

# Pengenalan Arduino IDE, ESP8266 Flasher, Controlling LED

D3 Teknologi Telekomunikasi

Fakultas Ilmu Terapan

Telkom University

# Install Aplikasi

- Arduino IDE

## Downloads

- <https://www.arduino.cc/en/software>
- Download



### Arduino IDE 1.8.13

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

#### SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

#### DOWNLOAD OPTIONS

**Windows** Win 7 and newer

**Windows** ZIP file

**Windows app** Win 8.1 or 10



**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32 bits

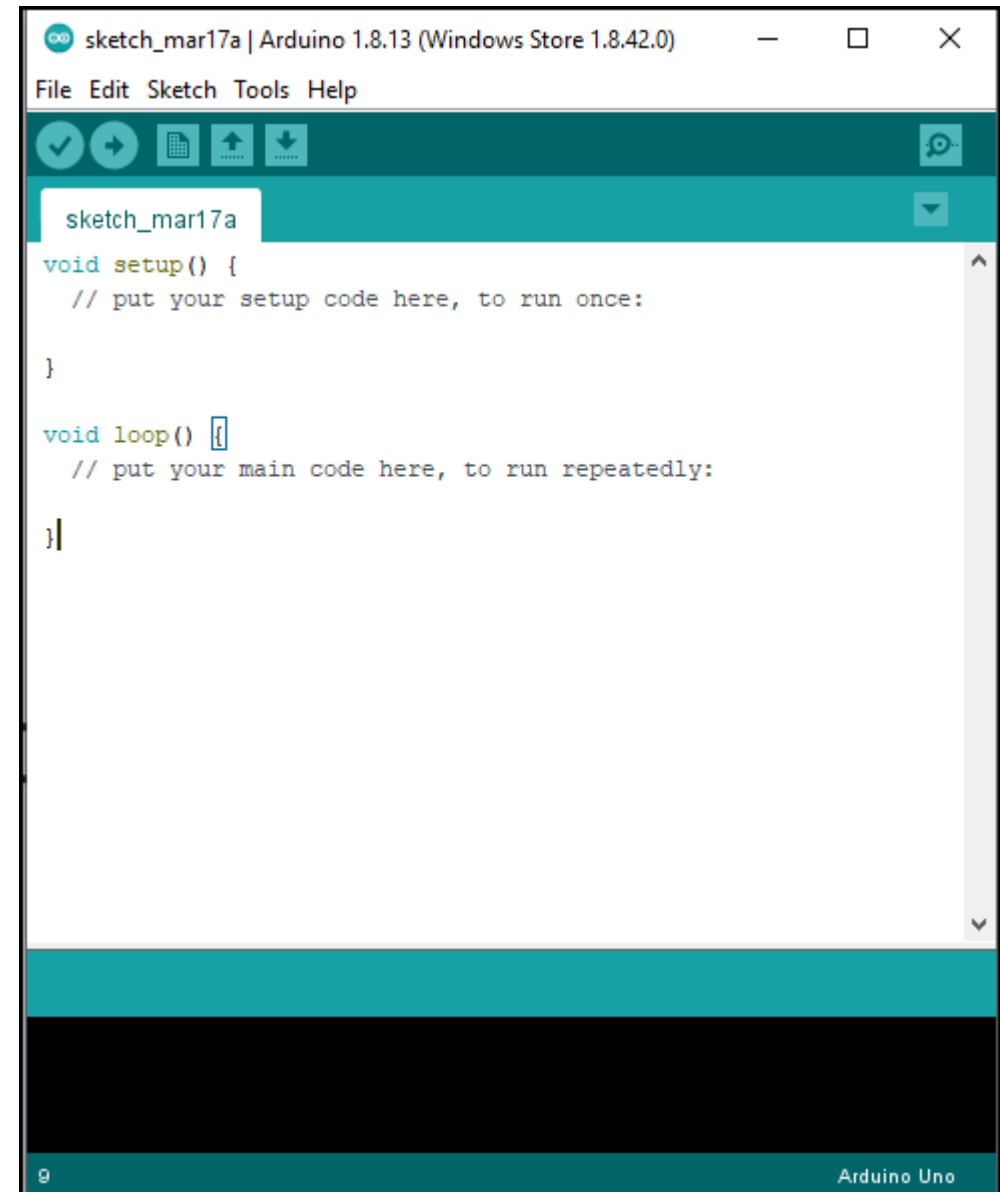
**Linux** ARM 64 bits

**Mac OS X** 10.10 or newer

[Release Notes](#) [Checksums \(sha512\)](#)

# Setting

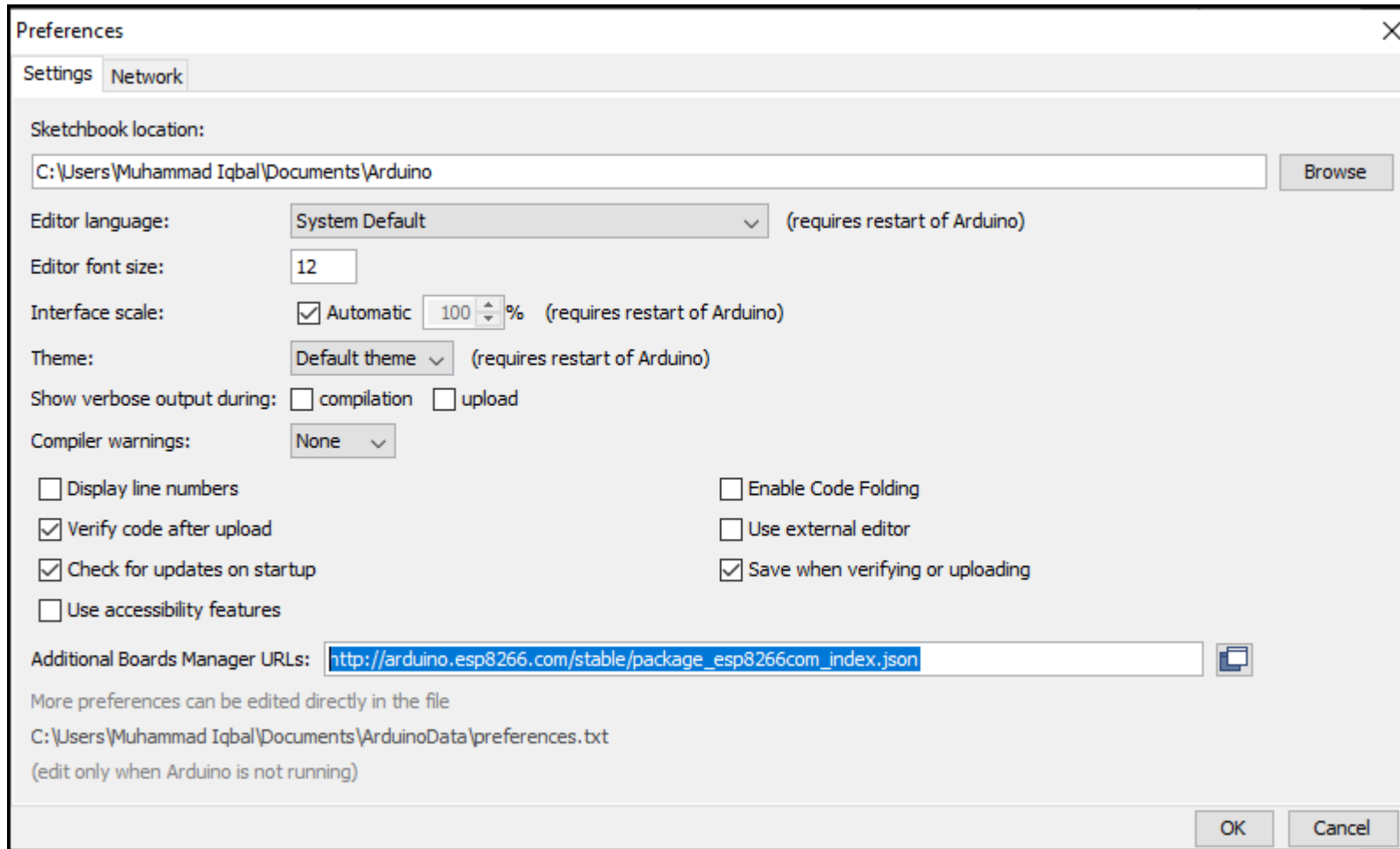
- Buka Aplikasi Arduino
- Pilih File → Preferences



# Additional Boards

- File → Preferences
- Tambahkan alamat url pada “Additional Boards Manager URLs” berikut :
  - `http://arduino.esp8266.com/stable/package\_esp8266com\_index.json`
- Link Sumber : <https://github.com/esp8266/Arduino>

# Preferences



Preferences

Settings Network

Sketchbook location:  
C:\Users\Muhammad Iqbal\Documents\Arduino Browse

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale:  Automatic 100% (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during:  compilation  upload

