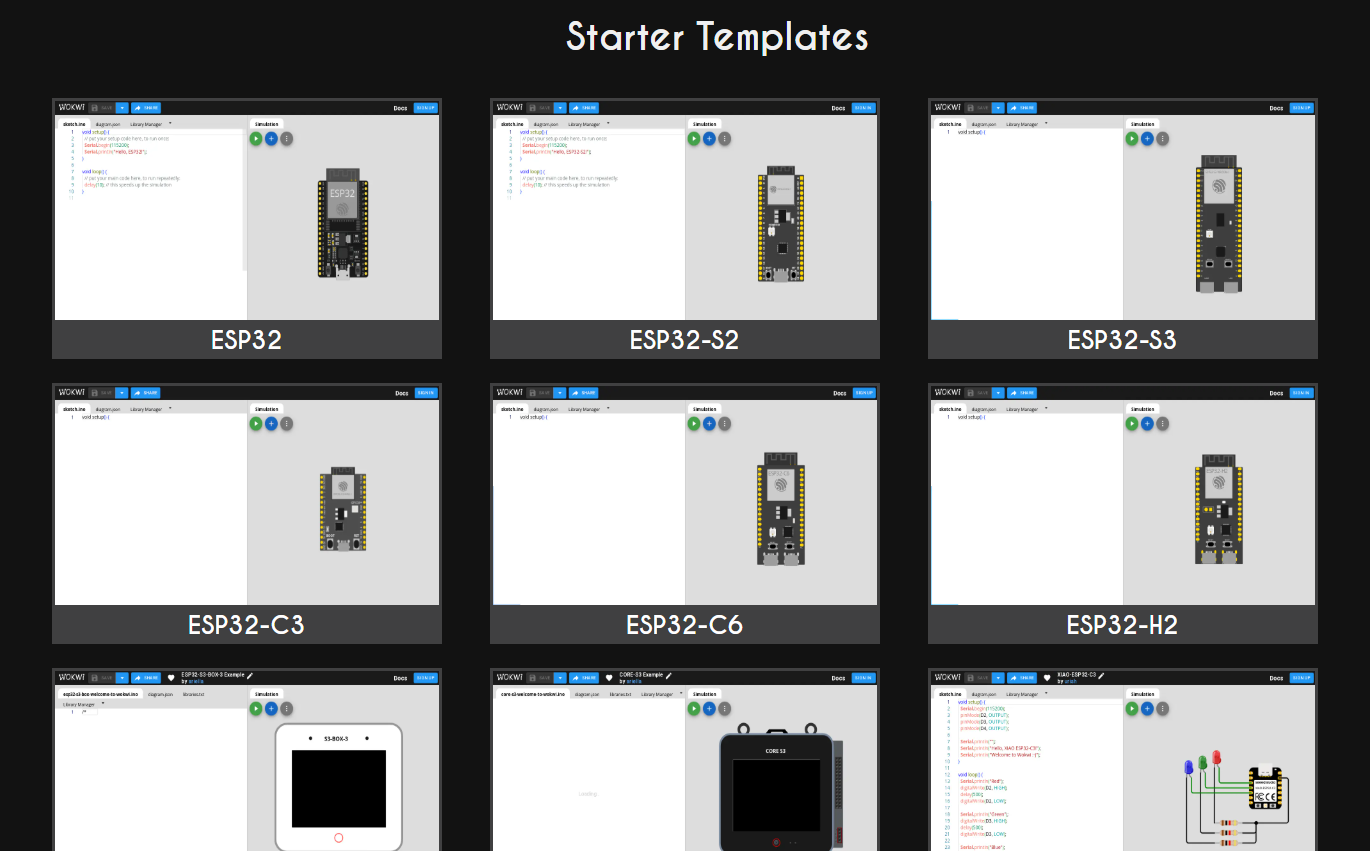
Sesuaikan nama project yang ingin dibuat, pilih Board DOIT ESP32 DEVKIT V1 dan gunakan framework Arduino, setelah itu tekan finish



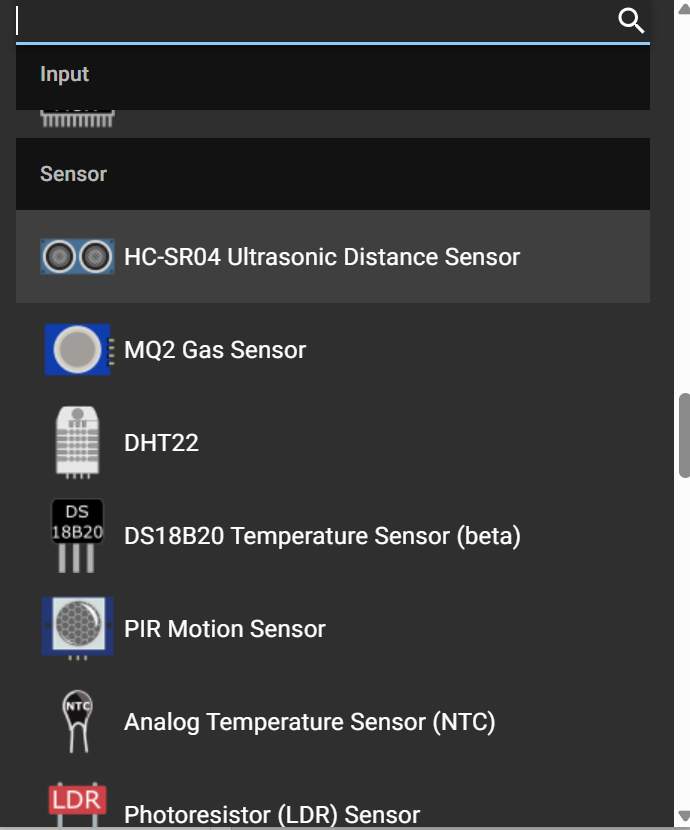
Buka situs wokwi dan pilih mikrokontroler ESP32

