LAPORAN PRAKTIKUM TEKNIK KENDALI DAN MESIN LISTRIK

DISUSUN OLEH:

Kelompok 2

ARM 2

Ade Surya Pramudya (19/441188/SV/16540)

Mahesa Audriansyah Agatha (19/441203/SV/16555)

DEPARTEMEN TEKNIK MESIN SEKOLAH VOKASI

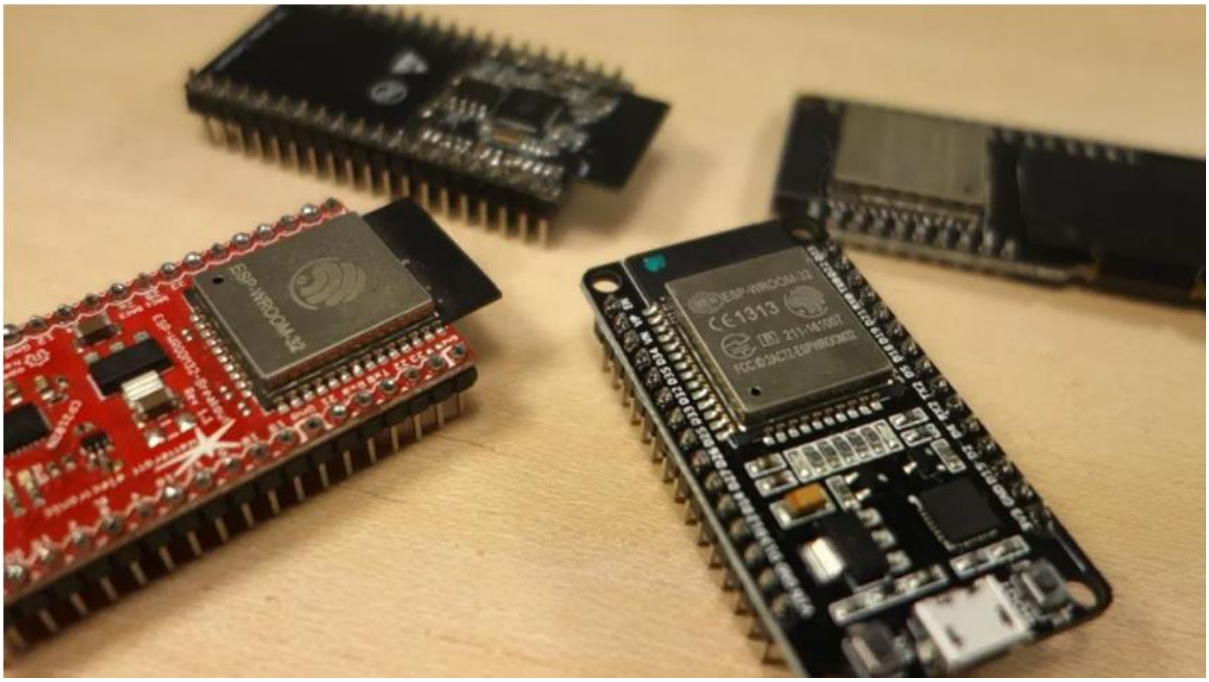
UNIVERSITAS GADJAH MADA

YOGYAKARTA

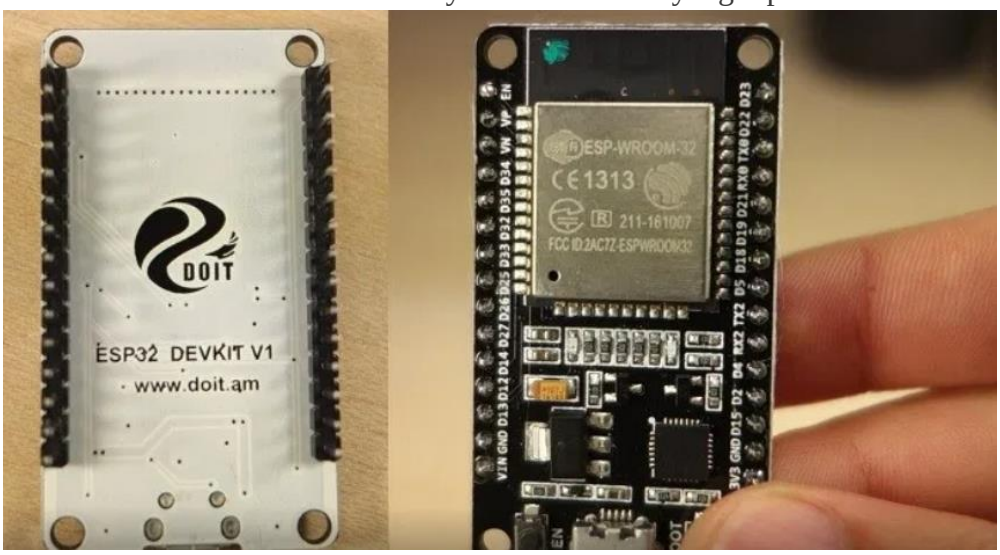
SEJARAH MIKROKONTROLER ESP32

ESP32 adalah mikrokontroller berharga rendah dan hemat energi dengan wifi dan dual-mode bluetooth terintegrasi. Generasi ESP32 menggunakan mikroprosesor Tensilica Xtensa LX6 sebagai inti. Baik dalam mode single-core maupun dual-core.

ESP32 dibuat oleh Espressif Systems, perusahaan berbasis di Shanghai, Tiongkok. Menggunakan TSMC sebagai pemproduksi inti dengan besar 40nm.

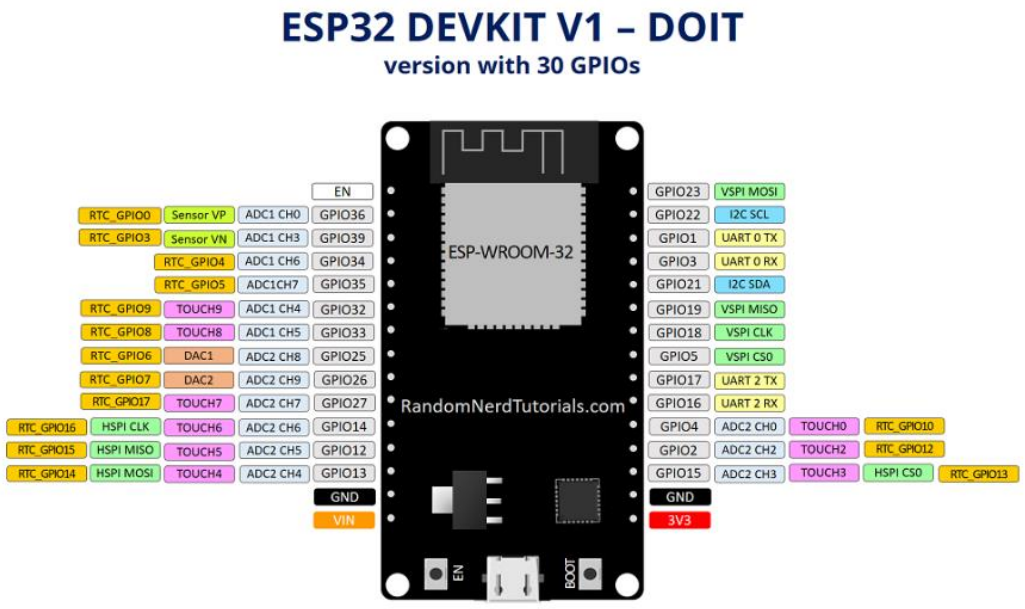


1. Mengetahui Spesifikasi Mikrokontroler ESP32
   Pada praktikum Teknik Kendali dan Mesin Listrik kami menggunakan board ESP32 yang memiliki chip ESP-WROOM-32, dan untuk referensi yang kami gunakan adalah board ESP32 DEVKIT DOIT menyesuaikan modul yang dipakai.



