

### Topik 3

# Praktik Basic Pemrograman Input, Output, PWM dengan Mikrokontroler beserta Troubleshoot Code

Digitalent Scholarship Professional Academy

**Isi dan elemen dari video ini memiliki hak kekayaan intelektual yang dilindungi oleh undang-undang**

**Dilarang menggunakan, merubah, memperbanyak, dan mendistribusikan video ini untuk tujuan komersil.**

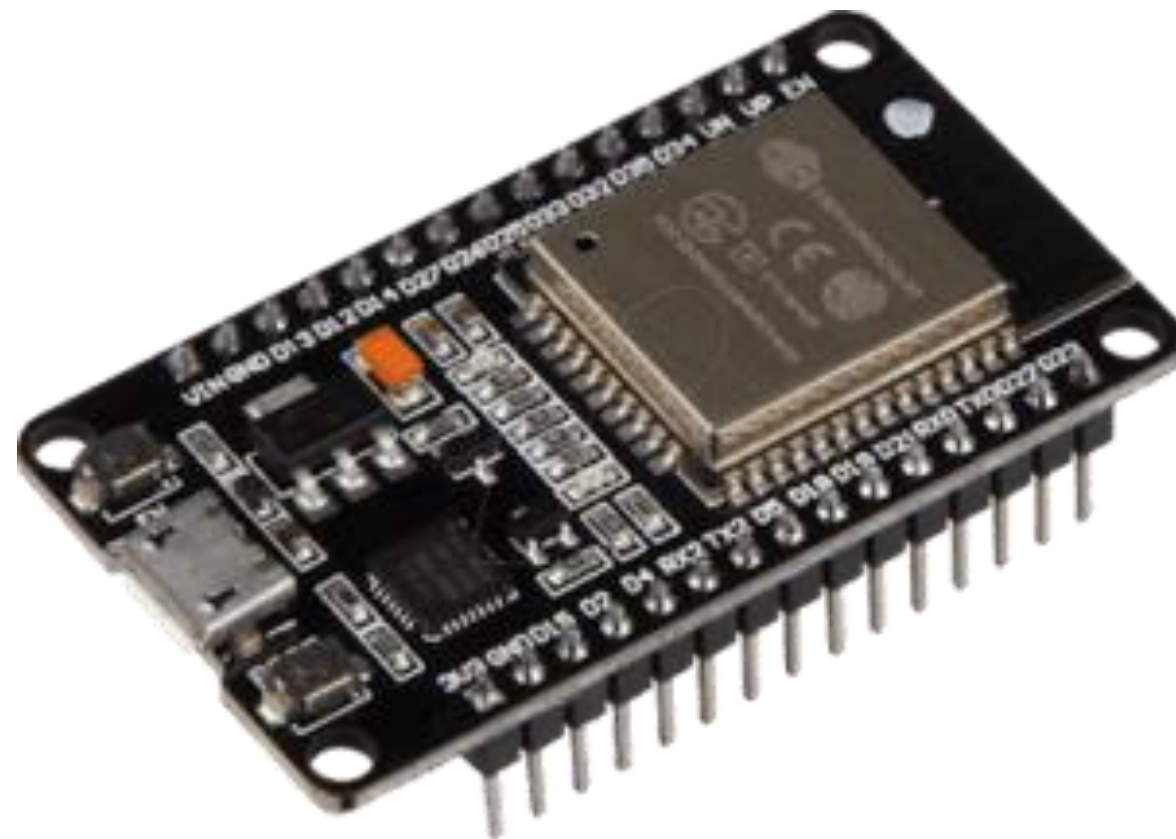
# Outline

- Pengenalan ESP32
- Datasheet dan Pinout ESP32
- Input
  - Button
  - Switch
  - Potensiometer
  - Sensor Ultrasonic
  - Aktif HIGH dan Aktif LOW
- Output
- PWM (Pulse Width Modulation)
- Serial Monitor
- Praktik dengan Wokwi
- Praktik menyalakan LED dengan Input Push Button
- Praktik menyalakan 3 LED dengan 1 Push Button
- Praktik mengendalikan LED dengan PWM
- Challenge : Membuat Running LED
- Praktik dengan Sensor Ultrasonic ditampilkan di Serial Monitor
- Challenges : Pengukuran Jarak dengan Indikator LED



# Pengenalan ESP32

ESP32 merupakan salah satu mikrokontroler yang dikenalkan oleh Espressif System. Pada mikrokontroler ini sudah tersedia modul WiFi dan Bluetooth didalam chip sehingga sangat mendukung untuk diaplikasikan untuk project berbasis Internet of Things



# Datasheet dan Pinout ESP32

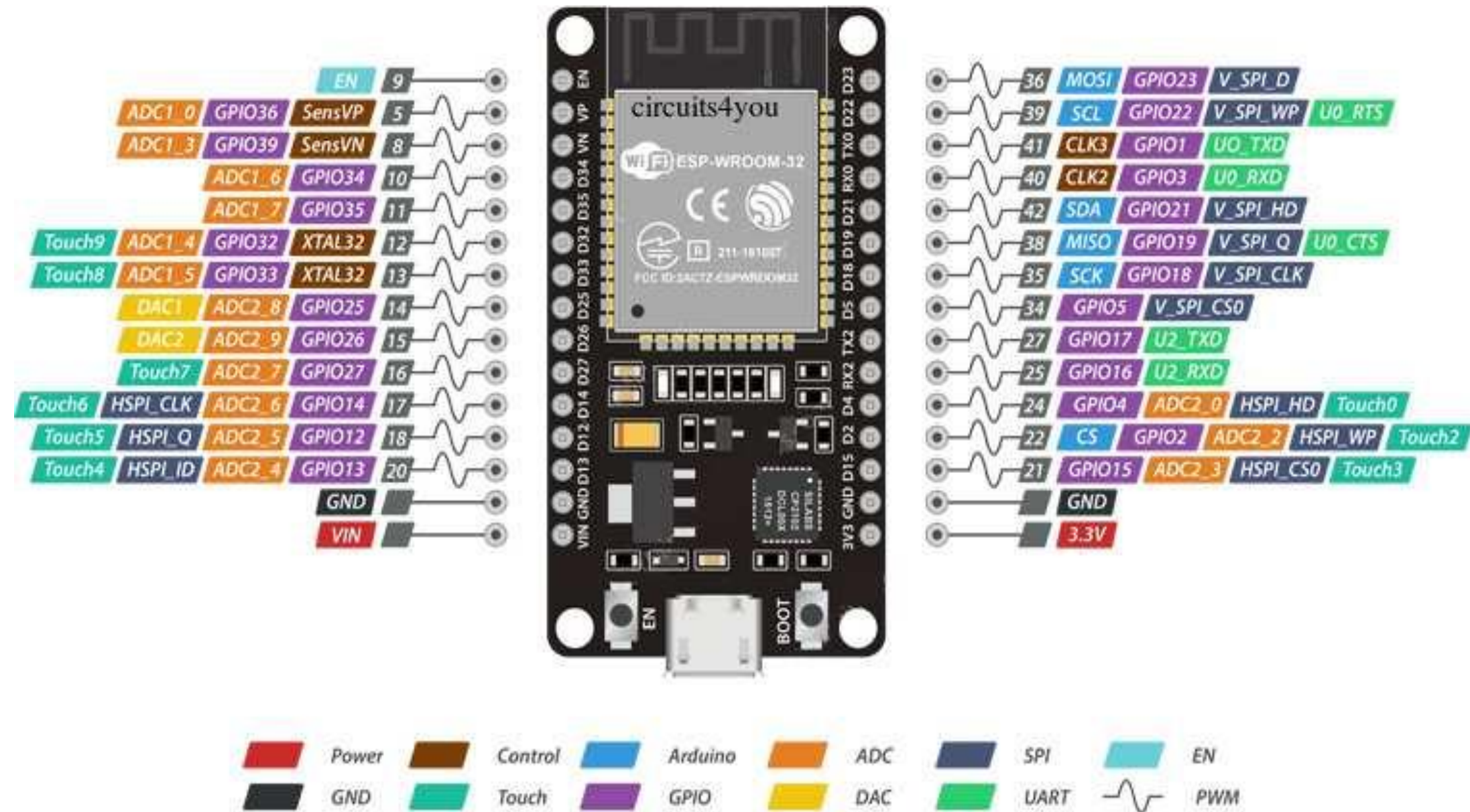
Tegangan Input	5 volt
Tegangan Operasi	5 volt
ADC Pin	18 buah
DAC Pin	2 buah
Flash Memory	128 Kb
SRAM	320 Kb
Clock Speed	240 MHz
Berat	25 gr
PXL	58,6 x 29 mm
Komunikasi	WiFi, Bluetooth, I2C, SPI ,Serial





# Datasheet dan Pinout ESP32

ESP32 memiliki total 48 pin yang multi fungsi. Penggunaan pin berbeda-beda tergantung fungsinya. Untuk detail pin dapat dilihat pada gambar disamping.

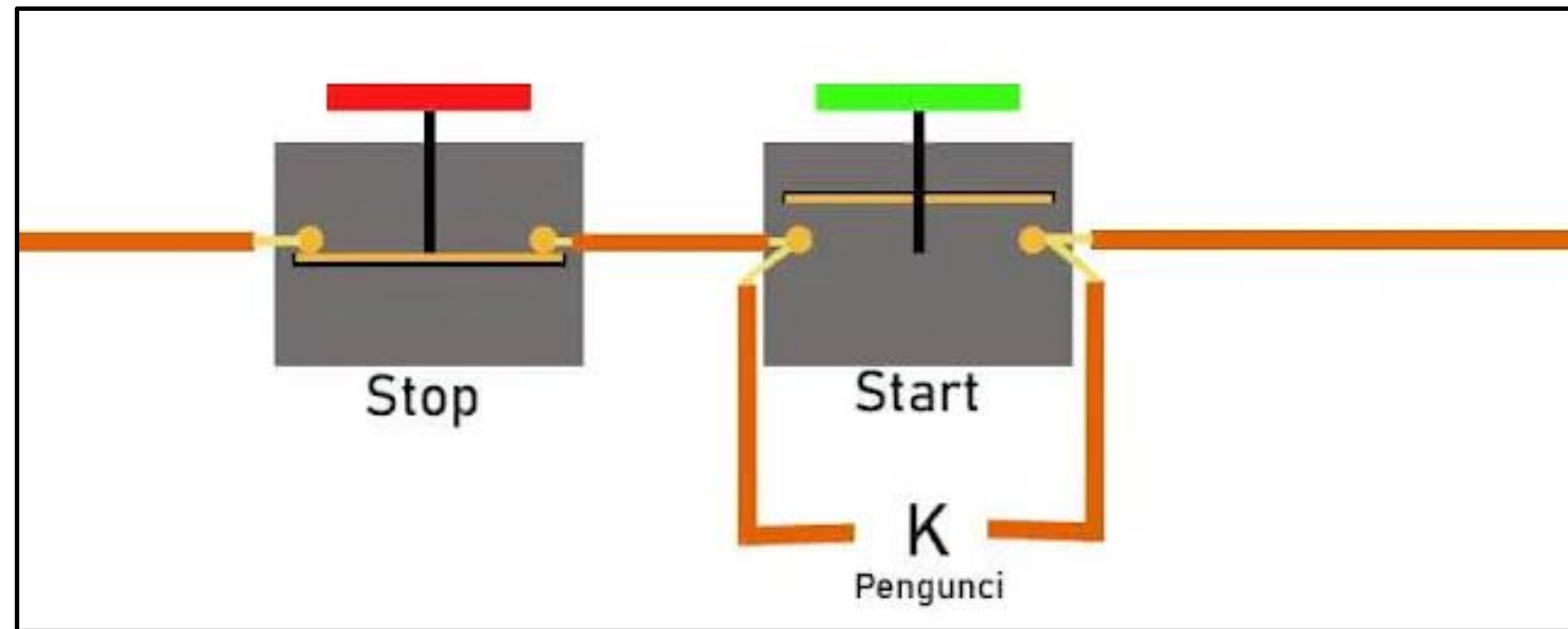


# Input

## Button

Push button merupakan tombol yang digunakan sebagai input pada rangkaian elektronika.

Push button bersifat aktif saat ditekan, namun non-aktif saat push button dilepas, artinya push button harus selalu mendapatkan tekanan untuk dapat bekerja.