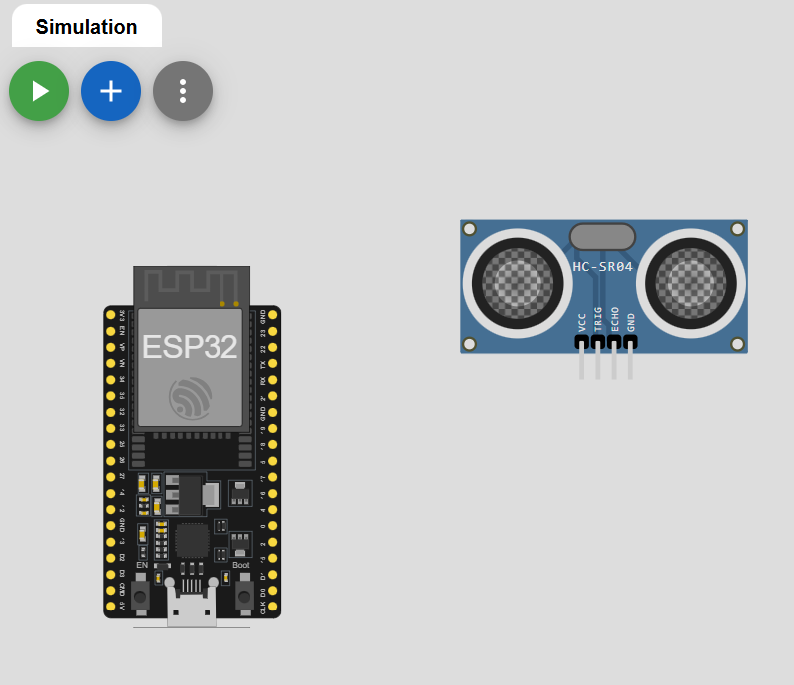


Pilih ESP32 di Starter Templates



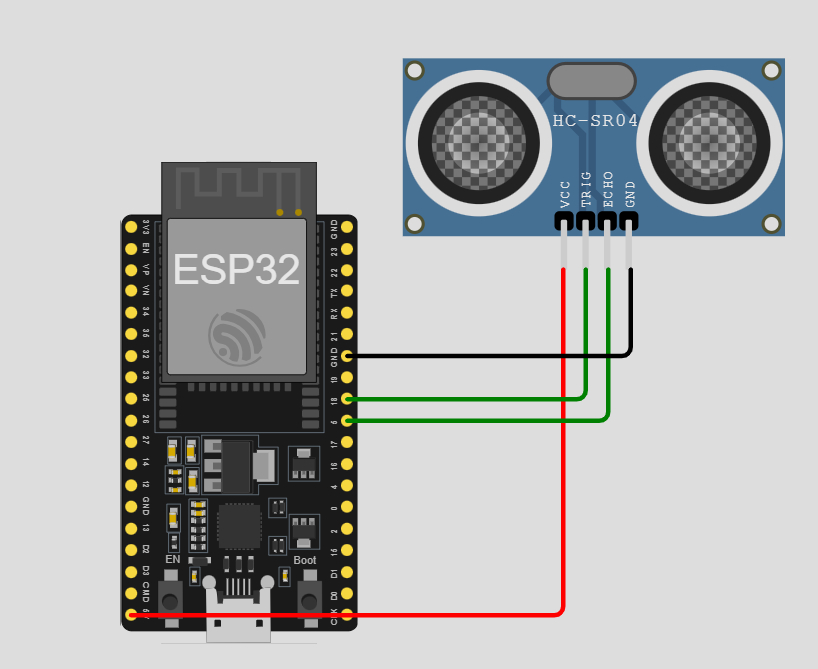
Tambahkan modul relay, *push button* dan LED di bagian simulation





Menghubungkan kabel sesuai intruksi berikut:

* VCC HC-SR04 → 3.3V atau 5V ESP32
* GND HC-SR04 → GND ESP32
* Trig HC-SR04 → GPIO5 ESP32
* Echo HC-SR04 → GPIO18 ESP32



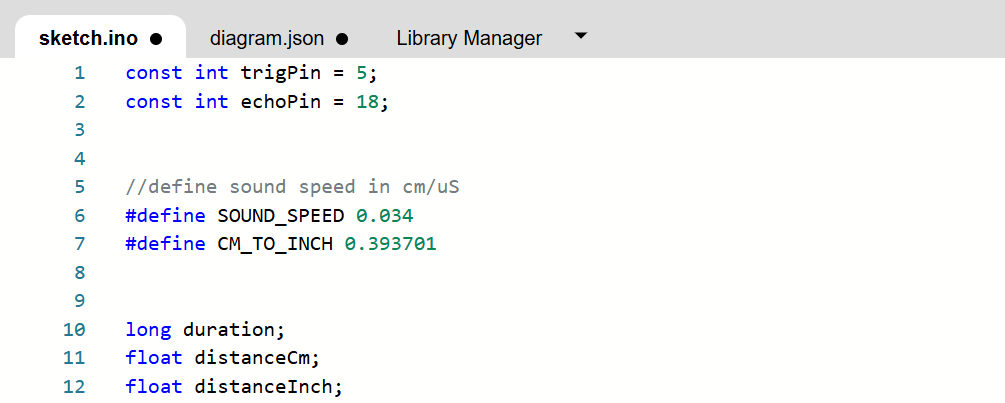
Inisialisasi dan Deklarasi Variabel

Menetapkan pin untuk sensor ultrasonik HC-SR04 pada ESP32. Konfigurasi pin pada ESP32:

* Trig Pin → GPIO5
* Echo Pin → GPIO18

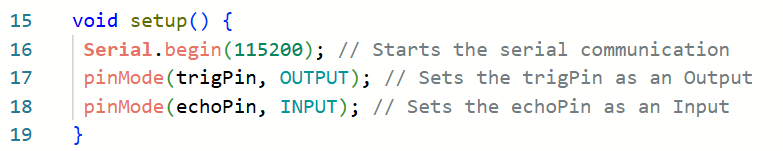
Variabel yang digunakan:

* duration: menyimpan waktu tempuh gelombang ultrasonik.
* distanceCm: menyimpan hasil perhitungan jarak dalam cm.
* distanceInch: menyimpan hasil perhitungan jarak dalam inci.



Konfigurasi Setup

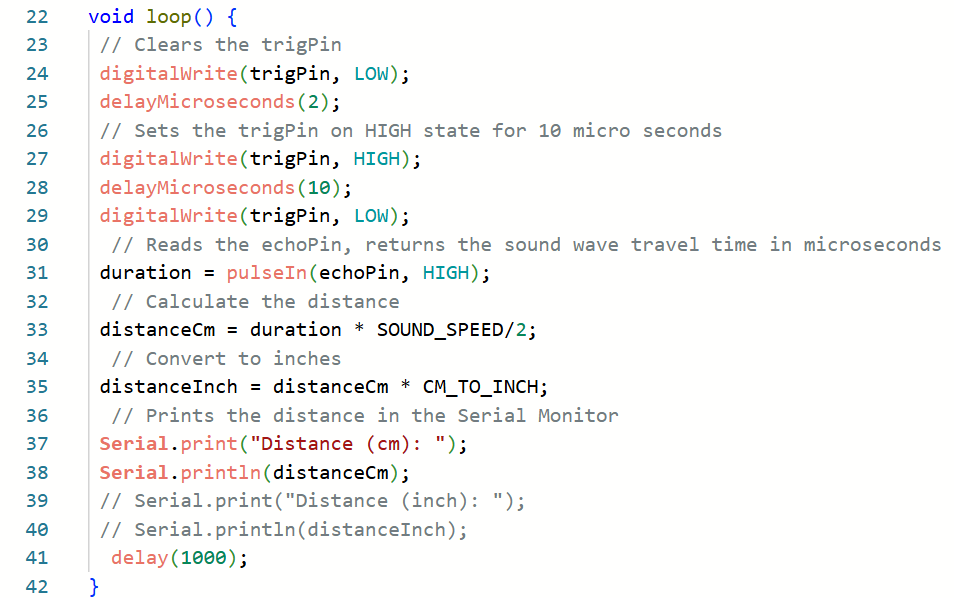
* Mengatur trigPin sebagai output dan echoPin sebagai input.
* Menginisialisasi komunikasi serial untuk menampilkan data hasil pengukuran.