Number of cores	2 (dual core)
Wi-Fi	2.4 GHz up to 150 Mbits/s
Bluetooth	BLE (Bluetooth Low Energy) and legacy Bluetooth
Architecture	32 bits
Clock frequency	Up to 240 MHz
RAM	512 KB
Pins	30 or 36 (depends on the model)
Peripherals	Capacitive touch, ADC (analog to digital converter), DAC (digital to analog converter), I2C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Netwokr), SPI (Serial Peripheral Interface), I2S (Integrated Inter-IC Sound), RMII (Reduced Media-Independent Interface), PWM (pulse width modulation), and more.

2. Pinout / kaki-kaki ESP32 beserta fungsinya
   ESP32 memiliki lebih banyak GPIO dengan lebih banyak fungsi dibandingkan dengan ESP826. Dengan ESP32 kami dapat memutuskan pin mana yang UART, I2C, atau SPI, kami hanya perlu mengaturnya pada kode. Hal ini dimungkinkan karena fitur multiplexing chip ESP32 yang memungkinkan untuk menetapkan beberapa fungsi ke pin yang sama. Jika tidak mengaturnya pada kode, pin akan digunakan sebagai default seperti yang ditunjukkan pada gambar di bawah ini (lokasi pin dapat berubah tergantung pada produsen).



Selain itu, ada pin dengan fitur khusus yang membuatnya cocok atau tidak untuk proyek tertentu. Tabel berikut menunjukkan pin apa yang terbaik untuk digunakan sebagai input, output dan mana yang perlu Anda hati-hati.

Pin yang disorot dengan warna hijau tidak apa-apa untuk digunakan. Yang disorot dalam warna kuning tidak apa-apa untuk digunakan, tetapi Anda perlu memperhatikan karena mereka mungkin memiliki perilaku yang tidak terduga terutama saat boot. Pin yang disorot dengan warna merah tidak disarankan untuk digunakan sebagai input atau output.

GPIO	Input	Output	Notes
0	pulled up	OK	outputs PWM signal at boot
1	TX pin	OK	debug output at boot
2	OK	OK	connected to on-board LED
3	OK	RX pin	HIGH at boot
4	OK	OK	
5	OK	OK	outputs PWM signal at boot
6	X	X	connected to the integrated SPI flash
7	X	X	connected to the integrated SPI flash
8	X	X	connected to the integrated SPI flash
9	X	X	connected to the integrated SPI flash
10	X	X	connected to the integrated SPI flash
11	X	X	connected to the integrated SPI flash
12	OK	OK	boot fail if pulled high
13	OK	OK	
14	OK	OK	outputs PWM signal at boot
15	OK	OK	outputs PWM signal at boot
16	OK	OK	
17	OK	OK	

18	OK	OK	
19	OK	OK	
21	OK	OK	
22	OK	OK	
23	OK	OK	
25	OK	OK	
26	OK	OK	
27	OK	OK	
32	OK	OK	
33	OK	OK	
34	OK		input only
35	OK		input only
36	OK		input only
39	OK		input only

A. Input Only Pins

GPIO 34 hingga 39 adalah GPU – hanya input pin. Pin ini tidak memiliki resistor pull-up atau pull-down internal. Mereka tidak dapat digunakan sebagai output, jadi gunakan pin ini hanya sebagai input:

- GPIO 34
- GPIO 35
- GPIO 36
- GPIO 39

B. SPI Flash Integrated on the ESP-WROOM-3

GPIO 6 hingga GPIO 11 diekspos di beberapa papan pengembangan ESP32. Namun, pin ini terhubung ke flash SPI terintegrasi pada chip ESP-WROOM-32 dan tidak direkomendasikan untuk penggunaan lain. Jadi, jangan gunakan pin ini dalam proyek Anda:

- GPIO 6 (SCK/CLK)
- GPIO 7 (SDO/SD0)
- GPIO 8 (SDI/SD1)

- GPIO 9 (SHD/SD2)
- GPIO 10 (SWP/SD3)
- GPIO 11 (CSC/CMD)

C. Analog Digital Converter (ADC)

ESP32 memiliki saluran input ADC 18 x 12 bit (sedangkan ESP8266 hanya memiliki ADC 1x 10 bit). Ini adalah GPIO yang dapat digunakan sebagai ADC dan saluran masing-masing:

- ADC1_CH0 (GPIO 36)
- ADC1_CH1 (GPIO 37)
- ADC1_CH2 (GPIO 38)
- ADC1_CH3 (GPIO 39)
- ADC1_CH4 (GPIO 32)
- ADC1_CH5 (GPIO 33)
- ADC1_CH6 (GPIO 34)
- ADC1_CH7 (GPIO 35)
- ADC2_CH0 (GPIO 4)
- ADC2_CH1 (GPIO 0)
- ADC2_CH2 (GPIO 2)
- ADC2_CH3 (GPIO 15)
- ADC2_CH4 (GPIO 13)
- ADC2_CH5 (GPIO 12)
- ADC2_CH6 (GPIO 14)
- ADC2_CH7 (GPIO 27)
- ADC2_CH8 (GPIO 25)
- ADC2_CH9 (GPIO 26)

D. Digital Analog Converter (DAC)

Ada saluran DAC 2 x 8 bit pada ESP32 untuk mengubah sinyal digital menjadi output sinyal tegangan analog. Ini adalah saluran DAC:

- DAC1 (GPIO25)
- DAC2 (GPIO26)

E. PWM

Pengontrol PWM LED ESP32 memiliki 16 saluran independen yang dapat dikonfigurasi untuk menghasilkan sinyal PWM dengan properti yang berbeda. Semua pin yang dapat bertindak sebagai output dapat digunakan sebagai pin PWM (GPIO 34 hingga 39 tidak dapat menghasilkan PWM).

Untuk mengatur sinyal PWM, Anda perlu menentukan parameter ini dalam kode:

- Frekuensi sinyal;
- Siklus tugas;
- Saluran PWM;
- GPIO di mana Anda ingin mengeluarkan sinyal.

F. I2C

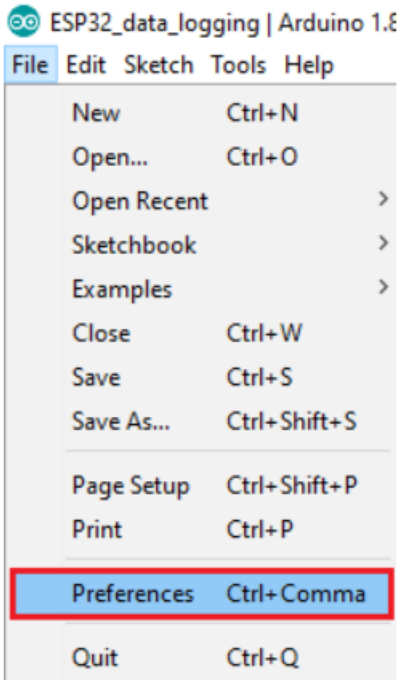
ESP32 memiliki dua saluran I2C dan pin apa pun dapat diatur sebagai SDA atau SCL. Saat menggunakan ESP32 dengan Arduino IDE, pin I2C default adalah:

- GPIO 21 (SDA)
- GPIO 22 (SCL)