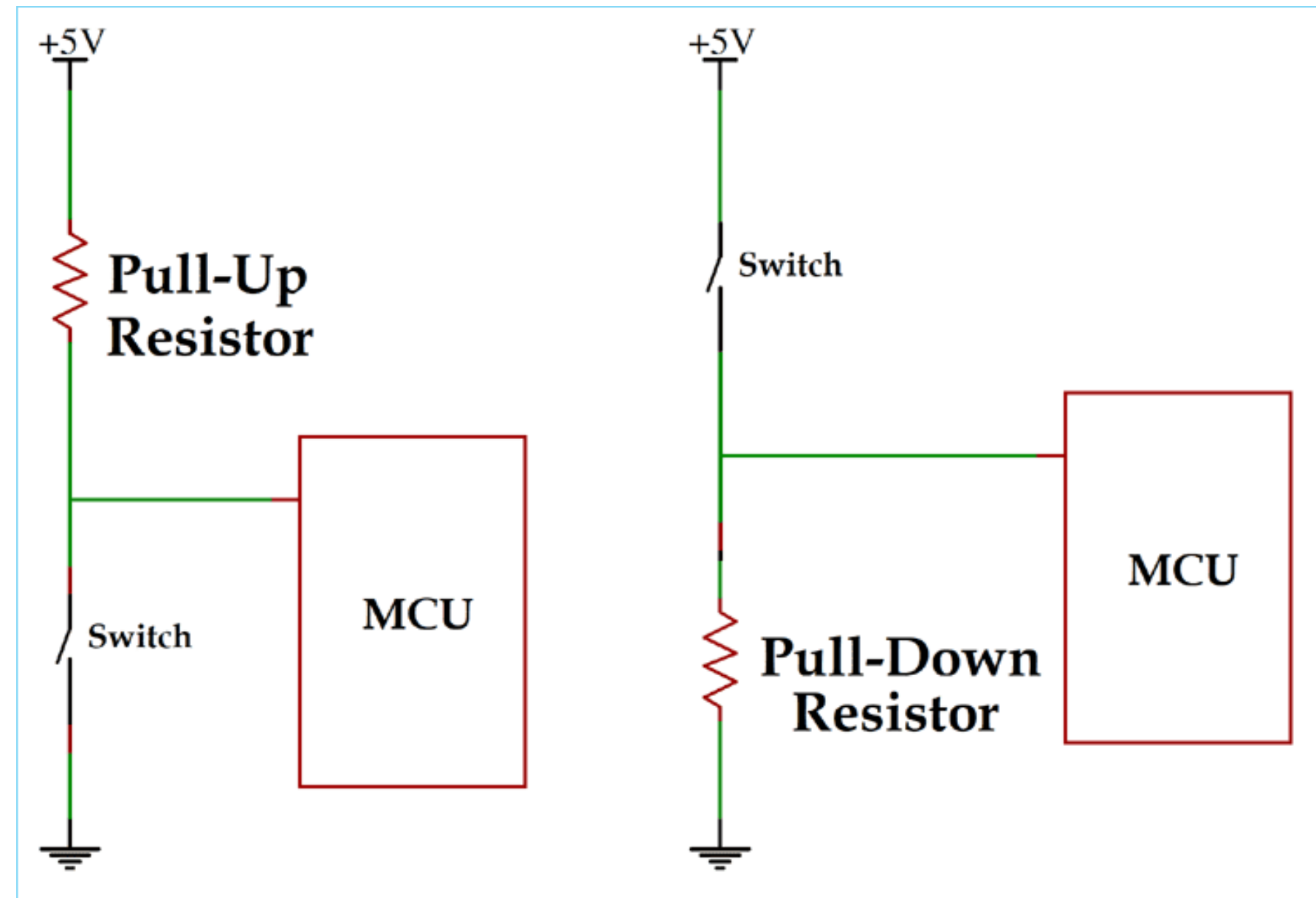
# Input

## Button

Pemasangan Push Button akan lebih jika menggunakan konfigurasi resistor secara pull up atau pull down tergantung kondisi. Resistor pull up dan pull down adalah resistor yang digunakan untuk memastikan input pada komponen benda pada kondisi HIGH atau LOW

Hal tersebut dikarenakan terdapat fenomena floating pada mikrokontroler karena tegangan yang diterima tidak memenuhi 5 volt dan juga 0 volt. Sehingga mikrokontroler tidak dapat menentukan apakah kondisinya HIGH atau LOW

Catatan : HIGH adalah kondisi aktif atau menyala. LOW adalah kondisi Padam

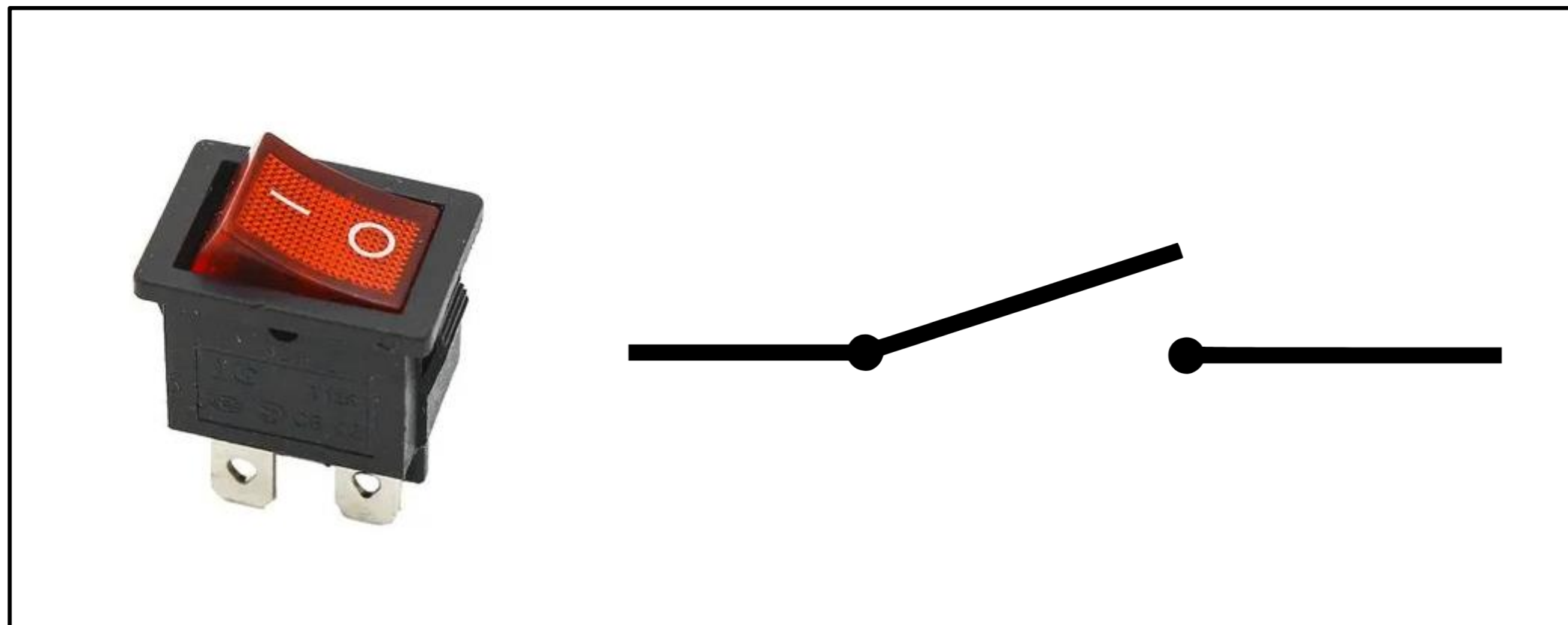




# Input

## Switch

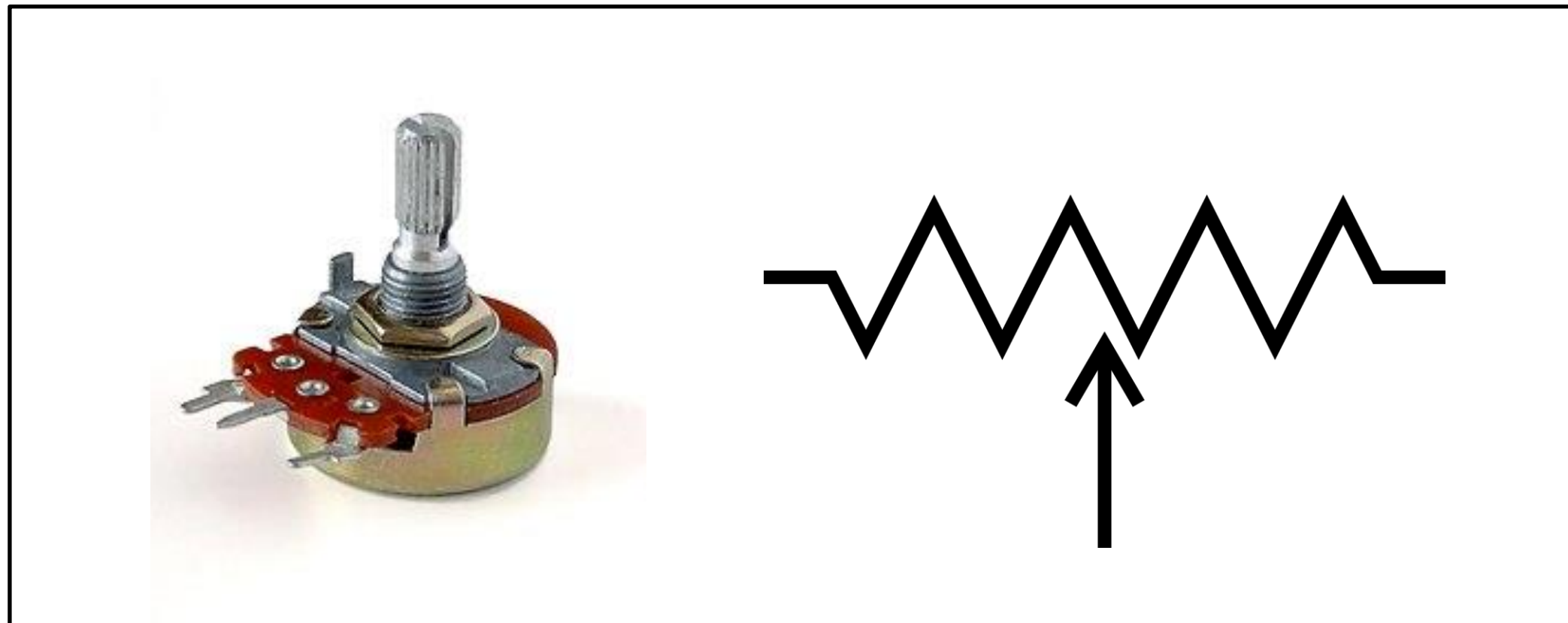
Pada dasarnya switch juga merupakan tombol yang digunakan sebagai input pada rangkaian elektronika. Perbedaannya dengan push button adalah switch bersifat aktif saat ditekan sekali, dan tetap aktif saat switch dilepas. Untuk mematikan switch dapat ditekan satu kali lagi atau ditekan pada bagian lawannya (tergantung jenis switch).



# Input

## Potensiometer

Potensiometer atau resistor variabel merupakan jenis resistor yang nilai resistansinya dapat diatur sesuai dengan kebutuhan penggunaannya. Potensiometer terdiri dari sebuah tuas yang berfungsi mengatur resistansi dan 3 kaki terminal.



# Input

## Sensor Ultrasonic

Sensor ultrasonik adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi/gelombang ultrasonik) menjadi besaran listrik dan sebaliknya.

Sensor ultrasonik dapat digunakan untuk mengukur jarak dengan suatu benda tertentu.

Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat didengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi ultrasonik bisa merambat melalui zat padat, cair dan gas.



# Input

## Cara Kerja Sensor Ultrasonic

Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu.

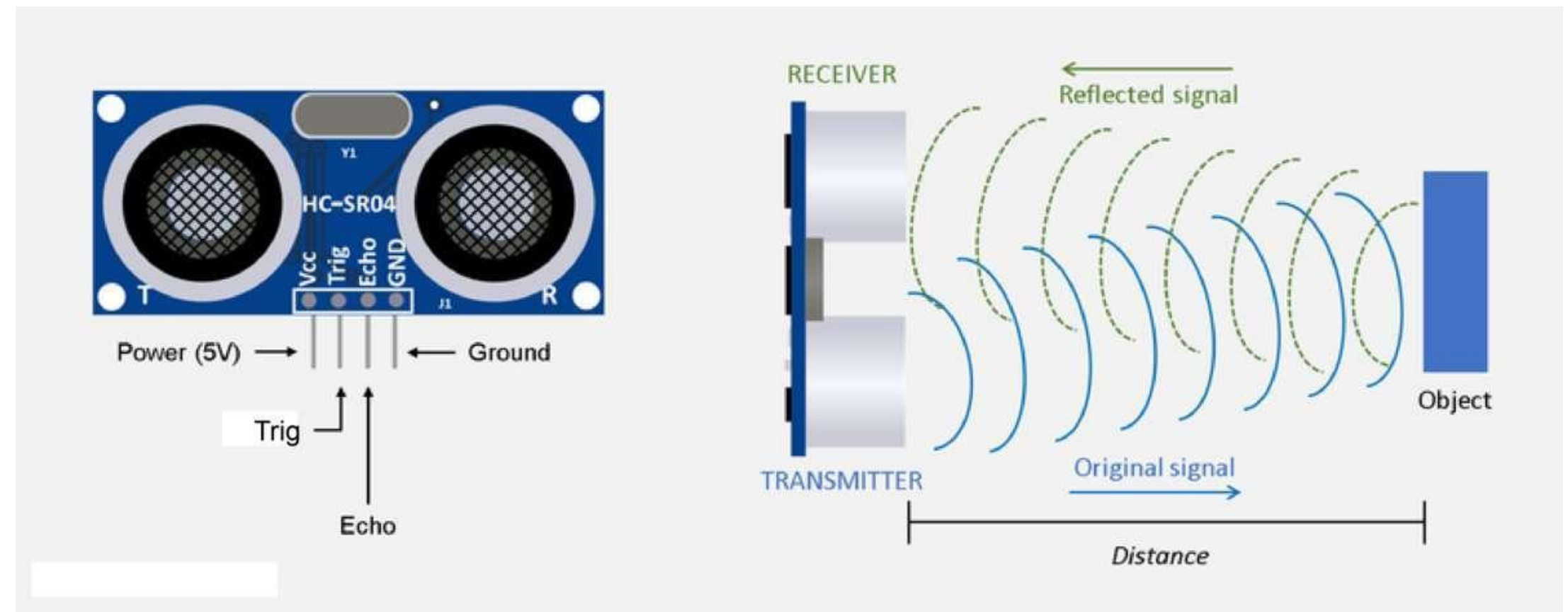
Kalkulasi pengukuran jarak dari sensor menggunakan rumus di bawah ini di mana:

