



*Train4best*

# PEREKAYASA PLATFORM IOT

CHAPTER #4: PROGRAM VISUAL ANTARMUKA PADA PERANGKAT BERBASIS WEB DENGAN MIKROKONTROLER

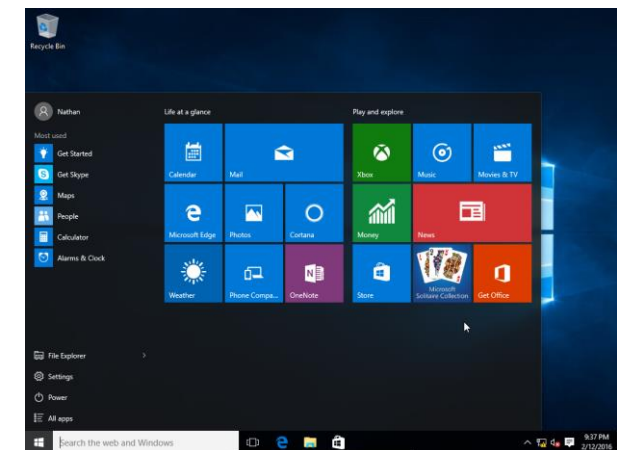
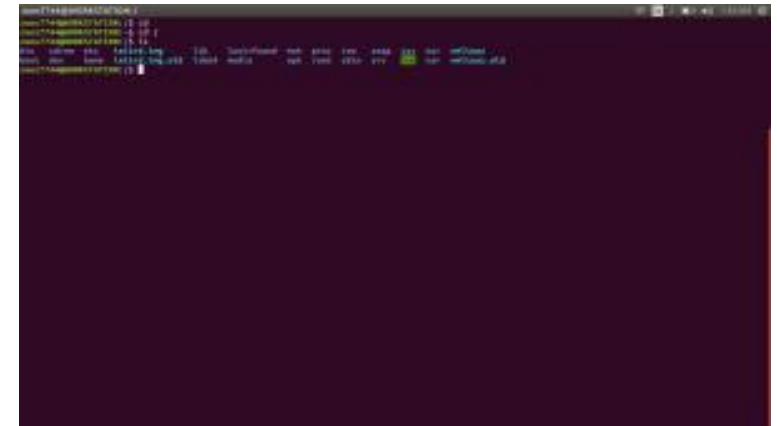
UI

# UI

- User Interface (UI) adalah tampilan visual sebuah produk yang menghubungkan sistem dengan pengguna (user).
- Sistem ini bisa berupa website, aplikasi atau lainnya.
- User interface atau UI adalah tampilan yang meliputi bentuk, warna, dan tulisan yang didesain semenarik mungkin.

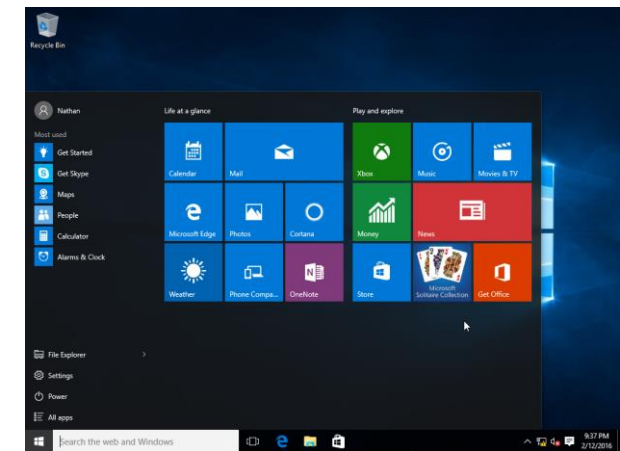
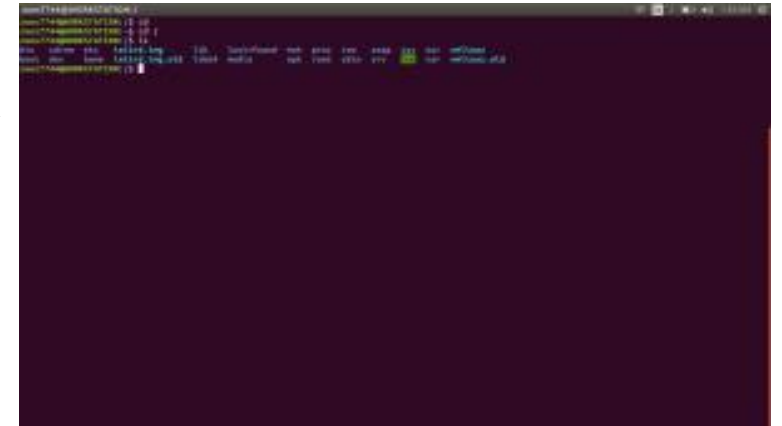
# APA PERBEDAAN ANTARA GUI DAN CUI?

- GUI adalah singkatan dari Graphical User Interface.
- Ini adalah jenis antarmuka pengguna di mana pengguna berinteraksi dengan komputer menggunakan grafik.
- Grafik termasuk ikon, bilah navigasi, gambar, dll.
- Mouse dapat digunakan saat menggunakan antarmuka ini untuk berinteraksi dengan grafik.
- Ini adalah antarmuka yang sangat ramah pengguna dan tidak memerlukan keahlian. Contoh: Windows memiliki GUI.



# APA PERBEDAAN ANTARA GUI DAN CUI?

- CUI: CUI adalah singkatan dari Character User Interface.
- Ini adalah antarmuka pengguna di mana pengguna berinteraksi dengan komputer hanya dengan menggunakan keyboard.
- CUI adalah pendahulu GUI dan digunakan di sebagian besar komputer primitif.



THINGSPEAK

# THINGSPEAK

- ThingSpeak memungkinkan kita untuk mempublikasikan pembacaan sensor kita ke situs web mereka dan memplotnya dalam grafik dengan stempel waktu.
- Kemudian, kita dapat mengakses pembacaan kita dari mana saja di seluruh dunia.
- Ada banyak cara untuk mengirim pembacaan sensor ke ThingSpeak.
  - API HTTP
  - MQTT
  - Matlab
  - Simulink

