

## 4.1 Teori dan Praktikum Mikrokontroler Arduino Uno, Serial Monitor, Fungsi Digital I/O, Analog I/O, Advanced I/O dan Time

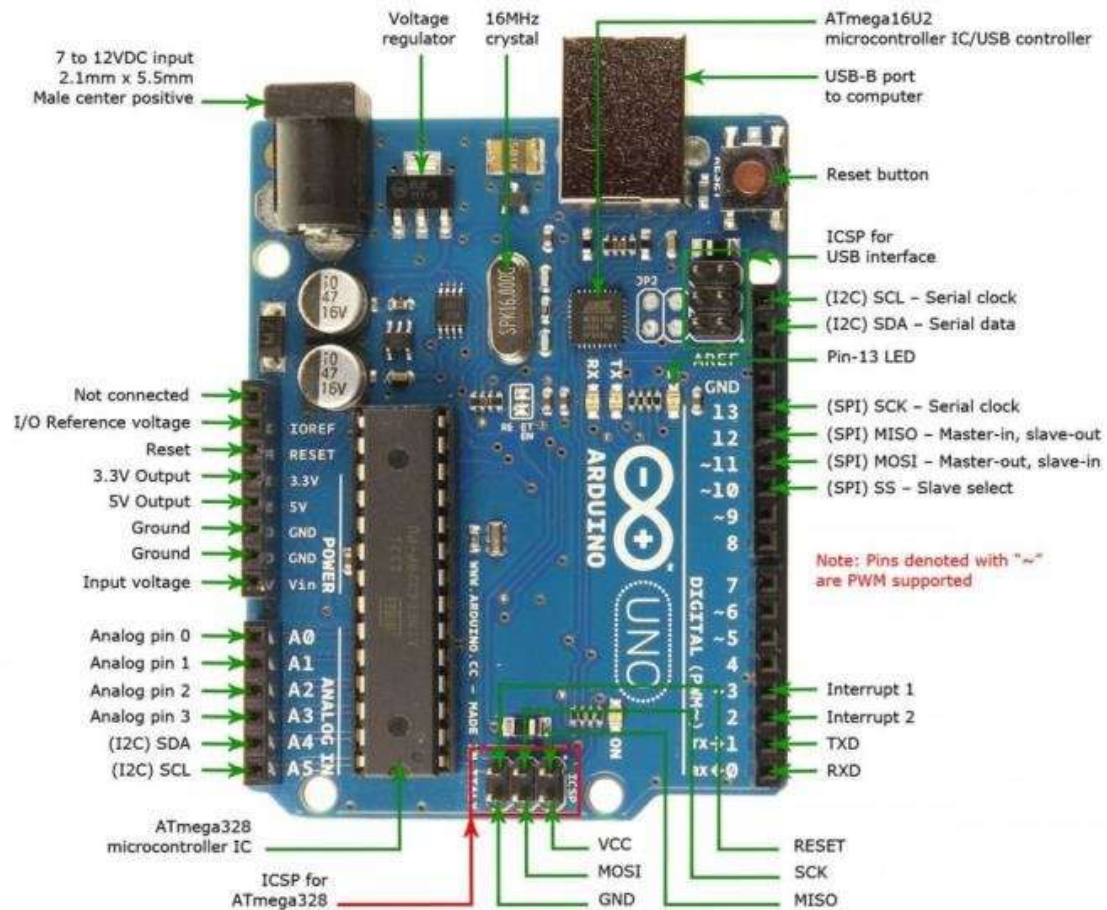


Indobot Academy 21 November 2022

### 1. Dasar Teori

#### 1.1. Mikrokontroler Arduino Uno

Arduino Uno adalah papan sirkuit berbasis mikrokontroler ATmega328. IC (integrated circuit) ini memiliki 14 input/output digital (6 output untuk PWM), 6 analog input, resonator kristal keramik 16 MHz, Koneksi USB, soket adaptor, pin header ICSP, dan tombol reset. Hal inilah yang dibutuhkan untuk mensupport mikrokontroler secara mudah terhubung dengan kabel power USB atau kabel power supply adaptor AC ke DC atau juga battery.



Gambar 1. Pin Out Arduino Uno

Spesifikasi Arduino Uno Sebagai Berikut :

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6

DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

## 1.2. Fungsi Digital I/O

Input Digital adalah sinyal yang akan diterima baik itu berupa nilai 1 atau 0, logika high atau low, maupun kondisi *on* atau *off*. dalam bentuk program untuk Arduino input digital ditulis sebagai `digitalRead` . Pada output digital sebenarnya hampir sama dengan input digital dikarenakan pada dasarnya sama, hal yang dikirimkan yaitu nilai 1 atau nilai 0. Yang membedakan hanya pada penggunaan fungsi pada saat pemrogramannya, kapan saat diatur sebagai input maupun output, dalam pemrograman untuk Arduino output digital ditulis sebagai `digitalWrite` .

## 1.3. Fungsi Analog I/O

Tidak seperti pin digital yang dapat difungsikan sebagai digital input-output, namun pada pin analog hanya dapat berfungsi sebagai analog input saja, apabila anda ingin membuat Arduino mengeluarkan output tegangan analog anda masih bisa melakukannya dengan menggunakan pin digital yang

berlabel PWM (3,5,6,9,10,11), sejatinya tegangan analog yang dikeluarkan Arduino adalah dengan memanipulasi output digital secara pulsa. Adapun dalam pemrograman analog input ditulis sebagai `analogRead` .

## 1.4. Fungsi Advanced I/O

### 1.4.1. `shiftOut()`

Merupakan fungsi untuk mengirim data secara serial sinkron setiap kali pin clock berlogika high. Karena data akan selalu tergeser keluar ke jalur pin data setiap kali transisi clock ke logika high maka sebelumnya pin clock harus dipastikan berlogika low dengan instruksi `digitalWrite (clockPin, LOW)`.

- **Syntax**

`shiftOut (dataPin, clockPin, bitOrder, value)`.

- **Parameter**

- `dataPin` : pin untuk jalur data keluar tiap bitnya dengan memakai tipe data integer.
- `clockPin` : pin di mana signal clock akan dibangkitkan untuk mengirim data lewat jalur `dataPin`.
- `bitOrder` : merupakan perintah bagaimana fungsi geser dilakukan, apakah LSB (`LSBFIRST`) atau MSB (`MSBFIRST`) dahulu.
- `value` : data yang digeser keluar pada jalur data dengan tipe data byte.