Compiler warnings: None

Display line numbers  Enable Code Folding

Verify code after upload  Use external editor

Check for updates on startup  Save when verifying or uploading

Use accessibility features

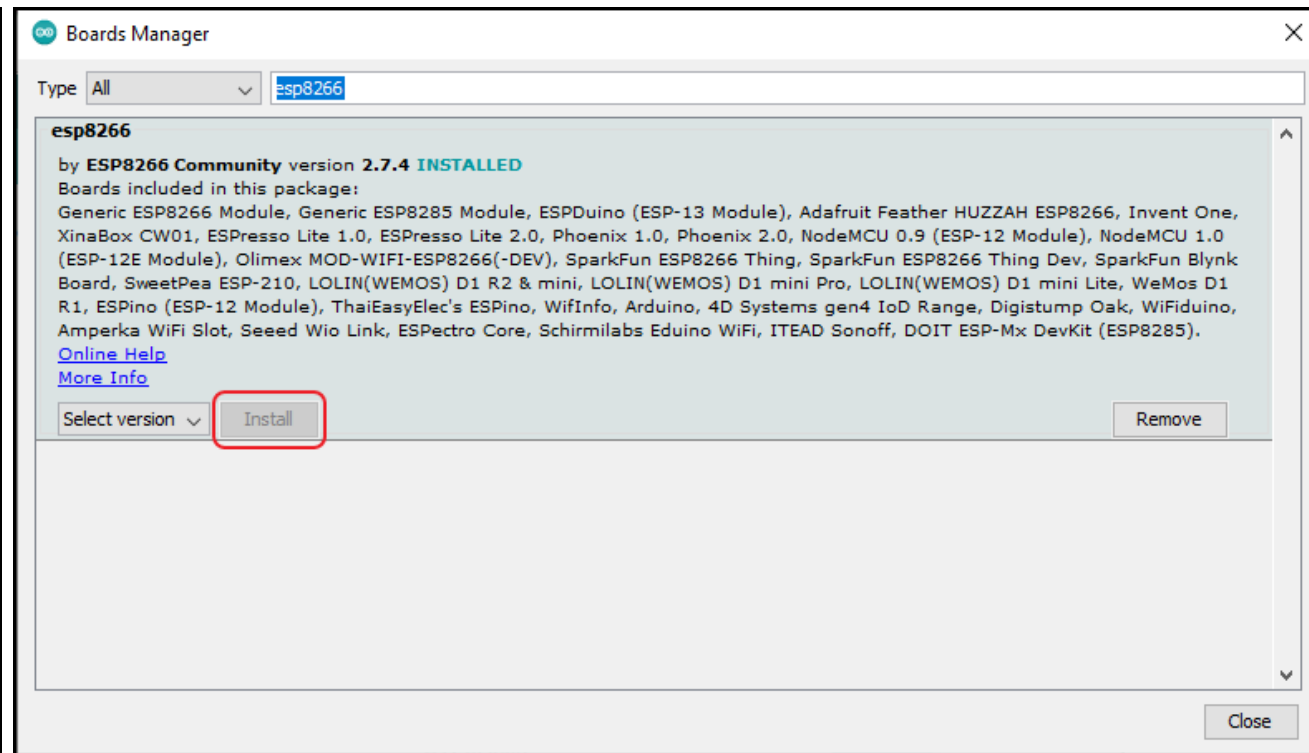
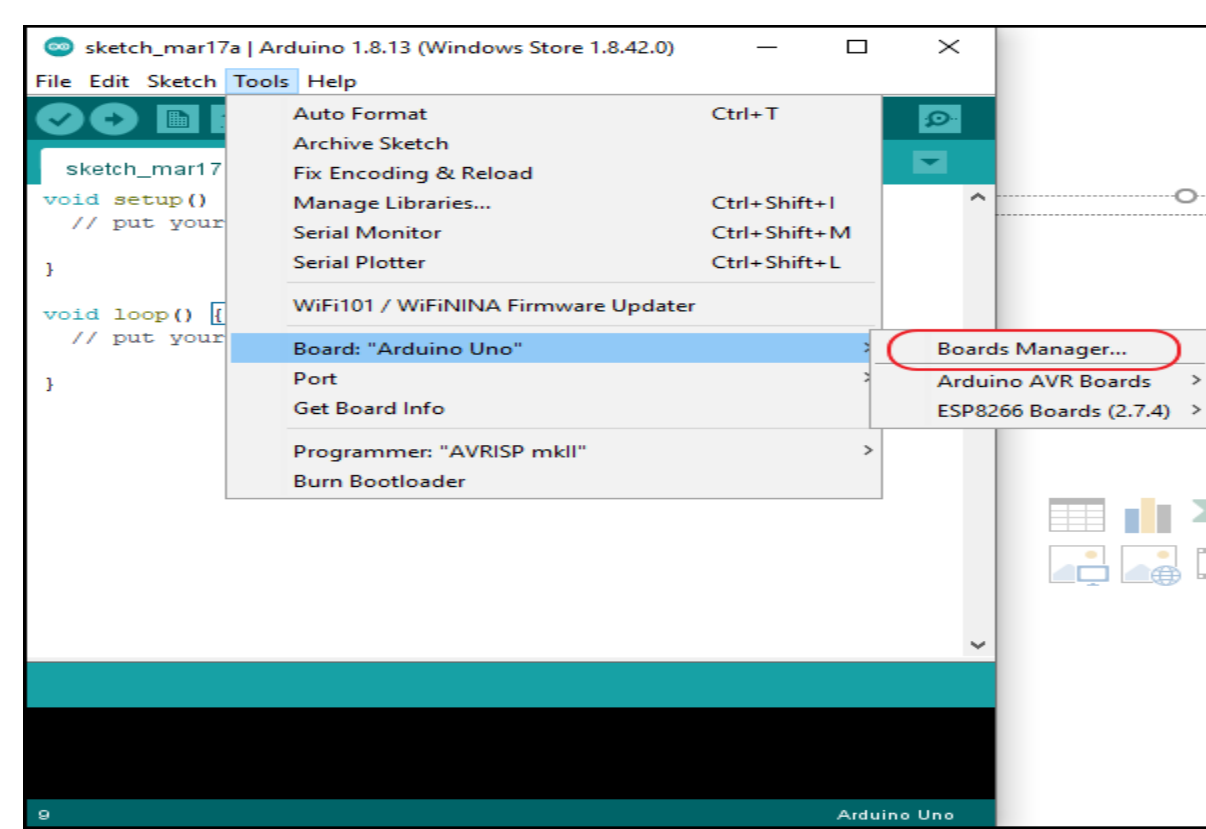
Additional Boards Manager URLs: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

More preferences can be edited directly in the file  
C:\Users\Muhammad Iqbal\Documents\ArduinoData\preferences.txt  
(edit only when Arduino is not running)

OK Cancel

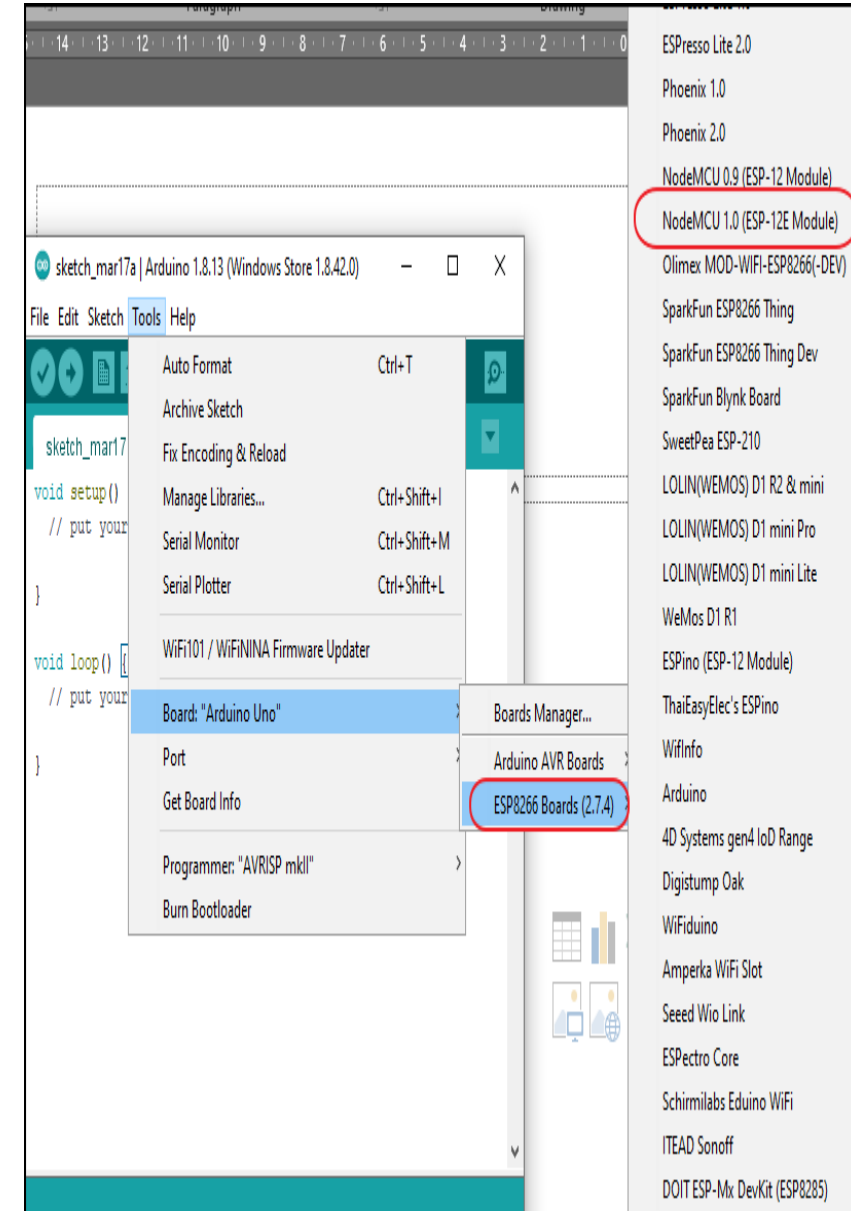
# Board Manager

- Pilih Tools → Board : “Arduino Uno” → Boards Manager...
- Pada Menu Search ketikkan : esp8266 dan pilih install