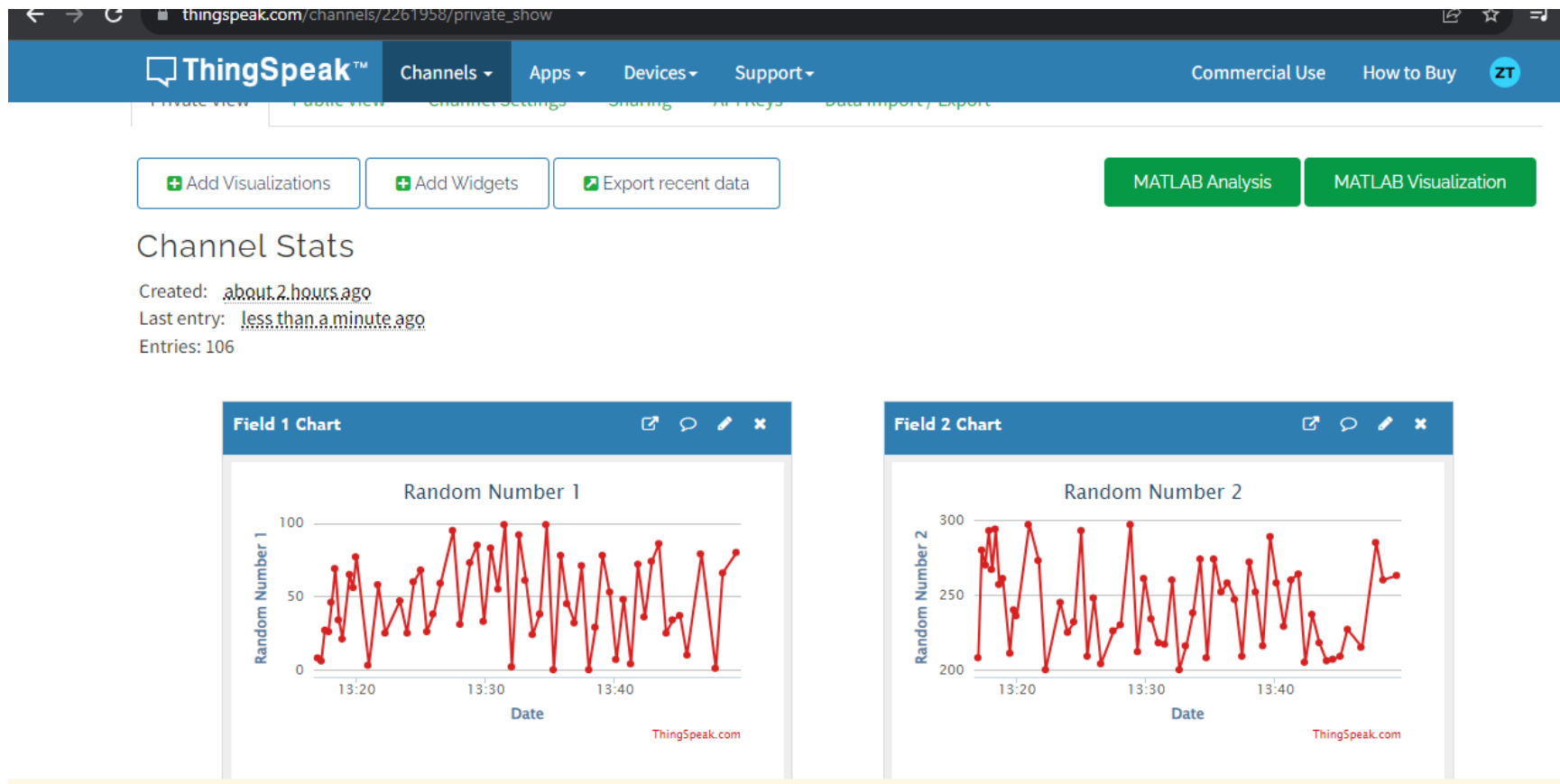
# THINGSPEAK

FEATURES	LICENSE				
	FREE	HOME	STUDENT	STANDARD	ACADEMIC
Number of messages	3 million/year (~8,200/day) <sup>(2)</sup>	33 million/year per unit (~90,000/day per unit) <sup>(2)</sup>	33 million/year per unit (~90,000/day per unit) <sup>(2)</sup>	33 million/year per unit (~90,000/day per unit) <sup>(2)</sup>	33 million/year per unit (~90,000/day per unit) <sup>(2)</sup>
Message update interval limit	Every 15 seconds	Every second	Every second	Every second	Every second
Number of channels	4	10 per unit	10 per unit	250 per unit	250 per unit

Ref: <https://randomnerdtutorials.com/esp32-thingspeak-publish-arduino/>



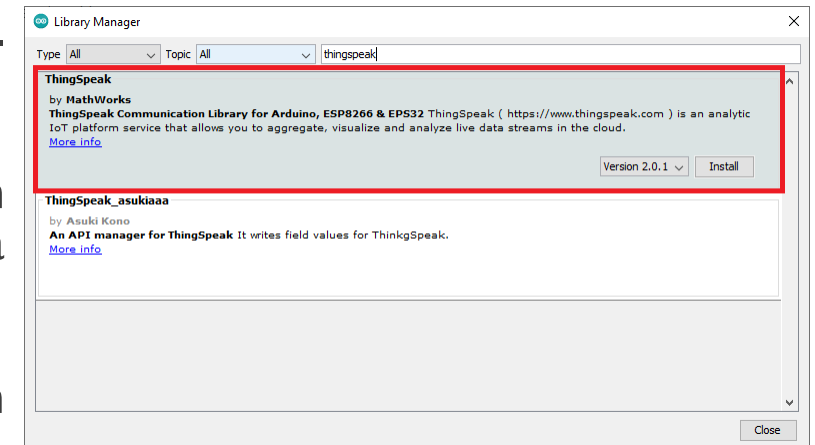
# ESP32 + THINGSPEAK



Ref: <https://randomnerdtutorials.com/esp32-thingspeak-publish-arduino/>

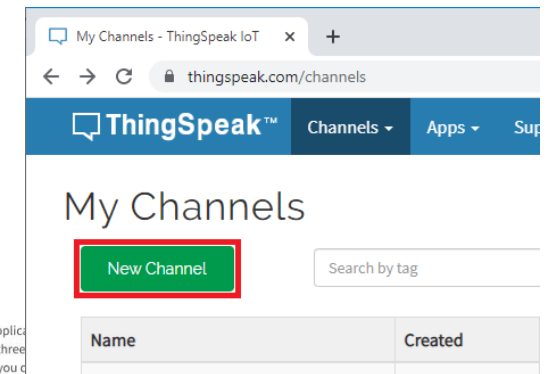
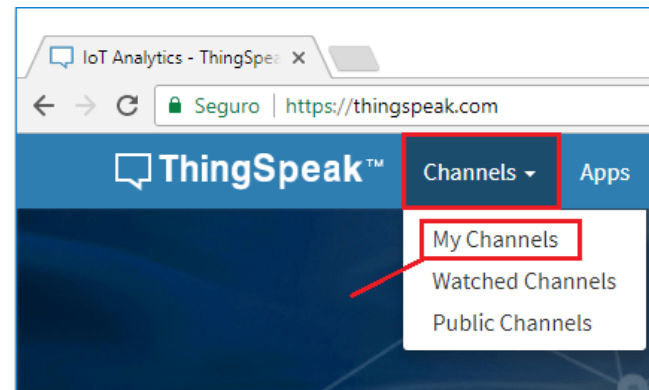
# ESP32 + THINGSPEAK

- Kali ini, kita akan menggunakan salah satu cara termudah-menggunakan library thingspeak-arduino.
- Library ini menyediakan metode untuk mempublikasikan pembacaan sensor dengan mudah ke satu bidang atau beberapa bidang.
- Untuk mengirim pembacaan sensor ke ThingSpeak, kita akan menggunakan library thingspeak-arduino.
- Kita dapat menginstal library ini melalui Arduino Library Manager.
- Buka Sketch > Include Library > Manage Libraries... dan cari "ThingSpeak" di Library Manager. Instal library ThingSpeak oleh MathWorks.



# MEMBUAT THINGSPEAK

- Buka ThingSpeak dan klik tombol "Get Started For Free" untuk membuat akun baru.
- Akun ini ditautkan ke akun Mathworks.
- Jadi, jika kita sudah memiliki akun Mathworks, kita bisa masuk dengan akun tersebut.