S = Jarak (Cm)

T = Waktu (detik)

Dan 340 adalah nilai kecepatan suara

$$S = \frac{340 \cdot T}{2}$$



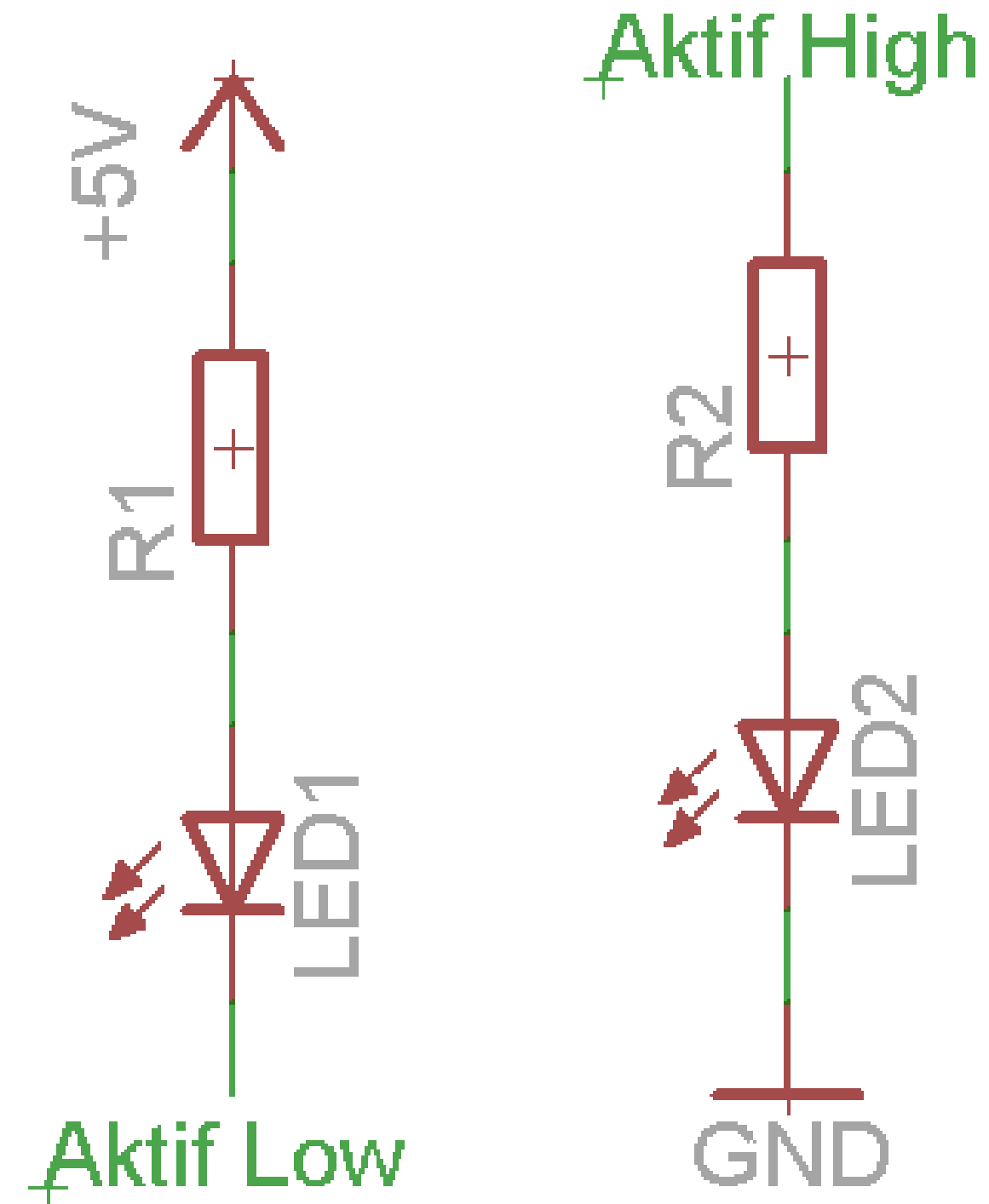


# Input

## Sensor Ultrasonic

Mikrokontroler dapat menerima input high dan low untuk melakukan suatu perintah. Active High merupakan rangkaian dimana saat diberikan logika high maka rangkaian akan aktif.

Sebaliknya, Active Low merupakan rangkaian dimana saat diberikan logika low maka rangkaian akan aktif.



# Output

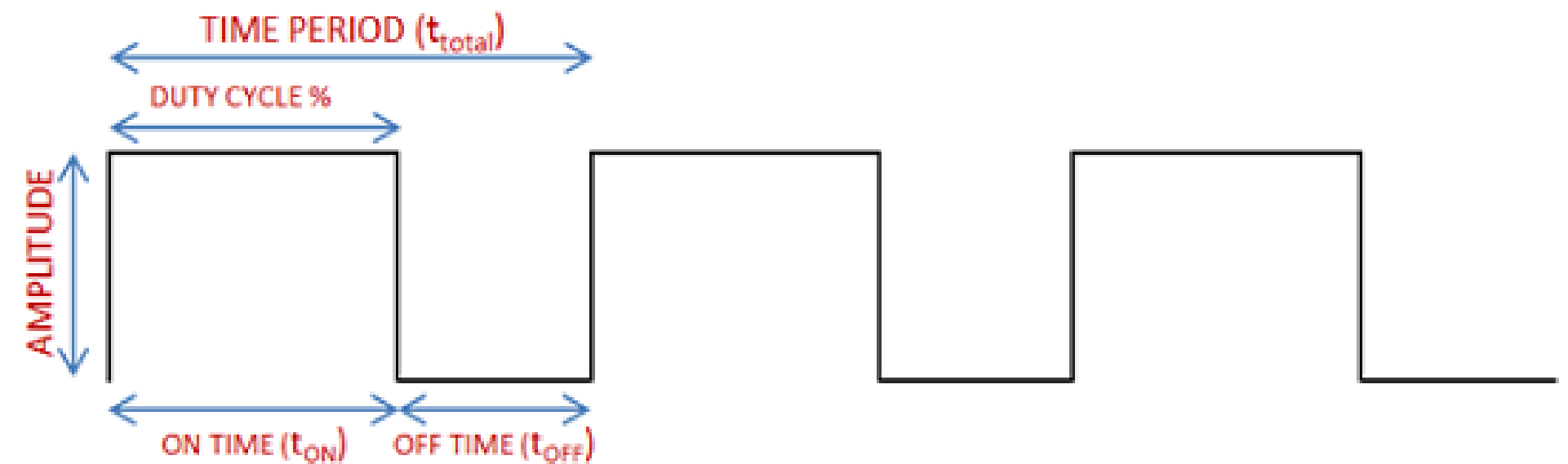
Mikrokontroler dapat mengeluarkan respon berdasarkan program yang kita buat. Keluaran tersebut dapat digunakan untuk mengontrol komponen dan juga perangkat lainnya seperti aktuator, lampu LED, buzzer dan lain-lain.



# PWM (Pulse Width Modulation)

PWM merupakan teknik modulasi yang mengubah lebar pulsa (pulse width) dengan nilai frekuensi dan amplitudo yang tetap. PWM adalah kebalikan dari ADC (Analog to Digital Converter) yang mengubah sinyal Analog ke Digital, PWM digunakan menghasilkan sinyal analog dari perangkat Digital seperti mikrokontroler msialnya.

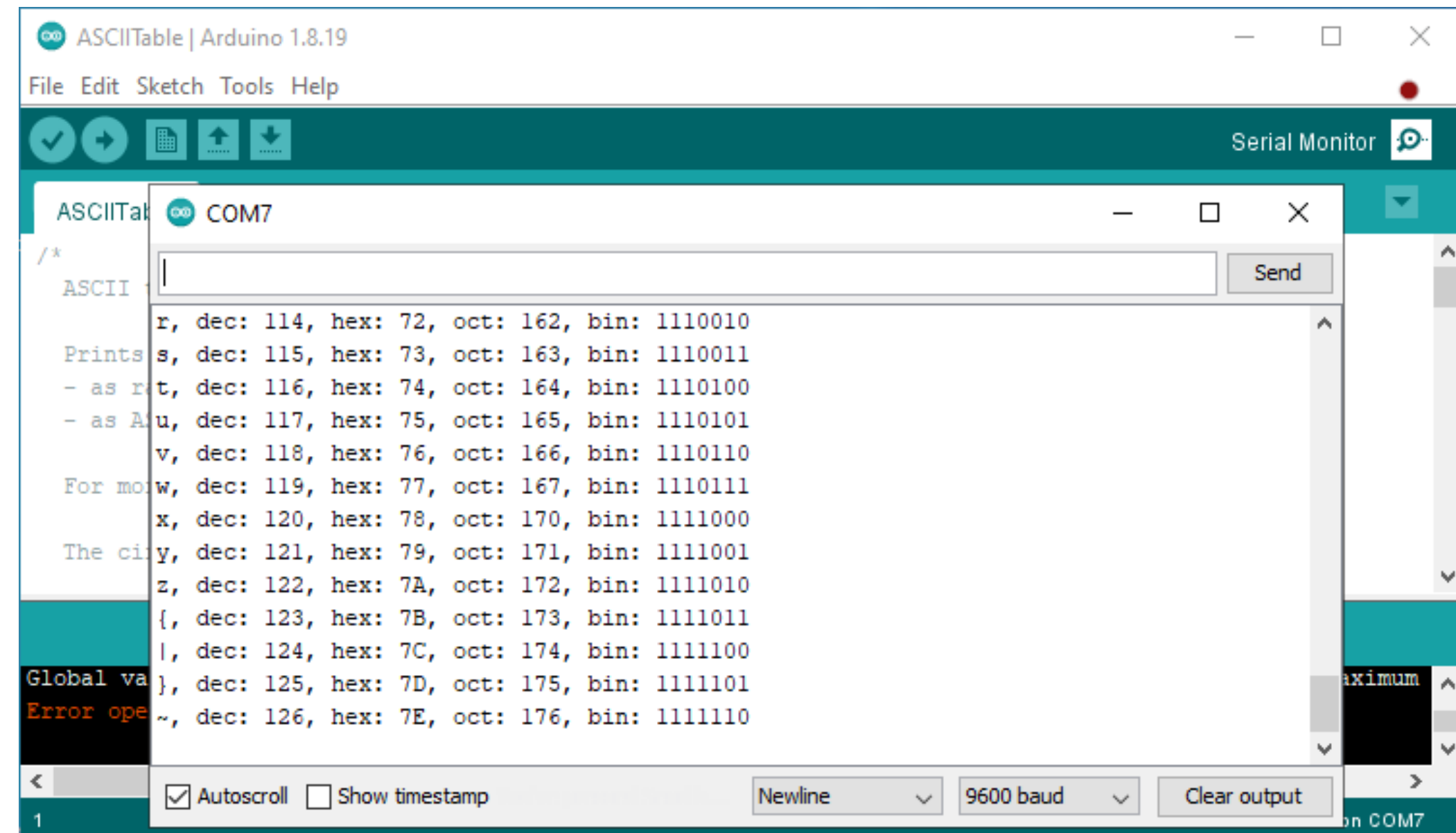
Sinyal PWM akan tetap HIGH untuk waktu tertentu dan kemudian terhenti atau LOW selama sisa periodenya. Yang membuat PWM ini istimewa dan lebih bermanfaat adalah kita dapat menetapkan berapa lama kondisi HIGH harus bertahan dengan cara mengendalikan siklus kerja atau Duty Cycle PWM.



# Serial Monitor

Serial Monitor merupakan alat yang dapat bekerja mengirim dan menerima data serial. Pertukaran data ini terjadi melalui kabel USB yang biasa kita gunakan untuk proses upload sketch dari laptop/PC ke Arduino.