* + - 1. Implementasi fungsi loop
* Mengaktifkan trigPin selama 10 mikrodetik untuk memicu gelombang ultrasonik.
* Membaca echoPin untuk mengukur waktu tempuh gelombang.
* Menghitung jarak berdasarkan rumus:

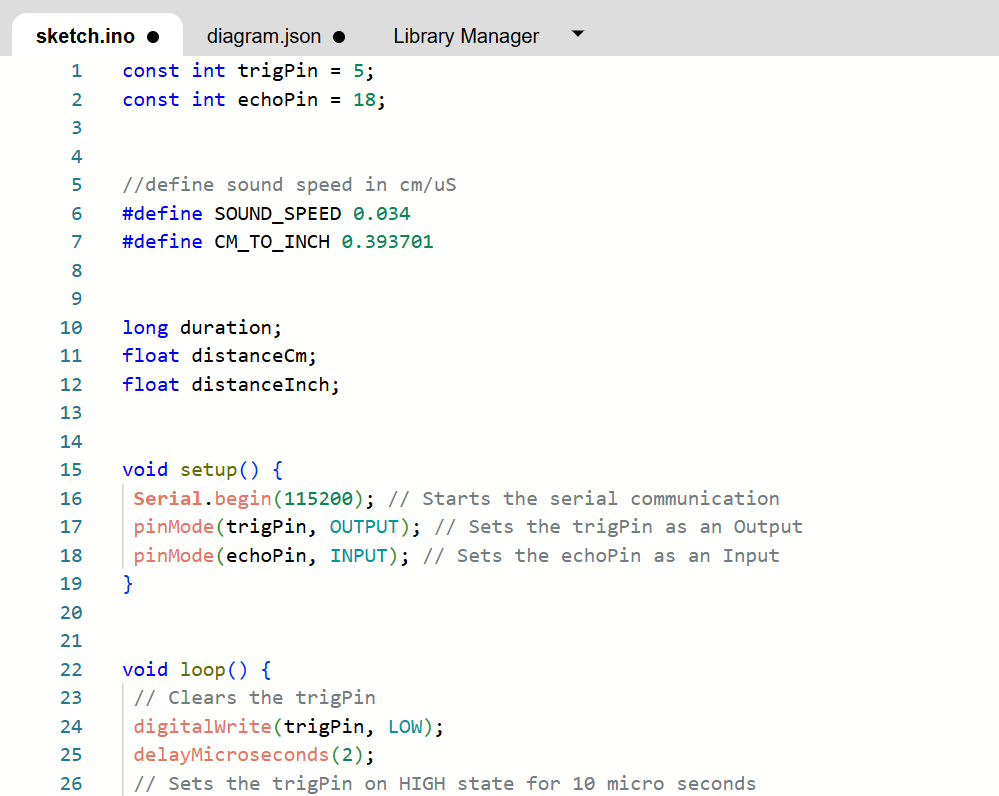
distanceCm = duration × 0.034 /2

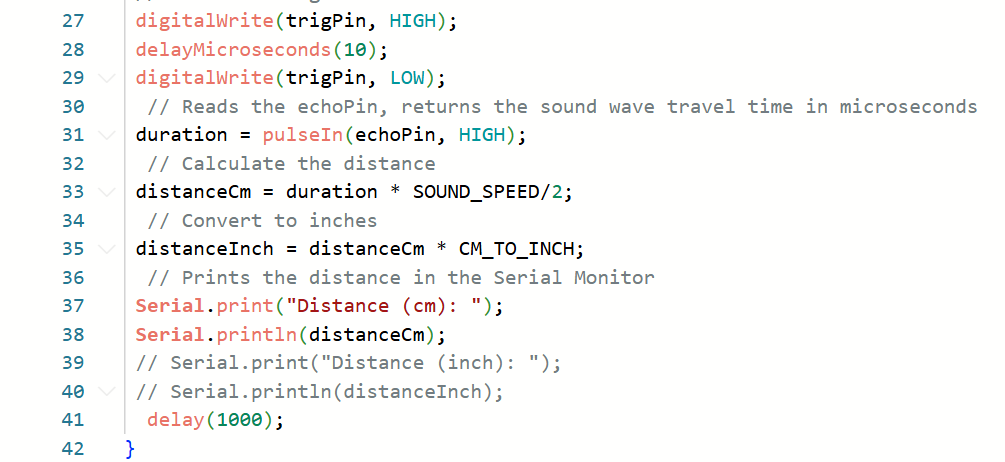
* Menampilkan hasil pengukuran pada Serial Monitor

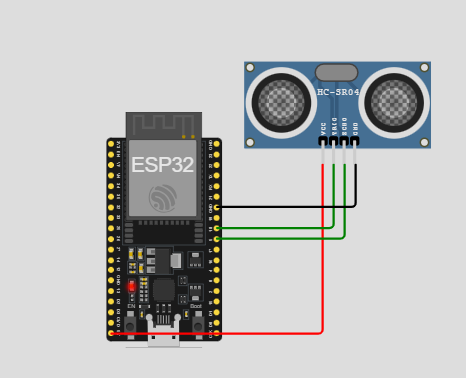


1. Pengujian dan Verifikasi

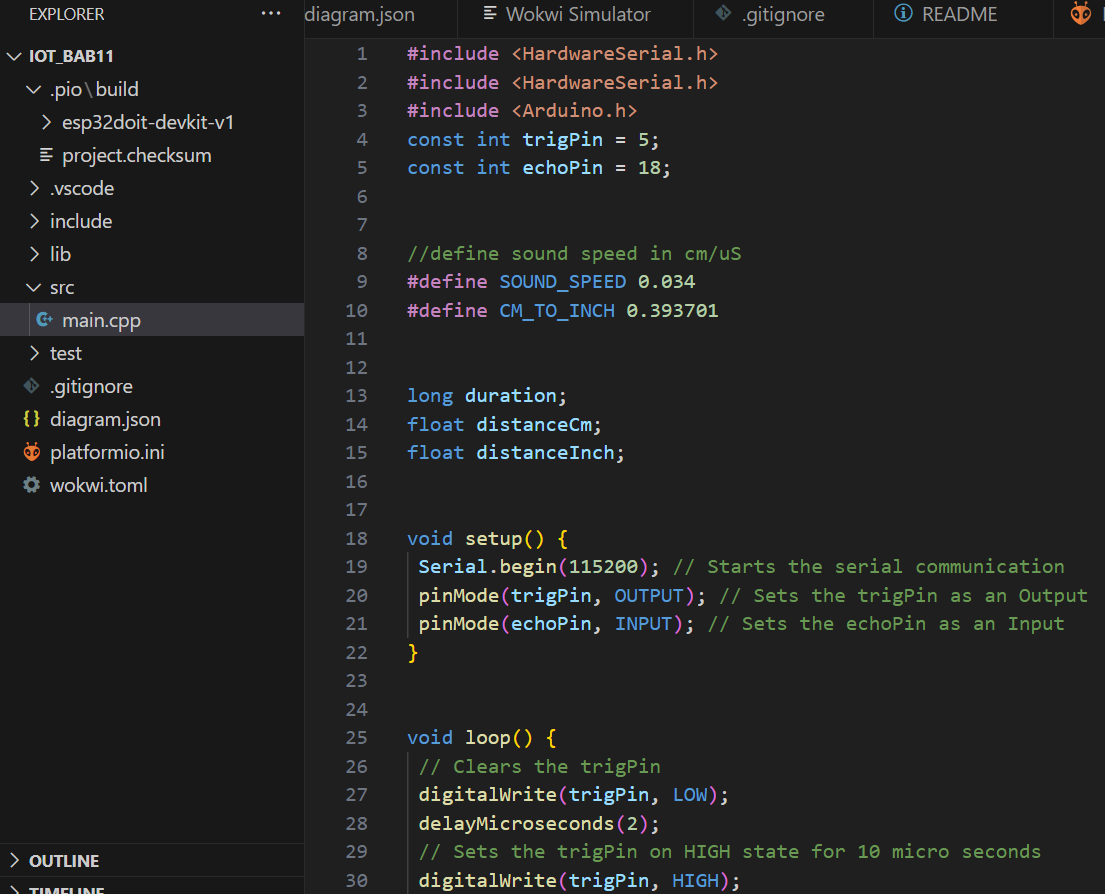
* Mengunggah kode ke ESP32.
* Mengamati Serial Monitor untuk memastikan pengukuran berjalan dengan baik.