## New Channel

Name	<input type="text" value="DHT11 reading"/>	
Description	<input type="text"/>	
Field 1	<input type="text" value="Temp"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text" value="Hum"/>	<input checked="" type="checkbox"/>
Field 3	<input type="text"/>	<input type="checkbox"/>
Field 4	<input type="text"/>	<input type="checkbox"/>

## Help

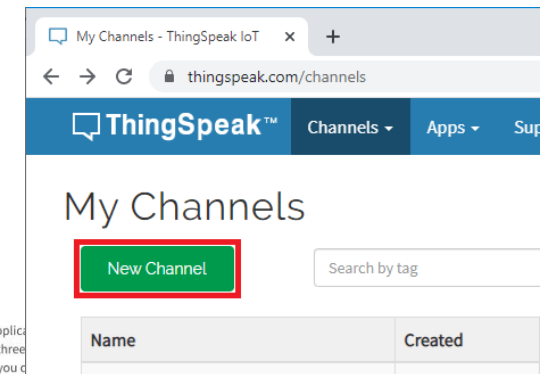
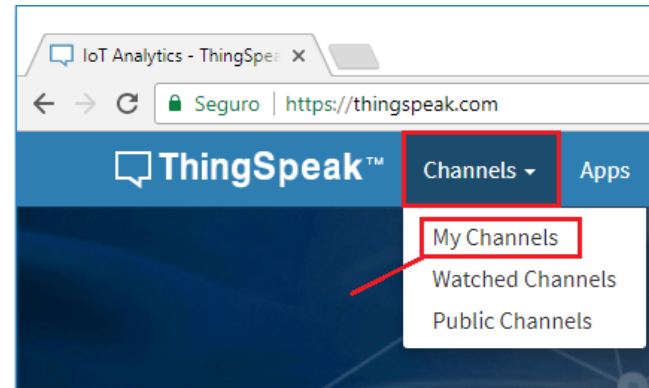
Channels store all the data that a ThingSpeak application sends. Each channel has eight fields that can hold any type of data, plus three status data. Once you collect data in a channel, you can visualize it.

### Channel Settings

- **Percentage complete:** Calculated based on data entered in a channel. Enter the name, description, location, URL, and channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a name for the channel. A channel can have up to 8 fields.

# MEMBUAT THINGSPEAK

- Setelah akun kita siap, masuk, buka tab " Channels" dan pilih "My Channels".
- Tekan tombol " New Channel" untuk membuat saluran baru
- Ketik nama untuk saluran kita dan tambahkan deskripsi.



## New Channel

Name

Description

Field 1  ☒

Field 2  ☒

Field 3  ☐

Field 4  ☐

## Help

Channels store all the data that a ThingSpeak application sends. Each channel has eight fields that can hold any type of data, plus three status data. Once you collect data in a channel, you can visualize it.

## Channel Settings

- **Percentage complete:** Calculated based on data entered in a channel. Enter the name, description, location, URL, and channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a name for the channel. A channel can have up to 8 fields.

# MEMBUAT THINGSPEAK

- Bagan dapat disesuaikan, buka tab Private View dan klik ikon edit
- Kita dapat memberikan judul pada grafik kita, menyesuaikan warna latar belakang, sumbu x dan y, dan masih banyak lagi.
- Setelah selesai, tekan tombol " Save ".

The screenshot shows the Thingspeak web interface. At the top, there are tabs: 'Private View' (highlighted with a red box), 'Public View', 'Channel Settings', 'Sharing', and 'API Keys'. Below the tabs are three buttons: 'Add Visualizations', 'Add Widgets', and 'Export recent data'. The 'Channel Stats' section shows 'Created: about a minute ago' and 'Entries: 0'. A 'Field 1 Chart' is displayed with the title 'BME280 Readings'. The chart has a blue header bar with an edit icon (pencil) highlighted by a red box and a close icon (X). To the right of the chart is the 'Field 1 Chart Options' dialog box. It contains various settings for the chart, including Title, X-Axis, Y-Axis, Color, Background, Type, Dynamic?, Days, Results, Timescale, Average, Median, Sum, Rounding, Data Min, Data Max, Y-Axis Min, and Y-Axis Max. The 'Save' button is green and the 'Cancel' button is light blue.

Private View Public View Channel Settings Sharing API Keys

Add Visualizations Add Widgets Export recent data

Channel Stats  
Created: about a minute ago  
Entries: 0

Field 1 Chart

BME280 Readings

Temperature

Date

ThingSpeak.com

Field 1 Chart Options

Title: BME280 Temperature Timescale:

X-Axis: Timestamp Average:

Y-Axis: Temperature °C Median:

Color: #d62020 Sum:

Background: #ffffff Rounding:

Type: line Data Min:

Dynamic?: true Data Max:

Days:

Results: 60 Y-Axis Min:

Y-Axis Max:

Save Cancel

# MEMBUAT THINGSPEAK

- Untuk mengirim nilai dari ESP32 ke ThingSpeak, kita memerlukan "Write API Key".
- Buka tab "API Keys" dan salin "Write API Key" ke tempat yang aman karena kita akan membutuhkannya sebentar lagi.

Private View

Public View

Channel Settings

Sharing

API Keys

## Write API Key

Key

HE00X0H.LT4V7Z

Generate New Write API Key

# COBA ESP32 + THINGSPEAK

- Ke [https://bit.ly/t4b\\_iot\\_code](https://bit.ly/t4b_iot_code)
- Klik folder day 4
- Download file randomNum\_thingspeak.ino
- Ubah hal2 yang diperlukan
- Upload kodenya

```
22  const char* ssid = "ganti"; // your network SSID (name)
23  const char* password = "ganti"; // your network password
24
```

```
27  unsigned long myChannelNumber = 1;
28  const char * myWriteAPIKey = "ganti";
29
```