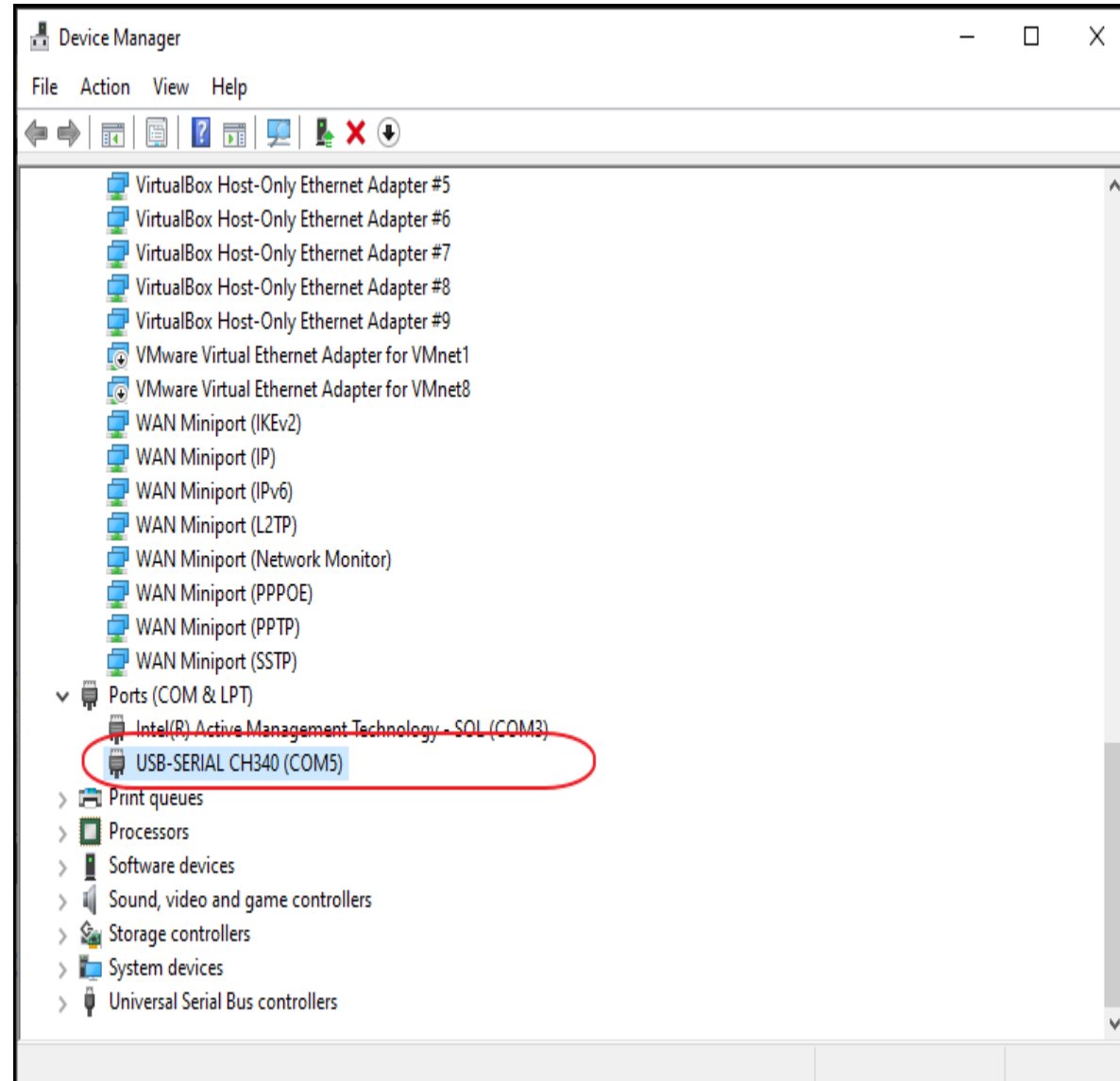
# ESP8266 Boards

- Indikasi instalasi telah berhasil dilakukan adalah
- Dengan adanya keterangan Board “Arduino Uno” : ESP8266 Boards (3.xx)  
Pilih NodeMCU 1.0 (ESP-12E Module)



# Cek Port

- Colokkan nodeMCU dengan kabel data
- Run → Device Manager
- Ports (COM & LPT)
  - USB-SERIAL CH340 → (COM~~X~~)
- Catatan : USB Serial akan berubah sesuai dengan ketersediaan port



# First Program (LED Blinking)

# LED Blinking

- Arduino IDE menyediakan banyak contoh/template penggunaan dari berbagai board yang ada.
- Blink merupakan salah satu program bawaan, yang mampu mati dan nyalakan LED sesuai dengan kebutuhan pengguna.
- Cara :
  - File → Examples → 01.Basics → Blink

# Program Blink

*// the setup function runs once when you press reset or power the b*

```
void setup() {
```

```
    // initialize digital pin LED_BUILTIN as an output.
```

```
    pinMode(LED_BUILTIN, OUTPUT);
```

```
}
```

*// the loop function runs over and over again forever*

```
void loop() {
```

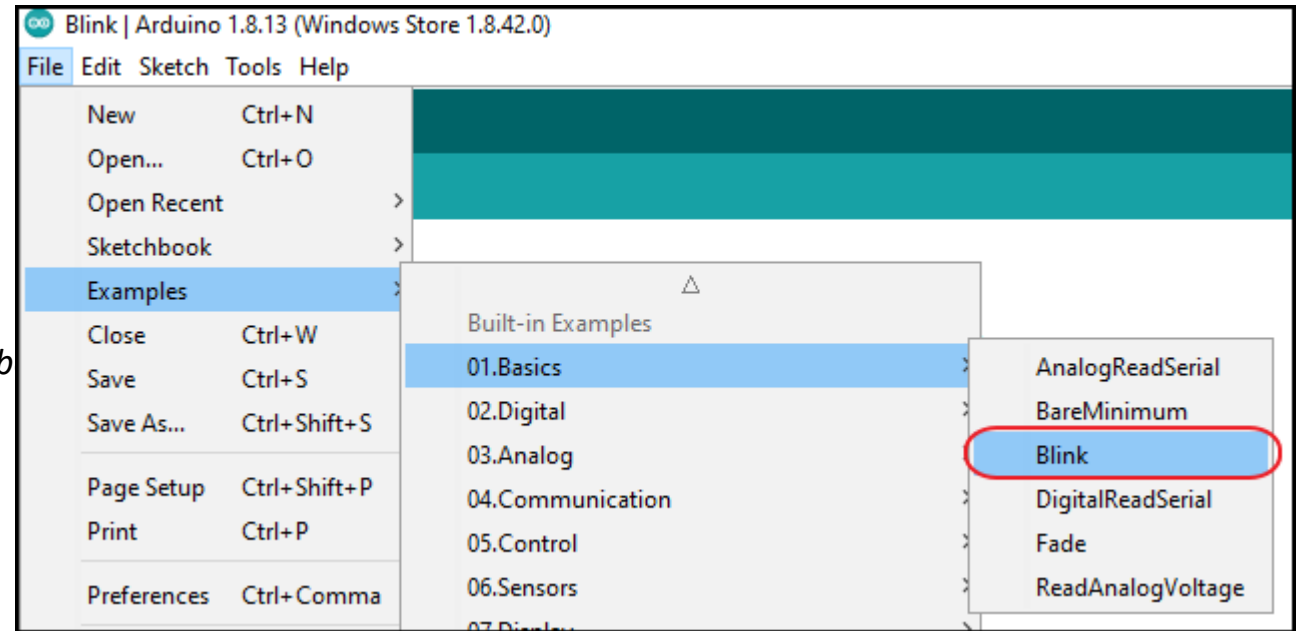
```
    digitalWrite(LED_BUILTIN, HIGH); //turn the LED on (HIGH is the voltage level)
```

```
    delay(1000);           // wait for a second
```

```
    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
```

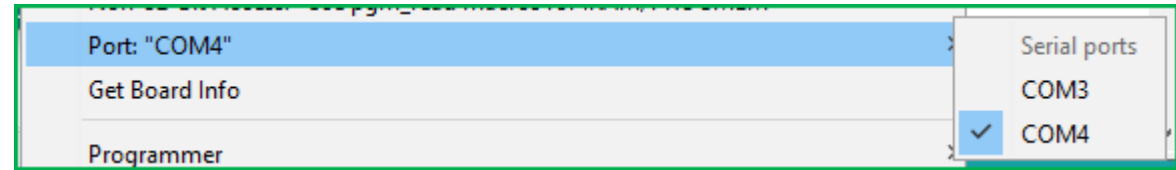
```
    delay(1000);           // wait for a second
```

```
}
```



# Compile Program

- Pastikan COM~~X~~ sesuai dengan device manager
  - Tools → Port : “COM~~X~~”
- Verify (optional) → tanda checklist
- Upload :
  - Control + U
  - Arrow Kanan



 A screenshot of the Arduino IDE interface showing the 'Blink' sketch. The 'Verify' button (a right-pointing arrow) is highlighted with a red box. The code editor displays the following code:
 

```

modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
  
```

# Keterangan Terminal

- Status Berhasil Upload

```
Done uploading.
```

```
Executable segment sizes:
```

```
IROM   : 228624      - code in flash          (default or ICACHE_FLASH_ATTR)
```

```
IRAM   : 26756 / 32768 - code in IRAM          (ICACHE_RAM_ATTR, ISRs...)
```

```
DATA   : 1248 )      - initialized variables (global, static) in RAM/HEAP
```

```
RODATA : 688 ) / 81920 - constants             (global, static) in RAM/HEAP
```

```
BSS    : 24880 )     - zeroed variables      (global, static) in RAM/HEAP
```

```
Sketch uses 257316 bytes (24%) of program storage space. Maximum is 1044464 bytes.
```

```
Global variables use 26816 bytes (32%) of dynamic memory, leaving 55104 bytes for local variables. Maximum is 81920 bytes.
```

```
esptool.py v2.8
```

```
Serial port COM5
```

```
Connecting...
```

```
Chip is ESP8266EX
```

```
Features: WiFi
```

```
Crystal is 26MHz
```

```
MAC: 8c:aa:b5:7c:da:4d
```

```
Uploading stub...
```

```
Running stub...
```

```
Stub running...
```

```
Configuring flash size...
```

```
Auto-detected Flash size: 4MB
```

```
Compressed 261472 bytes to 193134...
```

```
Wrote 261472 bytes (193134 compressed) at 0x00000000 in 17.0 seconds (effective 122.9 kbit/s)...
```