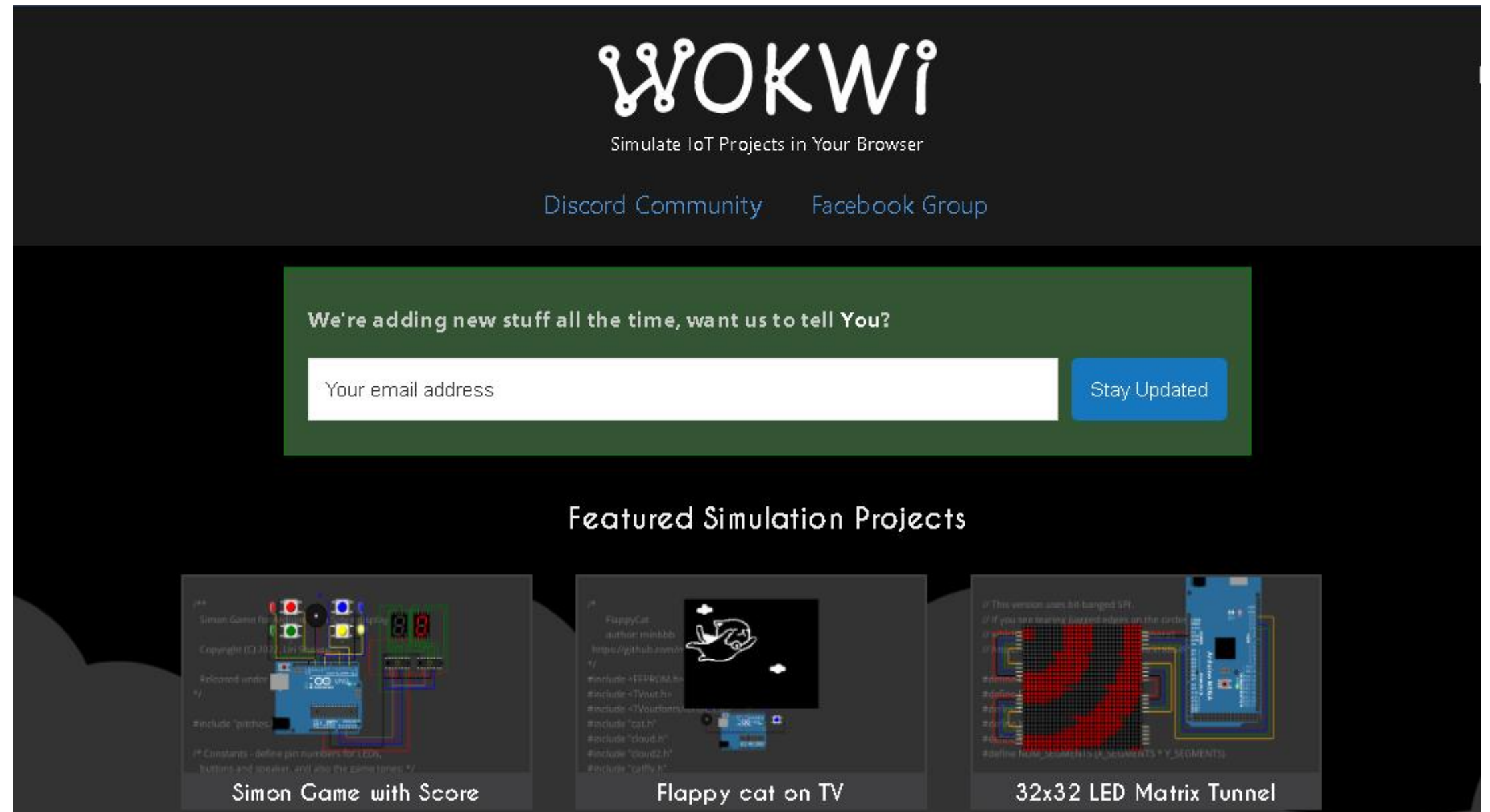
- **Autoscroll** : Merupakan opsi untuk scroll secara otomatis atau tidak.
- **Show Timestamp** : Merupakan opsi untuk menampilkan waktu saat ini atau tidak.
- **Ending selection** : Terdapat beberapa opsi untuk karakter terakhir yang dikirimkan ke Arduino. diantaranya No line ending, Newline, Carriage, Both NL and CR
- **Baudrate** : Terdapat opsi untuk kecepatan baudrate antara Arduino dengan PC. Pastikan nilai baudrate yang telah di deklarasikan sesuai dengan pilihan baudrate pada serial monitor
- **Clear Output** : Membersihkan consol.





# Praktikum Mikrokontroler dengan Wokwi

Wokwi adalah platform yang dapat digunakan sebagai simulator Electronics Development Board seperti Arduino, ESP32, dan Raspberry Pi Pico



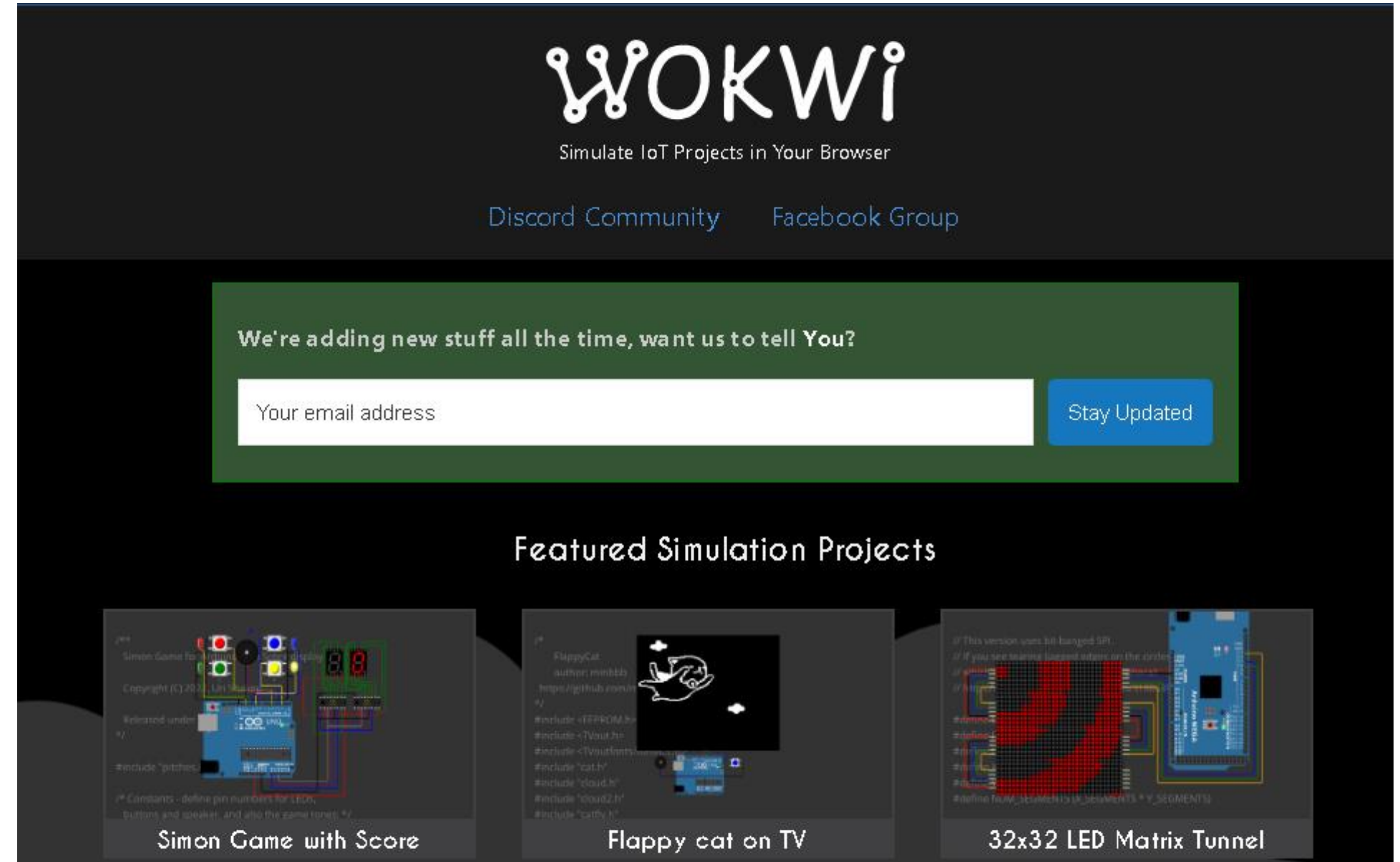
# Praktikum Mikrokontroler dengan Wokwi

Wokwi adalah platform yang dapat digunakan sebagai simulator Electronics Development Board seperti Arduino, ESP32, dan Raspberry Pi Pico

## Kelebihan Wokwi

Wokwi memiliki kelebihan dibandingkan platform simulasi yang lain yaitu:

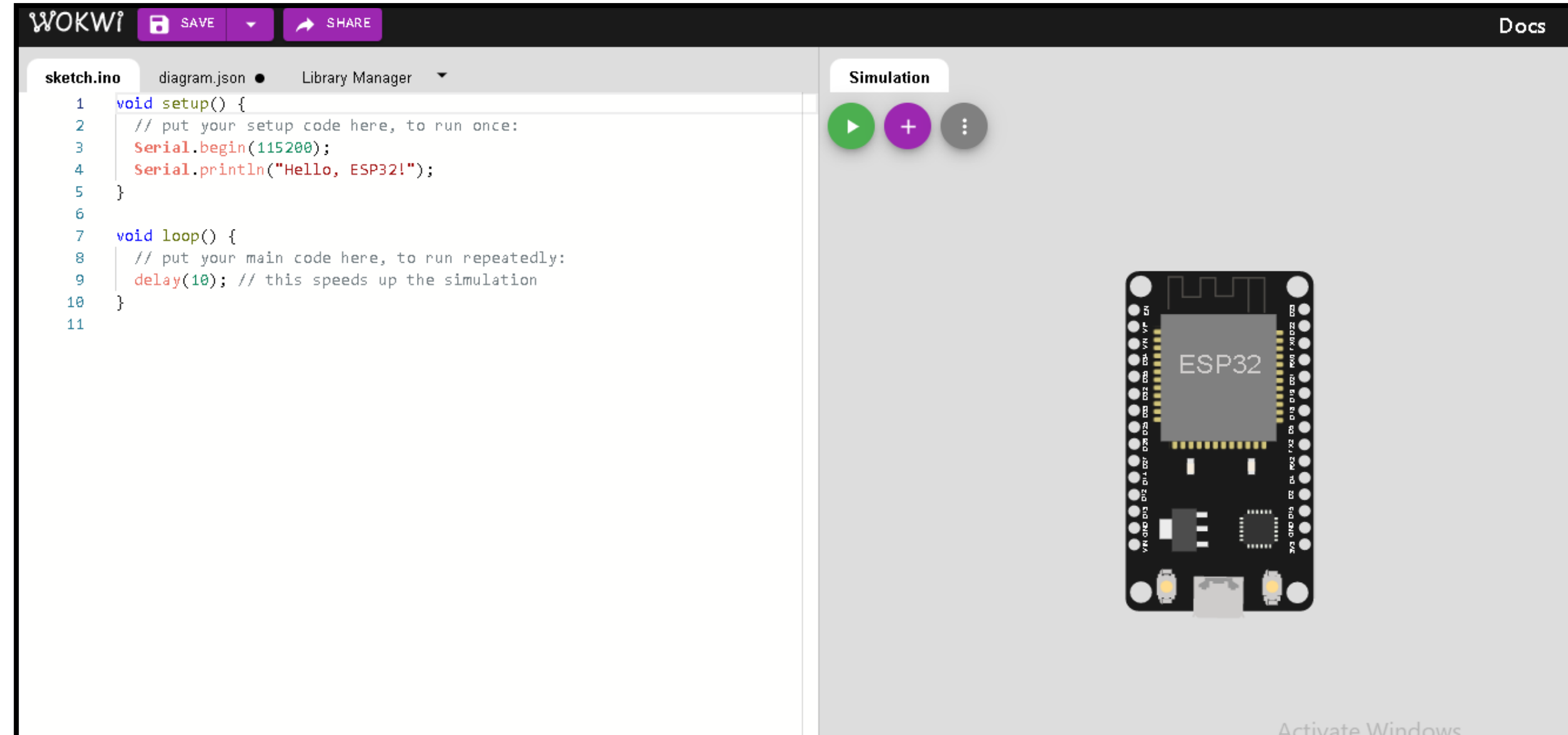
- Komponen pendukung lebih lengkap (sensor, tombol, led, dll)
- Simulasi dapat terhubung ke server asli
- Pilihan board lebih banyak (Arduino, ESP32, Raspberry Pi Pico)
- Hampir mirip dengan software pemrograman sebenarnya, salah satunya dengan fitur menambah library



# Praktikum Mikrokontroler dengan Wokwi

## Memulai Wokwi

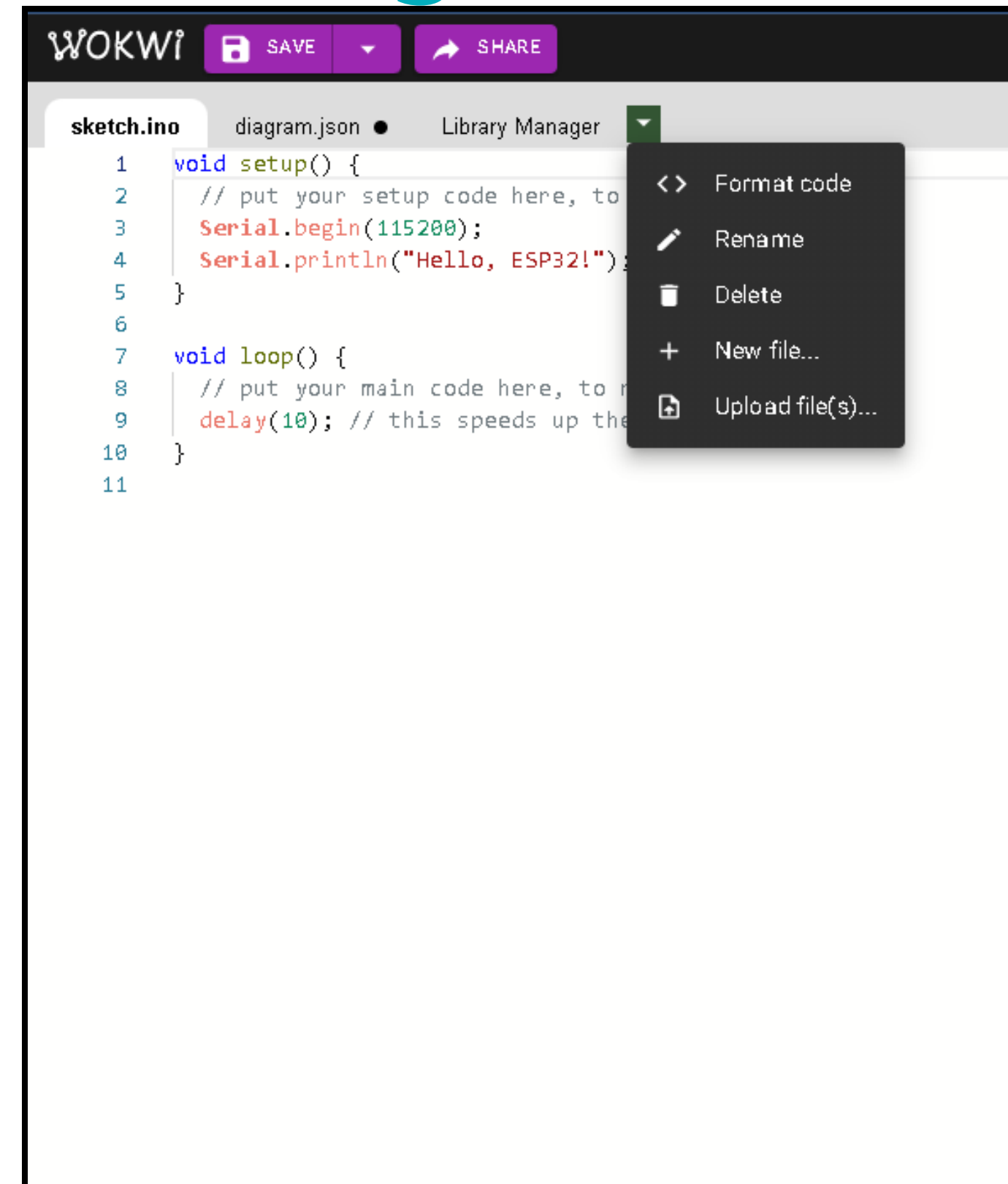
- Klik New Projects untuk membuat projek baru
- Pilih ESP32