### 1.4.2. `shiftIn()`

Merupakan fungsi untuk mengambil data secara serial sinkron setiap kali pin clock berlogika high. Karena data akan selalu tergeser masuk dari jalur pin data setiap kali transisi clock ke logika high maka sebelumnya pin clock harus dipastikan berlogika low dengan instruksi `digitalWrite (clockPin, LOW)`.

- **Syntax**

`byte incoming = shiftIn (dataPin, clockPin, bitOrder)`.

- **Parameter**

- `dataPin` : pin di mana jalur data akan diambil. Tipe data yang diizinkan yaitu integer.
- `clockPin` : pin di mana signal clock akan dibangkitkan untuk mengambil data agar terbaca di pin `dataPin`.
- `bitOrder` : merupakan perintah bagaimana fungsi geser dilakukan, apakah LSB (`LSBFIRST`) atau MSB (`MSBFIRST`) dahulu.

### 1.4.3. pulseInLong()

Merupakan fungsi alternatif dari pulseIn() yang lebih baik untuk handle pulsa yang lebih panjang dan skenario yang terpengaruh oleh interupsi.

Contoh untuk nilai HIGH akan menunggu pulsa dari LOW ke HIGH, start timing dan menunggu untuk kembali ke LOW untuk stop timing. Nilai yang diperoleh di sini adalah lebar pulsa dalam microsecond atau nilai nol apabila tidak diperoleh pulsa hingga time out.

Fungsi ini akan menampilkan error untuk pulsa yang lebih lebar. Hanya akan bekerja dalam nilai 10uS hingga 3 menit.

- **Syntax**

`pulseInLong(pin, value).`

`pulseInLong(pin, value, timeout).`

- **Parameters**

- pin : Nomor pin Arduino Board di mana pulsa akan dibaca. Tipe data yang diizinkan yaitu integer.

- value : tipe pulsa yang dibaca, HIGH atau LOW. Tipe data yang diizinkan yaitu integer.

- timeout(optional) : nilai dalam microsecond untuk menunggu pulsa dimulai, default dalam satu detik. Tipe data yang diizinkan yaitu unsigned long.

### 1.4.4 pulseIn()

Fungsi untuk membaca pulsa HIGH atau LOW pada pin. Contoh untuk nilai HIGH akan menunggu pulsa dari LOW ke HIGH, start timing dan menunggu untuk kembali ke LOW untuk stop timing. Nilai yang diperoleh di sini adalah lebar pulsa dalam microsecond atau nilai nol apabila tidak diperoleh pulsa hingga time out.

Fungsi ini akan menampilkan error untuk pulsa yang lebih lebar. Hanya akan bekerja dalam nilai 10uS hingga 3 menit.

- **Syntax**

`pulseIn(pin, value).`

`pulseIn(pin, value, timeout).`

- **Parameter**

- pin: Nomor pin Arduino Board di mana pulsa akan dibaca. Tipe data yang diizinkan yaitu integer.
- value: tipe pulsa yang dibaca, HIGH atau LOW. Tipe data yang diizinkan yaitu integer.
- timeout(optional): nilai dalam microsecond untuk menunggu pulsa dimulai, default dalam satu detik. Tipe data yang diizinkan yaitu unsigned long.

#### 1.4.5 noTone()

Merupakan fungsi untuk menghentikan fungsi tone(). Fungsi ini tidak akan membawa efek apapun apabila tidak ada tone yang dibangkitkan sebelumnya.

- **Syntax**

noTone(pin).

- **Parameters**

pin: pin Arduino Board di mana tone dimatikan.

#### 1.4.6 tone()

Merupakan fungsi untuk membangkitkan gelombang kotak dengan frekuensi tertentu (duty cycle 50%) di pin Arduino Board. Durasi untuk gelombang kotak ini juga dapat ditentukan atau juga dibuat kontinyu sampai adanya fungsi noTone(). Pin ini dapat dihubungkan ke buzzer atau speaker untuk menghasilkan suara dengan frekuensi tersebut.

Hanya ada satu signal tone yang dapat dibangkitkan dalam satu waktu. Apabila ada fungsi tone() lagi di pin yang berbeda, maka fungsi itu akan diabaikan, namun apabila ada fungsi tone() lagi di pin yang sama, maka fungsi ini akan mengupdate frekuensi yang dibangkitkan pada pin tersebut.

- **Syntax**

tone(pin, frequency).

tone(pin, frequency, duration).

- **Parameters**

- pin: pin Arduino yang dibangkitkan frekuensi tone nya.
- frequency: frekuensi tone yang dihasilkan dengan tipe data unsigned int.
- duration: durasi tone dalam milisecond dengan tipe data unsigned long.

## **1.5. Fungsi Time**

### **1.5.1. Delay()**

Delay adalah proses menghentikan suatu program yang berjalan di Arduino. Cara kerja fungsi **Delay** cukup sederhana. Fungsi ini menerima argumen bilangan bulat (bilangan). Bilangan ini merepresentasikan waktu (diukur dalam mili detik). Program harus menunggu sampai pindah ke baris kode berikutnya ketika bertemu fungsi ini.

### **1.5.2. Millis()**

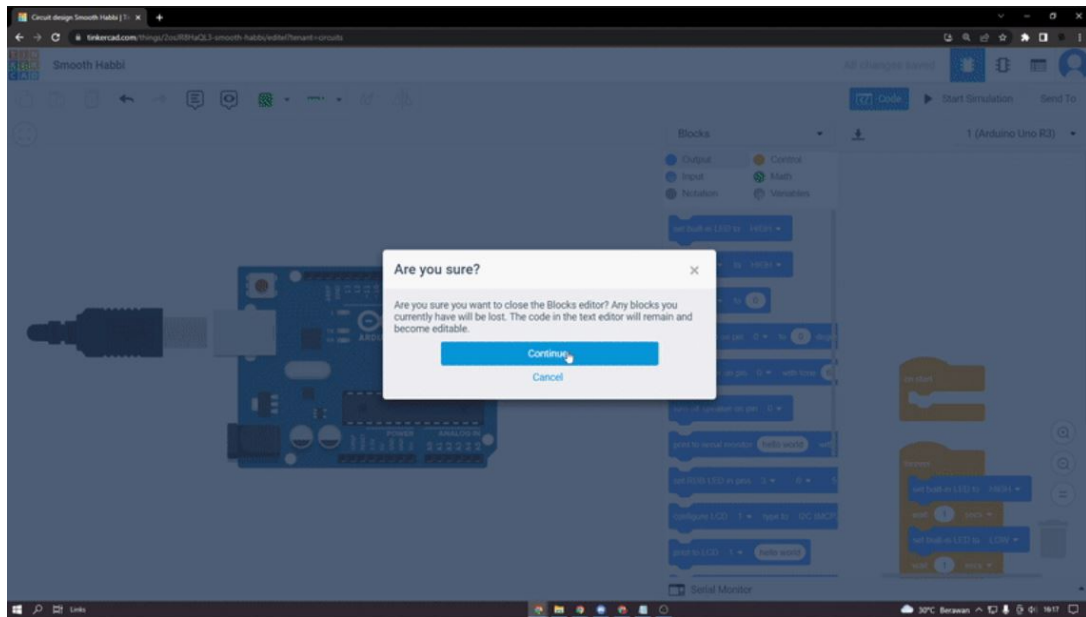
Millis adalah waktu yang terus berjalan menggunakan mili detik (Satuan dibawah detik) sehingga waktu ini walaupun kita gunakan bersamaan dengan fungsi **Delay** tidak akan berpengaruh karena millis akan tetap menghitung waktu dari awal arduino dijalankan.