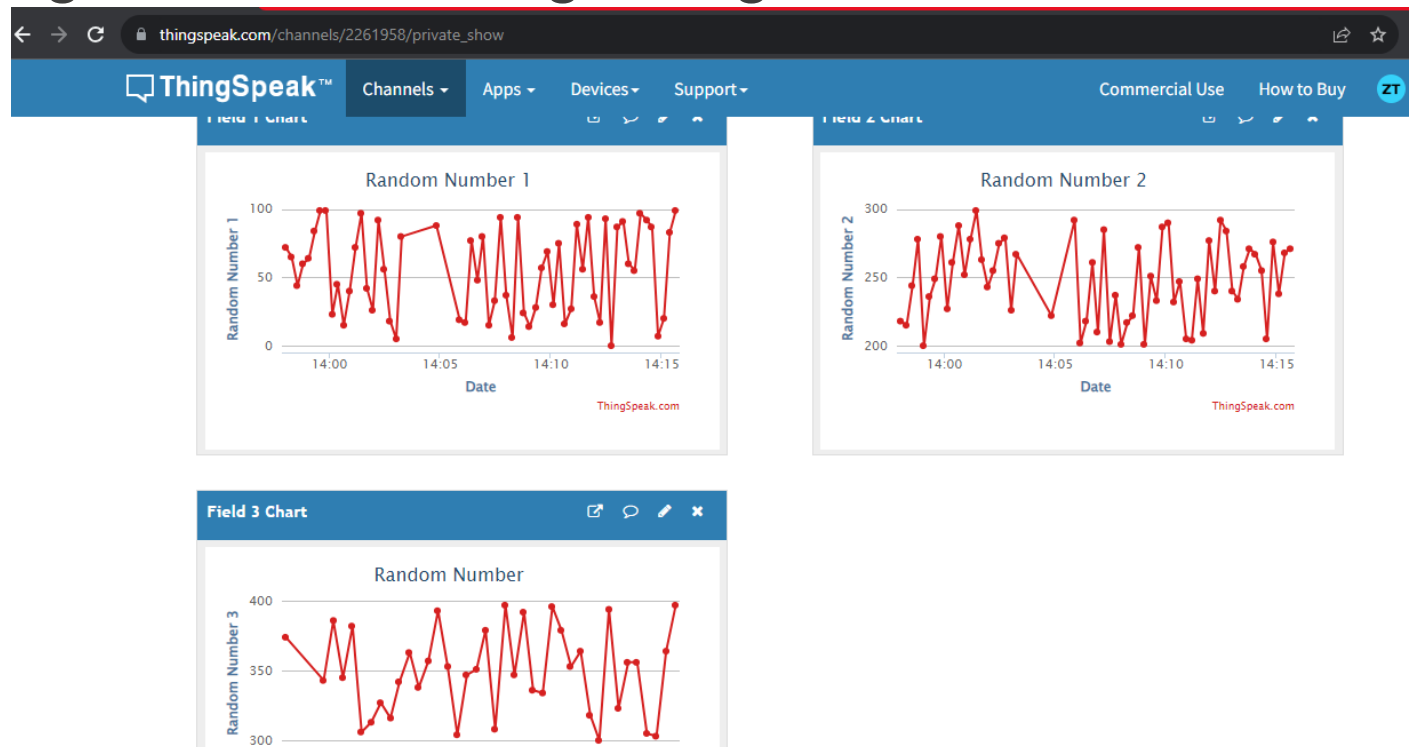
## PEMBAHASAN KODE

- [https://github.com/pidinplusplusultra/train4best\\_PENS\\_September/blob/main/Day%204/randomNum\\_thingspeak.ino](https://github.com/pidinplusplusultra/train4best_PENS_September/blob/main/Day%204/randomNum_thingspeak.ino)



# TUGAS ESP32 + THINGSPEAK

- Buat agar grafiknya menjadi 3
- Tambahkan 1 lagi data random dengan range 300 - 400



Ref: <https://www.geeksforgeeks.org/what-is-the-difference-between-gui-and-cui/>

---

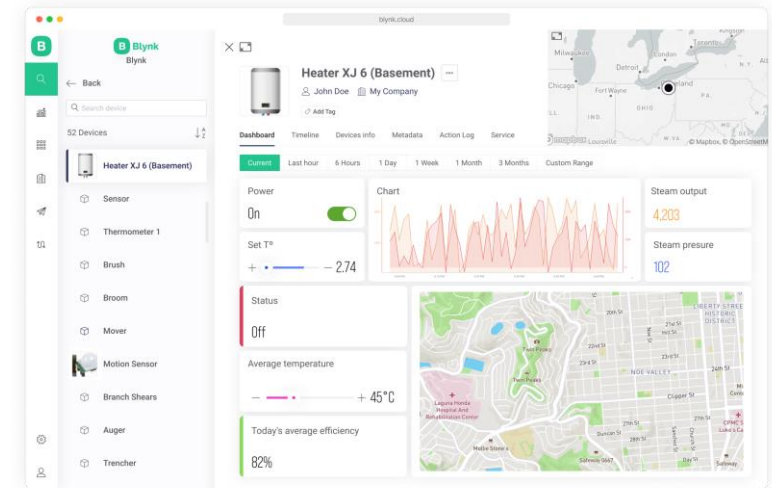
*Train4best*

BLYNK

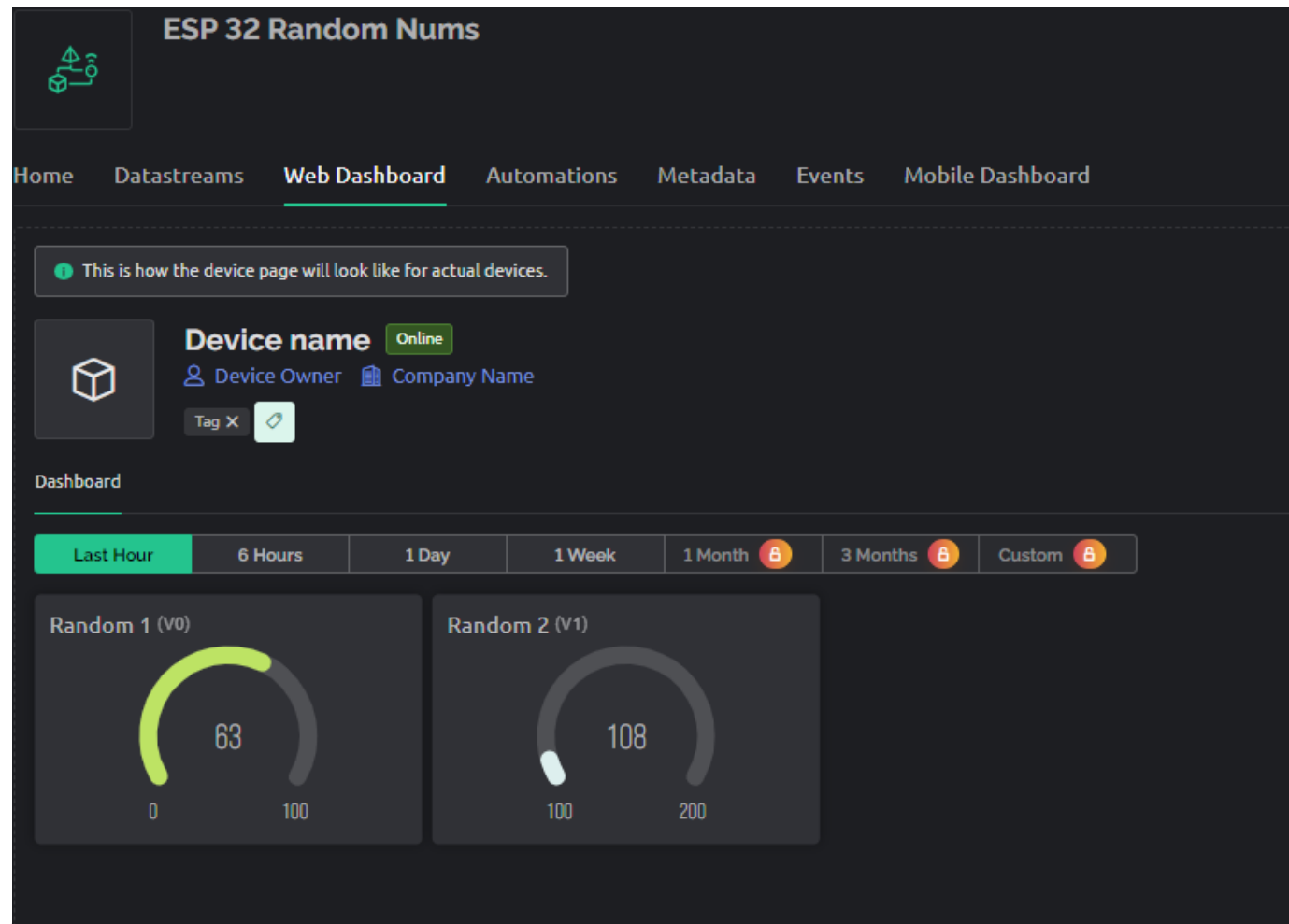
## Blynk.Console

Blynk.Console adalah aplikasi web yang kaya fitur yang melayani berbagai jenis pengguna. Fungsionalitas utamanya meliputi:

- Konfigurasi perangkat yang terhubung pada platform, termasuk pengaturan aplikasi.
- Manajemen perangkat, data, pengguna, organisasi, dan lokasi.
- Pemantauan dan kontrol perangkat dari jarak jauh



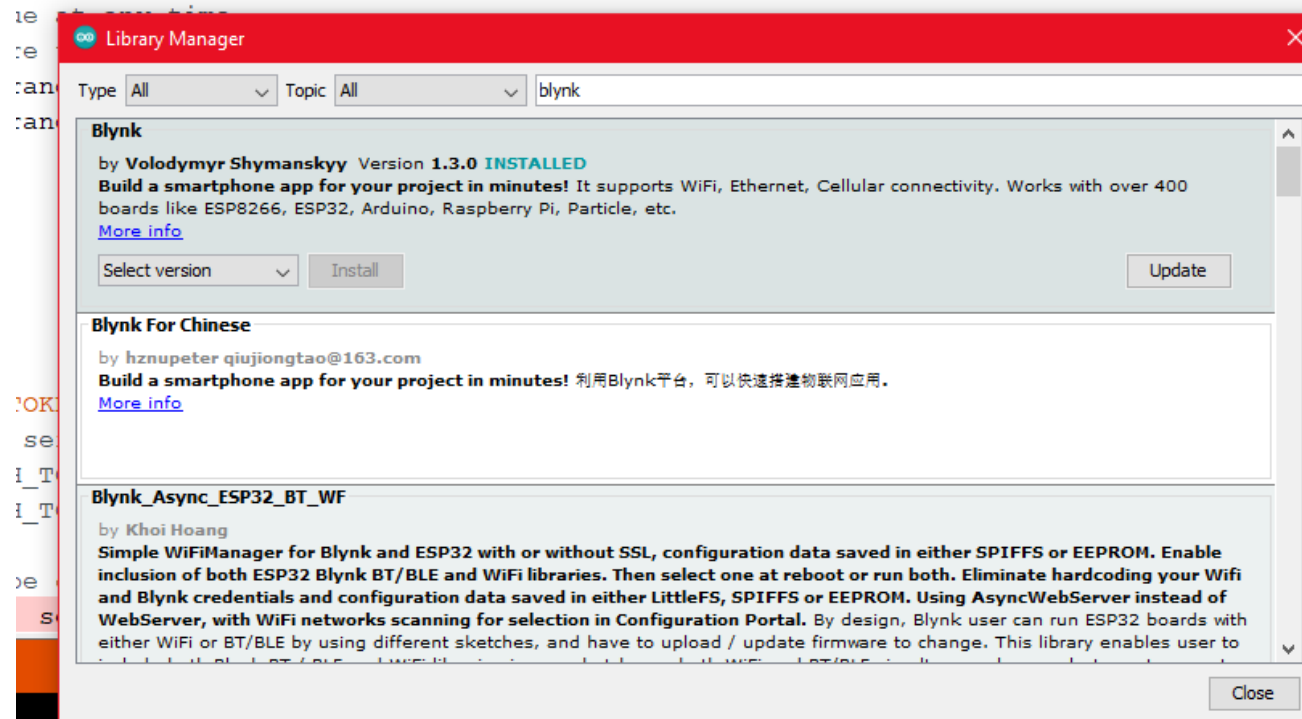
# HASIL COBA ESP32 + BLYNK



Ref: <https://docs.blynk.io/en/>

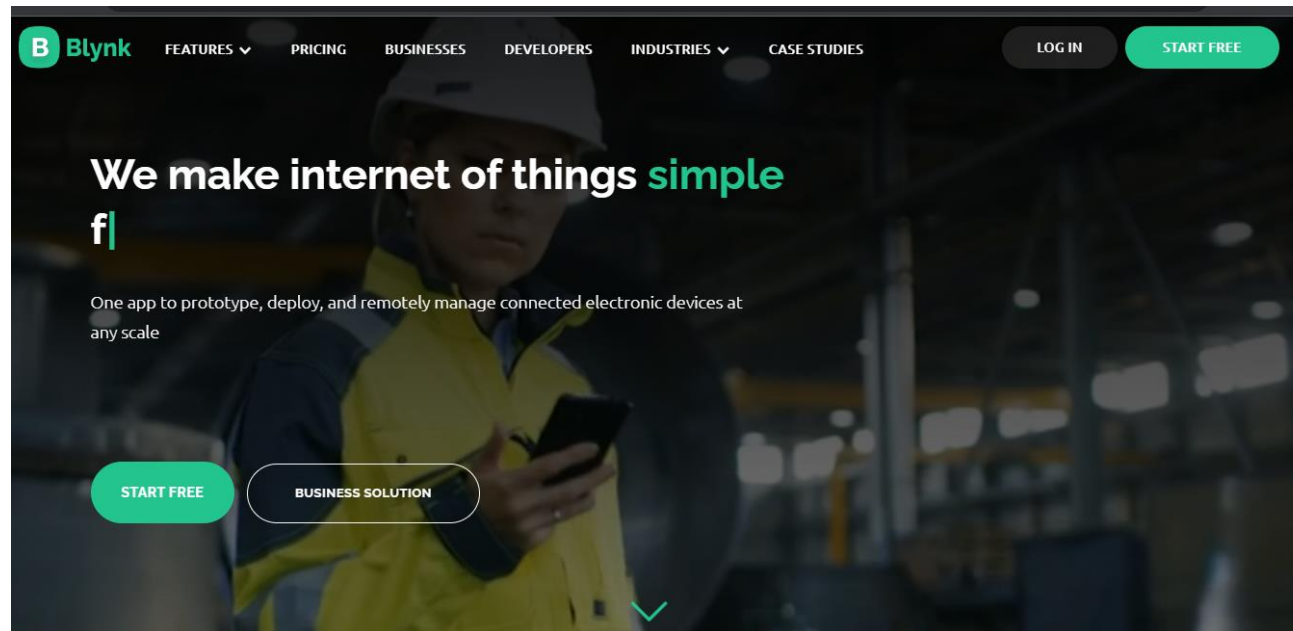
## ESP32 + Blynk

- Install library blynk di Arduino IDE



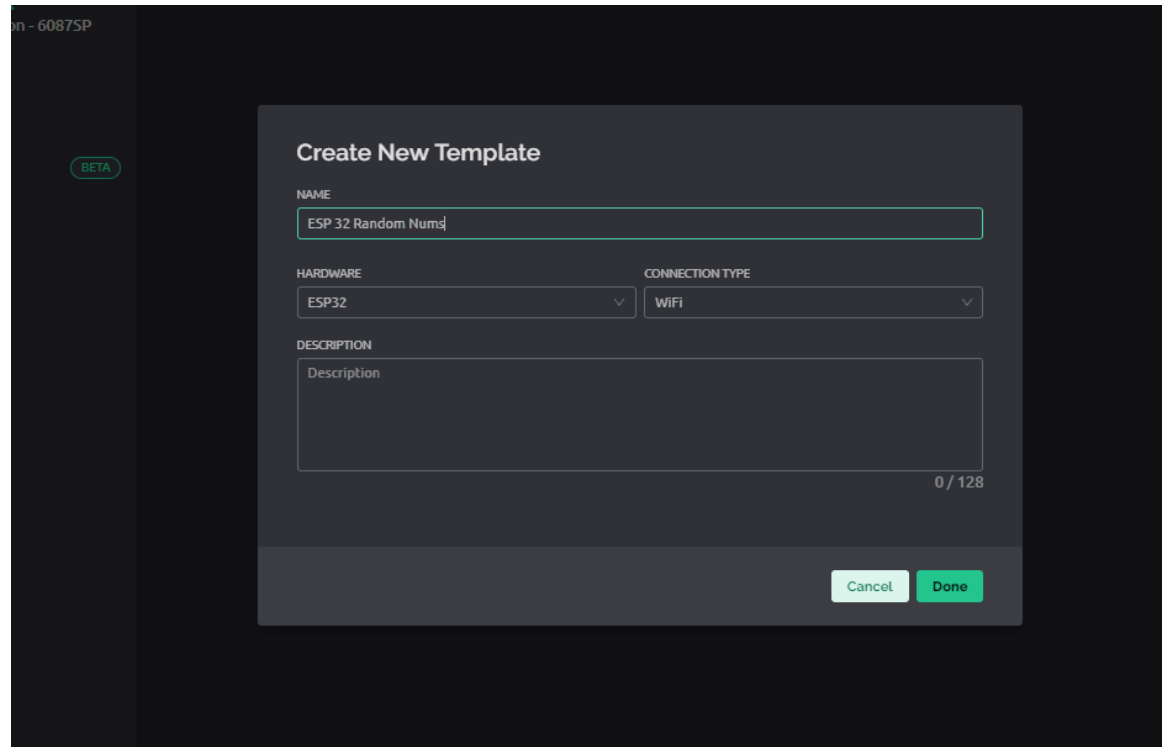
## ESP32 + Blynk

- Buat akun di <https://blynk.io/>
- Click started free



## ESP32 + Blynk

Buat template di web blynks



The screenshot shows the Blynk web interface with a dark theme. A 'Create New Template' dialog box is open in the center. The dialog has a title bar 'Create New Template'. Below the title, there are three main sections: 'NAME' with a text input field containing 'ESP 32 Random Num', 'HARDWARE' with a dropdown menu set to 'ESP32', and 'CONNECTION TYPE' with a dropdown menu set to 'WIFI'. Below these is a 'DESCRIPTION' section with a large text area containing the placeholder text 'Description'. At the bottom right of the text area, it says '0 / 128'. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Done'.