# Praktikum Mikrokontroler dengan Wokwi

## Mengenal Fitur Wokwi

- Sketch : tempat/workspace untuk membuat kode program
- Diagram.json : Bagian untuk mengubah rangkaian elektronika yang sedang dibuat
- Library Manager: Untuk menambahkan library, baik dari wokwi atau dari luar
- Option : Menggunakan fitur lain seperti rename, delete, dll.
- Save : Untuk menyimpan program
- Share : Untuk membagikan project

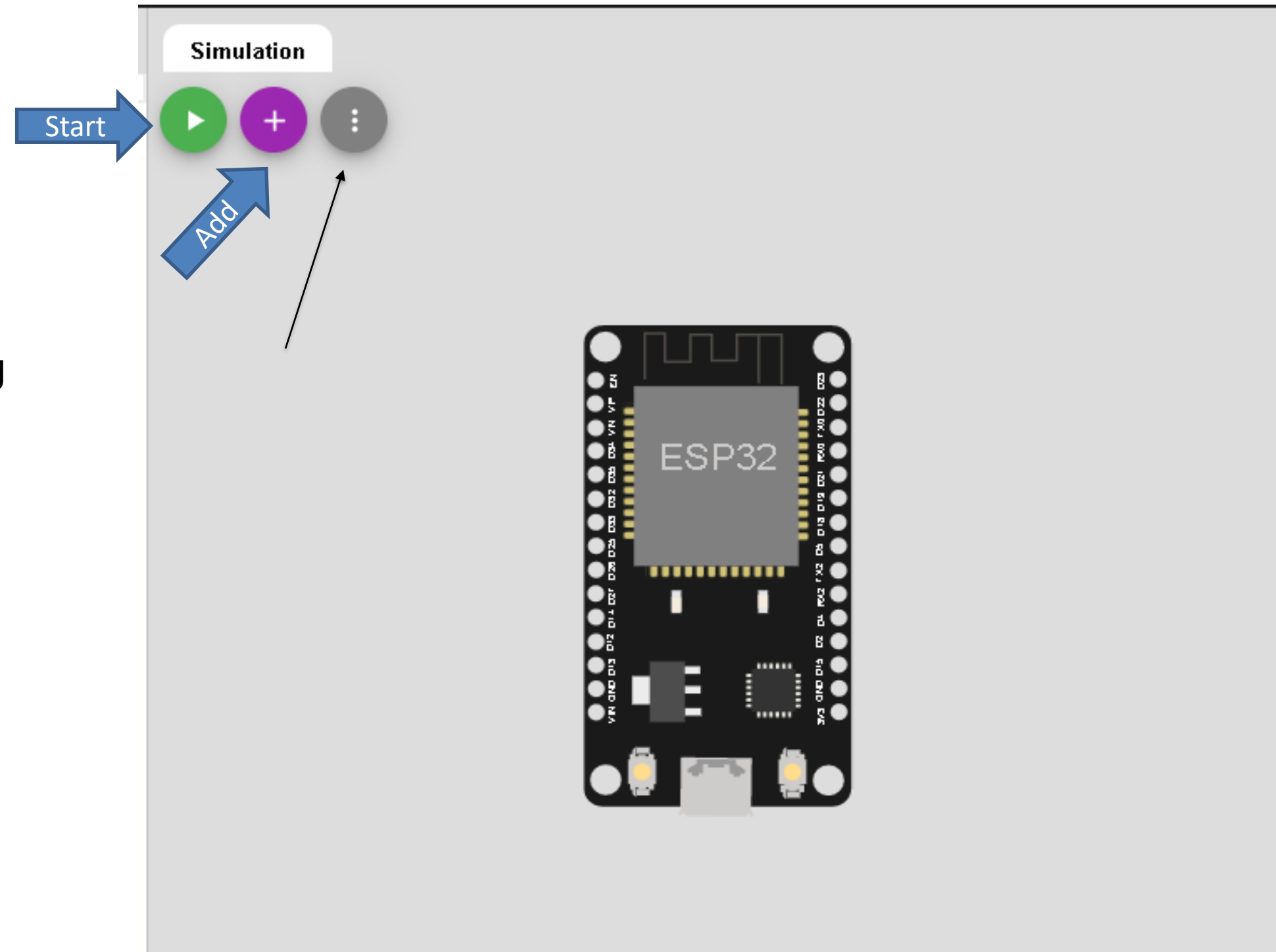




# Praktikum Mikrokontroler dengan Wokwi

## Mengenal Fitur Wokwi

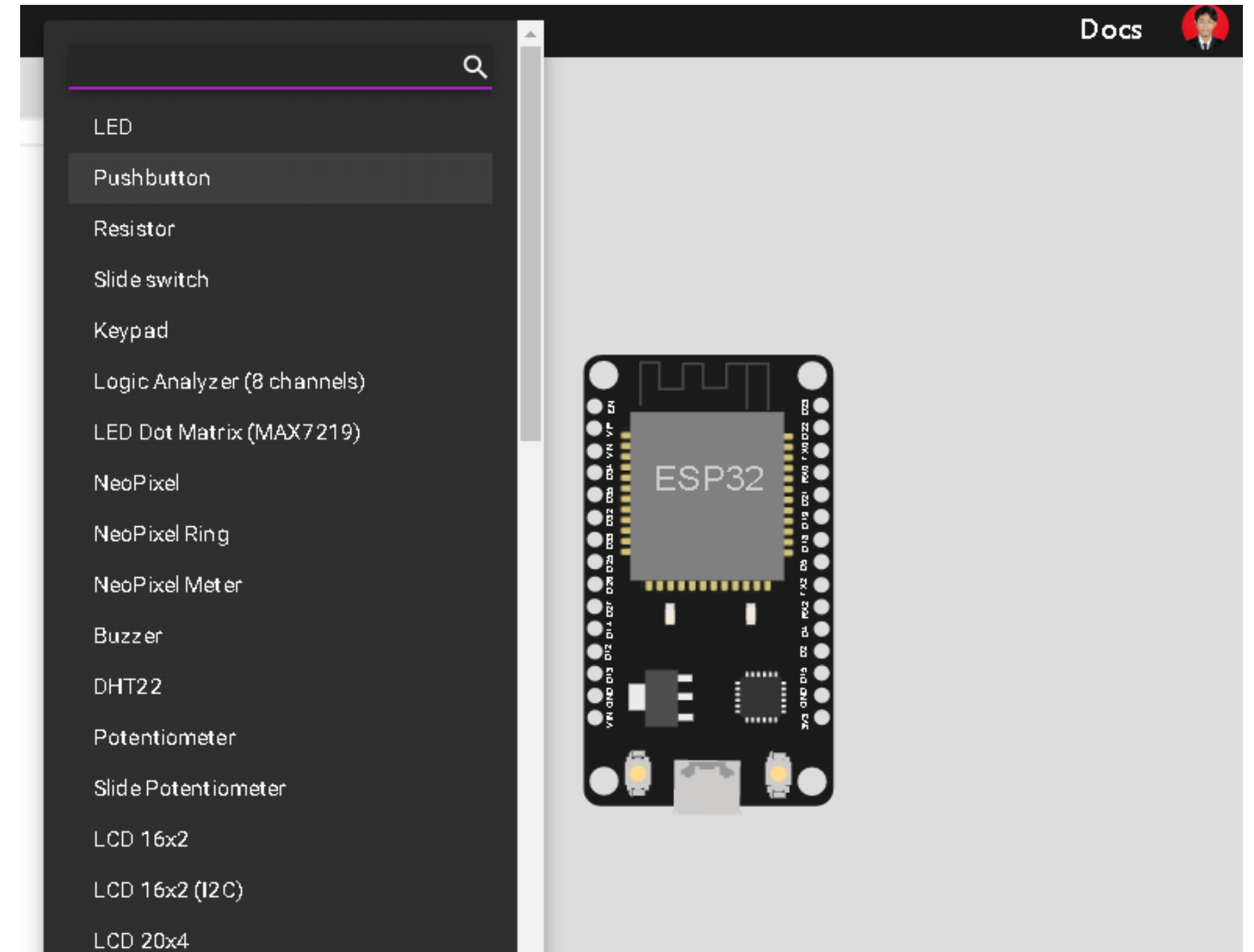
- **Start** : Untuk memulai simulasi
- **Add** : Untuk menambahkan komponen
- **Option** : Untuk melakukan fitur lain seperti zoom, fullscreen, dll.
- **Simulation workspace** : Tempat simulasi berdasarkan kode program yang sudah dibuat
- **Serial monitor** : Menampilkan komunikasi serial pada Board



# Praktikum Mikrokontroler dengan Wokwi

## Menyalakan LED dengan Push Button

- Tambahkan komponen baru dengan klik add (+)
- Pilih resistor, LED dan Push button

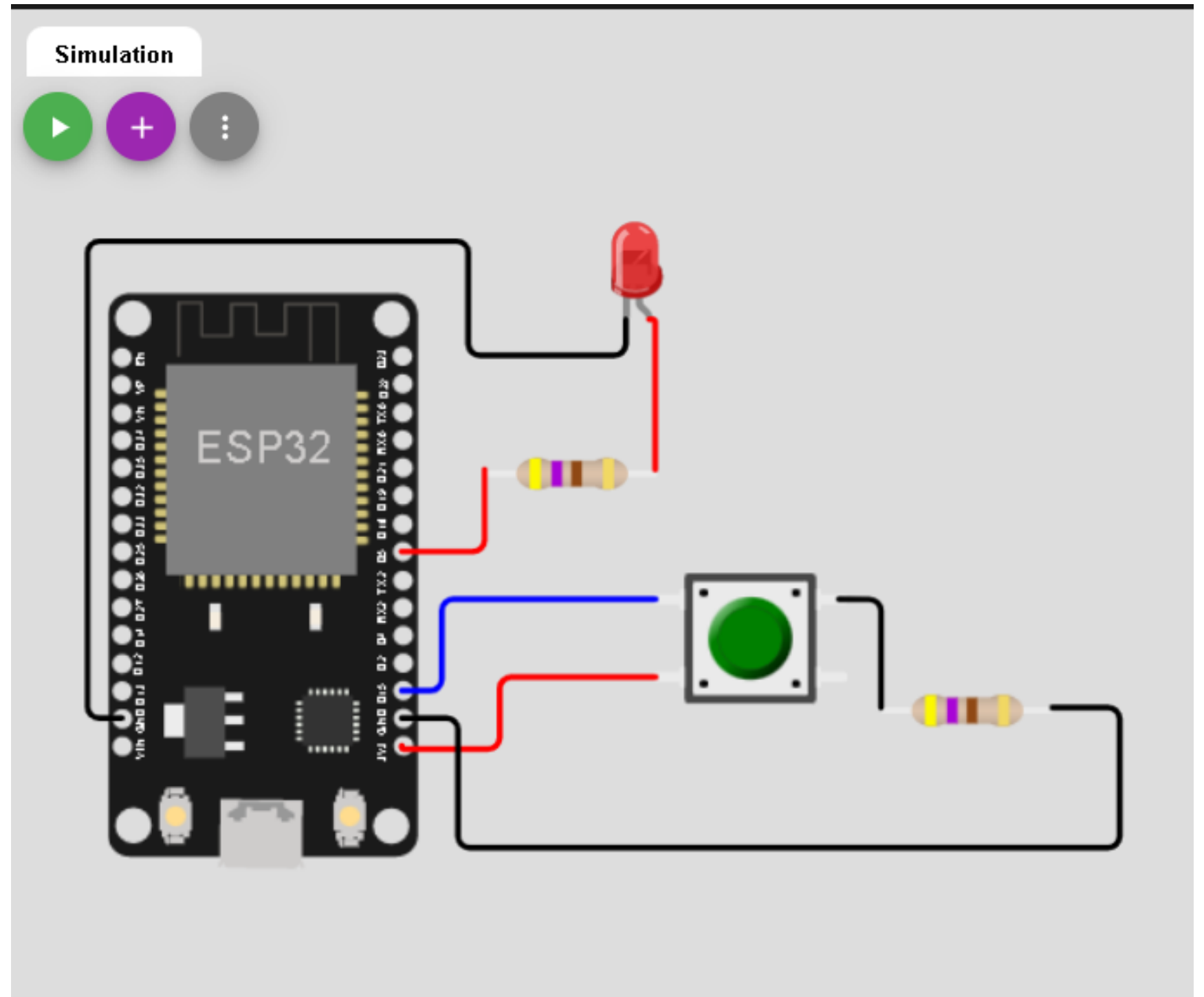


# Praktikum Mikrokontroler dengan Wokwi

## Menyalakan LED dengan Push Button

- Tambahkan push button yang digunakan untuk menyalakan LED.
- Tambahkan juga resistor yang digunakan sebagai PULLDOWN
- Hubungkan push button, resistor, dan arduino seperti pada gambar di samping
- Atur warna kabel melalui diagram.json agar mudah dalam membedakan koneksi pin berdasarkan warna kabel