## **2. Alat/Instrumen/Aparatus**

- Laptop
- Platform TinkerCAD

## **3. Langkah Praktikum 1 – Memulai Program Arduino**

Pertama masukan Arduino Uno dan ikuti langkah berikut ini:



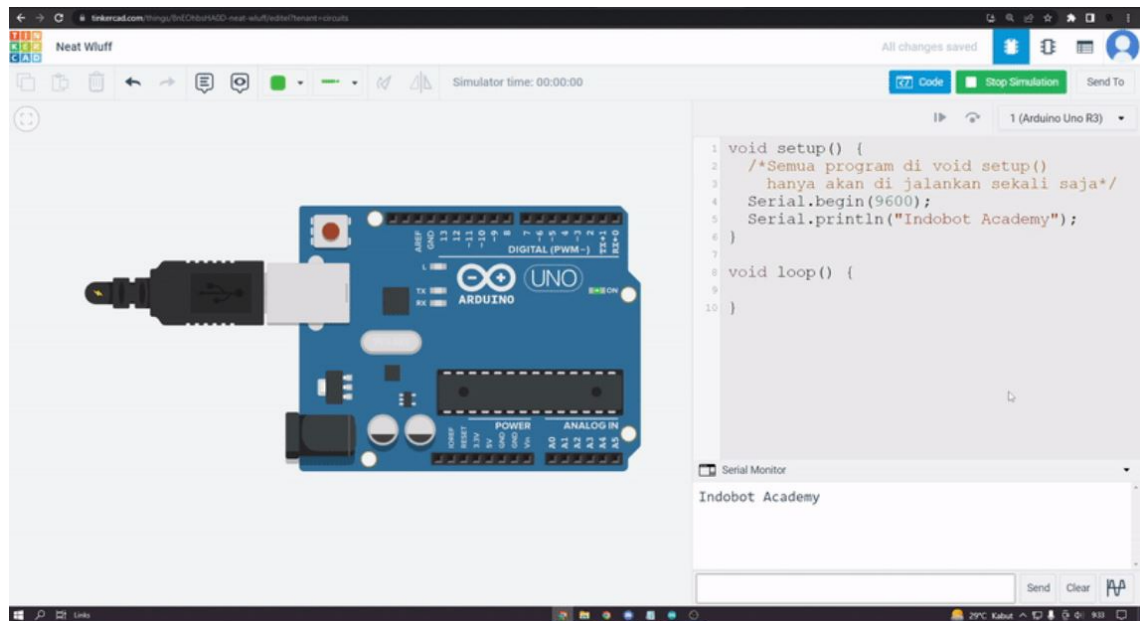
Penjelasan praktikum :

- Pertama, untuk mengaktifkan code pada tinkercad perlu mengganti dari mode block ke mode text.
- Kedua, secara default code pada Arduino Uno adalah example Blynk.
- Ketiga, kita start simulation untuk mencoba code Blynk tersebut.
- Keempat, kita juga dapat mengaktifkan debugger serial monitor dengan menarik keatas.
- Kelima, selamat mencoba.

### 3.1 Void Setup()

Fungsi void setup() dijalankan pada saat sketch atau program Arduino mulai. Fungsi ini digunakan untuk menginisiasi variabel, mendeklarasikan pin yang digunakan, menggunakan library, dll. Fungsi void setup() hanya akan dijalankan sekali saja setiap Arduino mulai atau saat direstart. Contoh :

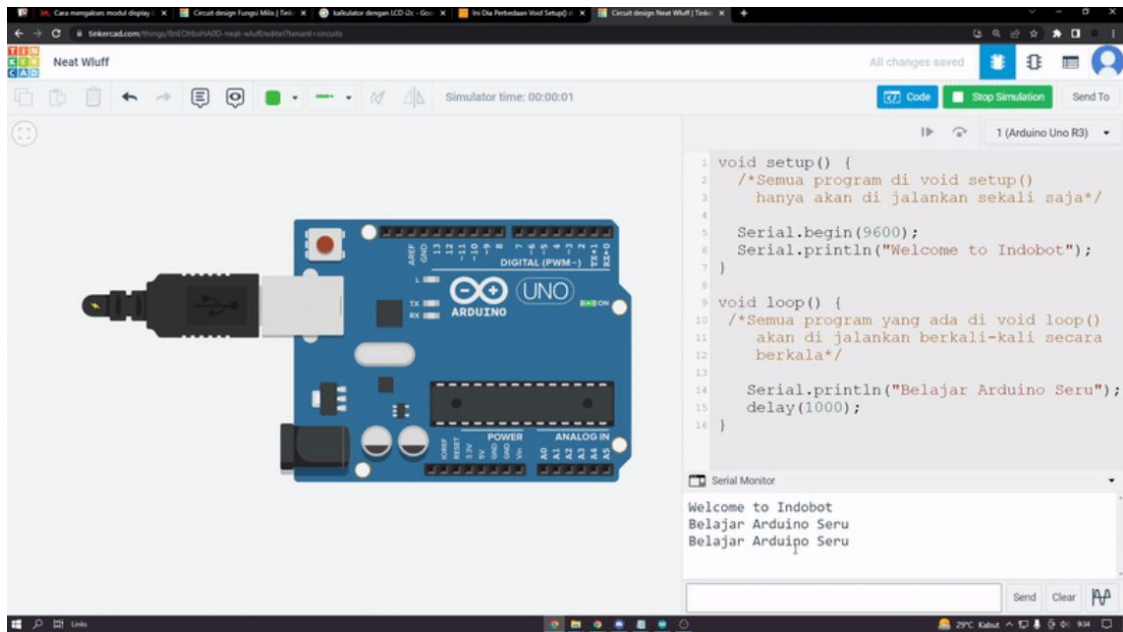




```
void setup() {  
  /*Semua program di void setup()  
  hanya akan dijalankan sekali saja*/  
  Serial.begin(9600);  
  Serial.println("Indobot Academy");  
}  
  
void loop() {  
  ....  
  ....  
}
```