3. Mengetahui Penggunaan Arduino IDE

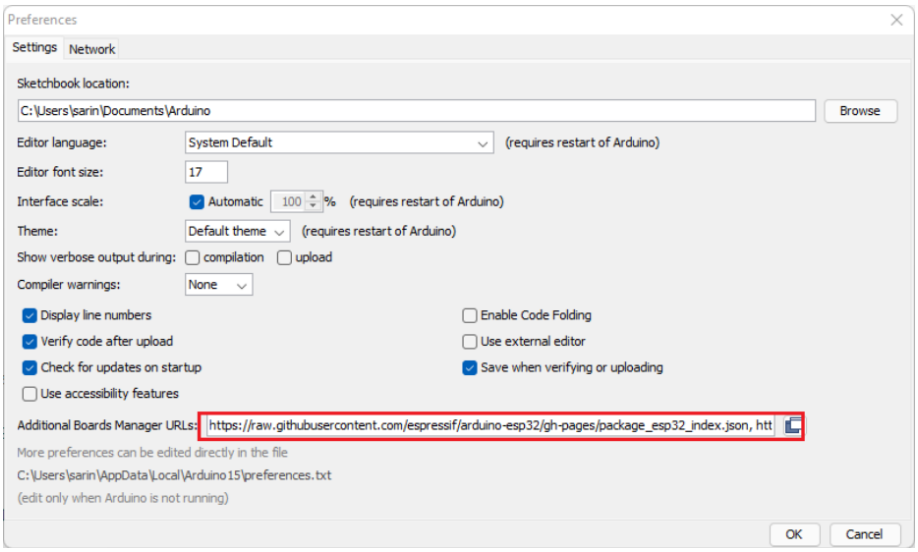
Penggunaan Arduino IDE berfungsi sebagai pembuatan perintah program yang akan digunakan pada mikrokontroler ESP32 yang kami gunakan, untuk proses penggunaannya berikut adalah langkah yang kami ikuti menurut referensi :

A. Download dan Install Arduino IDE versi Terbaru
B. Install ESP32 add-on di Arduino IDE
   - Jalankan Arduino IDE, pergi ke **file > preference**

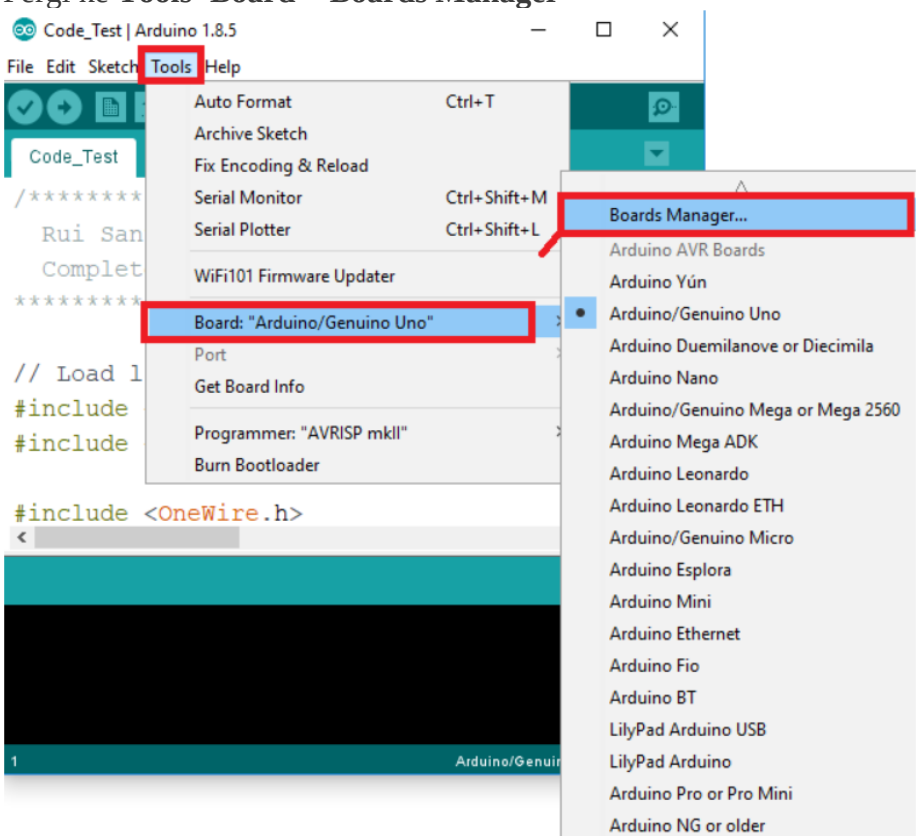


   - Masukkan link berikut pada **Addition Board Manager URLs** lalu Klik **OK**

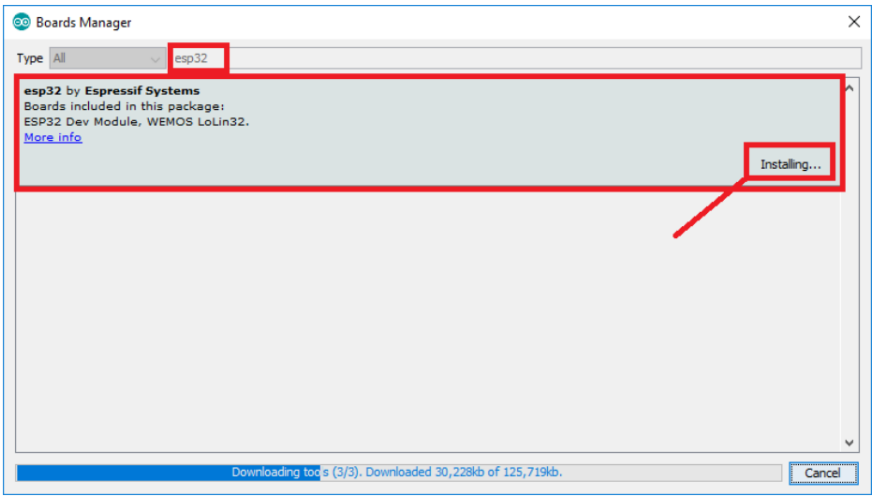
```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
```



- Pergi ke **Tools>Board > Boards Manager**

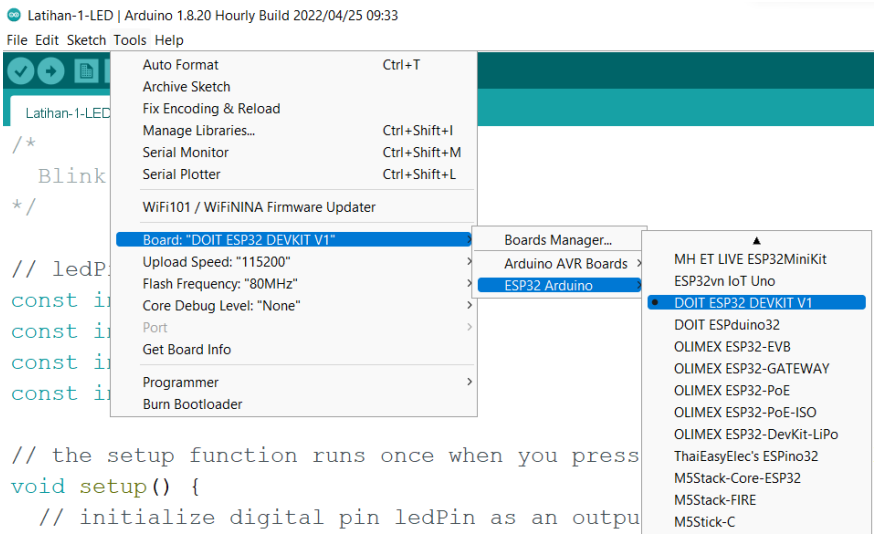


- Ketikkan **ESP32** lalu klik install untuk add-on "**ESP32 by Espressif Systems**"