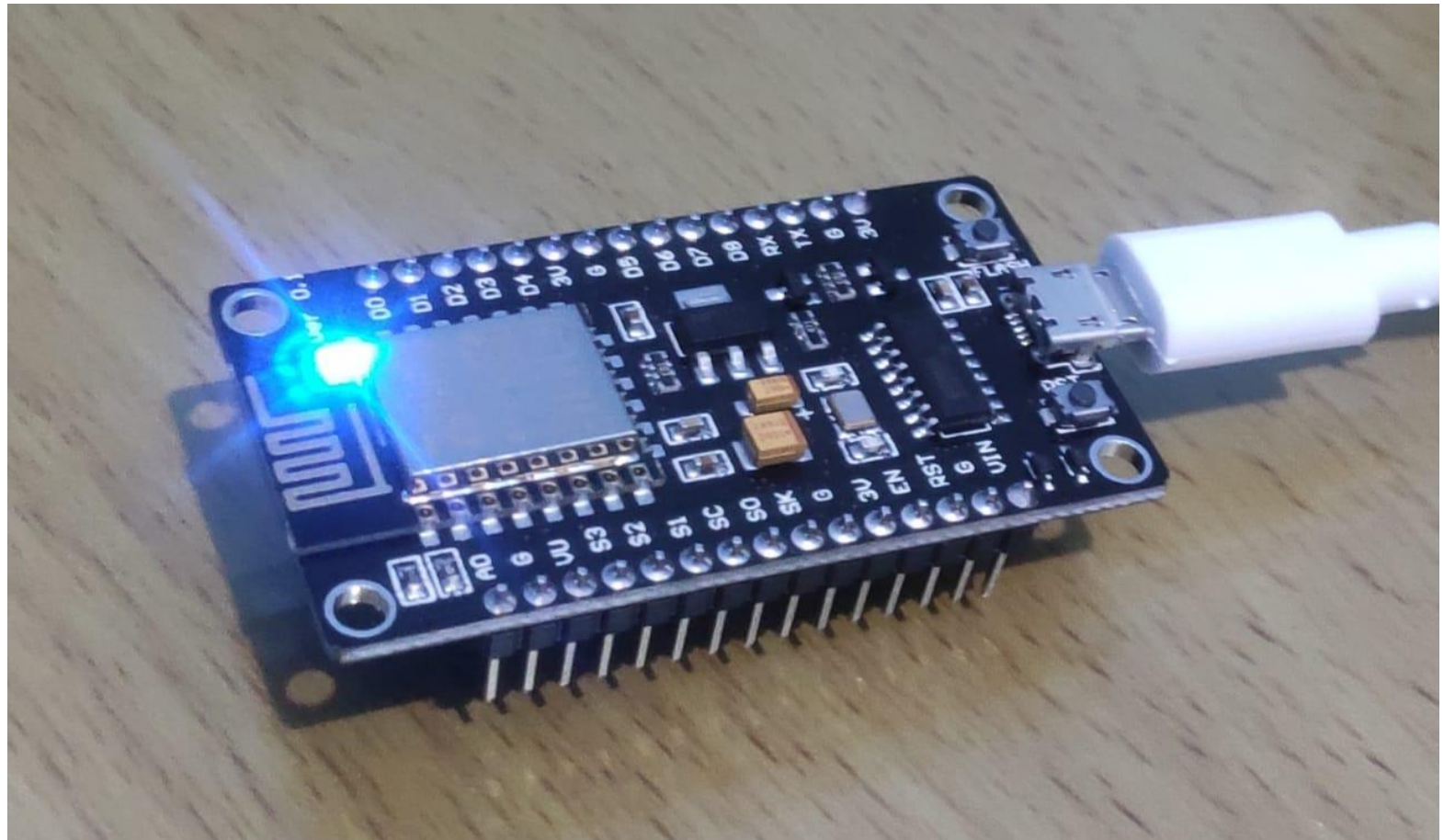
```
Hash of data verified.
```

```
Leaving...
```

```
Hard resetting via RTS pin...
```

# Output

- Indikator keberhasilan LED bawaan akan mengalami blinking secara terus menerus

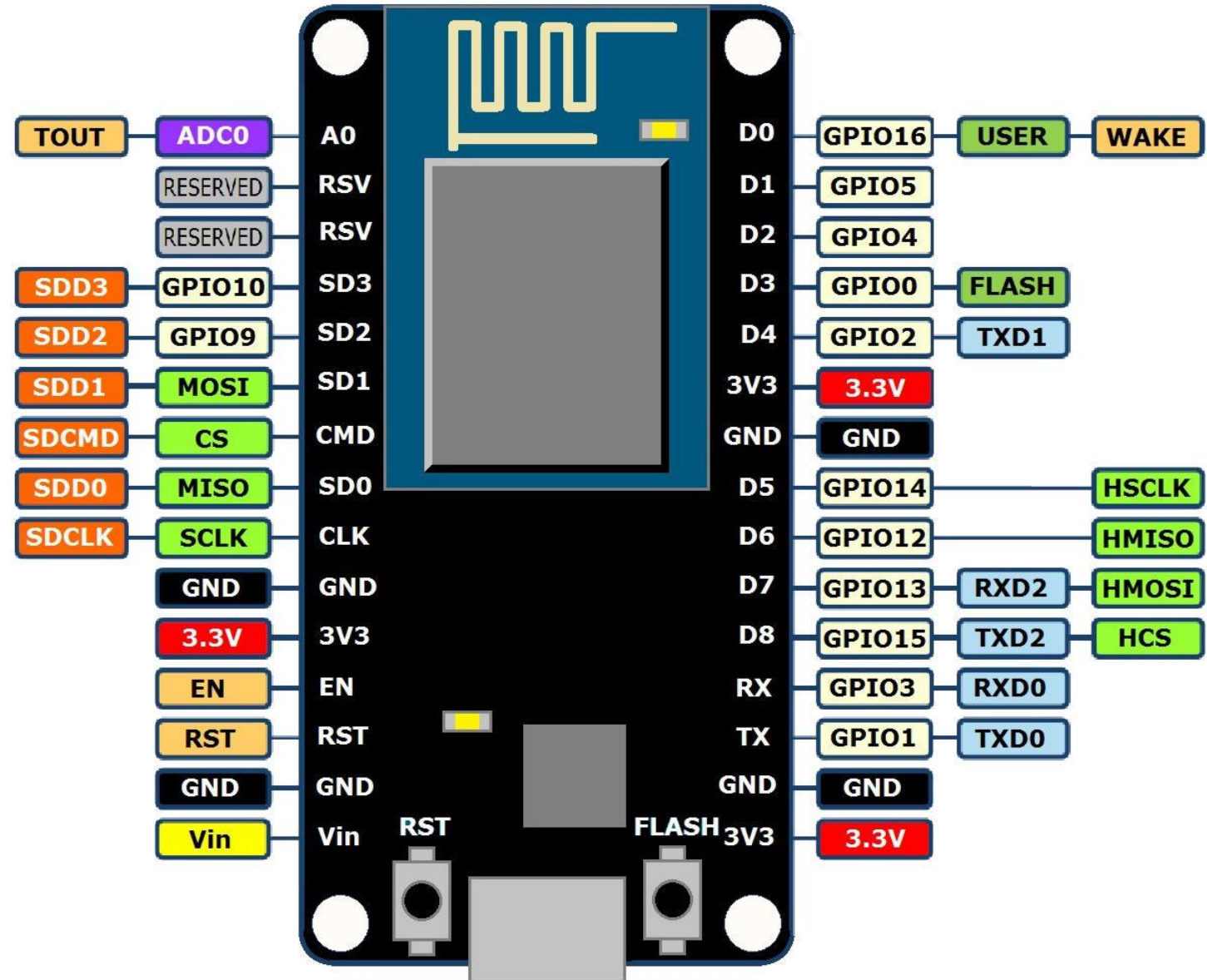


# Studi Kasus

- Buat blink (led) nyala lebih cepat dari defaultnya
- Buat blink (led) nyala lebih lambat
  
- Berapa nilai maksimum dan minimum blinking
- Amati berapa lama waktu yang terjadi dari proses upload sampai dengan selesai
  
- Hint : delay

# Blinking LED

# Board NodeMCU



# Persiapan LED dan NodeMCU

- Alat :
  - NodeMCU
  - Led (1 bh) → Kaki Pendek (Negatif), Kaki Panjang (positif)
  - Kabel jumper female (2 bh)
- Langkah Pemasangan kabel jumper
  - D2 → Kaki Panjang (Positif)
  - G → Kaki Pendek (Negatif)

Ket : D2 dan G (Ground) → lihat Board NodeMCU

# LED Control

```
int LedPin = 2;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LedPin, OUTPUT);
  Serial.begin(9600);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LedPin, LOW); // turn the LED off by making the voltage LOW
  Serial.println("Led Off");
  delay(1000);              // wait for a second
  digitalWrite(LedPin, HIGH); // turn the LED on (HIGH is the voltage level)
  Serial.println("Led On");
  delay(1000);              // wait for a second
}
```

