

LED PUSH BUTTON

A. Tujuan

Materi ini bertujuan untuk membuat suatu aplikasi sederhana yang melibatkan penggunaan tombol (push button) dan LED pada platform ESP32 agar dapat memahami cara membaca input dari tombol dan mengendalikan output pada perangkat mikrokontroler

B. Alat dan Bahan Praktikum

1. Laptop
2. Code Editor (Arduino IDE)
3. ESP32
4. BreadBoard
5. Jumper
6. Lampu LED
7. PushButton
8. Resistor

C. Penjelasan

1. Definisi Pin

```
const int buttonPin = 4;  
const int ledPin = 15;
```

Disini kita mendefinisikan nomor pin sesuai skema dan instalasi yang kita lakukan pada hardware atau mikrokontroler

2. Deklarasi Variabel

```
int buttonState = 0;  
int lastButtonState = 0;  
int ledState = LOW;
```

Kode diatas untuk meyimpan status tombol dan led.

- int buttonState = 0; : Menyimpan status saat ini pada tombol(HIGH atau LOW)
- int lastButtonState = 0; : Menyimpan status sebelumnya pada tombol
- int ledState = LOW; : Menyimpan status saat ini pada LED

3. Setup

```
void setup() {  
  pinMode(ledPin, OUTPUT);  
  pinMode(buttonPin, INPUT);  
}
```

- void setup() : ini fungsi yang dieksekusi sekali awal program
- pinMode(pin, mode) : Mengonfigurasi pin sebagai mode yang kita inginkan seperti 'INPUT' dan 'OUTPUT'

4. Loop

```
void loop() {

    buttonState = digitalRead(buttonPin);

    if (buttonState == HIGH && lastButtonState == LOW) {

        if (ledState == LOW) {
            ledState = HIGH;
        } else {
            ledState = LOW;
        }

        digitalWrite(ledPin, ledState);
    }

    lastButtonState = buttonState;
}
```

- void loop() : ini adalah fungsi yang dijalankan secara berulang terus menerus selama mikrokontroler berjalan
- digitalRead(buttonPin); digunakan untuk membaca status (nilai) dari suatu pin digital
- buttonState = digitalRead(buttonPin); Hasil dari digitalRead(buttonPin) disimpan dalam variabel buttonState.
- if (buttonState == HIGH && lastButtonState == LOW) Ini adalah kondisi yang mendeteksi perubahan tombol dari keadaan tidak ditekan (LOW) menjadi ditekan (HIGH).
- digitalWrite(ledPin, ledState); menetapkan status LED yang baru yang mana kita mengatur status pin 'ledPin' sesuai dengan nilai variabel 'ledState'
- lastButtonState = buttonState; Kode ini menyimpan status terakhir dari tombol (buttonState) ke dalam variabel lastButtonState

D. Kode dan Skema

1. Kode

a. PushButton tidak Kelock

```
#define LED_PIN 15
#define BUTTON_PIN 4

void setup() {
    pinMode(LED_PIN, OUTPUT);
    pinMode(BUTTON_PIN, INPUT);
}

void loop() {
    if(digitalRead(BUTTON_PIN) == HIGH)
    {
        digitalWrite(LED_PIN,HIGH);
    }
    else {
        digitalWrite(LED_PIN,LOW);
    }
}
```

b. PushButton Kelock

```
const int buttonPin = 4;
const int ledPin = 15;

int buttonState = 0;
int lastButtonState = 0;
int ledState = LOW;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {

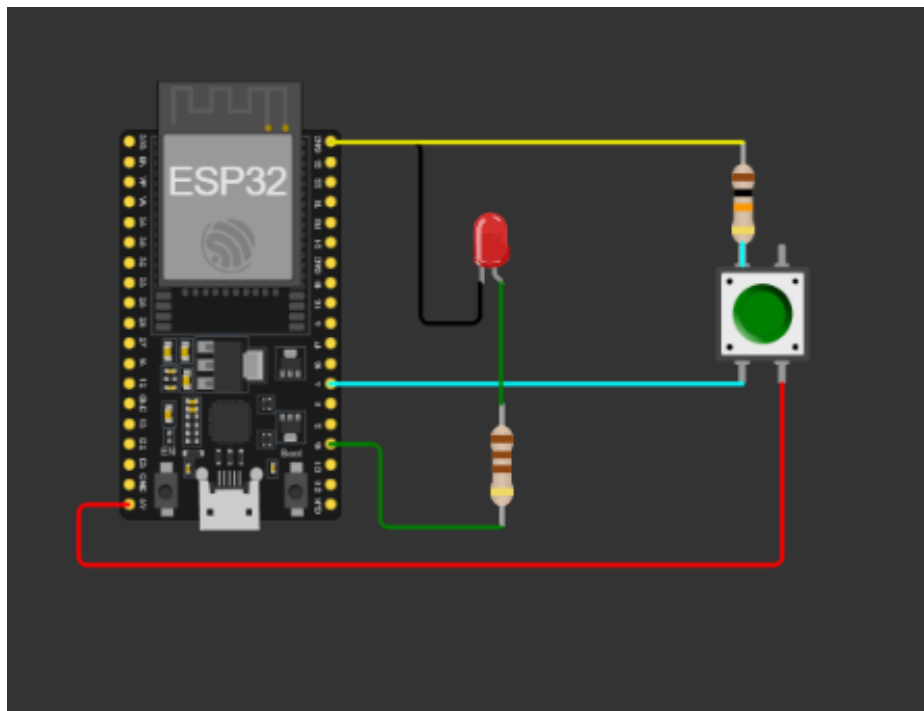
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH && lastButtonState == LOW) {

    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }

    digitalWrite(ledPin, ledState);
  }

  lastButtonState = buttonState;
}
```

2. Skema

3. TUGAS

Buatlah rangkaian seperti dibawah ini dengan menggunakan 2 push button yang berfungsi sebagai tombol on dan off (1 push button untuk saklar on dan 1 push button untuk saklar off)