### 3.2 Void Loop()

Fungsi void loop() dijalankan setelah fungsi setup() sudah selesai dijalankan, void loop() bertujuan untuk mengeksekusi dan menjalankan program yang sudah dibuat. Fungsi ini akan secara aktif mengontrol board Arduino baik membaca input atau mengubah output. Berbeda dengan void setup() yang hanya dijalankan sekali oleh Arduino, fungsi void loop() akan dijalankan berulang kali oleh Arduino secara berkala. Oleh sebab itu diberi nama loop() yang artinya berulang (lingkaran).



```
void setup() {  
  /*Semua program di void setup()  
  hanya akan dijalankan sekali saja*/
```

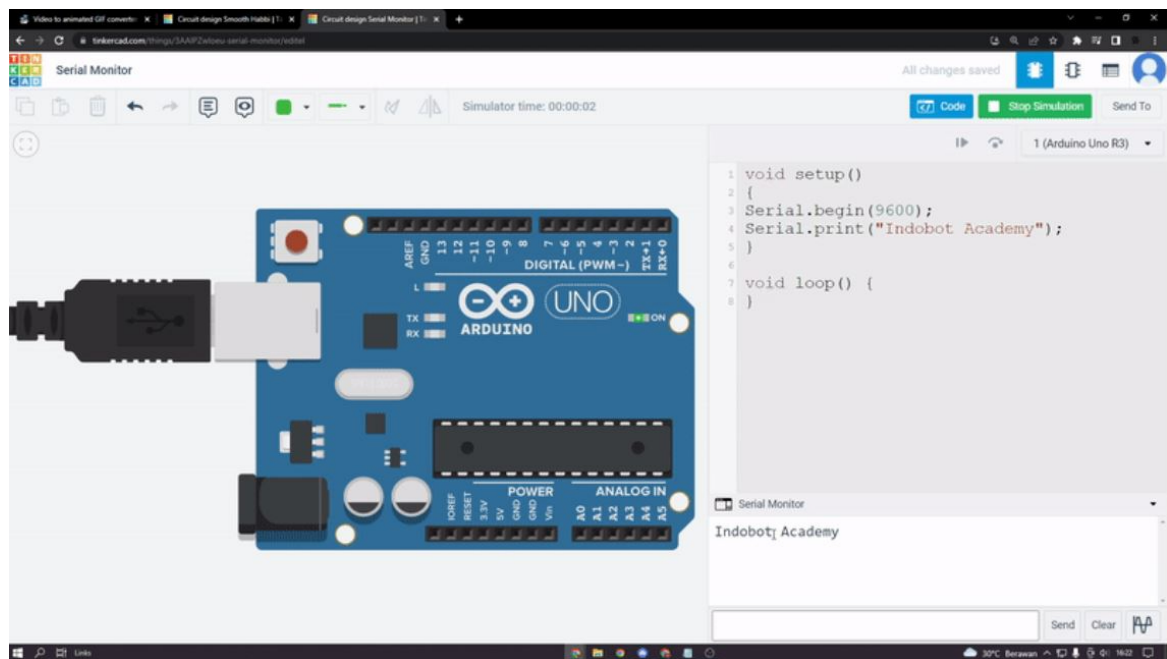
```
  Serial.begin(9600);  
  Serial.println("Indobot Academy");  
}
```

```
void loop() {  
  /*Semua program yang ada di void loop()  
  akan dijalankan berkali-kali secara  
  berkala*/
```

```
  Serial.println("Belajar Arduino Seru");  
  delay(1000);  
}
```

## 4. Langkah Praktikum 2 – Serial Monitor

Pada praktikum ini kita akan menampilkan teks pada serial monitor



Coding :

```

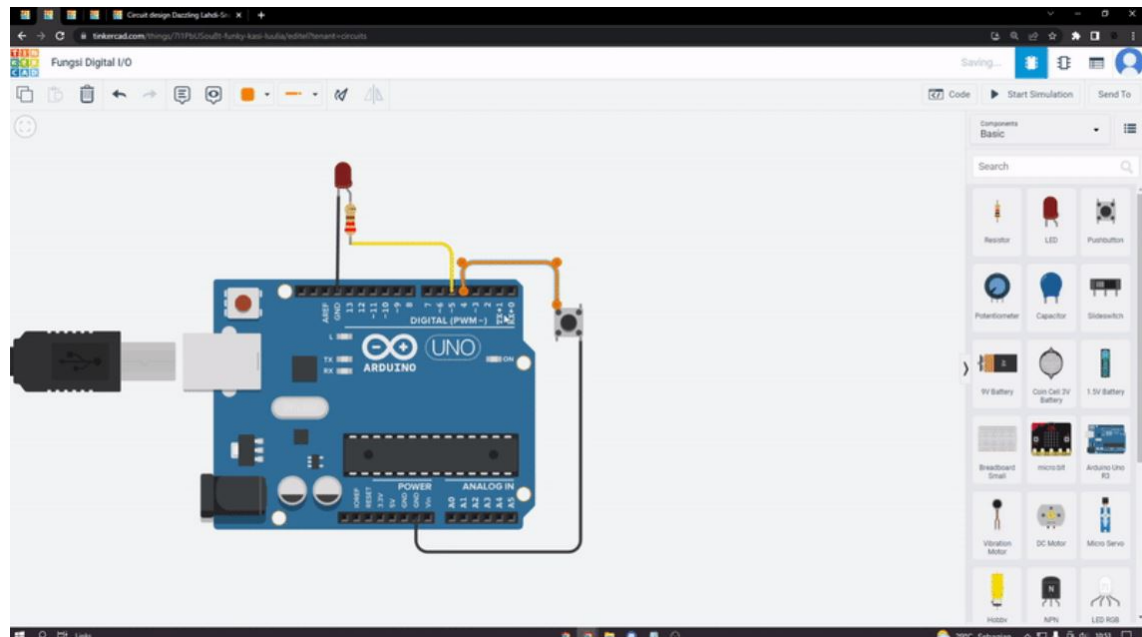
Text
1 void setup()
2 {
3   Serial.begin(9600);
4   Serial.print("Indobot Academy");
5 }
6
7 void loop() {
8 }
  