## ESP32 + Blynk

- Atur datastream buat 2 virtual pin, v0 untuk data random 1, dan v1 untuk data random 2.
- Ubah nilai min max-nya menjadi 0 dan 100 untuk v0
- Ubah nilai min max-nya menjadi 100 dan 200 untuk v1

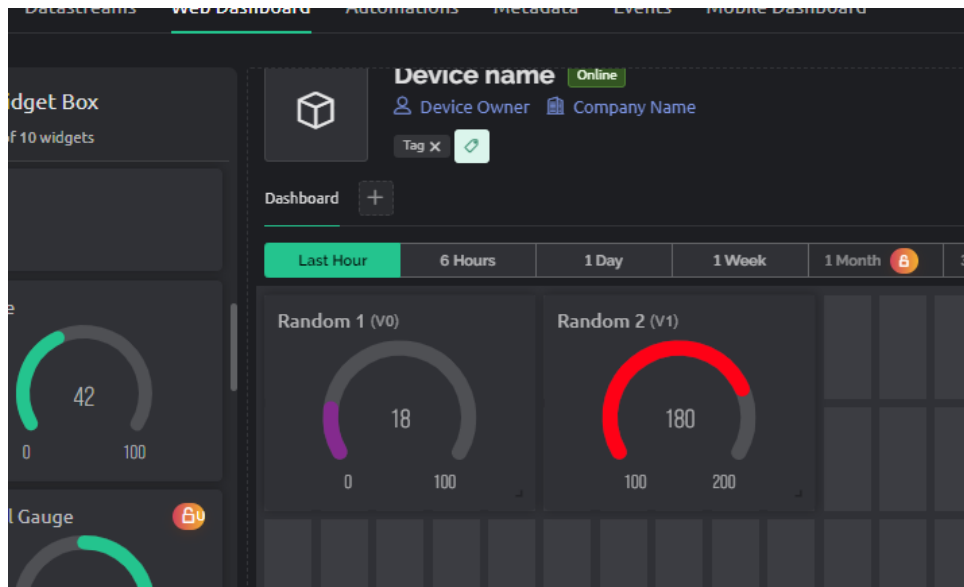
The screenshot shows the 'Virtual Pin Datastream' configuration window for a pin named 'Random 1'. The 'NAME' field is 'Random 1' and the 'ALIAS' field is 'Random 1'. The 'PIN' is set to 'V0' and the 'DATA TYPE' is 'Integer'. The 'UNITS' are set to 'None'. The 'MIN' value is 0 and the 'MAX' value is 100. The 'DEFAULT VALUE' is 0. There is an 'ADVANCED SETTINGS' button at the bottom left. At the bottom right, there are 'Cancel' and 'Save' buttons.

The screenshot shows the 'Virtual Pin Datastream' configuration window for a pin named 'Random 2'. The 'NAME' field is 'Random 2' and the 'ALIAS' field is 'Random 2'. The 'PIN' is set to 'V1' and the 'DATA TYPE' is 'Integer'. The 'UNITS' are set to 'None'. The 'MIN' value is 100 and the 'MAX' value is 200. The 'DEFAULT VALUE' is 0. There is an 'ADVANCED SETTINGS' button at the bottom left. At the bottom right, there are 'Cancel' and 'Save' buttons.