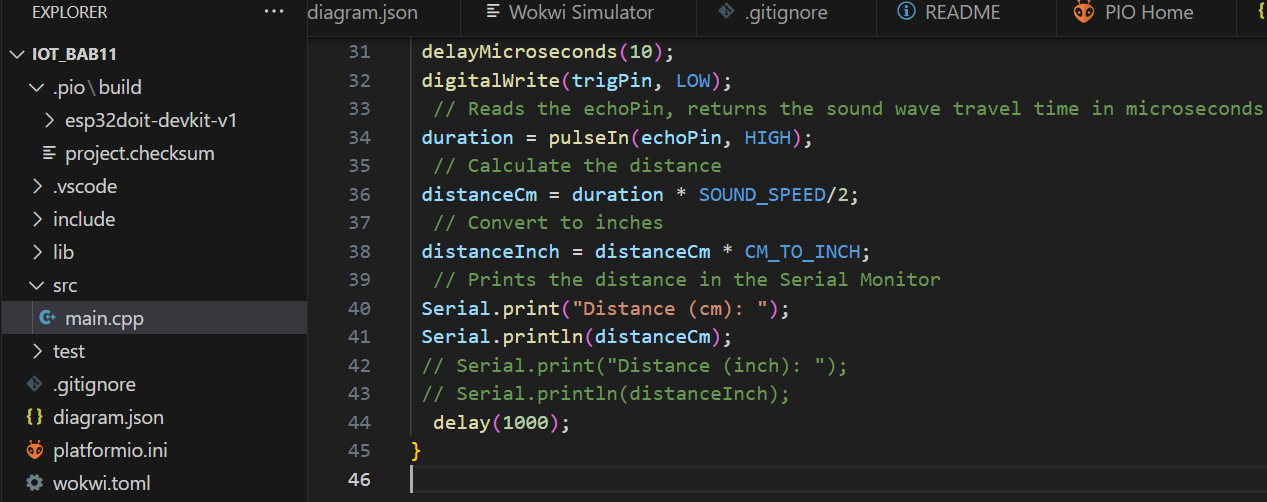






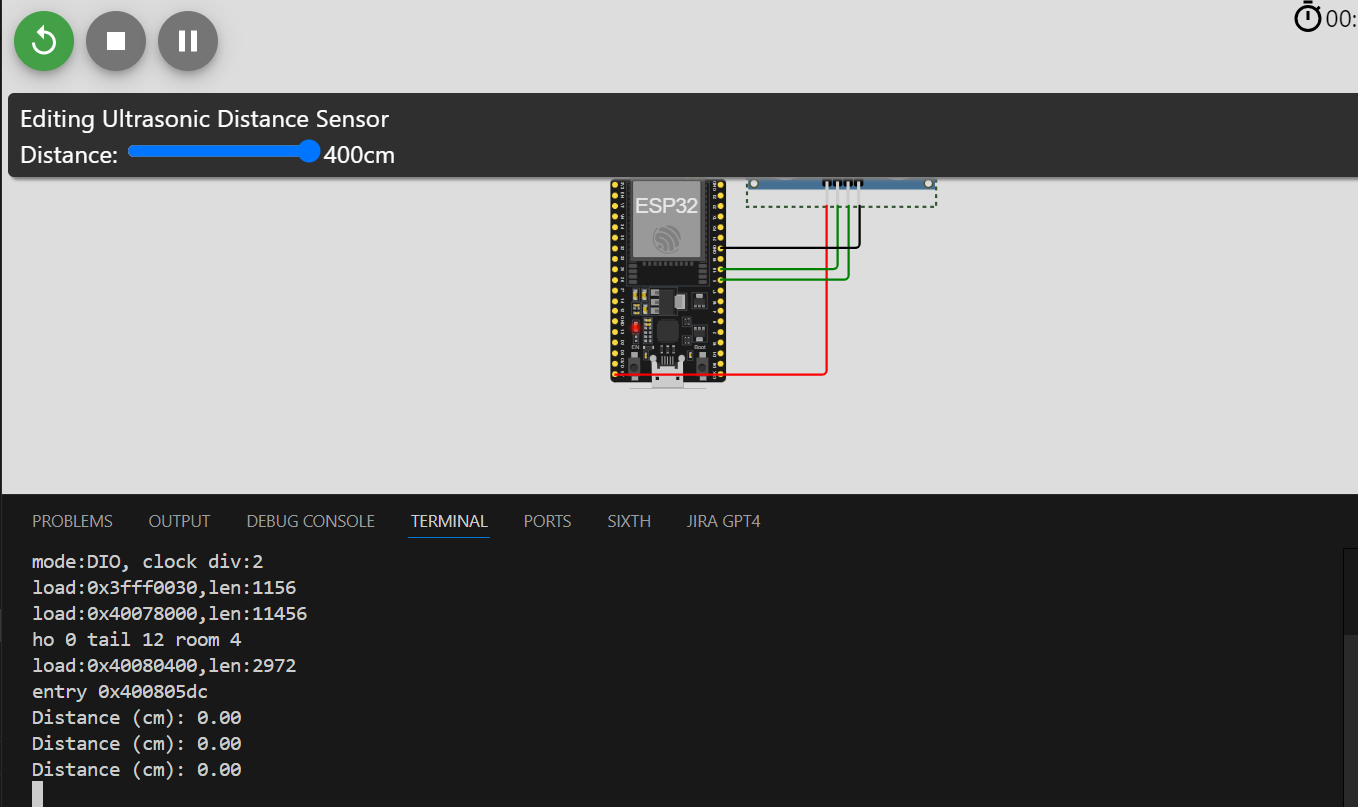
1. Salin kode program yang sudah dibuat di wokwi dan tempelkan ke dalam file main.cpp





* + - 1. **Hasil dan Pembahasan**
  1. **Hasil Eksperimen**

Hasil eksperimen menunjukkan bahwa sistem dapat mengukur jarak dengan cukup akurat dan menampilkan hasilnya pada Serial Monitor.