```

Penjelasan praktikum :

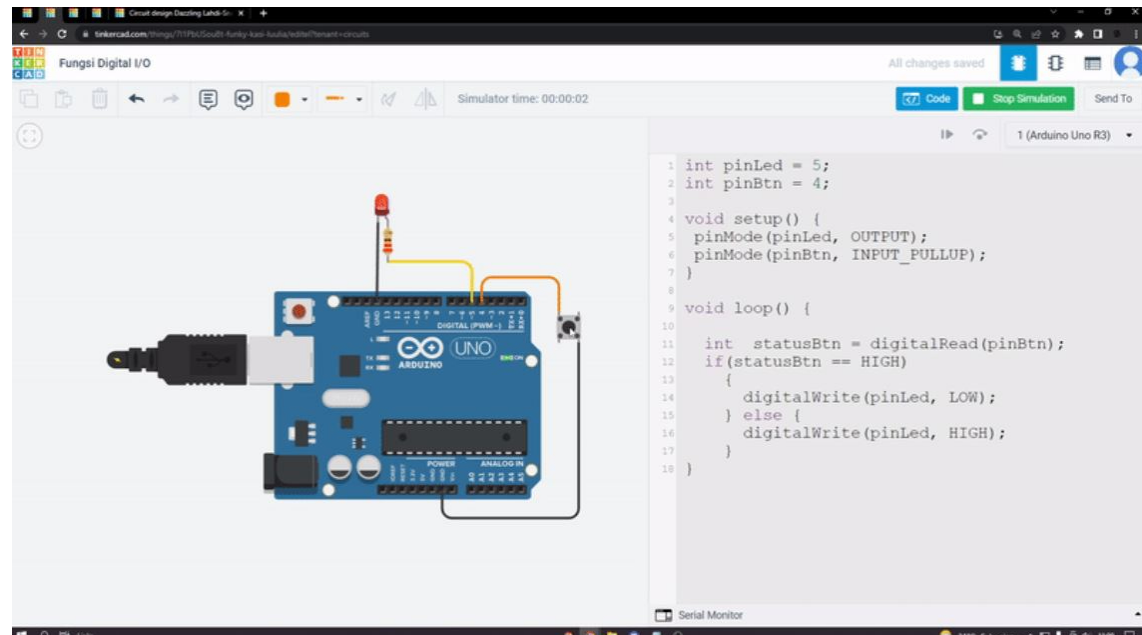
- Pertama, kita masukan arduino uno ke dalam Tinkercad.
- Kemudian, ganti code dari mode block ke mode text.
- Setelah itu, ketikkan program seperti pada gambar.
- Sebelum melakukan simulasi, pastikan jendela serial monitor telah terbuka.

## 5. Langkah Praktikum 3 – Fungsi Digital I/O

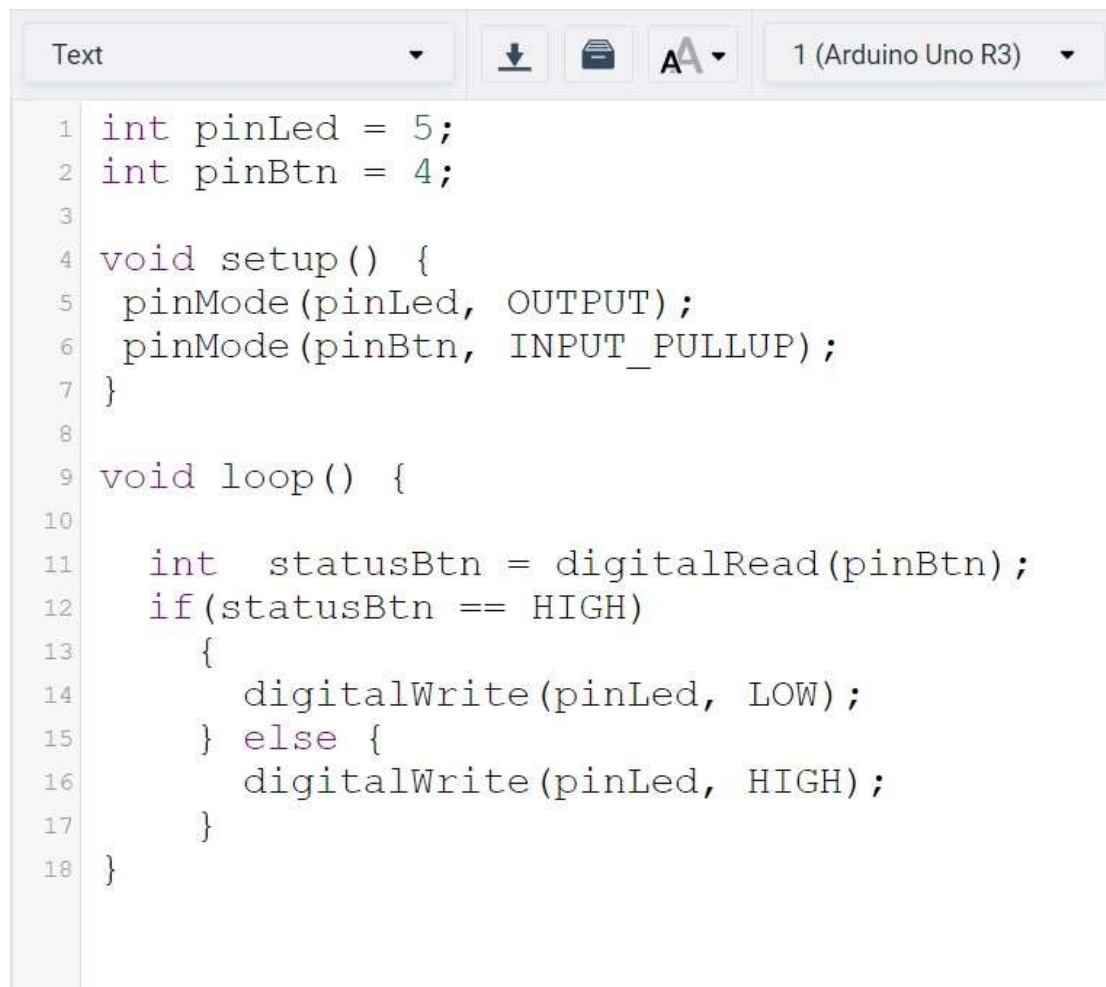
Buatlah rangkaian seperti pada gambar berikut ini :



Disini LED difungsikan sebagai Output, sedangkan untuk Push Buttonnya kita jadikan sebagai Input.