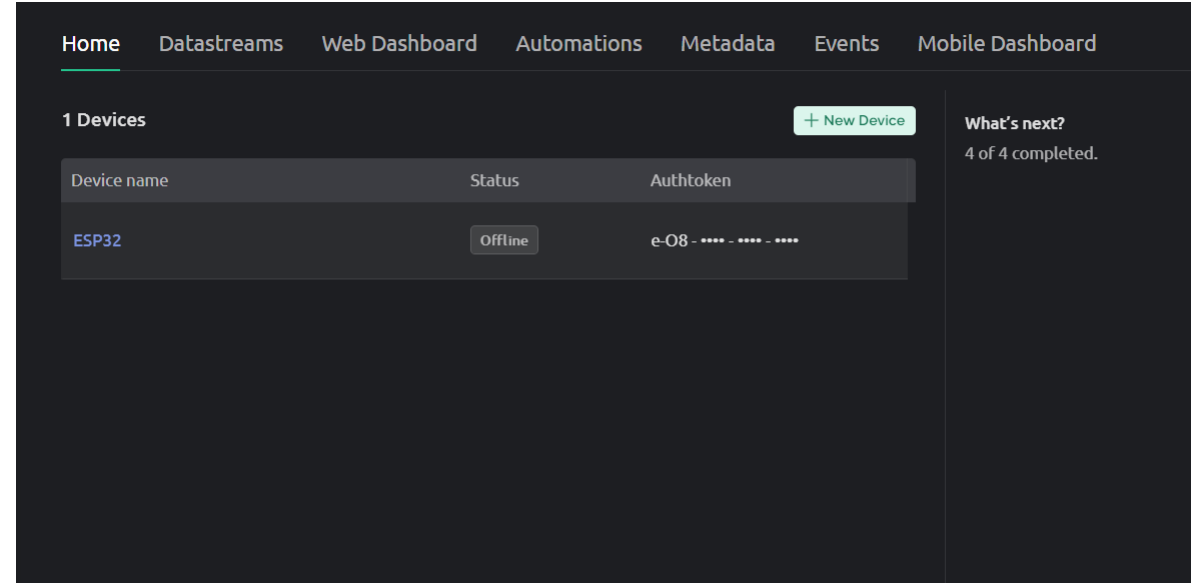
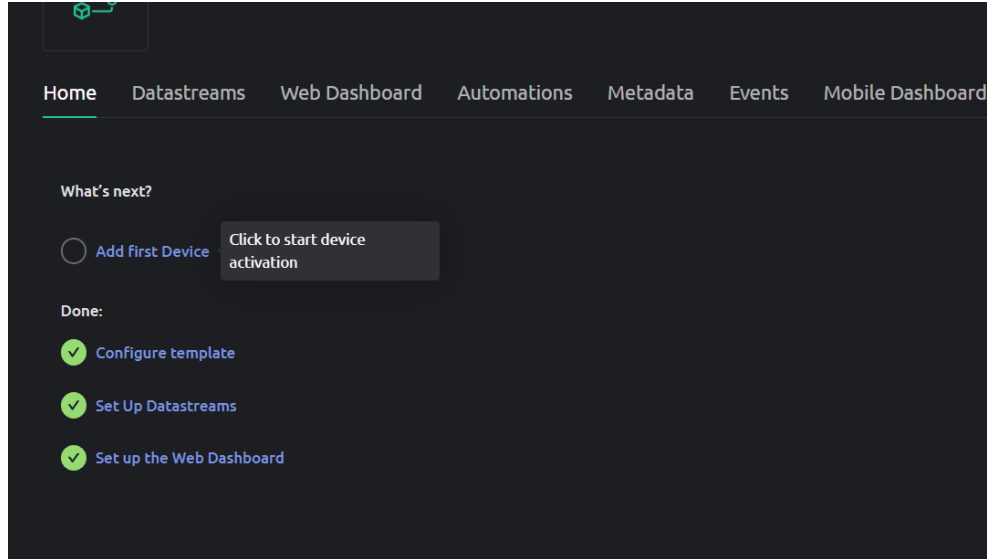
## ESP32 + Blynk

- Buat dashboard
- Dan atur widgetnya



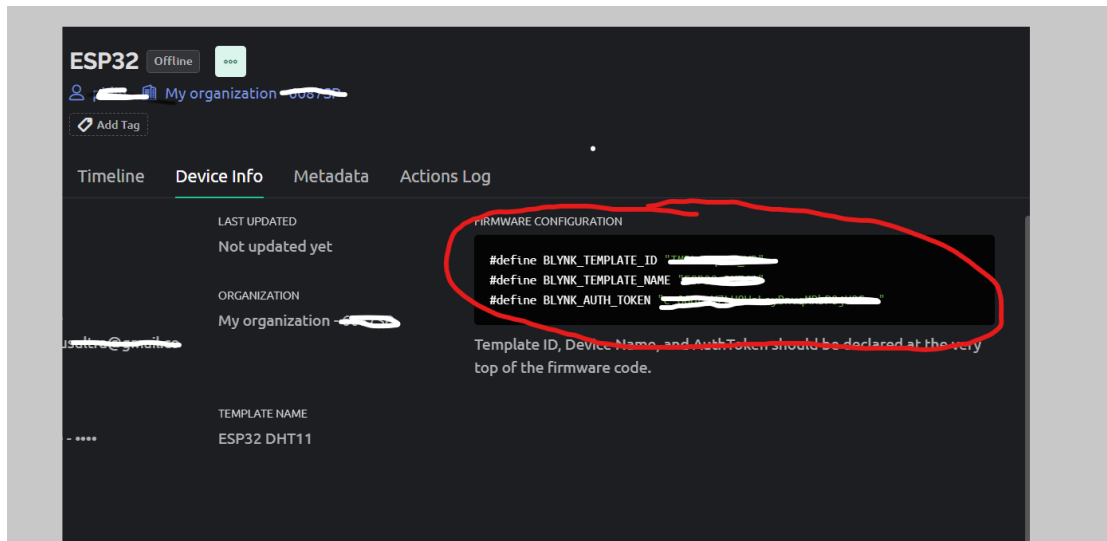
## ESP32 + Blynk

- Klik add first device
- Setelah menambahkan device, klik device tersebut di blynk kita



## ESP32 + Blynk

Lalu klik device info, dan copy firmware configuration kita



# COBA ESP32 + BLYNK

- Ke [https://bit.ly/t4b\\_iot\\_code](https://bit.ly/t4b_iot_code)
- Klik folder day 4
- Download file blynk\_random.ino
- Paste firmware configuration sesuai baris di kode
- Ubah hal2 yang diperlukan
- Upload kodenya

```
15
16     /* Fill-in information from Blynk Device Info here */
17     #define BLYNK_TEMPLATE_ID "ganti"
18     #define BLYNK_TEMPLATE_NAME "ganti"
19     #define BLYNK_AUTH_TOKEN "ganti"
30     // Your WiFi credentials.
31     // Set password to "" for open networks.
32     char ssid[] = "ganti";
33     char pass[] = "ganti";
34
```