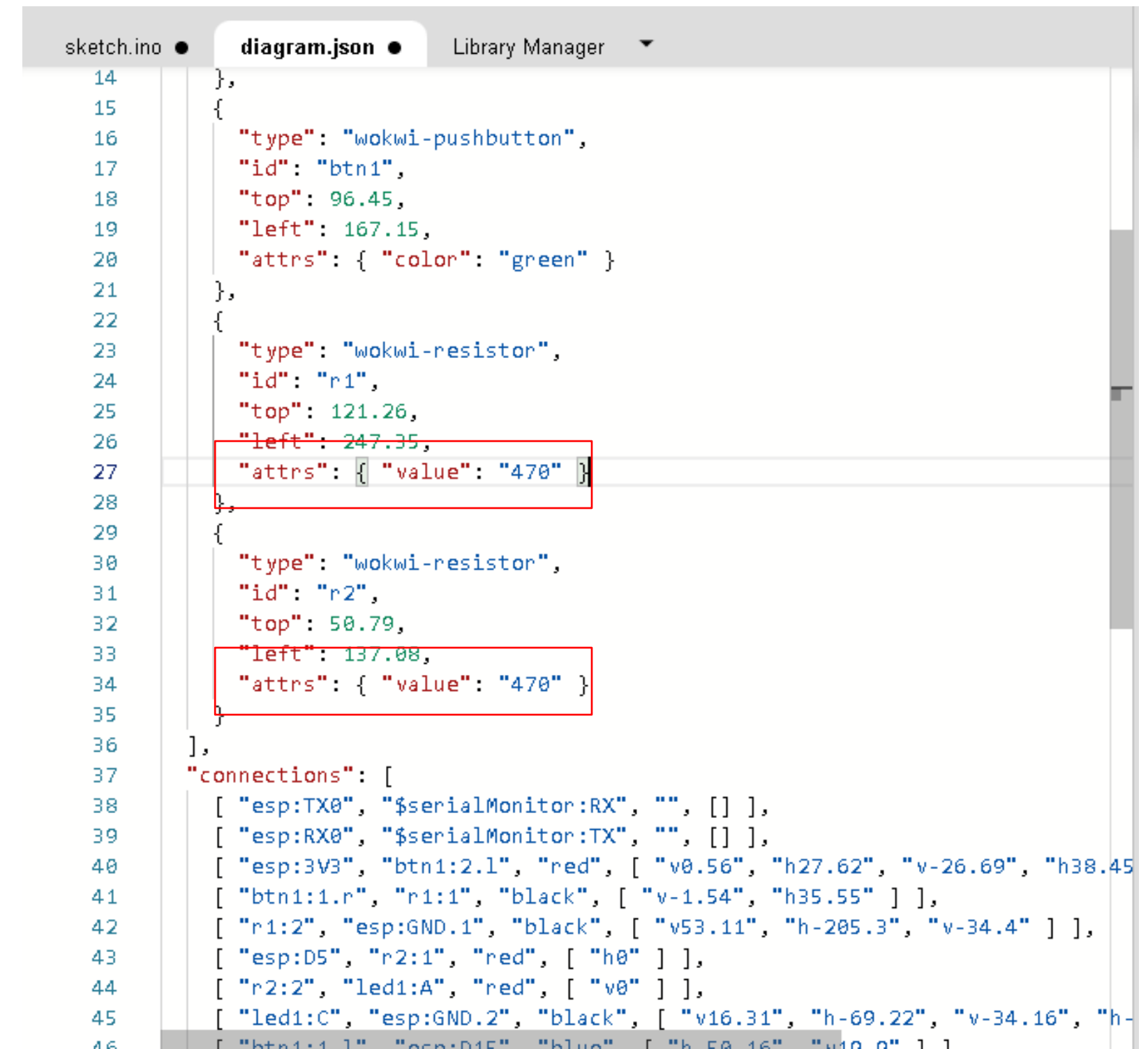
Coding : [Download](#)



# Praktikum Mikrokontroler dengan Wokwi

## Menyalakan LED dengan Push Button

- Untuk mengubah warna kabel, dll masuk ke menu diagram.json
- Ubah warna LED menjadi hijau pada "attrs" wokwi led menjadi green
- Ubah nilai resistor menjadi 470 ohm pada "attrs" menjadi 470
- Pada "connections" merupakan koneksi/kabel pada rangkaian. Kita bisa mengubah koneksi pin, warna kabel, dll.
- Ubah warna kabel GND dengan warna hitam, dll.



```
14 },
15 {
16   "type": "wokwi-pushbutton",
17   "id": "btn1",
18   "top": 96.45,
19   "left": 167.15,
20   "attrs": { "color": "green" }
21 },
22 {
23   "type": "wokwi-resistor",
24   "id": "r1",
25   "top": 121.26,
26   "left": 247.35,
27   "attrs": { "value": "470" }
28 },
29 {
30   "type": "wokwi-resistor",
31   "id": "r2",
32   "top": 50.79,
33   "left": 137.08,
34   "attrs": { "value": "470" }
35 },
36 ],
37 "connections": [
38   [ "esp:TX0", "$serialMonitor:RX", "", [] ],
39   [ "esp:RX0", "$serialMonitor:TX", "", [] ],
40   [ "esp:3V3", "btn1:2.1", "red", [ "v0.56", "h27.62", "v-26.69", "h38.45" ],
41   [ "btn1:1.r", "r1:1", "black", [ "v-1.54", "h35.55" ] ],
42   [ "r1:2", "esp:GND.1", "black", [ "v53.11", "h-205.3", "v-34.4" ] ],
43   [ "esp:D5", "r2:1", "red", [ "h0" ] ],
44   [ "r2:2", "led1:A", "red", [ "v0" ] ],
45   [ "led1:C", "esp:GND.2", "black", [ "v16.31", "h-69.22", "v-34.16", "h-
46   [ "btn1:1.l", "esp:D15", "blue", [ "h 50.16", "v10.0", "1 1
```



# Praktikum Mikrokontroler dengan Wokwi

## Praktik Menyalakan LED dengan Push Button

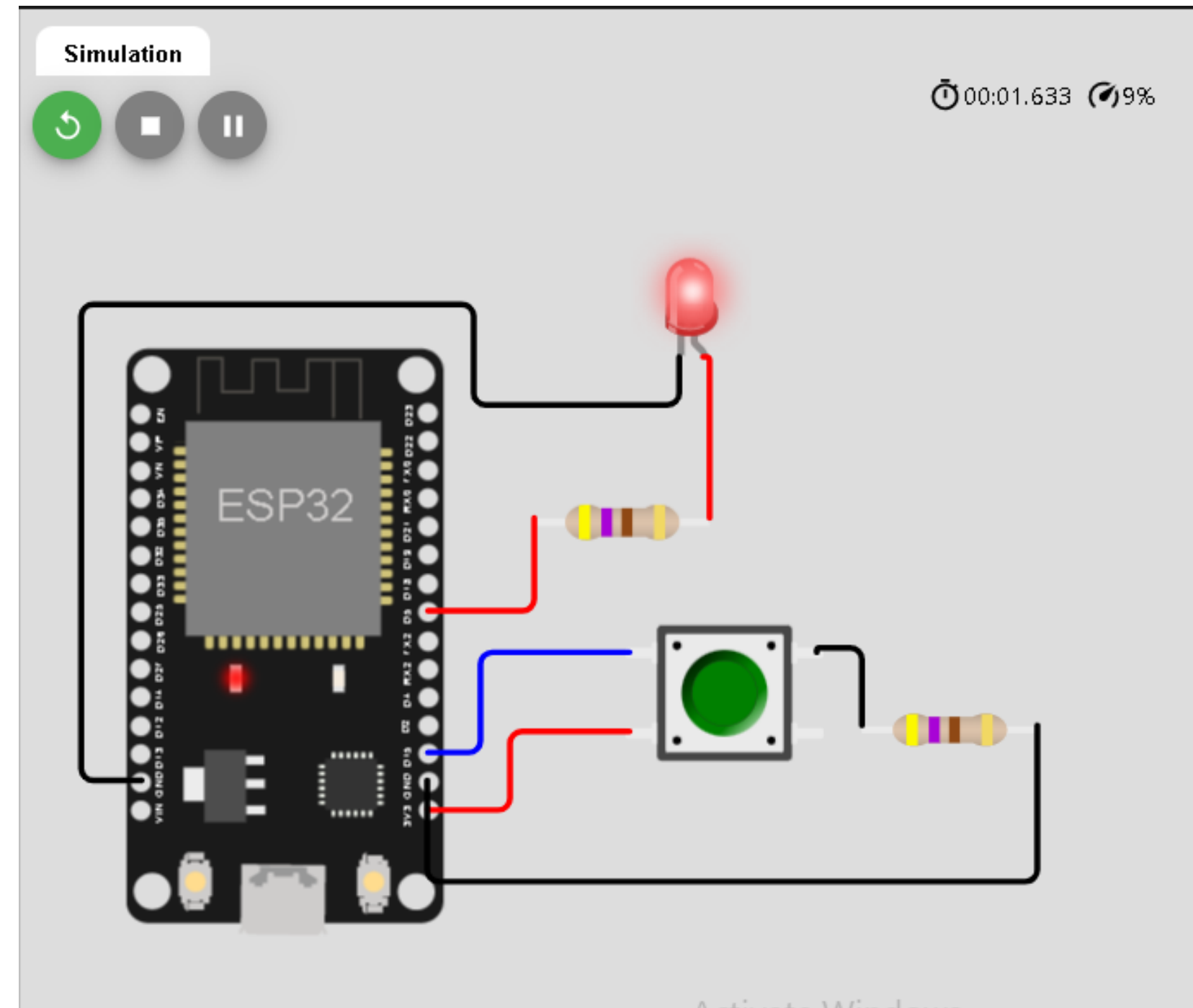
- Buatlah kode program seperti gambar disamping
- Program tersebut akan menyalakan LED saat Push button ditekan
- Jalankan simulasi dengan klik Start Simulation

```
sketch.ino • diagram.json • Library Manager ▼  
1 void setup() {  
2     pinMode(5, OUTPUT);  
3     pinMode(15, INPUT);  
4 }  
5  
6 void loop() {  
7     if(digitalRead(15) == HIGH){  
8         digitalWrite(5, HIGH);  
9     } else {  
10        digitalWrite(5, LOW);  
11    }  
12 }  
13
```

# Praktikum Mikrokontroler dengan Wokwi

## Menyalakan LED dengan Push Button

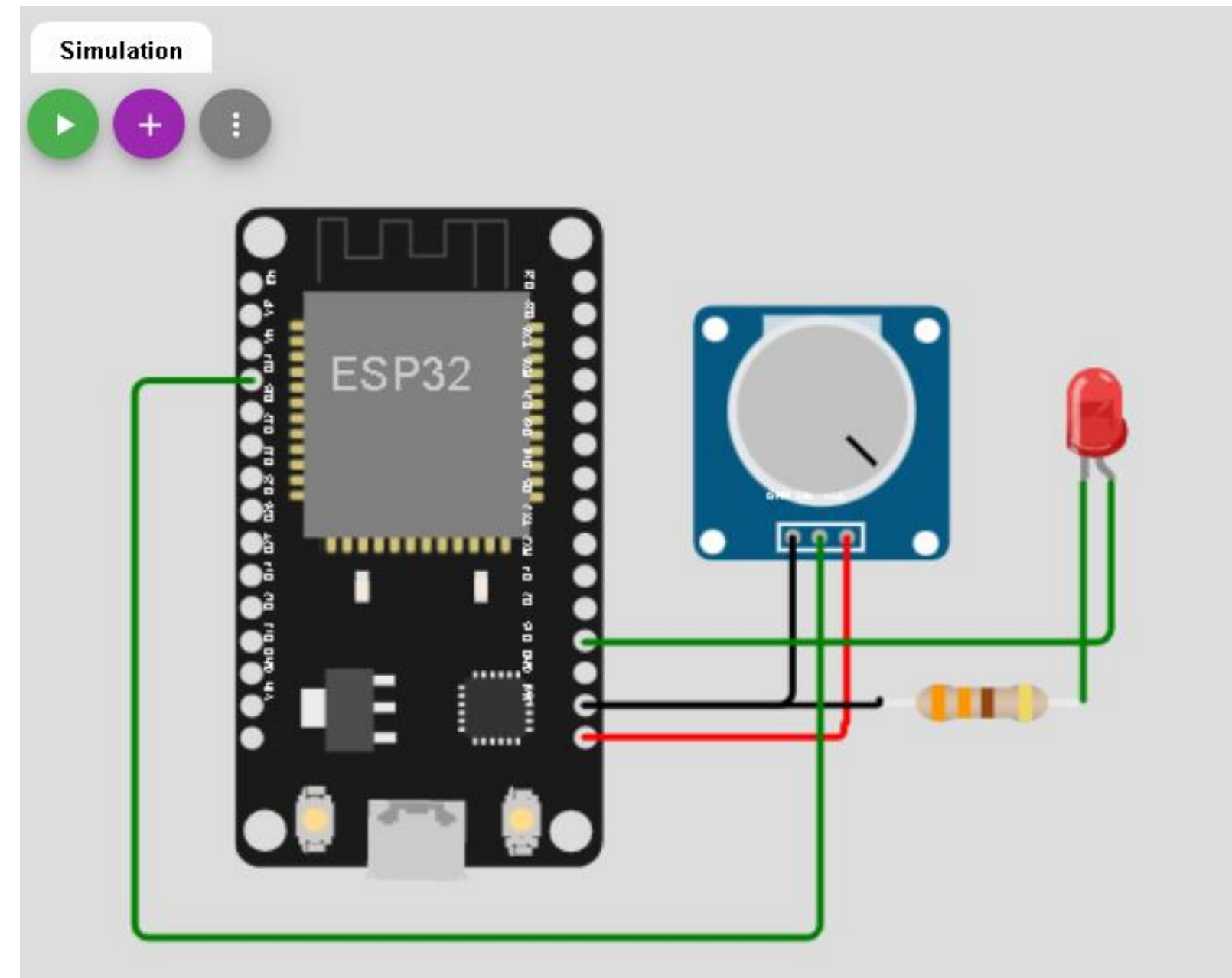
- Klik pushbutton maka LED akan menyala.
- Sebaliknya, saat push button tidak ditekan, maka LED akan mati.
- Klik Stop Simulation untuk menghentikan simulasi.