**PRAKTIK PEMBUATAN API MENGGUNAKAN LARAVEL 11 DAN NGROK**

**Abstrak**

Internet of Things (IoT) memerlukan komunikasi yang efektif antara perangkat dan server untuk mengelola serta menganalisis data sensor secara real-time. Praktikum ini membahas implementasi API menggunakan Laravel 11 sebagai backend, dengan Ngrok sebagai perantara untuk menghubungkan API ke internet. Metode yang digunakan meliputi pembuatan database, pembuatan endpoint API, serta pengujian menggunakan Postman dan Wokwi. Hasil menunjukkan bahwa API dapat menerima dan menyimpan data sensor dengan baik, serta dapat diakses dari berbagai platform melalui Ngrok. Praktikum ini membuktikan bahwa Laravel dan Ngrok dapat digunakan untuk membangun infrastruktur komunikasi IoT yang efisien dan mudah diimplementasikan.

*Kata kunci: API, Ngrok, Laravel, IoT, sensor.*

**Pendahuluan**

* 1. **Latar Belakang**

Dalam perkembangan Internet of Things (IoT), penggunaan API (Application Programming Interface) menjadi sangat penting untuk memungkinkan komunikasi antara perangkat IoT dan server. Laravel sebagai salah satu framework PHP populer digunakan untuk membangun API yang dapat diakses dari berbagai platform. Ngrok digunakan untuk menyediakan akses publik ke API lokal, sehingga dapat diuji dan digunakan dalam simulasi atau perangkat IoT nyata.

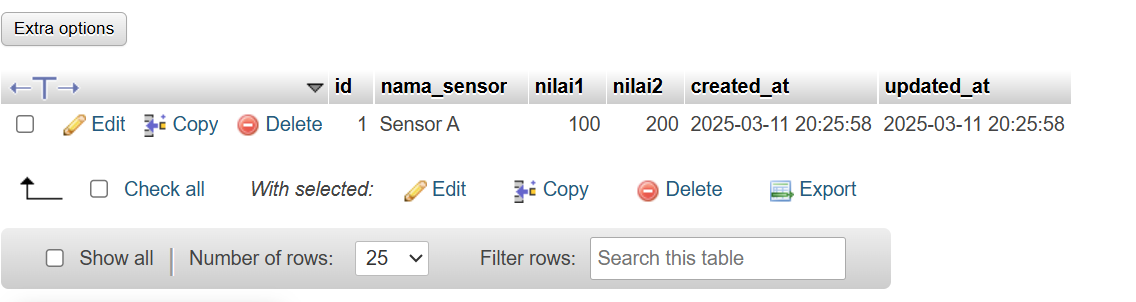
* 1. **Tujuan Eksperimen**

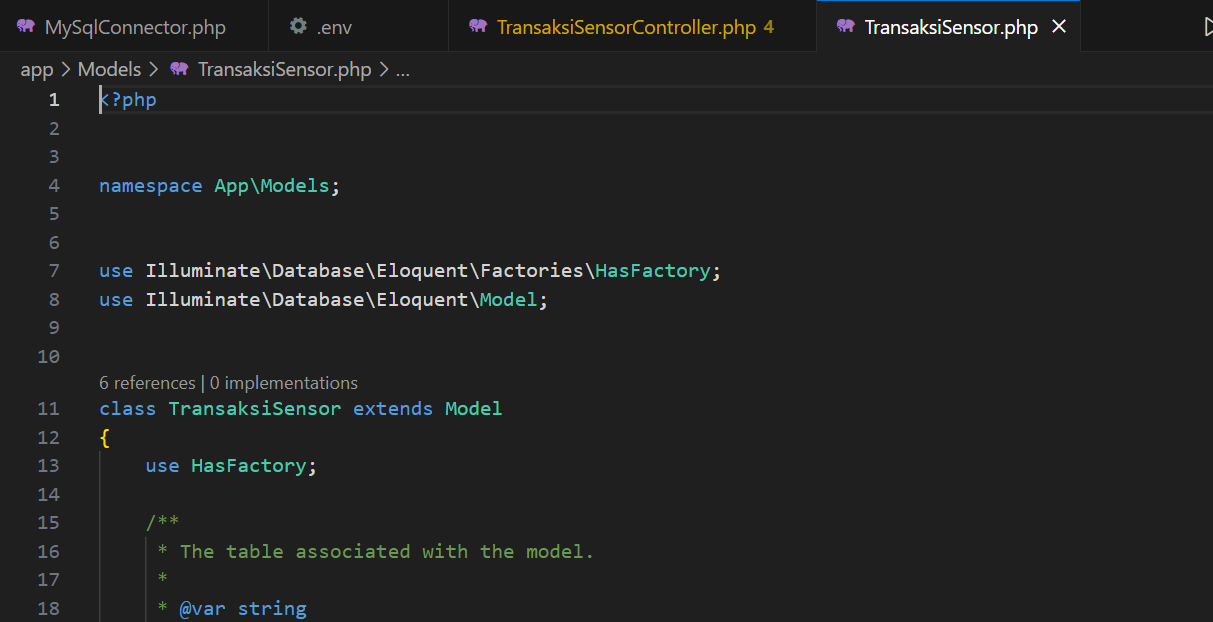
1. Mempelajari dasar-dasar pembuatan API menggunakan Laravel 11
2. Mengimplementasikan database untuk menyimpan data dari sensor
3. Menggunakan Ngrok untuk menghubungkan API dengan perangkat IoT atau simulator Wokwi