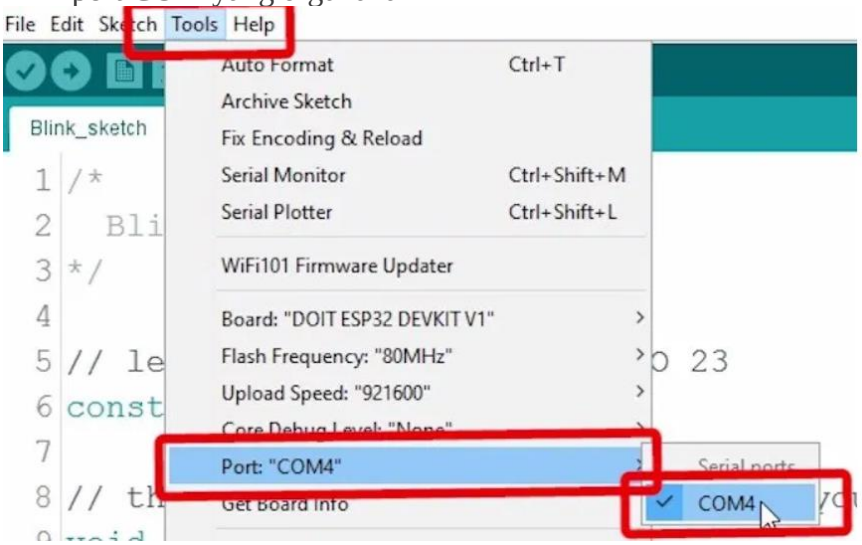


C. Testing Code di Arduino IDE dalam pembuatan LED Blink

- Pergi ke **Tools** > **Board**, cari ESP32 section dan pilih model ESP32 board yang dipakai. Disini kami menggunakan ESP32 DEVKIT V1 board.



- Pilih port COM yang digunakan



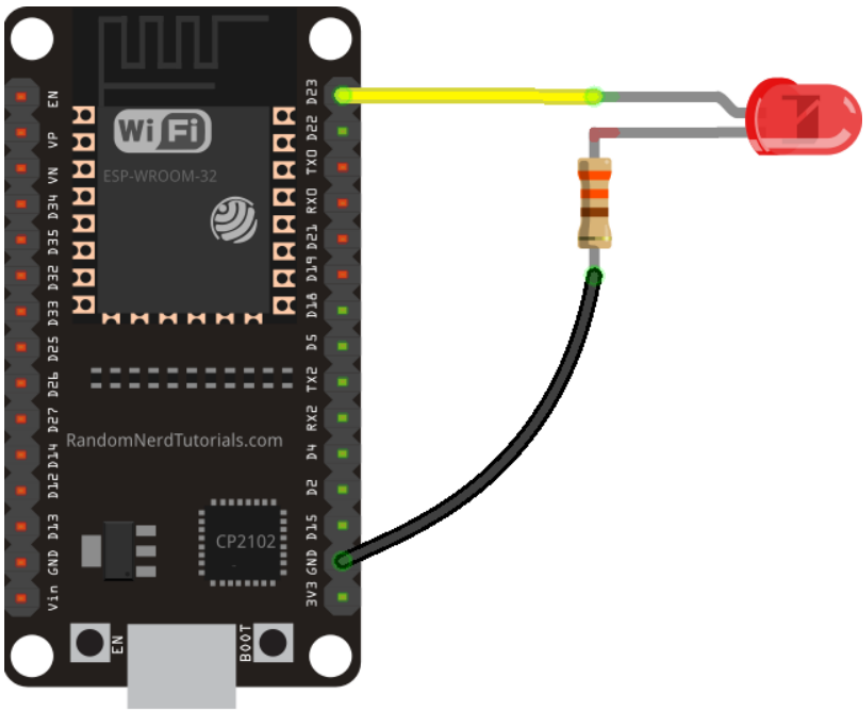
- Masukkan Code Perintah yang diinginkan

```
/*
  Blink
*/

// ledPin refers to ESP32 GPIO 23
const int ledPin1 = 27;
const int ledPin2 = 26;
const int ledPin3 = 25;
const int ledPin4 = 33;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin ledPin as an output.
  Serial.begin(115200);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  //Turn On
  digitalWrite(ledPin1, HIGH);   // turn the LED on (HIGH is the voltage level)
  Serial.println("LED1 is on");
  delay(1000);                   // wait for a second
  digitalWrite(ledPin2, HIGH);   // turn the LED on (HIGH is the voltage level)
  Serial.println("LED2 is on");
  delay(1000);                   // wait for a second
  digitalWrite(ledPin3, HIGH);   // turn the LED on (HIGH is the voltage level)
  Serial.println("LED3 is on");
  delay(1000);                   // wait for a second
  digitalWrite(ledPin4, HIGH);   // turn the LED on (HIGH is the voltage level)
  Serial.println("LED4 is on");
  delay(1000);                   // wait for a second

  //Turn Off
  digitalWrite(ledPin1, LOW);    // turn the LED off by making the voltage LOW
  Serial.println("LED1 is off");
  delay(1000);                   // wait for a second
  digitalWrite(ledPin2, LOW);    // turn the LED off by making the voltage LOW
  Serial.println("LED2 is off");
  delay(1000);                   // wait for a second
```

```
  digitalWrite(ledPin3, LOW);      // turn the LED off by making the
voltage LOW
  Serial.println("LED3 is off");
  delay(1000);                 // wait for a second
  digitalWrite(ledPin4, LOW);      // turn the LED off by making the
voltage LOW
  Serial.println("LED4 is off");
  delay(1000);                 // wait for a second
}
```

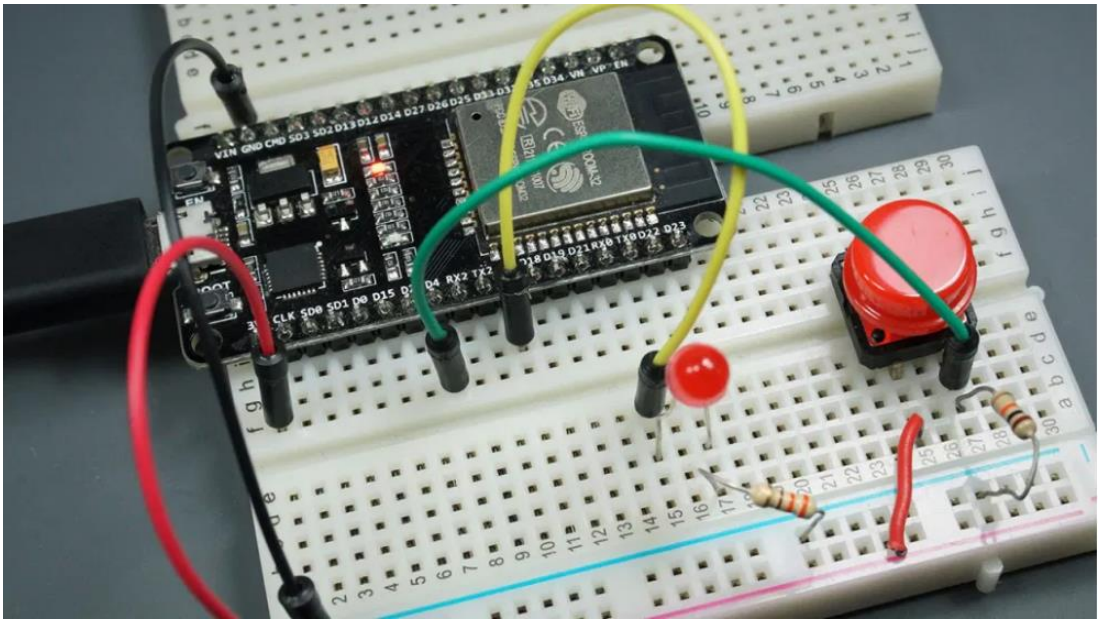
- Klik **Upload** lalu tunggu hingga program diunngah ke kontroler dan mendapatkan pesan **Done Uploading**, maka program telah bisa digunakan