# Praktikum Mikrokontroler dengan Wokwi

## Praktik Mengendalikan LED dengan PWM

- Buatlah rangkaian seperti disamping dengan menggunakan 1 LED, 1 potensiometer, 1 resistor
- Ubah warna kabel, warna LED, dan nilai resistor dengan edit pada diagram.json.
- LED terhubung dengan pin 2 ESP32
- 3 kaki potensiometer dihubungkan ke pin ESP32 :
  - VCC ke 3V3
  - GND ke pin GND
  - SIG ke D34



# Praktikum Mikrokontroler dengan Wokwi

## Mengendalikan LED dengan PWM

- Buatlah program seperti di tunjukan pada gambar.
- Terdapat penambahan variabel status dengan tipe data integer yang berfungsi untuk menunjukkan status atau nilai yang dihasilkan dari push button.

Coding : [Download](#)

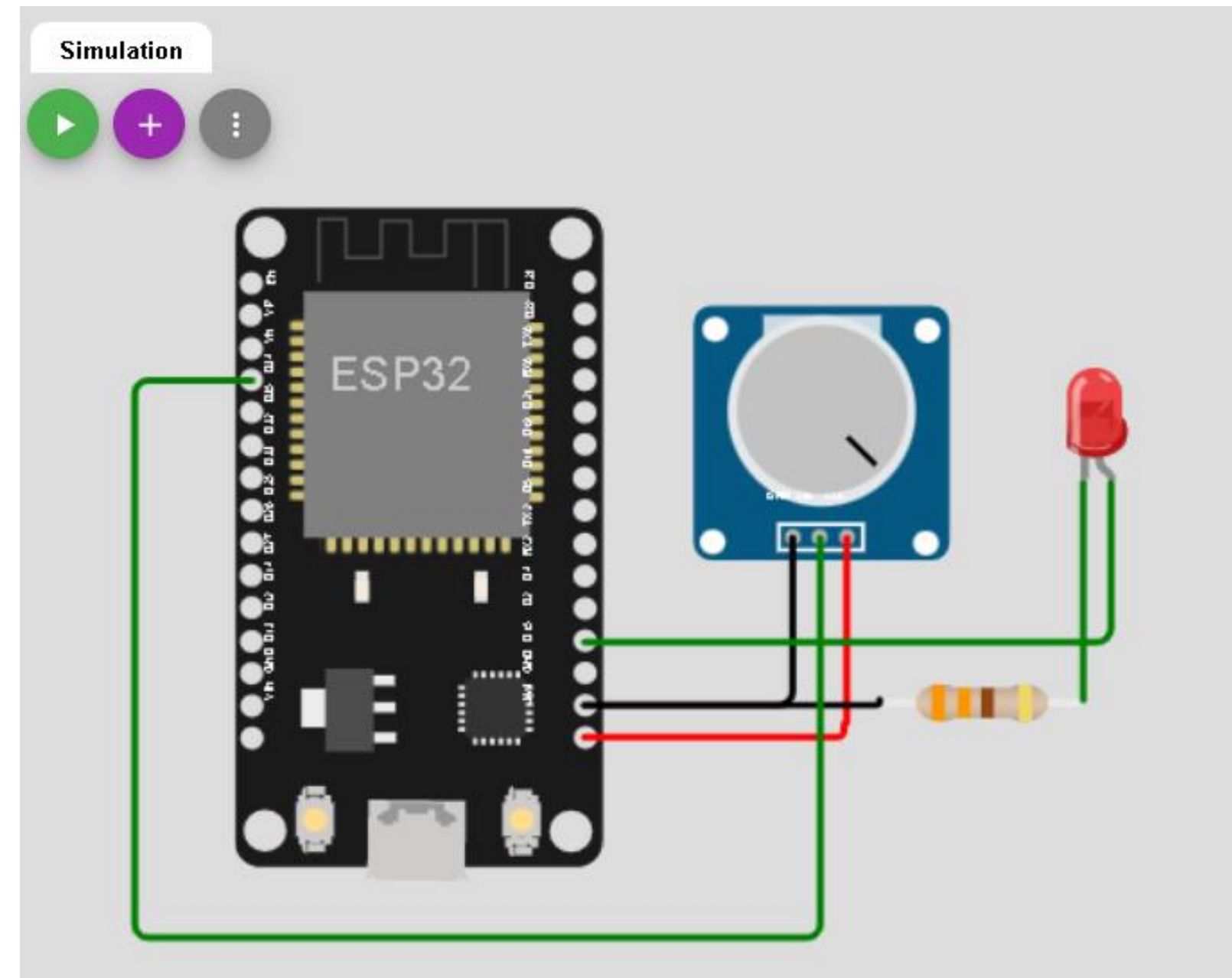
```
sketch.ino ● diagram.json ● Library Manager ▼
1  int ledPin = 2;
2  int potensioPin = 34;
3  int val = 0;
4  void setup()
5  {
6      Serial.begin(9600);
7      pinMode(ledPin, OUTPUT);
8  }
9
10 void loop(){
11     val = analogRead(potensioPin);
12     analogWrite(ledPin, val / 4);
13     Serial.println(val);
14 }
15
```



# Praktikum Mikrokontroler dengan Wokwi

## Mengendalikan LED dengan PWM

- Buatlah rangkaian seperti disamping dengan menggunakan 1 LED, 1 potensiometer, 1 resistor
- Ubah warna kabel, warna LED, dan nilai resistor dengan edit pada diagram.json.
- LED terhubung dengan pin 2 ESP32
- 3 kaki potensiometer dihubungkan ke pin ESP32 :
  - VCC ke 3V3
  - GND ke pin GND
  - SIG ke D34



# Praktikum Mikrokontroler dengan Wokwi

## Mengendalikan LED dengan PWM

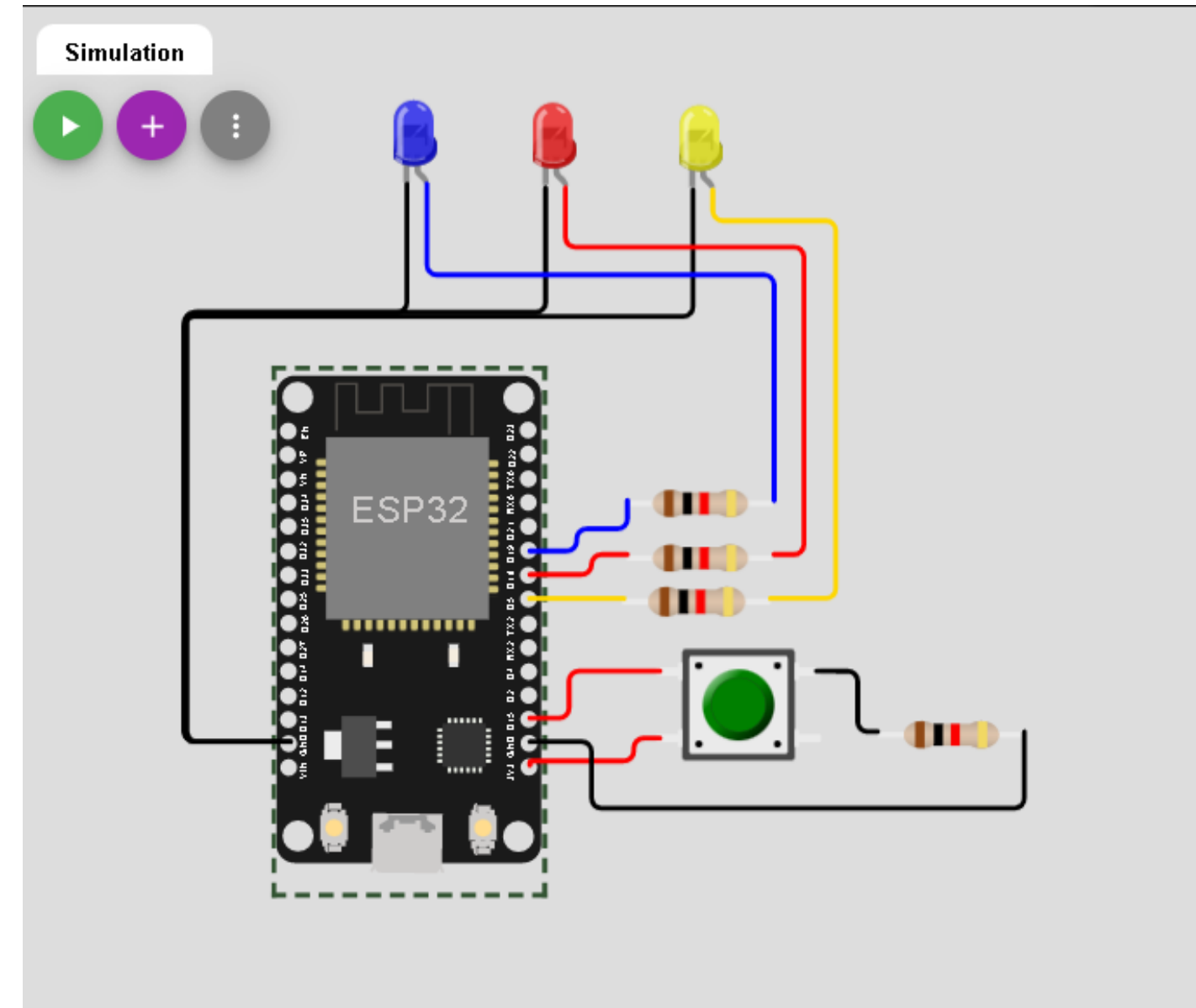
- Buatlah program seperti di tunjukan pada gambar.
- Terdapat penambahan variabel status dengan tipe data integer yang berfungsi untuk menunjukkan status atau nilai yang dihasilkan dari push button.

```
sketch.ino ● diagram.json ● Library Manager ▼
1  int ledPin = 2;
2  int potensioPin = 34;
3  int val = 0;
4  void setup()
5  {
6      Serial.begin(9600);
7      pinMode(ledPin, OUTPUT);
8  }
9
10 void loop(){
11     val = analogRead(potensioPin);
12     analogWrite(ledPin, val / 4);
13     Serial.println(val);
14 }
15
```