Coding :



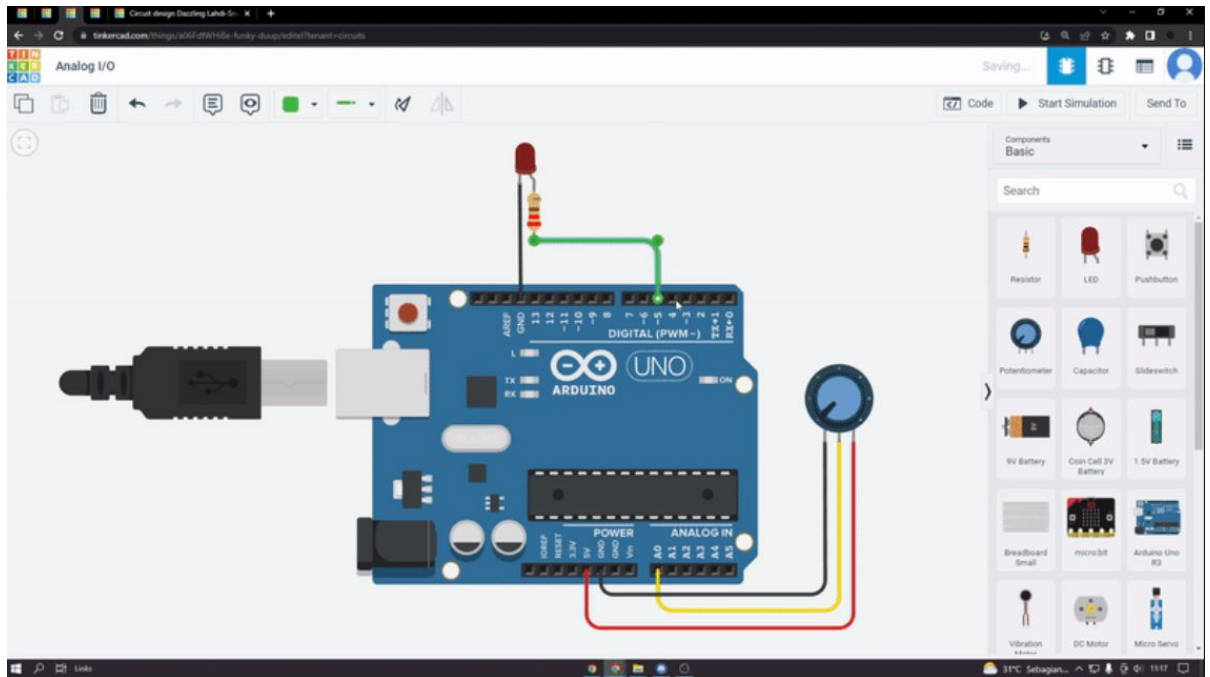
```
Text [v] [Download] [Save] [Font Size] 1 (Arduino Uno R3) [v]
1 int pinLed = 5;
2 int pinBtn = 4;
3
4 void setup() {
5   pinMode(pinLed, OUTPUT);
6   pinMode(pinBtn, INPUT_PULLUP);
7 }
8
9 void loop() {
10
11   int statusBtn = digitalRead(pinBtn);
12   if(statusBtn == HIGH)
13   {
14     digitalWrite(pinLed, LOW);
15   } else {
16     digitalWrite(pinLed, HIGH);
17   }
18 }
```

Penjelasan Praktikum :

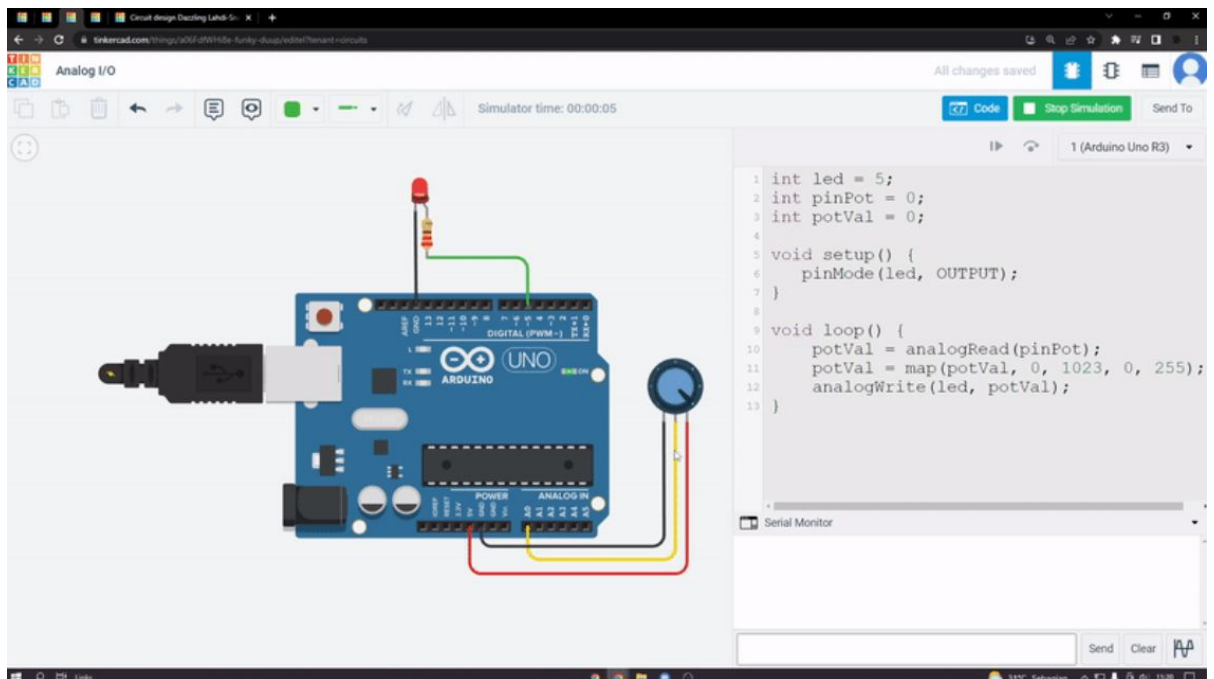
- Setelah membuat rangkaian, kita bisa memasukan coding sesuai gambar.
- Hasilnya adalah LED akan menyala ketika push button ditekan
- Apabila push button dilepaskan maka LED akan mati