- Schematic demonstration



4. Mengetahui tentang ESP32 digital input dan digital output
   Dalam pembelajaran pada input dan output, kami menganalogikan program perintah on/off LED menggunakan Button dengan cara membaca input digital seperti tombol sakelar tombol dan mengontrol output digital seperti LED menggunakan ESP32 dengan Arduino IDE. Berikut ini adalah langkah yang kami lakukan dalam menyalakan 4 LED menggunakan 4 button :

   A. Pembuatan skema jalur jumper yang menghubungkan LED, BUTTON, dan Mikrokontroler ESP32



B. Mengatur dalam pemberian kontrol kondisi atas nama, pin yang dipakai serta input dan output yang akan digunakan

- Inisialisasi nama dan pin yang dipakai

```
const int ledPin1 = 27;
const int buttonPin1 = 21;
```

- Inisialisasi kondisi awal button yang dipakai

```
//Button Condition
int buttonState1 = 0;
```

- Inisisalisasi input dan output yang dipakai

```
// initialize digital pin ledPin as an output.
Serial.begin(115200);
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
pinMode(ledPin4, OUTPUT);
pinMode(buttonPin1, INPUT);
pinMode(buttonPin2, INPUT);
pinMode(buttonPin3, INPUT);
pinMode(buttonPin4, INPUT);
```

Berfungsi untuk mengkonfirmasi perintah input dan output yang dipakai

```
buttonState1 = digitalRead(buttonPin1);
digitalWrite(ledPin1, HIGH);
```

C. Pembuatan perintah kode pada Arduino IDE

- Membuat beberapa nama sebagai variabel untuk membedakan pin yang dipakai

```
const int ledPin1 = 27;
const int ledPin2 = 26;
const int ledPin3 = 25;
const int ledPin4 = 33;
const int buttonPin1 = 21;
const int buttonPin2 = 19;
const int buttonPin3 = 18;
const int buttonPin4 = 5;
```

- Membuat variable atas kondisi button yang mana jika 0 maka button tersebut tidak tertekan

```
//Button Condition
int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;
```

- Menginisialisasi **Button** sebagai Input dan **LED** sebagai Output

```
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
pinMode(ledPin4, OUTPUT);
pinMode(buttonPin1, INPUT);
pinMode(buttonPin2, INPUT);
pinMode(buttonPin3, INPUT);
pinMode(buttonPin4, INPUT);
```

- Mengkonfirmasi pembacaan status kondisi butoon menggunakan variabel **buttonstate** dan menggunakan fungsi **digitalread()**

```
buttonState1 = digitalRead(buttonPin1);
buttonState2 = digitalRead(buttonPin2);
buttonState3 = digitalRead(buttonPin3);
buttonState4 = digitalRead(buttonPin4);
```

- Memberikan pernyataan apabila **buttonstate** adalah **High,** LED akan menyala dengan fungsi **digitalwrite()** yang menerima status dengan kondisi **ledpin,** dan status **High.**

```
if (buttonState3 == HIGH) {
  // turn LED on
  digitalWrite(ledPin3, HIGH);
  Serial.println("LED3 is on");
```

- Jika kondisi **buttonstate** adalah **Low,** kami memberikan status kedua pada LED agar tidak menyala dengan fungsi **digitalwrite()** yang menerima status **ledpin,** dan status **Low.**

```
else {
// turn LED off
digitalWrite(ledPin3, LOW);
Serial.println("LED3 is on");
```

D. Uploading Program Perintah menggunakan Arduino IDE

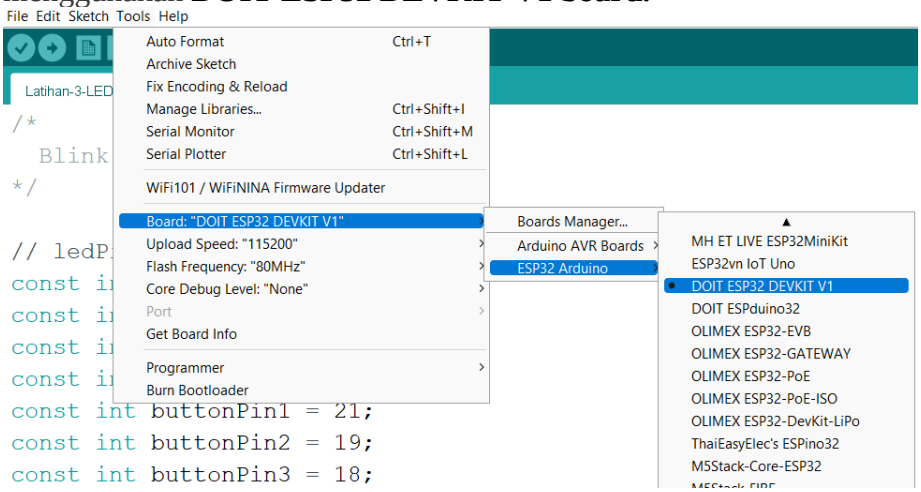
- Siapkan kode perintah yang sudah disiapkan

```
/*
  Blink
*/

// ledPin and buttonPin refers to ESP32 GPIO 23
const int ledPin1 = 27;
const int ledPin2 = 26;
const int ledPin3 = 25;
const int ledPin4 = 33;
const int buttonPin1 = 21;
const int buttonPin2 = 19;
const int buttonPin3 = 18;
const int buttonPin4 = 5;

//Button Condition
int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin ledPin as an output.
  Serial.begin(115200);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
```

```
  pinMode(buttonPin3, INPUT);
  pinMode(buttonPin4, INPUT);
}

// the loop function runs over and over again forever
void loop() {

  //Rangkaian 1
  buttonState1 = digitalRead(buttonPin1);
  Serial.println(buttonState1);
  if (buttonState1 == HIGH) {
    // turn LED on
    digitalWrite(ledPin1, HIGH);
    Serial.println("LED1 is on");
  } else {
    // turn LED off
    digitalWrite(ledPin1, LOW);
    Serial.println("LED1 is on");
  }

  //Rangkaian 2
  buttonState2 = digitalRead(buttonPin2);
  Serial.println(buttonState2);
  if (buttonState2 == HIGH) {
    // turn LED on
    digitalWrite(ledPin2, HIGH);
    Serial.println("LED2 is on");
  } else {
    // turn LED off
    digitalWrite(ledPin2, LOW);
    Serial.println("LED2 is on");
  }

 //Rangkaian 3
  buttonState3 = digitalRead(buttonPin3);
  Serial.println(buttonState3);
  if (buttonState3 == HIGH) {
    // turn LED on
    digitalWrite(ledPin3, HIGH);
    Serial.println("LED3 is on");
  } else {
    // turn LED off
    digitalWrite(ledPin3, LOW);
    Serial.println("LED3 is on");
  }

  //Rangkaian 4
  buttonState4 = digitalRead(buttonPin4);
  Serial.println(buttonState4);
  if (buttonState4 == HIGH) {
```

```
    // turn LED on
    digitalWrite(ledPin4, HIGH);
    Serial.println("LED4 is on");
  } else {
    // turn LED off
    digitalWrite(ledPin4, LOW);
    Serial.println("LED4 is on");
  }
}
```