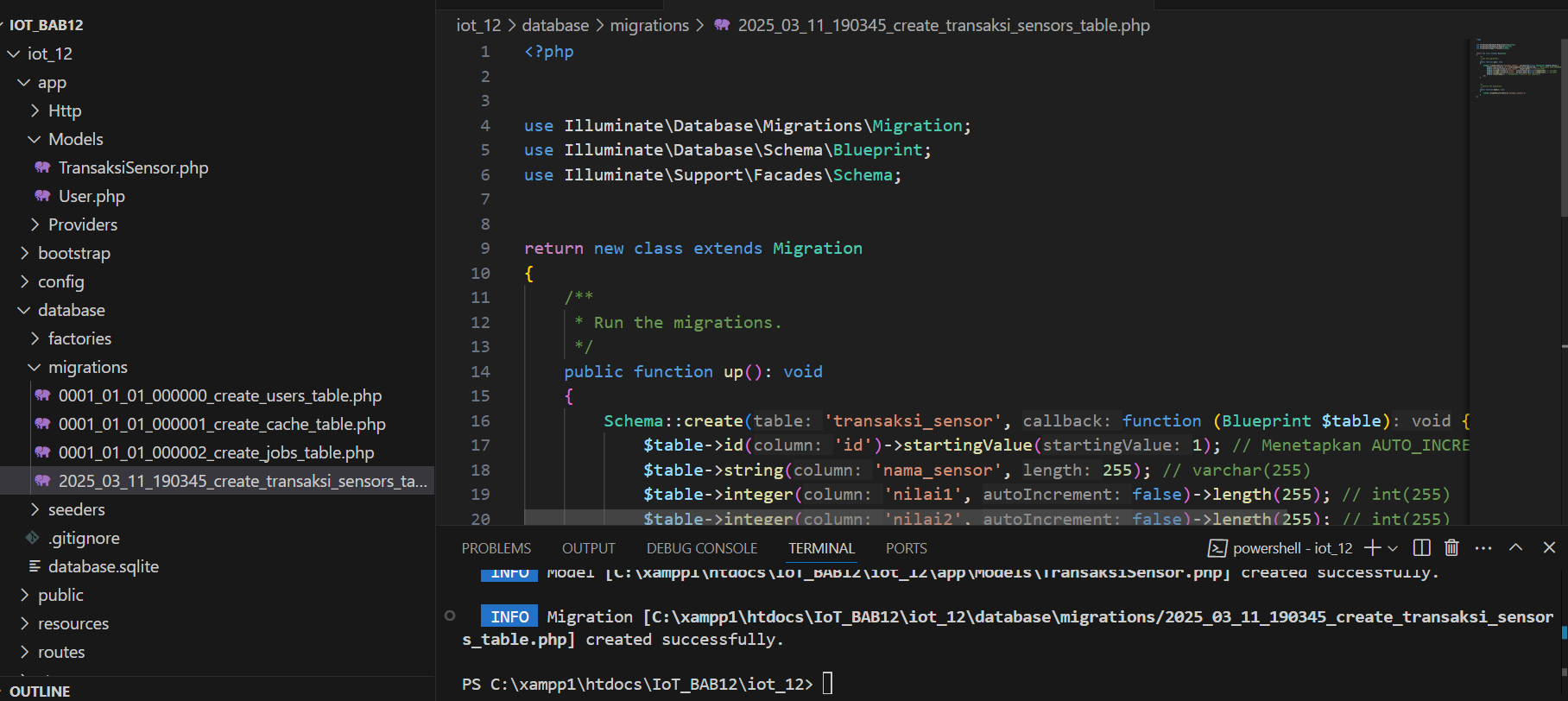
**Metodologi**

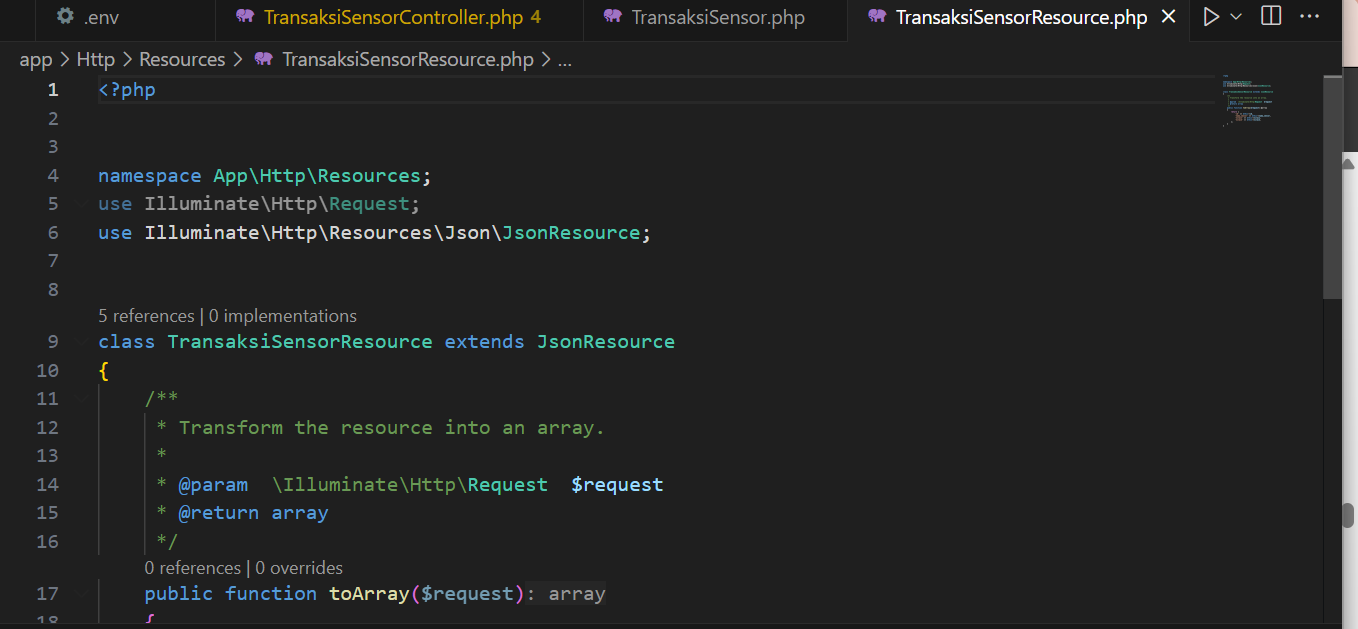
* 1. **Alat dan Bahan**
     + 1. Perangkat komputer dengan koneksi internet
       2. Software: VS Code, Laravel 11, Ngrok, MySQL (phpMyAdmin), Postman, Wokwi Simulator.Modul relay 5V
       3. Akses ke situs Wokwi (<https://wokwi.com>)