NodeMCU\_dgn\_LED

```
#include <Blynk.h>

/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

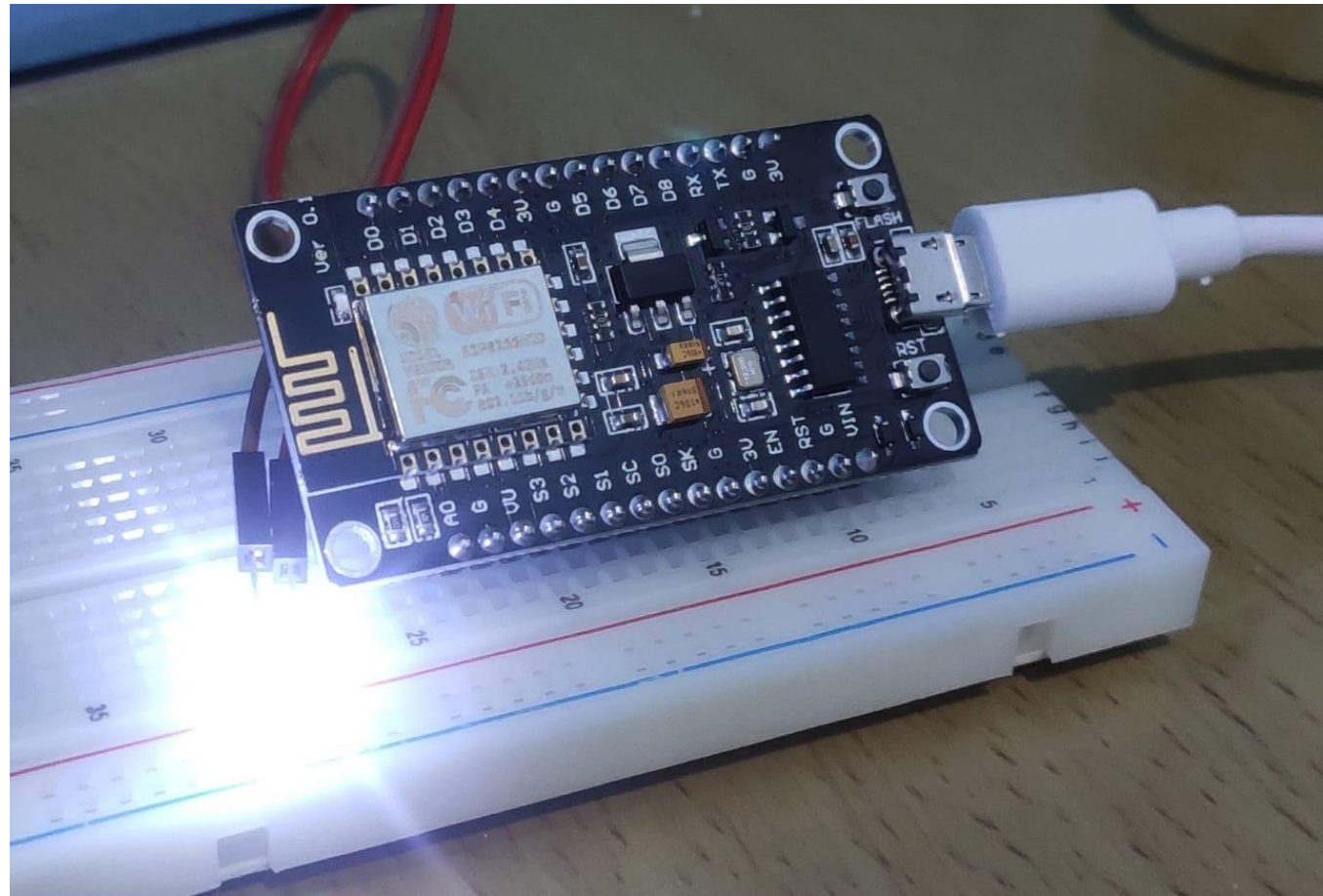
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/
int LedPin = 4;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LedPin, OUTPUT);
  Serial.begin(9600);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LedPin, LOW); // turn the LED off by making the voltage LOW
  Serial.println("Led Off");
  delay(1000);              // wait for a second
  digitalWrite(LedPin, HIGH); // turn the LED on (HIGH is the voltage level)
  Serial.println("Led On");
  delay(1000);              // wait for a second
}
```

# Hasil

- Indikator LED yang terpasang akan mengalami blinking

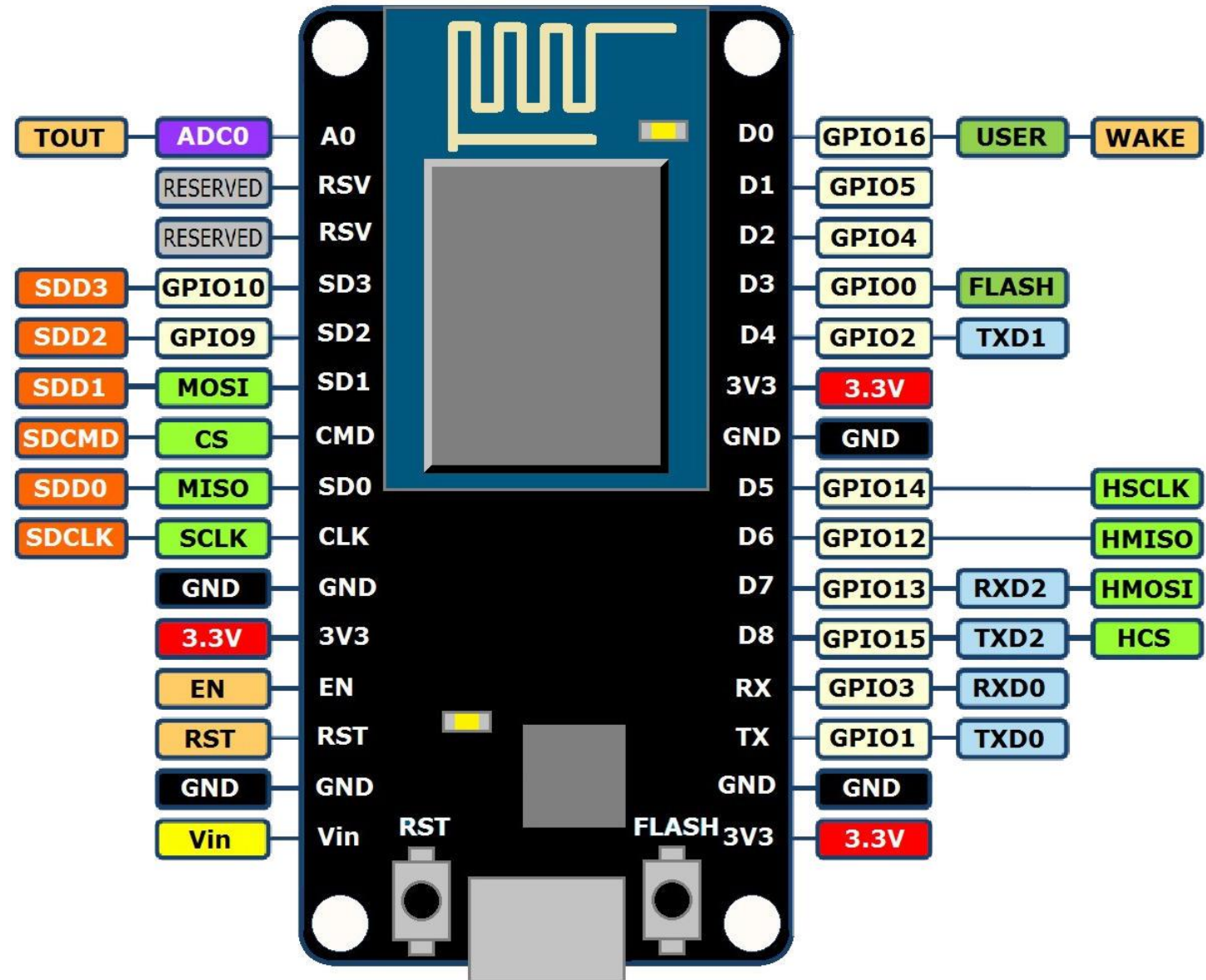


# Studi Kasus

- Tambahkan jenis warna LED lainnya (2 warna lainnya)
  - Perhatikan LedPin
- Skenario
  1. Jalankan ketiga warna LED secara bersamaan
  2. Jalankan ketiga warna LED secara bergantian