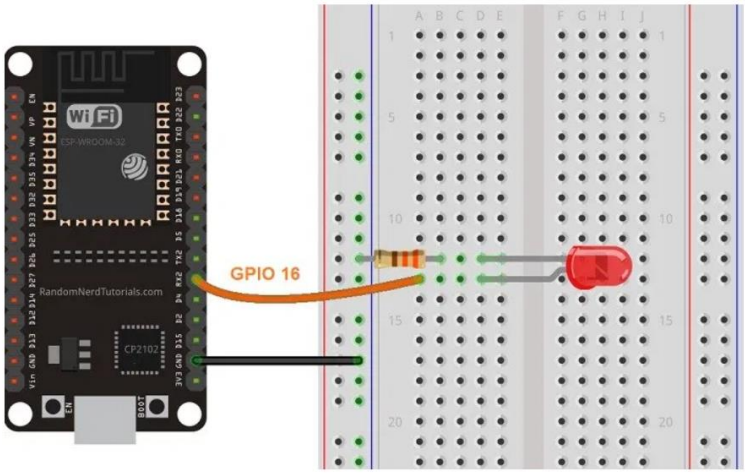
- Pergi ke **Tools > Board** pilih board yang akan digunakan, kami menggunakan **DOIT ESP32 DEVKIT V1 board.**



- Melakukan penyesuaian port, Buka **Tools > Port** dan pilih port **COM** yang terhubung dengan ESP32. Kemudian, tekan tombol unggah dan tunggu pesan "**Done Uploading**".

5. Mengetahui tentang ESP32 PWM menggunakan Arduino IDE
   Menghasilkan sinyal PWM dengan ESP32 menggunakan Arduino IDE. Sebagai contoh kita akan membangun sirkuit sederhana yang meredupkan LED menggunakan pengontrol PWM LED dari ESP32.

   A. Mempersiapkan komponen part yang dibutuhkan
   B. Pembuatan skematik jalur yang menyambungkan LED dan ESP32



C. Pembuatan Perintah Kode pada Arduino IDE

- Pemberian definisi pin yang akan dipakai dan disambungkan antara LED dan ESP32

```
// the number of the LED pin
const int ledPin1 = 27;
const int ledPin2 = 26;
const int ledPin3 = 25;
const int ledPin4 = 33;
```

- Kemudian, mengatur properti sinyal PWM. menentukan frekuensi 5000 Hz, pilih **channel** 0 untuk menghasilkan sinyal, dan atur **resolution** 8 bit. Anda dapat memilih properti lain, berbeda dari ini, untuk menghasilkan sinyal PWM yang berbeda.

```
// setting PWM properties
const int freq = 5000;
const int ledChannel = 0;
const int resolution = 8;
```

- mengonfigurasi LED PWM dengan properti yang telah ditentukan sebelumnya dengan menggunakan fungsi **ledcSetup()** yang menerima sebagai argumen, **ledChannel**, **frequency**, dan **resolution**.

```
void setup(){
  // configure LED PWM functionalitites
  ledcSetup(ledChannel, freq, resolution);
```

- memilih GPIO untuk mendapatkan sinyal. Dan menggunakan fungsi **ledcAttachPin()** yang menerima sebagai argumen GPIO untuk mendapatkan sinyal, dan channel yang menghasilkan sinyal.

```
// attach the channel to the GPIO to be controlled
ledcAttachPin(ledPin1, ledChannel);
ledcAttachPin(ledPin2, ledChannel);
ledcAttachPin(ledPin3, ledChannel);
ledcAttachPin(ledPin4, ledChannel);
```

- Dalam loop, membuat kode perintah untuk memvariasikan siklus tugas mulai dari 0 sampai 255 untuk meningkatkan kecerahan LED. Dan kemudian, mulai dari 255 sampai 0 untuk mengurangi kecerahan.

```
void loop(){
  // increase the LED brightness
  for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++){
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
  }

  // decrease the LED brightness
  for(int dutyCycle = 255; dutyCycle >= 0; dutyCycle--){
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
  }
}
```

- Untuk mengatur kecerahan LED, menggunakan fungsi **ledcWrite()** untuk menerima perintah sebagai argumen saluran yang menghasilkan sinyal, dan **dutycycle**.

```
ledcWrite(ledChannel, dutyCycle);
```

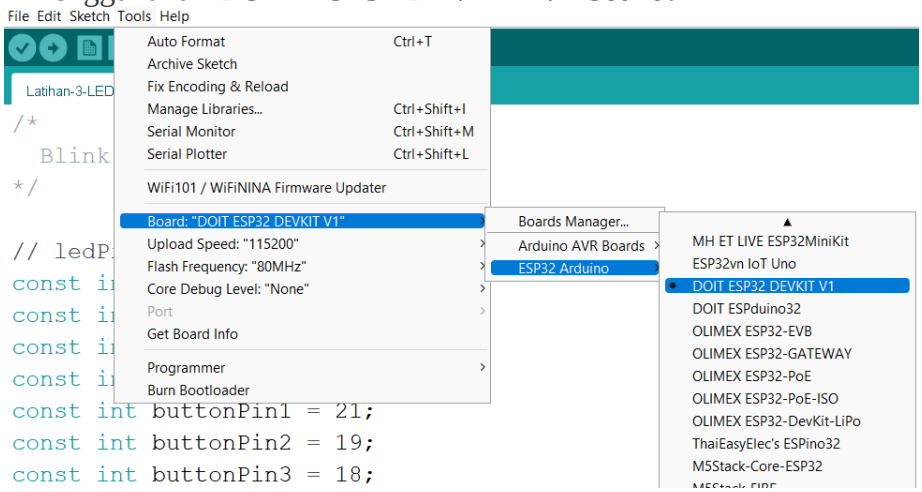
D. Uploading Program Perintah menggunakan Arduino IDE

- Siapkan kode perintah yang sudah disiapkan

```
// the number of the LED pin
const int ledPin1 = 27;
const int ledPin2 = 26;
const int ledPin3 = 25;
const int ledPin4 = 33;

// setting PWM properties
const int freq = 5000;
const int ledChannel = 0;
const int resolution = 8;

void setup(){
 // configure LED PWM functionalitites
 ledcSetup(ledChannel, freq, resolution);

 // attach the channel to the GPIO to be controlled
 ledcAttachPin(ledPin1, ledChannel);
 ledcAttachPin(ledPin2, ledChannel);
 ledcAttachPin(ledPin3, ledChannel);
 ledcAttachPin(ledPin4, ledChannel);
}

void loop(){
 // increase the LED brightness
 for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++){
  // changing the LED brightness with PWM
```

```
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
  }

  // decrease the LED brightness
  for(int dutyCycle = 255; dutyCycle >= 0; dutyCycle--){
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
  }
}
```

- Pergi ke **Tools > Board** pilih board yang akan digunakan, kami menggunakan **DOIT ESP32 DEVKIT V1 board.**



- Melakukan penyesuaian port, Buka **Tools > Port** dan pilih port **COM** yang terhubung dengan ESP32. Kemudian, tekan tombol unggah dan tunggu pesan "**Done Uploading**".