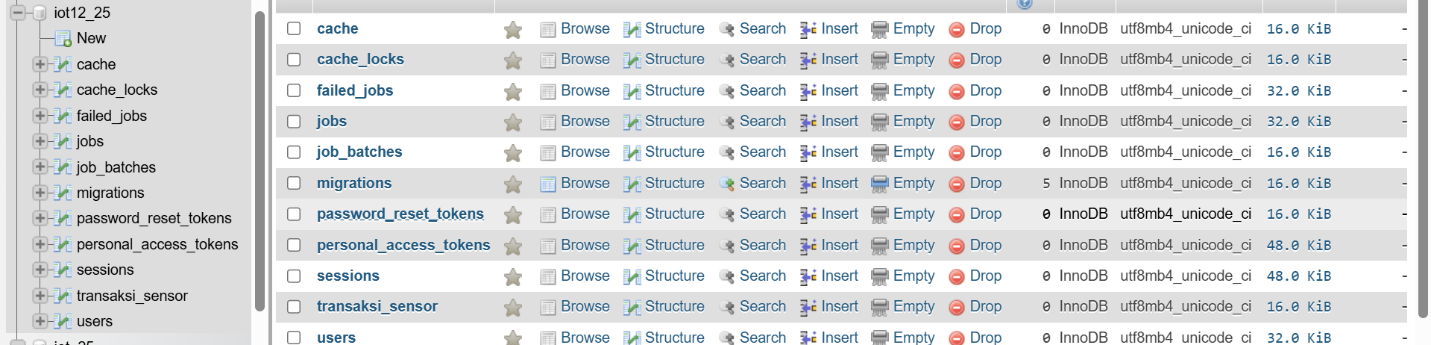
**2.2 Langkah Implementasi**

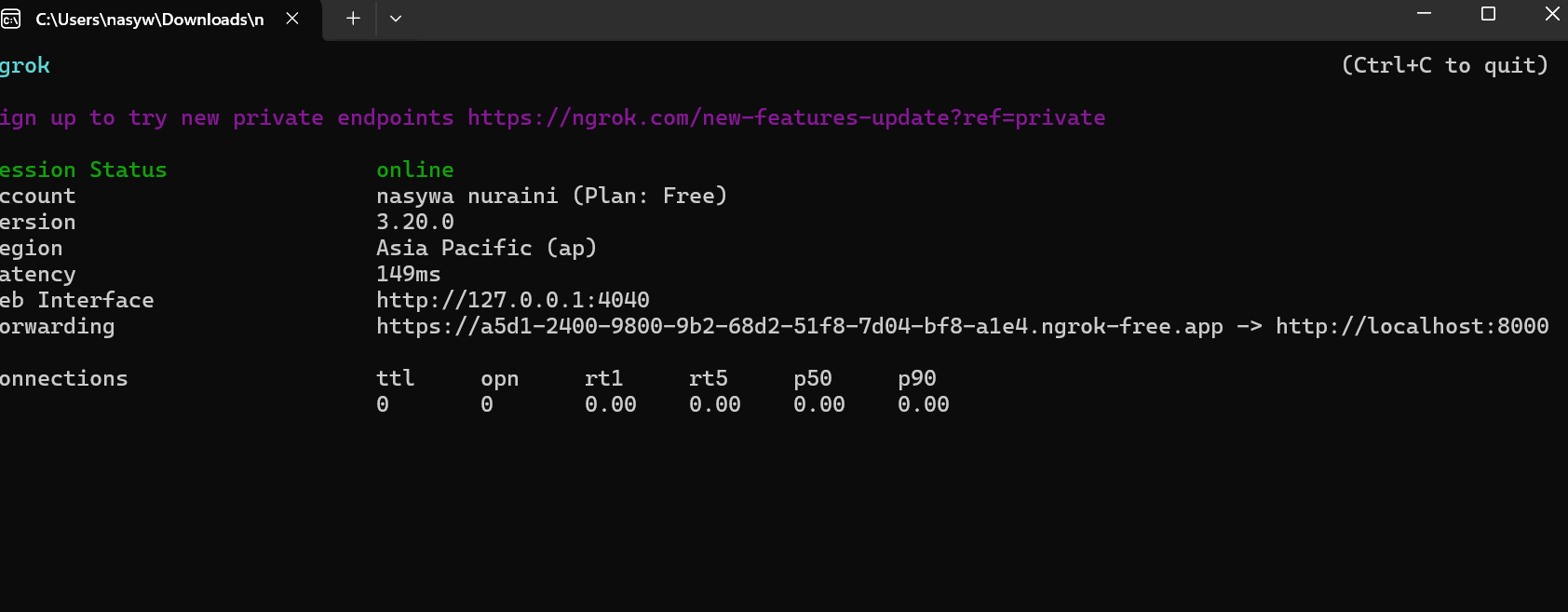


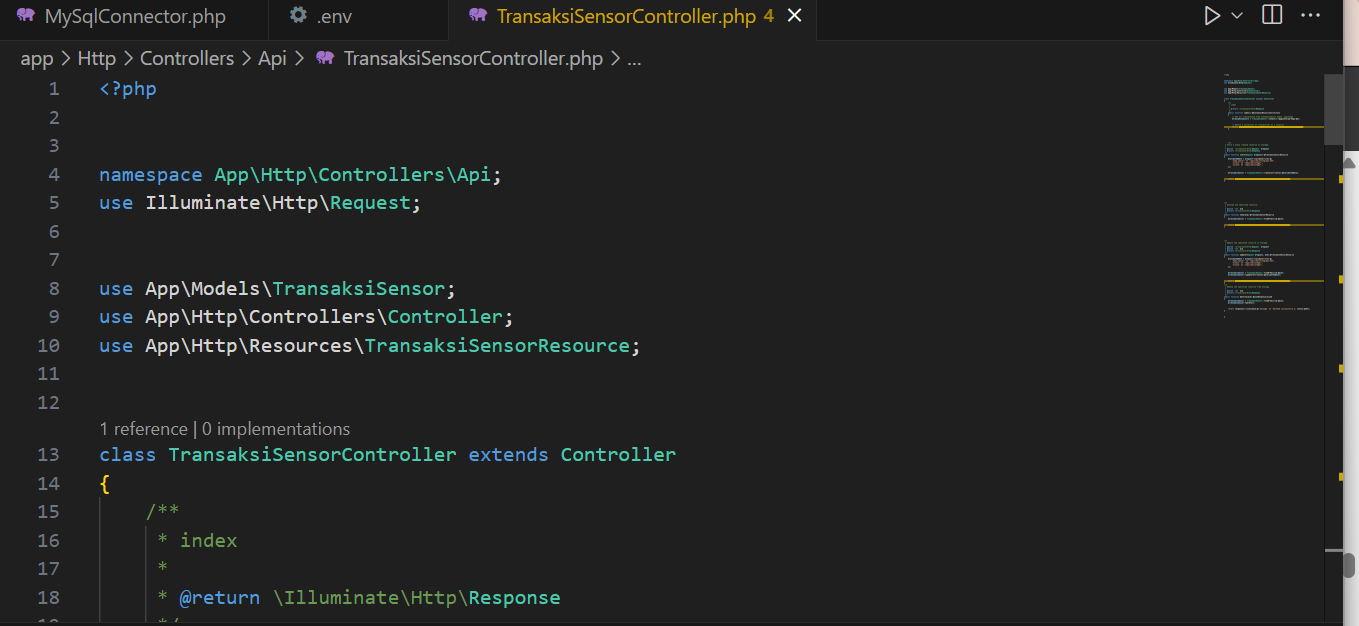








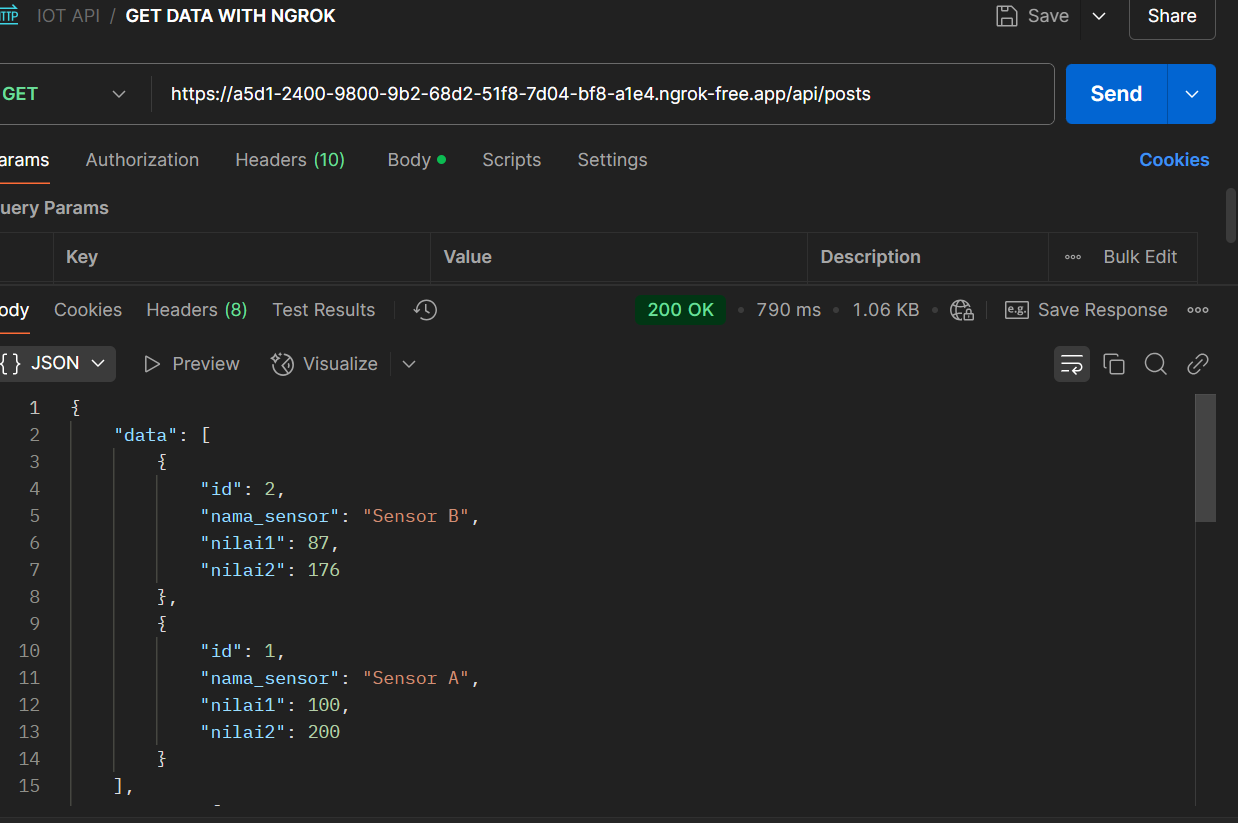


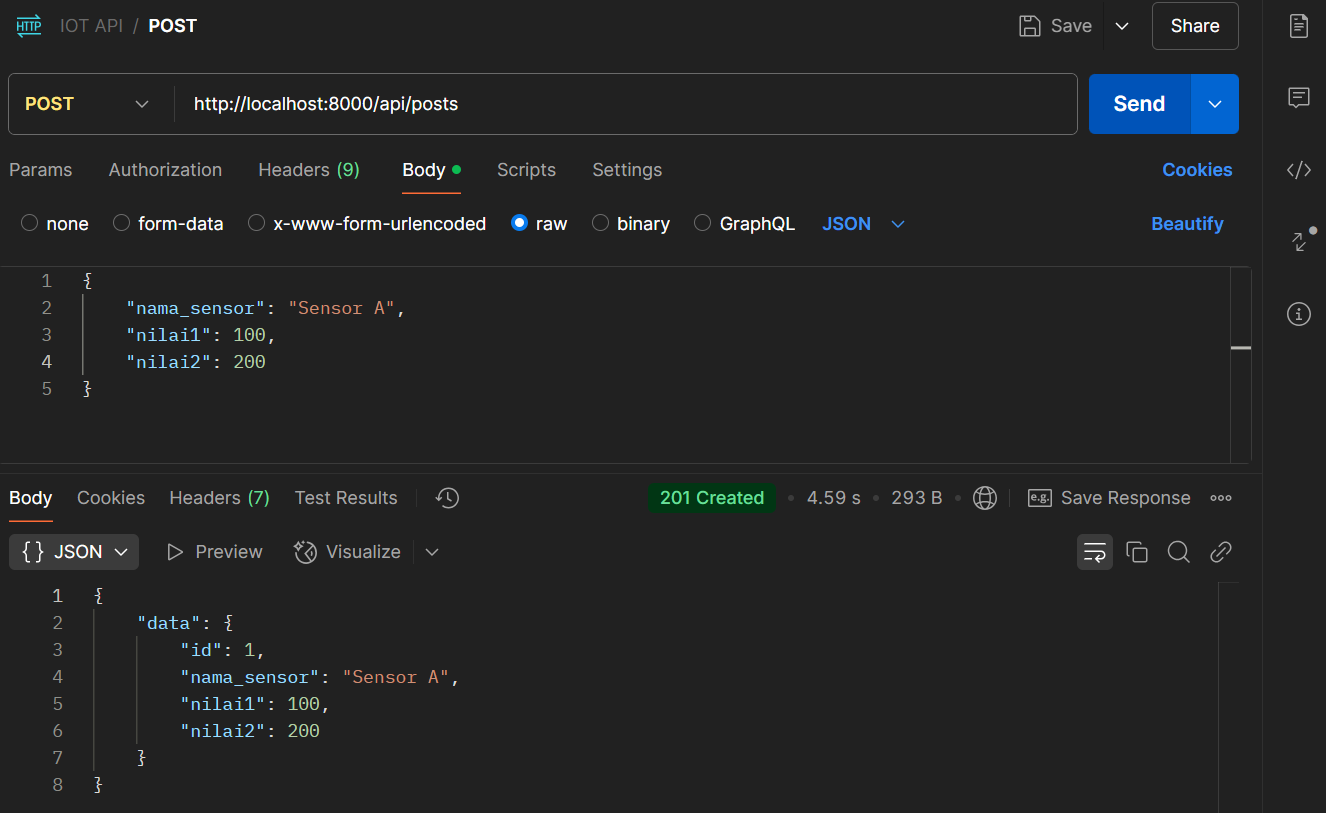


**Hasil dan Pembahasan**

**3.1 Hasil Eksperimen**

* API Laravel berhasil dibuat dan dapat menerima data sensor.
* Ngrok berhasil menghubungkan API ke internet, sehingga dapat diakses dari perangkat lain.
* Data yang dikirim dari simulator Wokwi berhasil masuk ke database MySQL.





**3.2 Pembahasan**

* Penggunaan Laravel 11 dalam pembuatan API sangat efisien karena memiliki fitur bawaan untuk manajemen database dan routing.
* Ngrok mempermudah testing API tanpa harus melakukan konfigurasi server secara manual.
* Pengujian dengan Postman membantu memastikan API berfungsi dengan baik sebelum diterapkan pada perangkat IoT.