## 6. Langkah Praktikum 4 – Fungsi Analog I/O

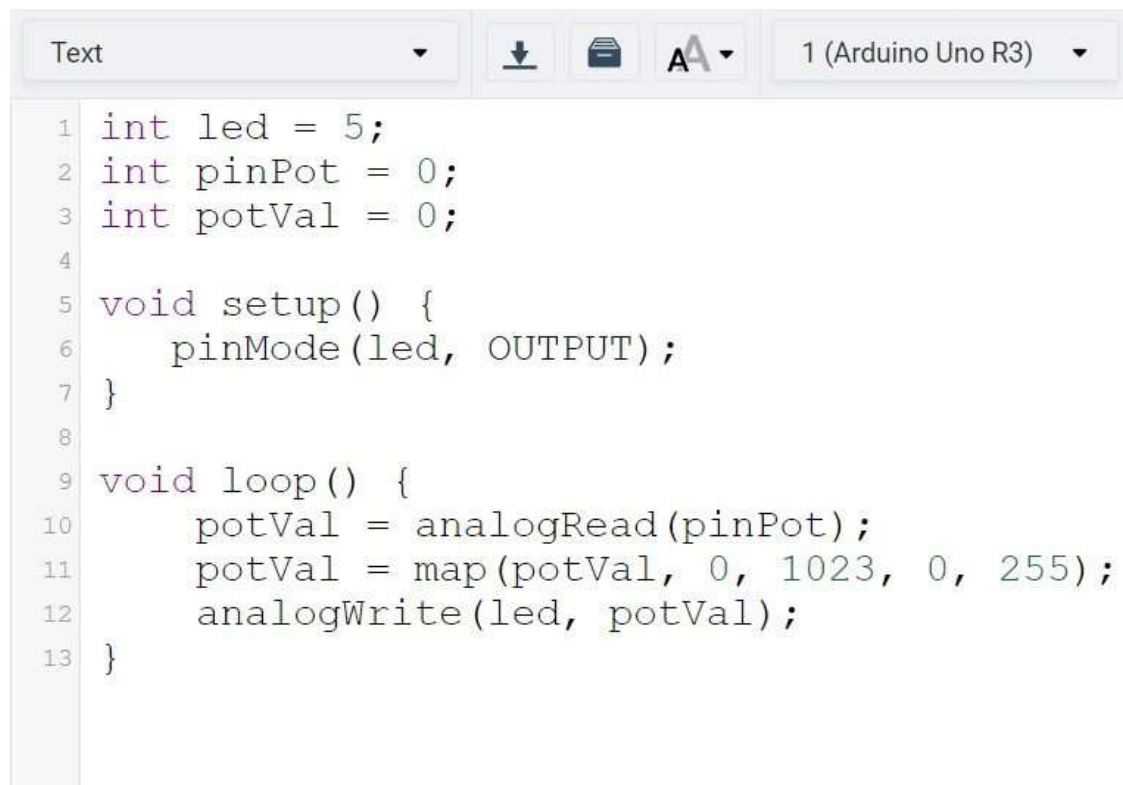
buatlah rangkaian seperti pada gambar berikut ini :



Berikutnya kita akan memasukan program dan melakukan simulasi.



Coding :



```
Text [dropdown] [download icon] [save icon] [font size icon] 1 (Arduino Uno R3) [dropdown]
1 int led = 5;
2 int pinPot = 0;
3 int potVal = 0;
4
5 void setup() {
6     pinMode(led, OUTPUT);
7 }
8
9 void loop() {
10     potVal = analogRead(pinPot);
11     potVal = map(potVal, 0, 1023, 0, 255);
12     analogWrite(led, potVal);
13 }
```

Penjelasan Praktikum :

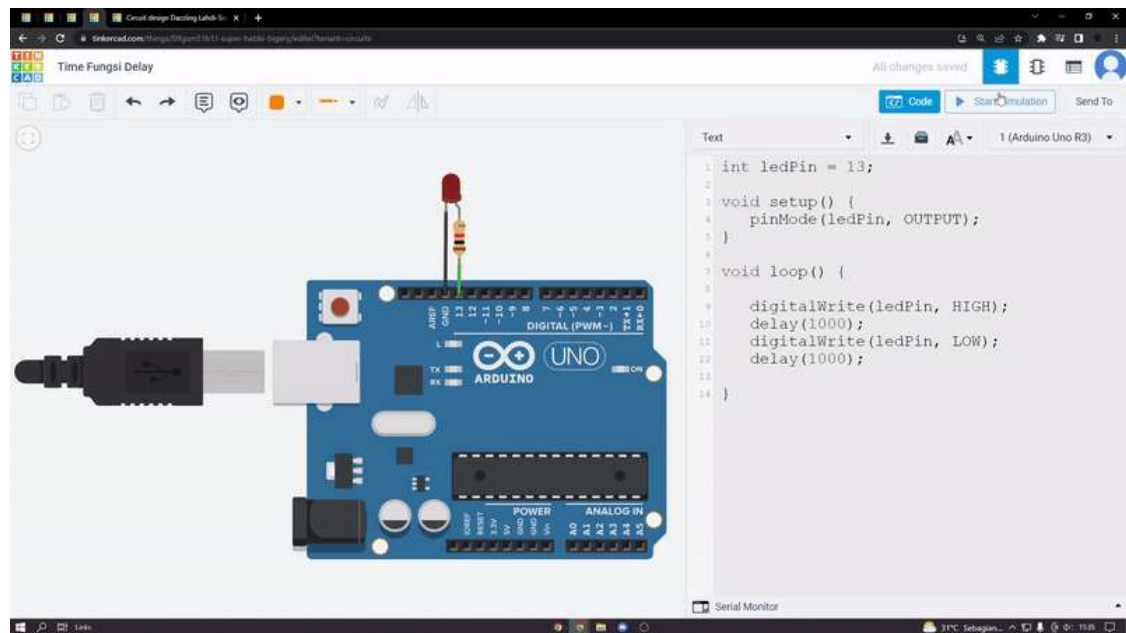
- Setelah melakukan uploading program, kita dapat melihat hasil dari simulasi.
- Di sini potensiometer kita jadikan sebagai Input.
- Mikrokontroller akan mengubah Input dari Potensiometer menjadi intensitas terang dari LED.
- Sehingga ketika Potensiometer diputar ke kanan, maka nyala LED akan semakin terang. Sebaliknya ketika diputar ke kiri maka nyala LED akan semakin redup dan mati.