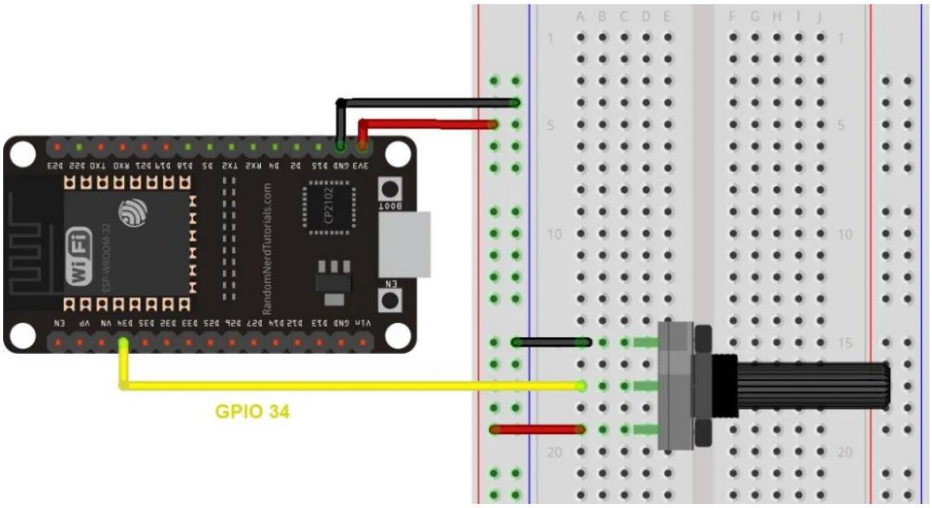
6. Mengetahui ESP32 ADC- Pembacaan Analog Values menggunakan Arduino IDE
   Pembacaan input analog dengan ESP32 menggunakan Arduino IDE. Pembacaan analog berguna untuk membaca nilai dari resistor variabel seperti potensiometer, atau sensor analog.
   A. Analog Input (ADC)
      Membaca nilai analog dengan ESP32 berarti Anda dapat mengukur tingkat tegangan yang bervariasi antara 0 V dan 3,3 V.

      Tegangan yang diukur kemudian ditetapkan ke nilai antara 0 dan 4095, di mana 0 V sesuai dengan 0, dan 3,3 V sesuai dengan 4095. Setiap tegangan antara 0 V dan 3,3 V akan diberikan nilai yang sesuai di antaranya.

   B. AnalogRead() Function
      Membaca input analog dengan ESP32 menggunakan Arduino IDE semudah menggunakan fungsi **analogRead()**. Ini menerima sebagai argumen, GPIO yang ingin dibaca

```
analogRead(GPIO);
```

C. Pembuatan Skematik



D. Pembuatan Perintah Kode pada Arduino IDE

- Dalam pembuatan kode, mulai dengan mendefinisikan GPIO potensiometer yang terhubung. Dalam contoh ini, GPIO 34.

```
const int potPin = 34;
```

- pada **setup()**, inisialisasi serial communication dengan baud rate adalah 115200.

```
Serial.begin(115200);
```

- Dalam **loop()**, gunakan fungsi **analogRead()** untuk membaca input analog dari **potPin**.

```
potValue = analogRead(potPin);
```

- Terakhir, cetak nilai yang dibaca dari potensiometer di serial monitor.

```
Serial.println(potValue);
```

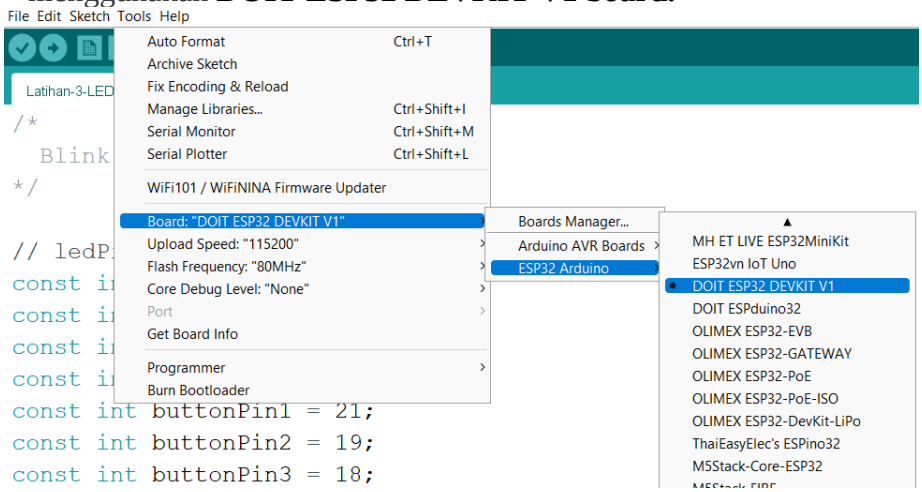
E. Uploading Program Perintah menggunakan Arduino IDE

- Siapkan kode perintah yang sudah disiapkan

```
// Potentiometer is connected to GPIO 34 (Analog ADC1_CH6)
const int potPin = 34;

// variable for storing the potentiometer value
int potValue = 0;

void setup() {
  Serial.begin(115200);
  delay(1000);
}

void loop() {
  // Reading potentiometer value
```

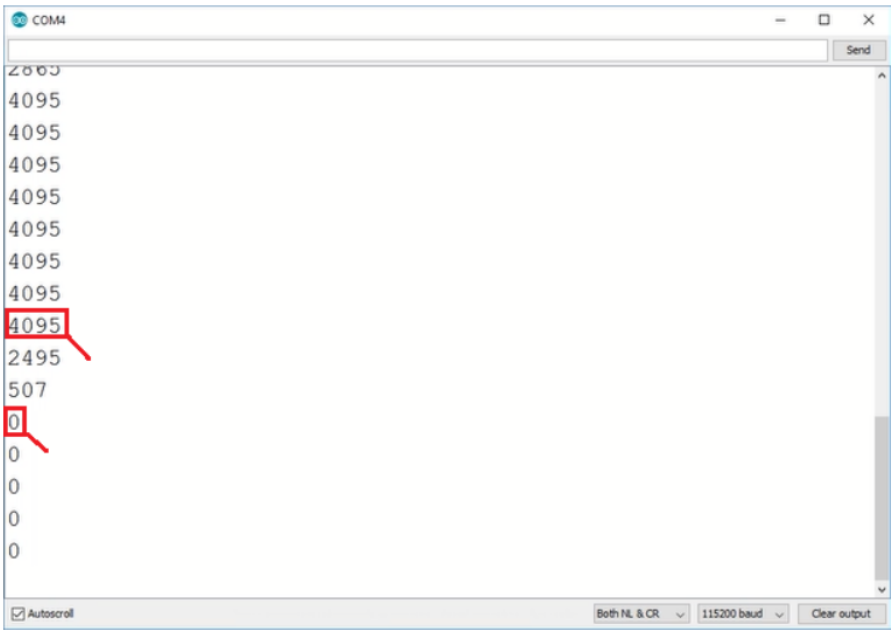
```
  potValue = analogRead(potPin);
  Serial.println(potValue);
  delay(500);
}
```

- Pergi ke **Tools > Board** pilih board yang akan digunakan, kami menggunakan **DOIT ESP32 DEVKIT V1 board.**



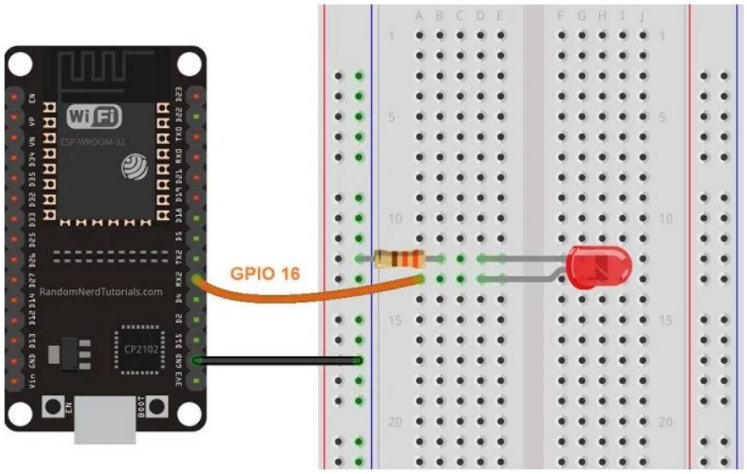
- Melakukan penyesuaian port, Buka **Tools > Port** dan pilih port **COM** yang terhubung dengan ESP32. Kemudian, tekan tombol unggah dan tunggu pesan "**Done Uploading**".