# Aplikasi Blynk

# Board NodeMCU



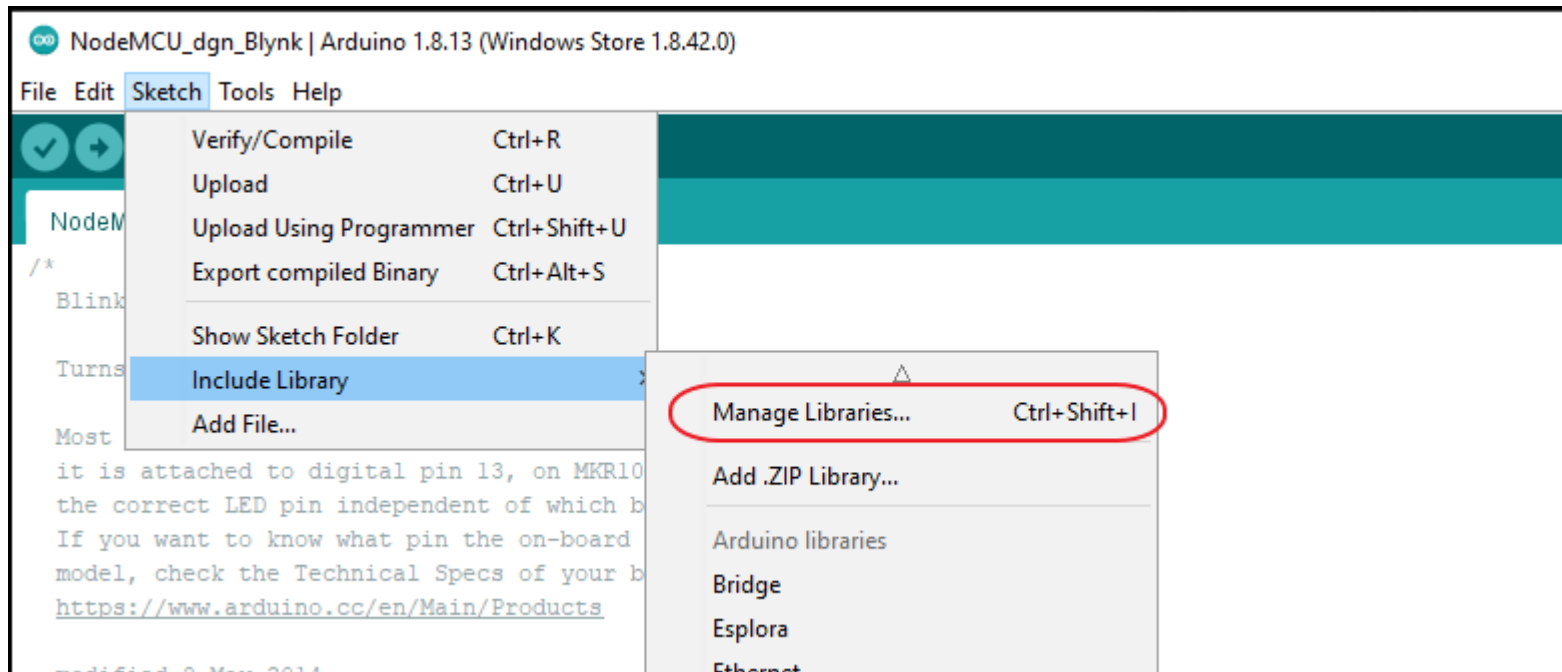
# Persiapan LED dan NodeMCU

- Alat :
  - NodeMCU
  - Led (1 bh) → Kaki Pendek (Negatif), Kaki Panjang (positif)
  - Kabel jumper female (2 bh)
- Langkah Pemasangan kabel jumper
  - D2 → Kaki Panjang (Positif)
  - G → Kaki Pendek (Negatif)

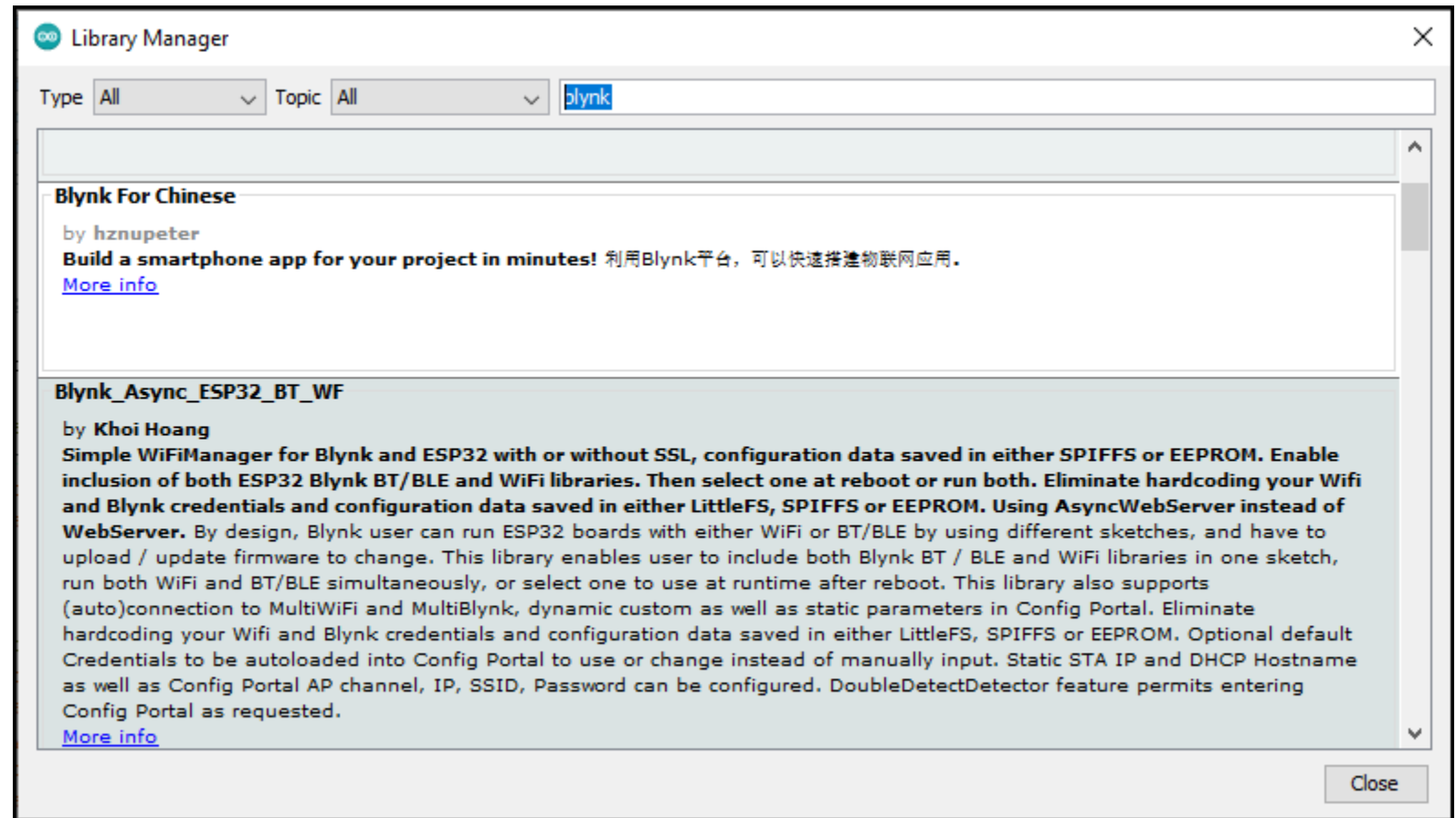
Ket : D2 dan G (Ground) → lihat Board NodeMCU