# Praktikum Mikrokontroler dengan Wokwi

## Challenges : Membuat Running LED

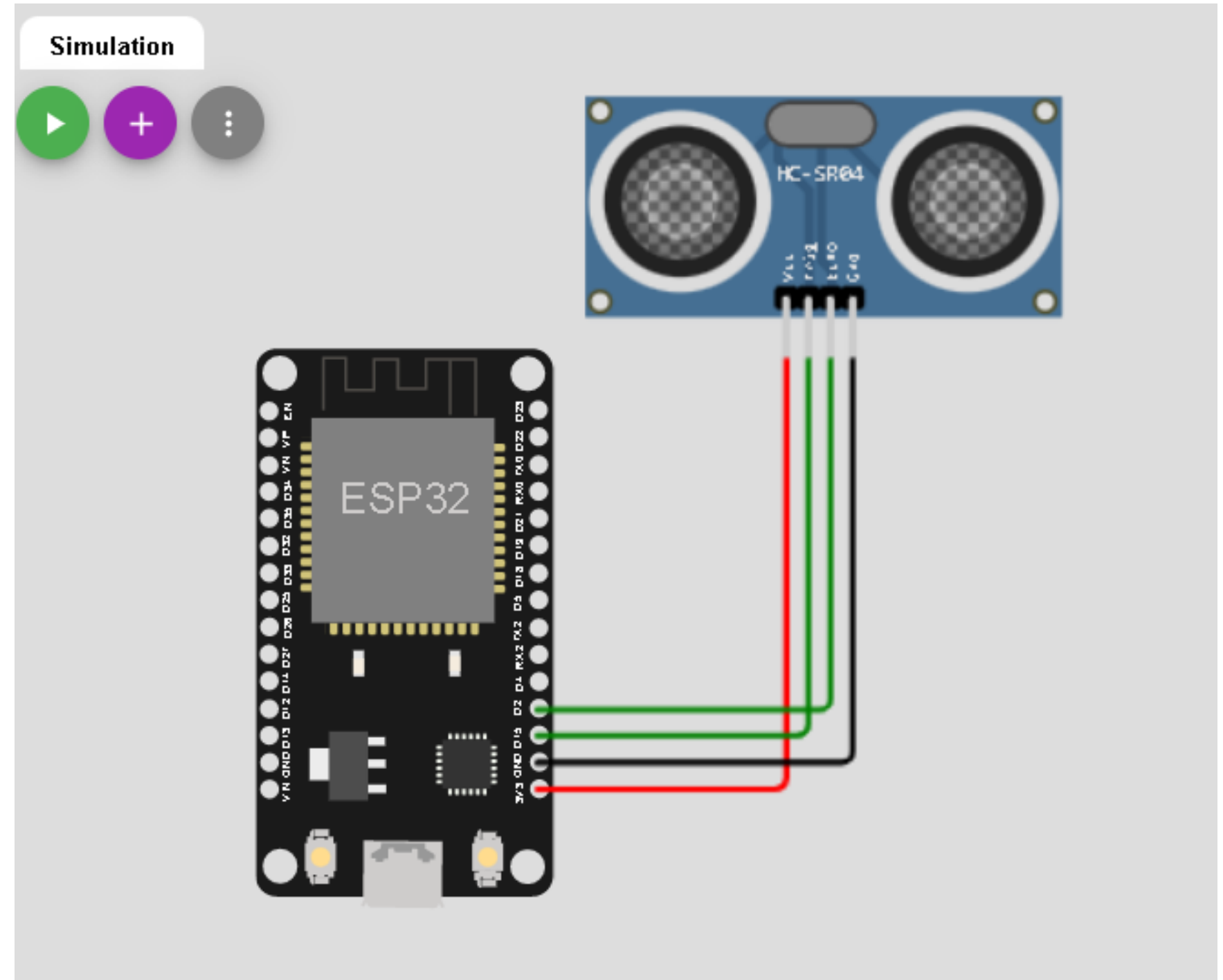
- Buatlah Program untuk menyalakan LED dari kiri ke kanan menggunakan push button
- Buatlah rangkaian seperti disamping dengan menggunakan 3 LED, 3 resistor, 1 pushbutton
- Ubah warna kabel, warna LED, dan nilai resistor dengan edit pada diagram.json.
- LED biru terhubung dengan pin D19 ESP32
- LED merah terhubung dengan pin D18 ESP32
- LED kuning terhubung dengan pin D5 ESP32
- Push button terhubung dengan pin D15 ESP32



# Praktikum Mikrokontroler dengan Wokwi

## Praktik Input Sensor Ultrasonic pada Serial Monitor

- Buatlah rangkaian seperti disamping dengan menggunakan HC-SR04 Ultrasonic Sensor
- Hubungkan pin sensor ultrasonic :
  - VCC ke 3V3
  - GND ke GND
  - Trig ke D15
  - Echo ke D2





# Praktikum Mikrokontroler dengan Wokwi

## Praktik Input Sensor Ultrasonic pada Serial Monitor

- Buatlah program seperti ditunjukkan pada gambar.
- Program tersebut digunakan untuk menampilkan hasil pembacaan data sensor ultrasonik.
- Hasil pembacaan ditampilkan di serial monitor
- Jalankan simulasi dengan klik Start Simulation.

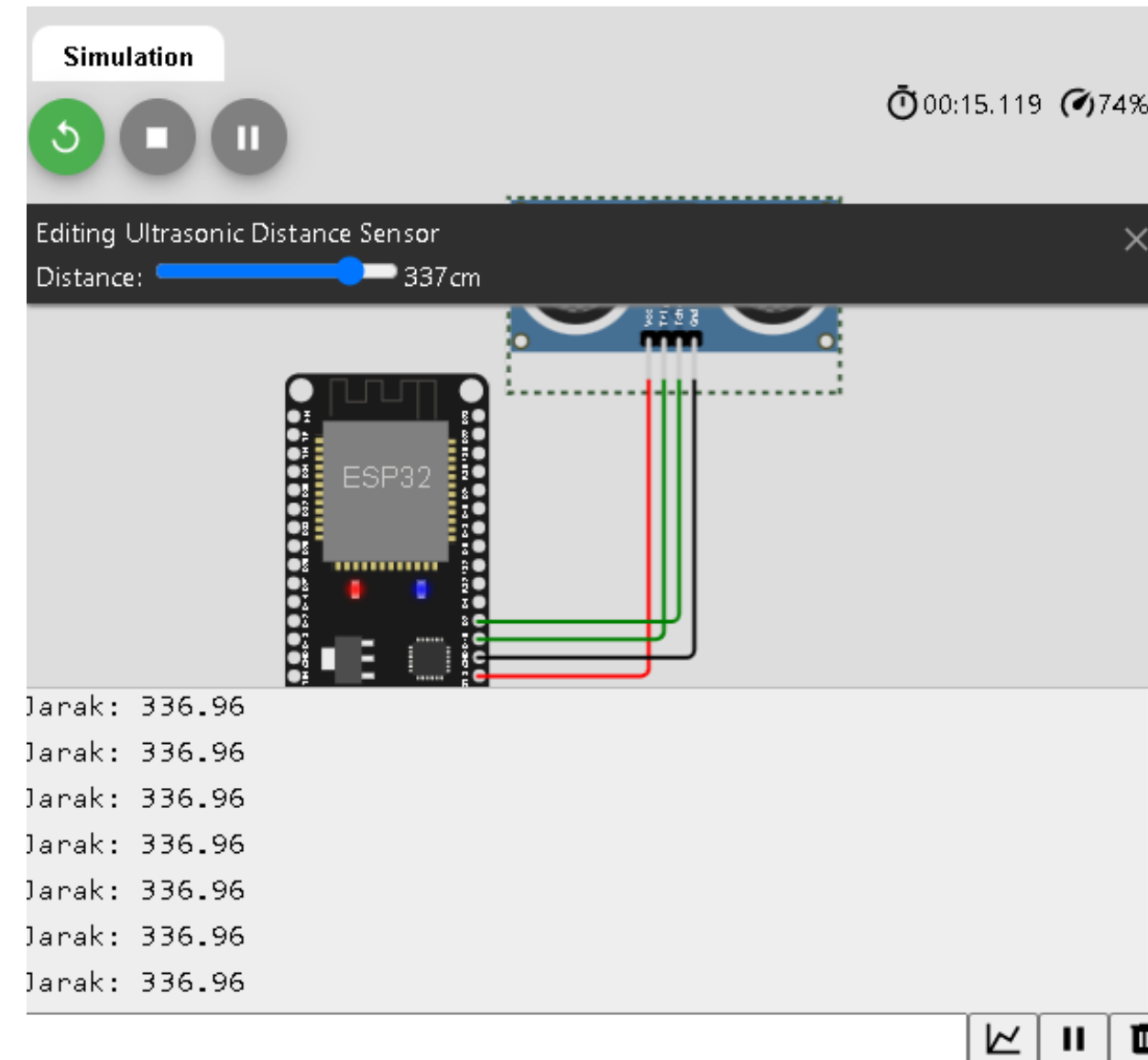
Coding : [Download](#)

```
sketch.ino • diagram.json • Library Manager
1  #define ECHO_PIN 2
2  #define TRIG_PIN 15
3
4  void setup() {
5      Serial.begin(115200);
6      pinMode(LED_BUILTIN, OUTPUT);
7      pinMode(TRIG_PIN, OUTPUT);
8      pinMode(ECHO_PIN, INPUT);
9  }
10
11 float readDistanceCM() {
12     digitalWrite(TRIG_PIN, LOW);
13     delayMicroseconds(2);
14     digitalWrite(TRIG_PIN, HIGH);
15     delayMicroseconds(10);
16     digitalWrite(TRIG_PIN, LOW);
17     int duration = pulseIn(ECHO_PIN, HIGH);
18     return duration * 0.034 / 2;
19 }
20
21 void loop() {
22     digitalWrite(TRIG_PIN, LOW);
23     delayMicroseconds(2);
24     digitalWrite(TRIG_PIN, HIGH);
25     delayMicroseconds(10);
26     digitalWrite(TRIG_PIN, LOW);
27     int duration = pulseIn(ECHO_PIN, HIGH);
28     float distance = duration * 0.034 / 2;
29
30     Serial.print("Jarak: ");
31     Serial.println(distance);
32
33     delay(400);
34 }
```

# Praktikum Mikrokontroler dengan Wokwi

## Praktik Input Sensor Ultrasonic pada Serial Monitor

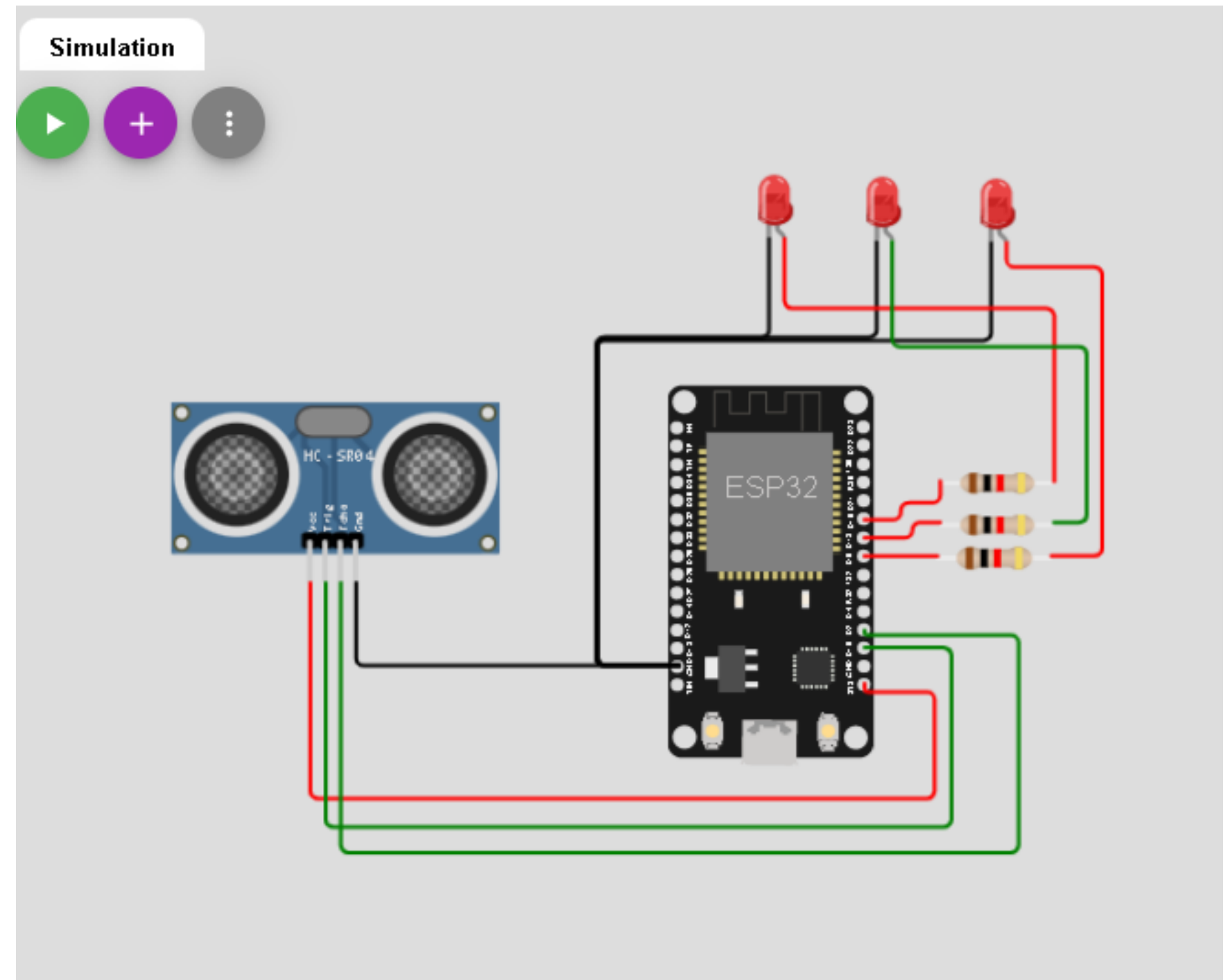
- Klik Serial Monitor (bawah) untuk melihat hasil pembacaan sensor.
- Ubah jarak dengan klik pada sensor, lakukan pengubahan jarak.
- Jika jarak berhasil muncul, berarti pembacaan data sensor sudah berhasil.
- Klik Stop simulation untuk menghentikan simulasi.



# Praktikum Mikrokontroler dengan Wokwi

## Challenges : Pengukuran Jarak dengan Indikator LED

- Buatlah sistem pengukuran jarak menggunakan sensor ultrasonik, 4 lampu LED
- Buatlah rangkaian seperti gambar disamping.
- Cara kerja program :
  - Ketika Jarak > 50 cm, maka 1 lampu indikator akan menyala
  - Ketika Jarak > 100 cm, maka 2 lampu indikator akan menyala
  - Ketika Jarak > 200 cm, maka 3 lampu indikator akan menyala
  - Ketika Jarak > 300 cm, maka 4 lampu indikator akan menyala dan Buzzer juga menyala



Sekian Materi

**Praktik Basic Pemrograman Input,  
Output, PWM dengan Mikrokontroler  
beserta Troubleshoot Code**