- Run Serial Monitor yang ada di bagian kanan atas



7. Mengetahui tentang ESP32 PWM with Button menggunakan Arduino IDE
   Menghasilkan sinyal PWM dengan ESP32 menggunakan Arduino IDE. Sebagai contoh kita akan membangun sirkuit sederhana yang meredupkan LED menggunakan pengontrol PWM LED dari ESP32 ditambah dengan menggunakan button untuk control tambahan.

A. Pembuatan skematik jalur yang menyambungkan LED dan ESP32



B. Pembuatan Perintah Kode pada Arduino IDE

- Pemberian definisi pin yang akan dipakai dan disambungkan antara LED, Button dan ESP32

```
//Introduce LED and Button PIN
const int ledPin1 = 27;
const int ledPin2 = 26;
const int ledPin3 = 25;
const int ledPin4 = 33;
const int buttonPin1 = 21;
const int buttonPin2 = 19;
```

- Kemudian, mengatur properti sinyal PWM. menentukan frekuensi 5000 Hz, pilih **channel** 0 untuk menghasilkan sinyal, dan atur **resolution** 8 bit. Anda dapat memilih properti lain, berbeda dari ini, untuk menghasilkan sinyal PWM yang berbeda.

```
// setting PWM properties
const int freq = 5000;
const int ledChannel = 0;
const int resolution = 8;
```

- Membuat variable atas kondisi button yang mana jika 0 maka button tersebut tidak tertekan

```
//Button Condition
int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;
```

- mengonfigurasi LED PWM dengan properti yang telah ditentukan sebelumnya dengan menggunakan fungsi **ledcSetup()** yang menerima sebagai argumen, **ledChannel**, **frequency**, dan **resolution**.

```
void setup(){
  // configure LED PWM functionalitites
  ledcSetup(ledChannel, freq, resolution);
```

- memilih GPIO untuk mendapatkan sinyal. Dan menggunakan fungsi **ledcAttachPin()** yang menerima sebagai argumen GPIO untuk mendapatkan sinyal, dan channel yang menghasilkan sinyal serta menyatakan status dari button sebagai input.

```
// put your setup code here, to run once:
Serial.begin(115200);
pinMode(buttonPin1, INPUT);
pinMode(buttonPin2, INPUT);
ledcSetup(ledChannel, freq, resolution);
ledcAttachPin(ledPin1, ledChannel);
ledcAttachPin(ledPin2, ledChannel);
ledcAttachPin(ledPin3, ledChannel);
ledcAttachPin(ledPin4, ledChannel);
```

- Mengkonfirmasi pembacaan status kondisi butoon menggunakan variabel **buttonstate** dan menggunakan fungsi **digitalread()**

```
buttonState1 = digitalRead(buttonPin1);
buttonState2 = digitalRead(buttonPin2);
```

- Dalam loop, membuat kode perintah untuk memvariasikan siklus tugas mulai dari 0 sampai 255 untuk meningkatkan kecerahan LED. Dan kemudian, mulai dari 255 sampai 0 untuk mengurangi kecerahan.

```
void loop() {
  // put your main code here, to run repeatedly:
  //Rangkaian 1
  buttonState1 = digitalRead(buttonPin1);
  Serial.println(buttonState1);
  if (buttonState1 == LOW) {
    for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++){
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
    }
  }
  buttonState2 = digitalRead(buttonPin2);
  Serial.println(buttonState2);
  if (buttonState2 == LOW) {
    for(int dutyCycle = 255; dutyCycle >= 0; dutyCycle--){
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
    }
  }
```

- Untuk mengatur kecerahan LED, menggunakan fungsi **ledcWrite()** untuk menerima perintah sebagai argumen saluran yang menghasilkan sinyal, dan **dutycycle**.

```
ledcWrite(ledChannel, dutyCycle);
```

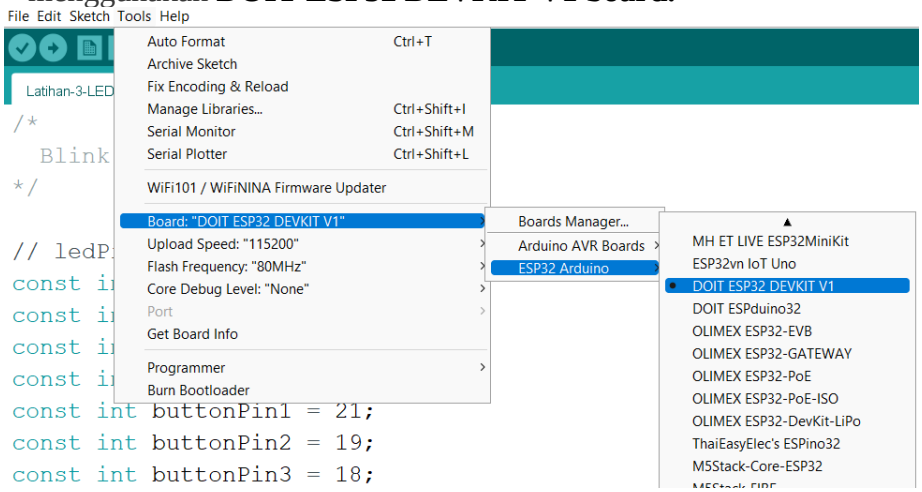
C. Uploading Program Perintah menggunakan Arduino IDE

- Siapkan kode perintah yang sudah disiapkan

```
/*Bismillahirrahmanirrahim
 * Cobi Mahesa dan Ade
 * ARM2 UHUYYYYYYYYYY
*/

//Introduce LED and Button PIN
const int ledPin1 = 27;
const int ledPin2 = 26;
const int ledPin3 = 25;
const int ledPin4 = 33;
const int buttonPin1 = 21;
const int buttonPin2 = 19;

//Setting PWM Properties
const int freq = 5000;
const int ledChannel = 0;
const int resolution = 8;

//Introduce Button State
int buttonState1 = 0;
int buttonState2 = 0;
```

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
  ledcSetup(ledChannel, freq, resolution);
  ledcAttachPin(ledPin1, ledChannel);
  ledcAttachPin(ledPin2, ledChannel);
  ledcAttachPin(ledPin3, ledChannel);
  ledcAttachPin(ledPin4, ledChannel);
}

void loop() {
  // put your main code here, to run repeatedly:
  //Rangkaian 1
  buttonState1 = digitalRead(buttonPin1);
  Serial.println(buttonState1);
  if (buttonState1 == LOW) {
    for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++){
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
    }
  }
  buttonState2 = digitalRead(buttonPin2);
  Serial.println(buttonState2);
  if (buttonState2 == LOW) {
    for(int dutyCycle = 255; dutyCycle >= 0; dutyCycle--){
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
    }
  }
}
```

- Pergi ke **Tools > Board** pilih board yang akan digunakan, kami menggunakan **DOIT ESP32 DEVKIT V1 board.**



Melakukan penyesuaian port, Buka **Tools > Port** dan pilih port **COM** yang terhubung dengan ESP32. Kemudian, tekan tombol unggah dan tunggu pesan "**Done Uploading**".