# Library Blynk (1)

- Install library Blynk pada Arduino IDE

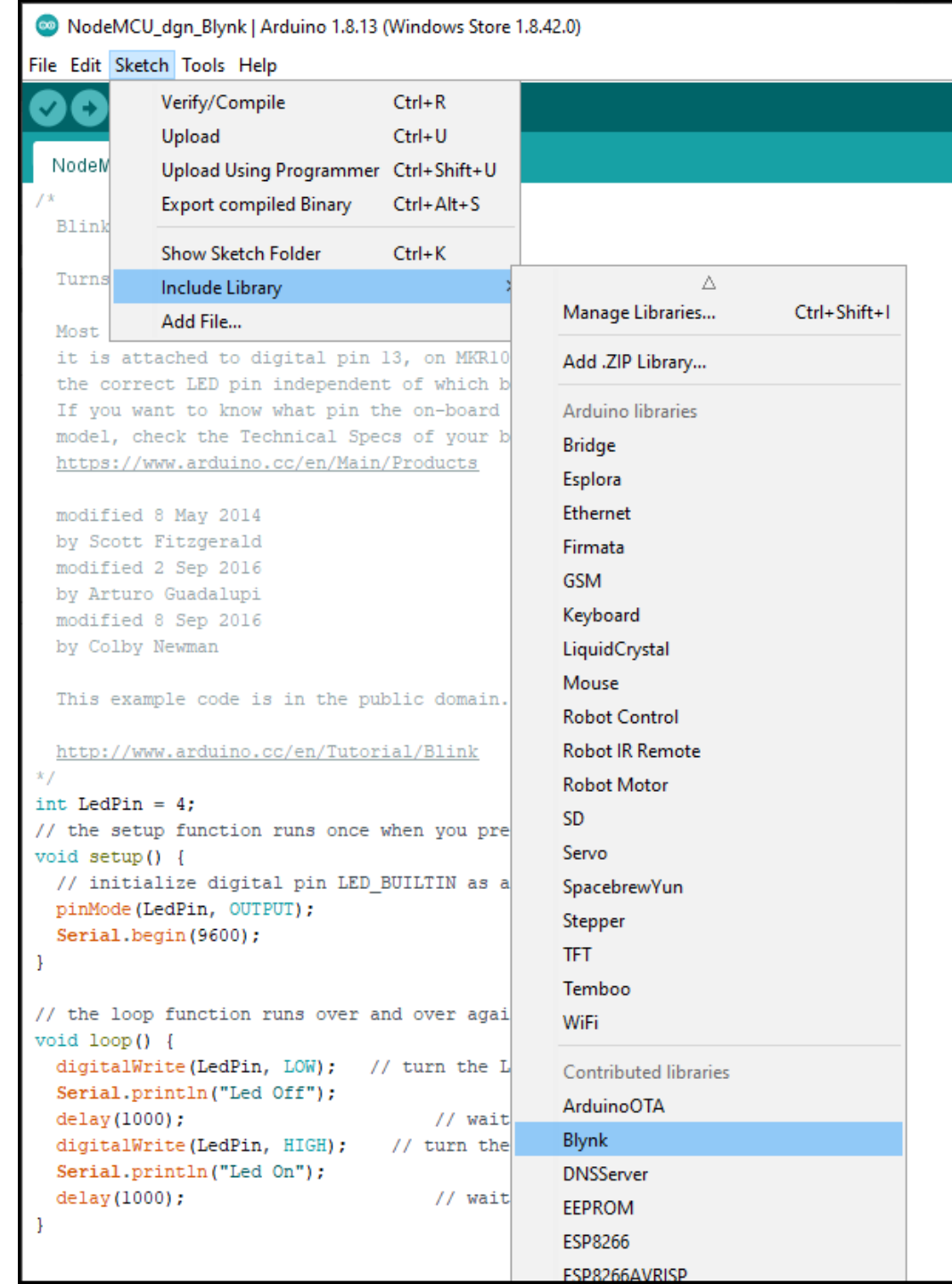


# Library Blynk (2)



# Library Blynk (2)

- Pastikan Library yang digunakan sudah tersedia



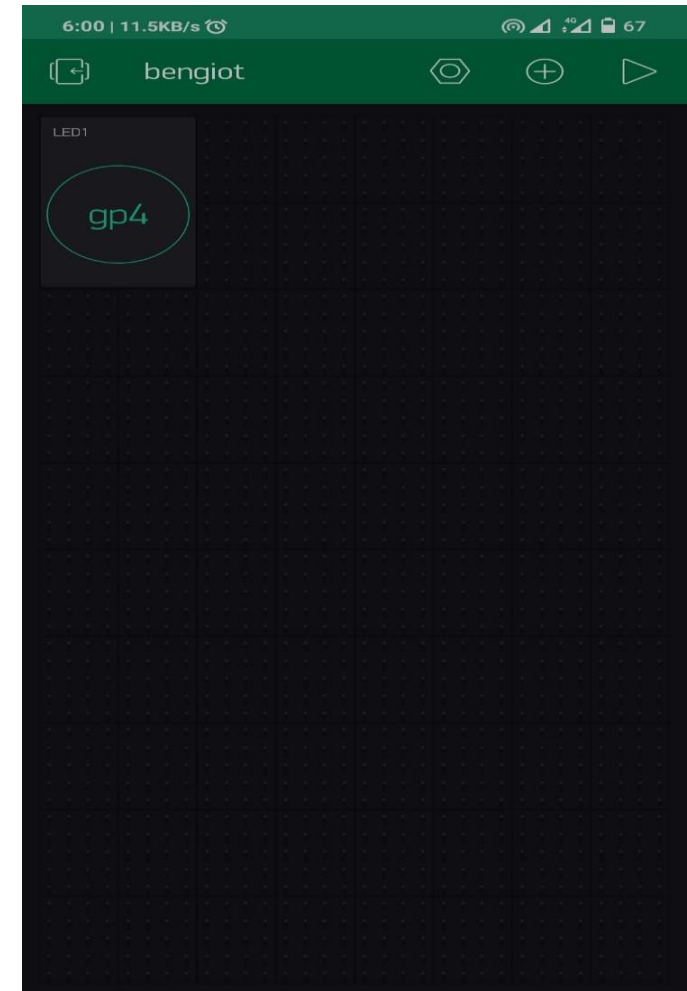
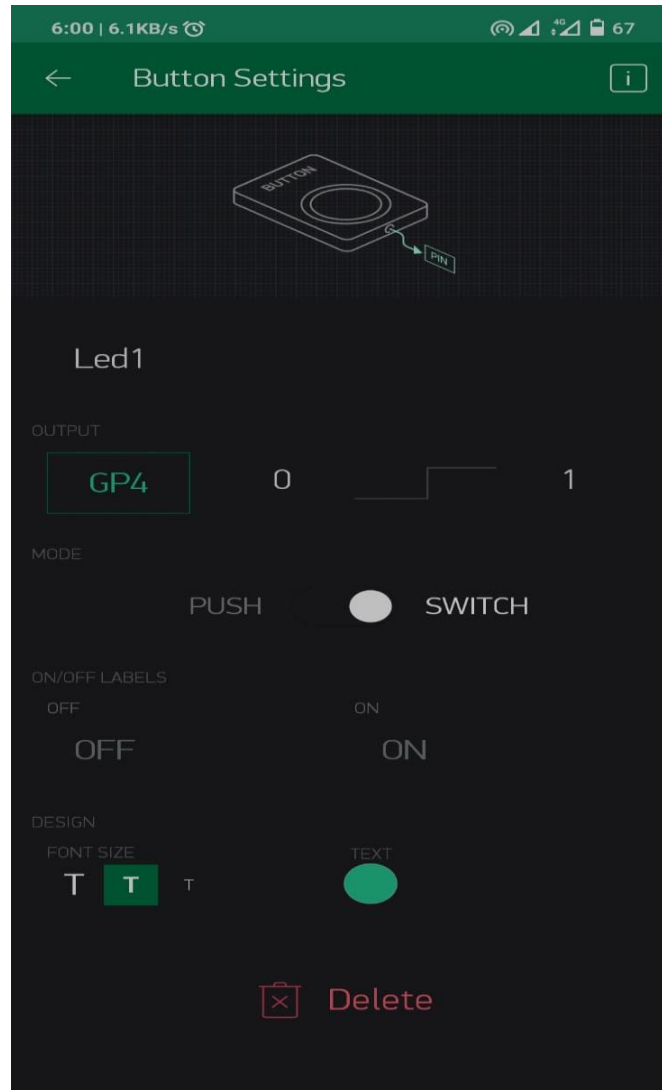
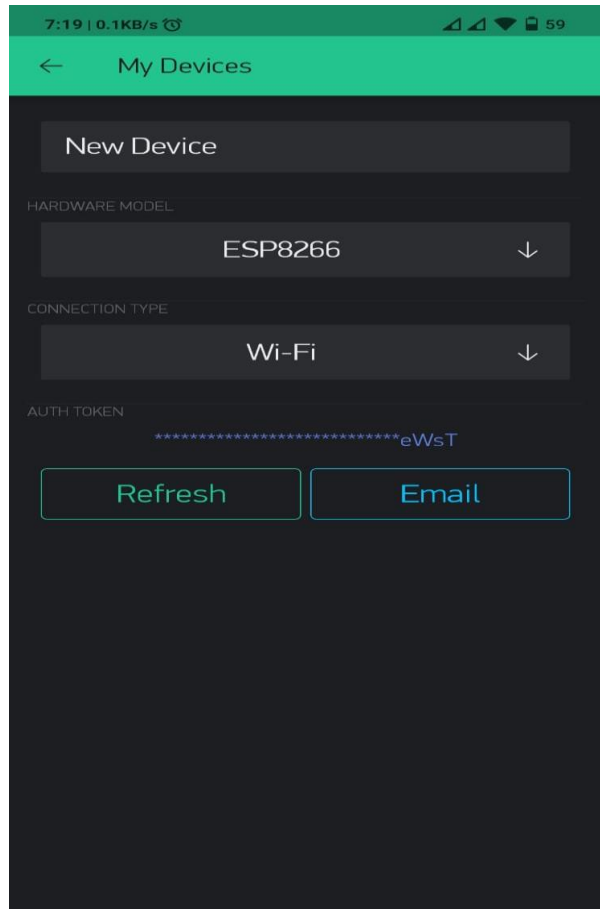
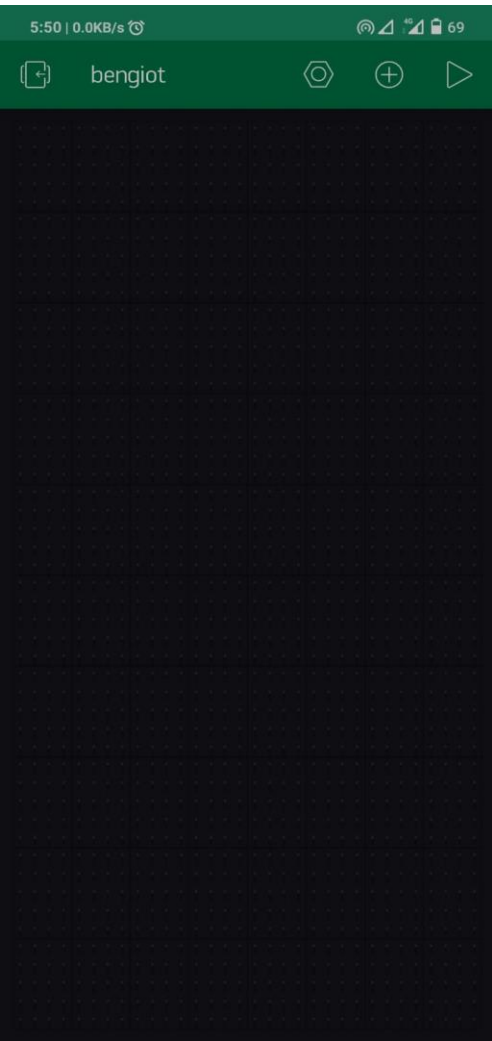
# Download Aplikasi

- Download PlayStore atau IOS
- Blynk – IoT for Arduino ESP8266
- Open dan Registrasi akun