## 7. Langkah Praktikum 5 – Fungsi Time

### 7.1 Praktikum dengan Fungsi Delay

Buatlah rangkaian seperti gambar dibawah.





Coding :

```
Text
1 int ledPin = 13;
2
3 void setup() {
4   pinMode(ledPin, OUTPUT);
5 }
6
7 void loop() {
8
9   digitalWrite(ledPin, HIGH);
10  delay(1000);
11  digitalWrite(ledPin, LOW);
12  delay(1000);
13
14 }
```

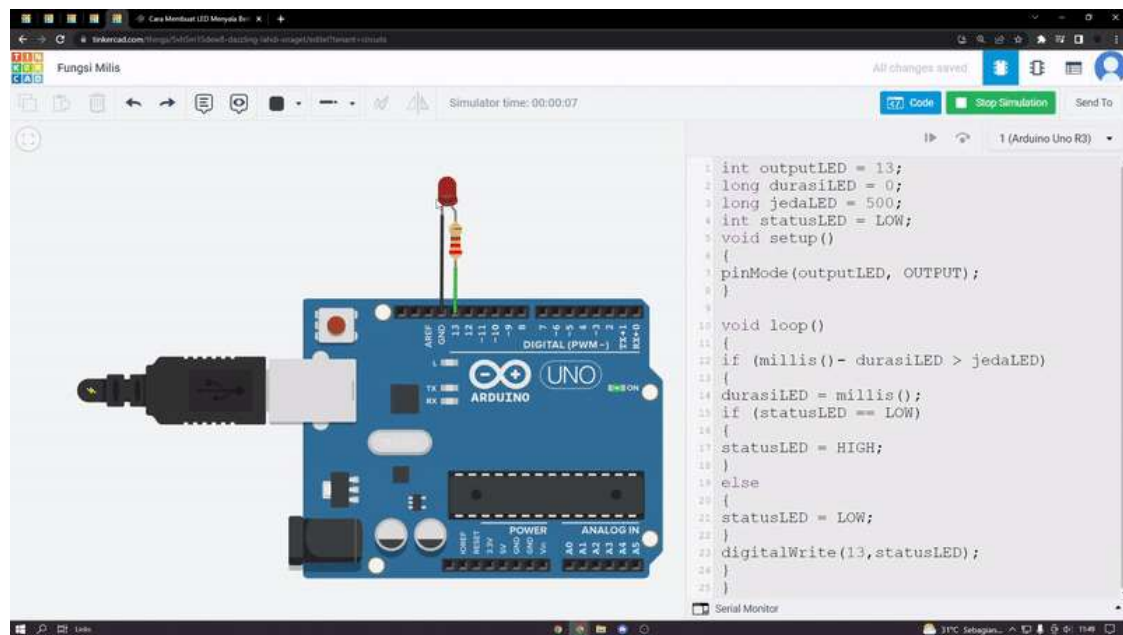
Penjelasan Praktikum :

- Pertama, kita buat terlebih dahulu untuk rangkaiannya.
- Setelah itu, kita tuliskan kode programnya.
- Bagian delay (1000) ini merupakan jeda untuk baris kode, yaitu 1000 milisecond atau 1 detik.
- Delay ini bisa kita sesuaikan dengan kebutuhan.



## 7.2 Praktikum dengan Fungsi Millis

Buatlah rangkaian seperti gambar dibawah.



Coding :

```
Text 1 (Arduino Uno R3)
1 int outputLED = 13;
2 long durasiLED = 0;
3 long jedaLED = 500;
4 int statusLED = LOW;
5 void setup()
6 {
7   pinMode(outputLED, OUTPUT);
8 }
9
10 void loop()
11 {
12   if (millis() - durasiLED > jedaLED)
13   {
14     durasiLED = millis();
15     if (statusLED == LOW)
16     {
17       statusLED = HIGH;
18     }
19     else
20     {
21       statusLED = LOW;
22     }
23     digitalWrite(13, statusLED);
24   }
25 }
```

Penjelasan Praktikum :

- Fungsi `delay()` terkadang dianggap kurang efisien. Hal ini karena fungsi ini mencegah perintah lain untuk dieksekusi, sehingga tidak bisa diterapkan di program multitasking.
- Fungsi `millis()` merupakan fungsi yang cocok untuk kondisi seperti ini.

## 8. Tugas dan Tantangan

Setelah mengerjakan praktikum diatas, sebelum dikumpulkan kerjakan tugas dibawah.

### **Tugas Untuk Langkah Praktikum 2 – Serial Monitor**

- Setelah berhasil melakukan praktikum seperti contoh, Buatlah tampilan di serial monitor sesuai dengan “Nama Lengkap” kalian masing-masing.

### **Tugas Untuk Langkah Praktikum 3 – Fungsi Digital I/O**

- Sebagai challenge untuk kalian, buatlah Digital I/O menggunakan 4 LED dan 4 Push Button.

### **Tugas Untuk Langkah Praktikum 4 – Fungsi Analog I/O**

- Untuk menguji pemahaman kalian, buatlah sebuah Analog I/O menggunakan Variable Resistor yang lainnya yaitu sensor LDR.

### **Tugas Untuk Langkah Praktikum 5 – Fungsi Time**

- Selanjutnya, buatlah jeda nyala dan mati LED menjadi 1,5 detik.

Penjelasan Praktikum :

- Fungsi delay() terkadang dianggap kurang efisien. Hal ini karena fungsi ini mencegah perintah lain untuk dieksekusi, sehingga tidak bisa diterapkan di program multitasking.
- Fungsi millis() merupakan fungsi yang cocok untuk kondisi seperti ini.

### **Challenge**

- Selanjutnya, buatlah jeda nyala dan mati LED menjadi 1,5 detik.

## **8. Tugas Pembelajaran**

- Membuat tampilan di serial monitor sesuai dengan "Nama Lengkap" masing-masing.
- Membuat Digital I/O menggunakan 4 LED dan 4 Push Button.
- Membuat Sebuah Analog I/O menggunakan Sensor "LDR".
- Membuat 2 LED dengan nyala waktu 2 detik dan 1 Detik dengan menggunakan delay.
- Membuat jeda nyala dan mati LED menjadi 1,5 detik menggunakan millis.