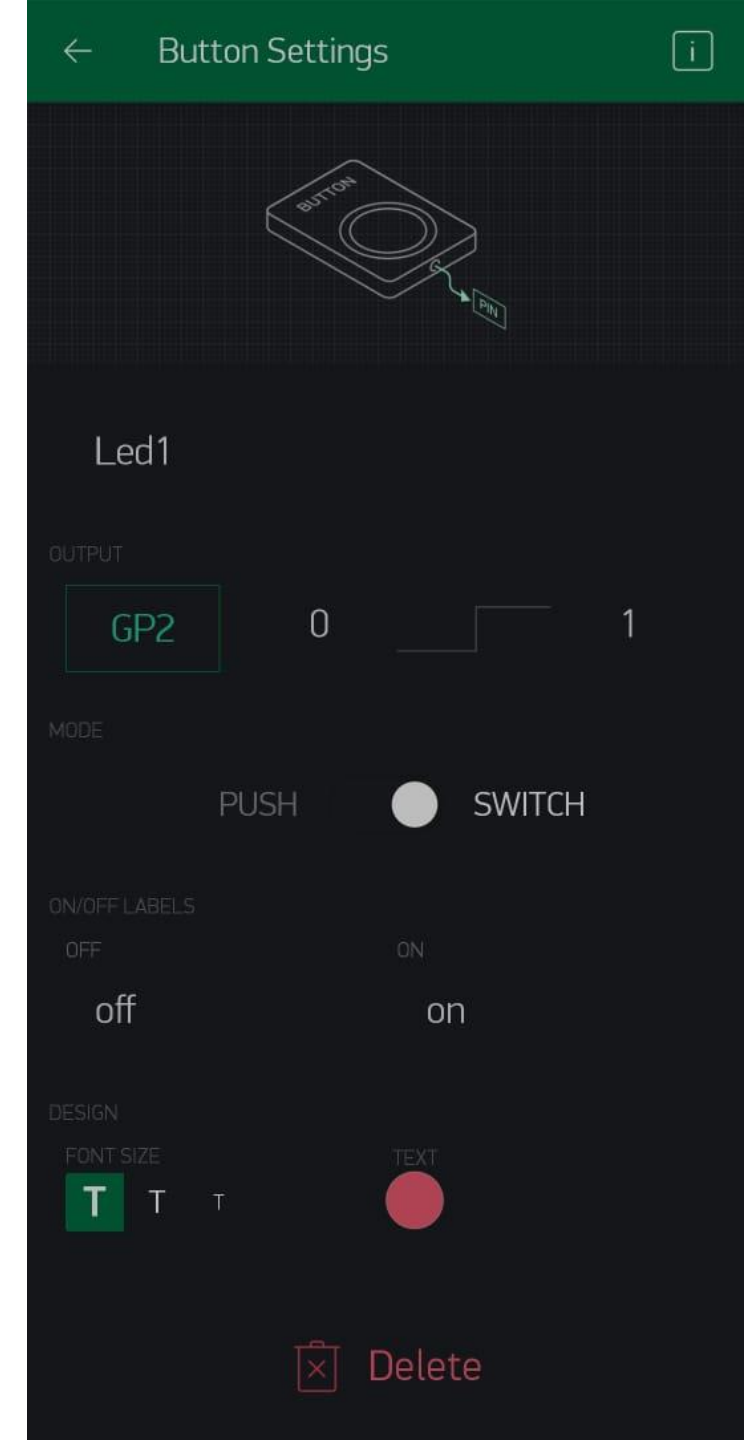
# Langkah pada Blynk

- Create New Project : bengiot
  - Hardware Model : ESP8266/NodeMCU
  - Connection Type : Wi-Fi
  - Auth Token : xxxxxxxxxxxxxxxxxxxxxxx
  - Cek Email untuk autentikasi token
- 
- Catatan : Ada beberapa layanan ISP yang tidak support, gunakan tethering HP dari operator telekomunikasi jika tidak berhasil menggunakan ISP



# Widget Box

- Pilih Button
  - Name → Led1 (bebas)
  - Output → GP4
  - Mode → Switch
  - ON/OFF LABELS → off dan on



# Cek Email

- Auth Token : xxxxxxxxxxxxxxxxxxxx
- Token yang didapatkan di copy pada ESP8266\_Standalone

Auth Token for First Project project and device First Project ▶ Inbox x

**Blynk** <dispatcher@blynk.io> [Unsubscribe](#)

9:13 AM (11 minutes ago) ★

to me ▼

Auth Token : gH3NvtkwLVjLo42wN-vY4Y0XSPxTnbGS

Happy Blynking!

-

Getting Started Guide -> <https://www.blynk.cc/getting-started>

Documentation -> <http://docs.blynk.cc/>

Sketch generator -> <https://examples.blynk.cc/>

# Format Blynk

NodeMCU\_dgn\_Blynk | Arduino 1.8.13 (Windows Store 1.8.42.0)

File Edit Sketch Tools Help

- New Ctrl+N
- Open... Ctrl+O
- Open Recent >
- Sketchbook >
- Examples >
- Close Ctrl+W
- Save Ctrl+S
- Save As... Ctrl+Shift+S
- Page Setup Ctrl+Shift+P
- Print Ctrl+P
- Preferences Ctrl+Comma
- Quit Ctrl+Q

by Scott Fitzgerald modified 2 Sep 2016  
 by Arturo Guadalupi modified 8 Sep 2016  
 by Colby Newman

This example code is in <http://www.arduino.cc/en>

```

*/
int LedPin = 4;
// the setup function runs
void setup() {
  // initialize digital pin
  pinMode(LedPin, OUTPUT);
  Serial.begin(9600);
}

// the loop function runs
void loop() {
  digitalWrite(LedPin, LOW);
  Serial.println("Led Off");
  delay(1000);
  digitalWrite(LedPin, HIGH);
  Serial.println("Led On");
  delay(1000);
}
  
```

- LiquidCrystal >
- SD >
- Stepper >
- Temboo >
- RETIRED >
- Examples for NodeMCU 1.0 (ESP-12E Module)
- ArduinoOTA >
- DNSServer >
- EEPROM >
- ESP8266 >
- ESP8266AVRISP >
- ESP8266HTTPClient >
- ESP8266httpUpdate >
- ESP8266HTTPUpdateServer >
- ESP8266LLMNR >
- ESP8266mDNS >
- ESP8266NetBIOS >
- ESP8266SdFat >
- ESP8266SSDP >
- ESP8266WebServer >
- ESP8266WiFi >
- ESP8266WiFiMesh >
- EspSoftwareSerial >
- Ethernet(esp8266) >
- Hash >
- LittleFS(esp8266) >
- SD(esp8266) >
- Servo(esp8266) >
- SPISlave >
- TFT\_Touch\_Shield\_V2 >
- Ticker >
- Wire >
- Examples from Custom Libraries
- Blynk >

- Blynk.Inject >
- Boards\_Bluetooth >
- Boards\_Ethernet >
- Boards\_GSM >
- Boards\_USB\_Serial >
- Boards\_WiFi >
- Boards\_With\_HTTP\_API >
- GettingStarted >
- More >
- Widgets >
- tests >

- Adafruit\_Feather\_M0\_WiFi
- Arduino\_MKR1000
- Arduino\_MKR1010
- Arduino\_org\_UNO\_WiFi
- Arduino\_WiFi\_Shield
- Arduino\_WiFi\_Shield\_101
- Arduino\_Yun
- CC3000
- Digistump\_Oak
- Energia\_WiFi
- ESP32\_WiFi
- ESP32\_WiFi\_SSL
- ESP8266\_Shield
- ESP8266\_Standalone
- ESP8266\_Standalone\_Manual\_IP
- ESP8266\_Standalone\_SmartConfig
- ESP8266\_Standalone\_SSL
- Fishino
- Intel\_Edison\_WiFi
- LinkitONE
- NodeMCU
- RedBear\_Duo\_WiFi
- RN\_XV\_WiFly
- Sparkfun\_Blynk\_Board
- TheAirBoard\_WiFly
- TinyDuino\_WiFi
- WildFire\_V3
- WildFire\_V4
- WizFi310

# Setting Wi-Fi dan Token

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = "E70x6reH_O3_eLkSMsBDFCeqPv8zqe3o";
char ssid[] = "Baymax";
char pass[] = "12345678";
void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
}
void loop()
{
  Blynk.run();
}
```

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "E70x6reH_O3_eLkSMsBDFCeqPv8zqe3o";

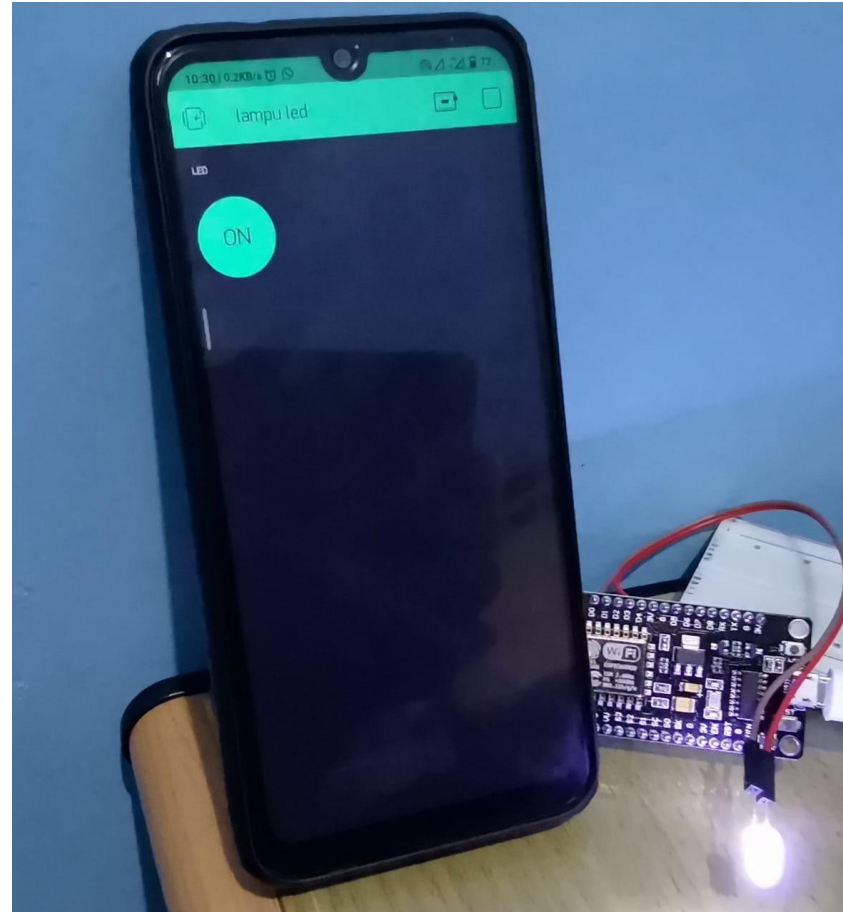
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Baymax";
char pass[] = "12345678";
```



# Hasil (1)



# Hasil (2)



# Studi Kasus

- Gp4 diganti dengan GP2, apa yang terjadi
- Buat rangkaian 3 LED, dan control dengan menggunakan aplikasi Blynk

**Thank You**