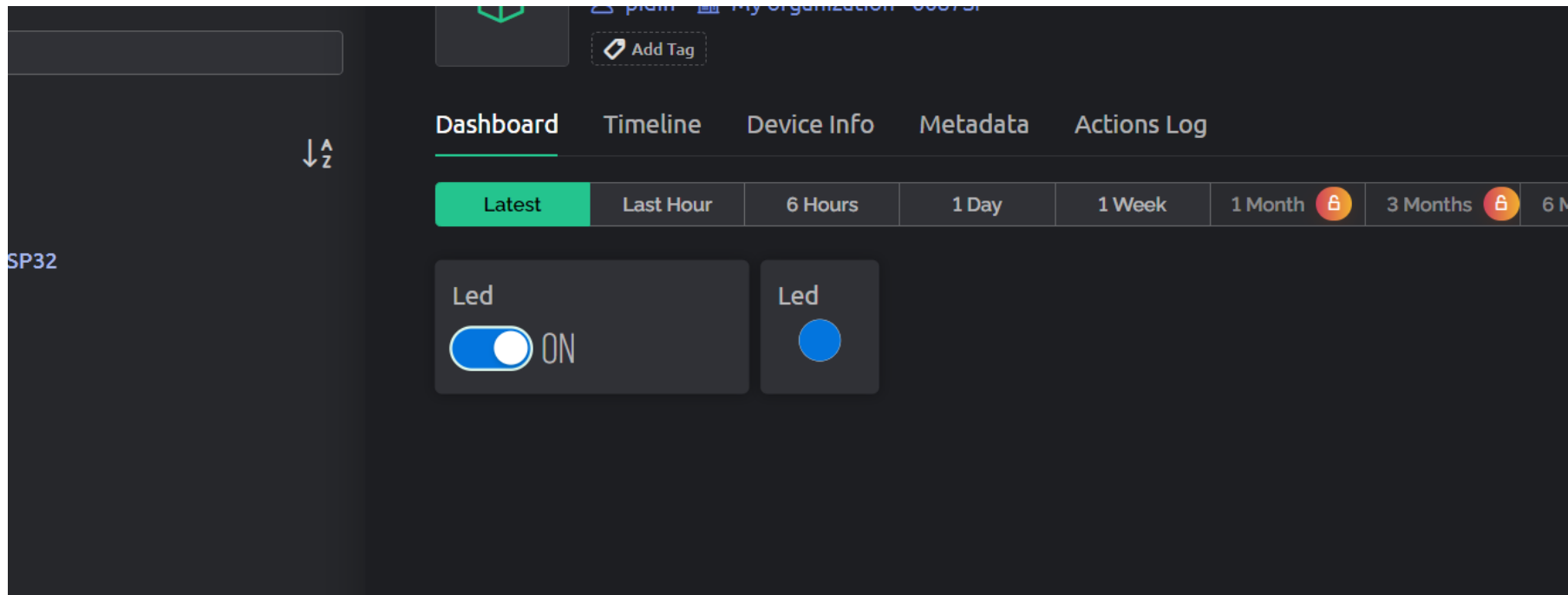
Ref: <https://docs.blynk.io/en/>

# PEMBAHASAN KODE ESP32 + BLYNK DHT11

- [https://github.com/pidinplusplusultra/train4best\\_PENS\\_September/blob/main/Day%204/blynk\\_random.ino](https://github.com/pidinplusplusultra/train4best_PENS_September/blob/main/Day%204/blynk_random.ino)

# HASIL COBA ESP32 + BLYNK LED

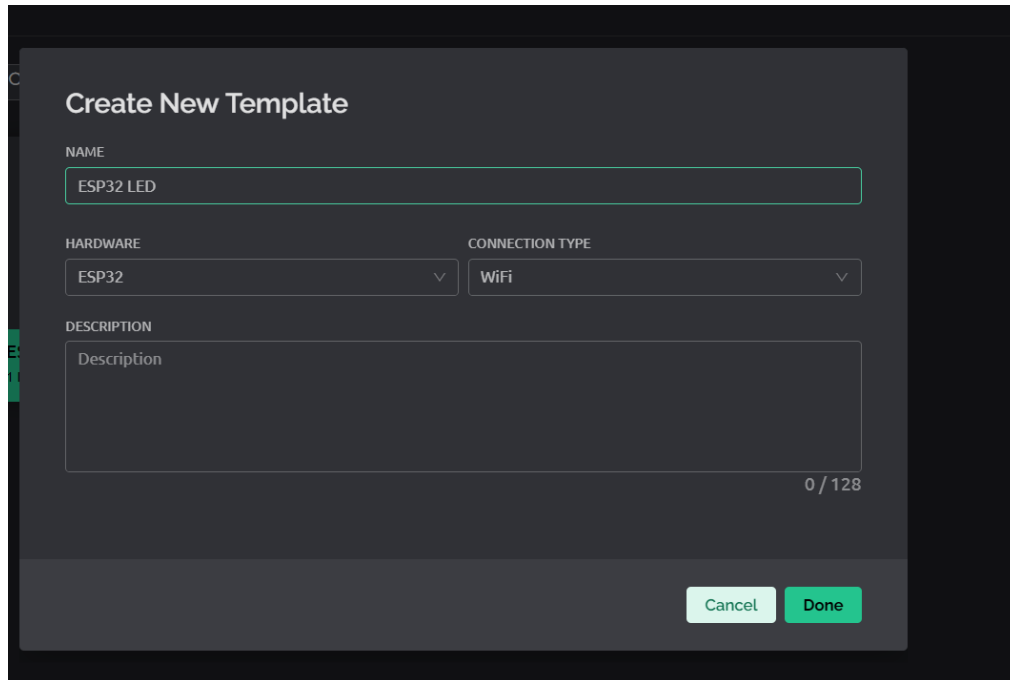
LED berubah setiap switch di klik



Ref: <https://docs.blynk.io/en/>

## ESP32 + Blynk LED

Buat template di web blynk



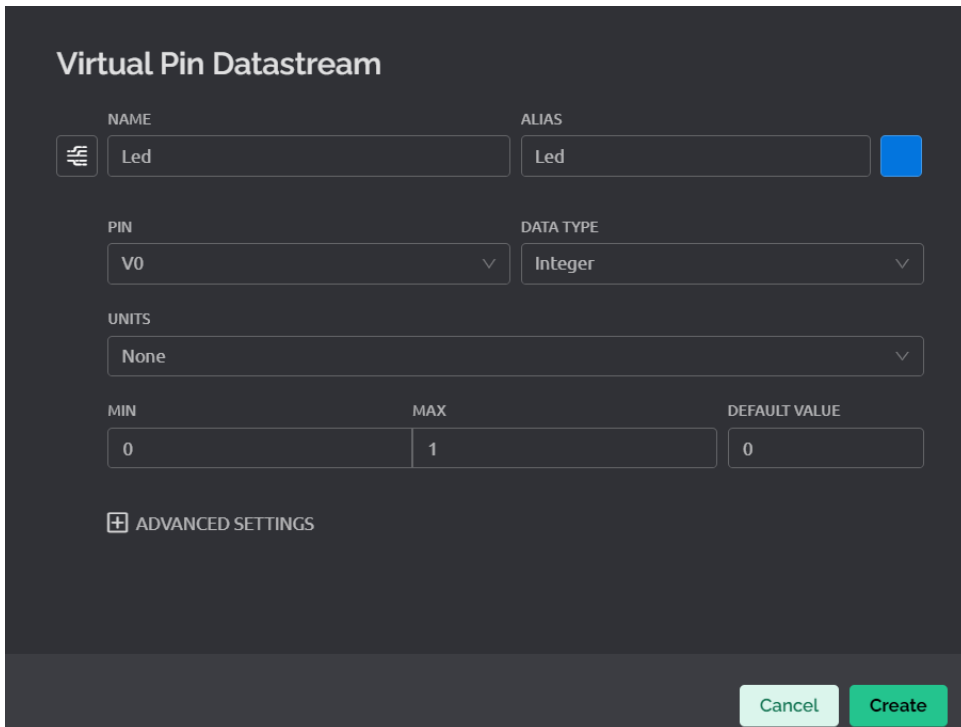
The screenshot shows the 'Create New Template' interface on the Blynk web platform. The form is dark-themed and contains the following fields:

- NAME:** A text input field containing 'ESP32 LED'.
- HARDWARE:** A dropdown menu with 'ESP32' selected.
- CONNECTION TYPE:** A dropdown menu with 'WiFi' selected.
- DESCRIPTION:** A large text area with the placeholder text 'Description' and a character count '0 / 128' at the bottom right.

At the bottom right of the form, there are two buttons: 'Cancel' (light blue) and 'Done' (orange).

## ESP32 + Blynk LED

Atur datastream, buat 1 virtual pin untuk led dengan datatype integer



The screenshot shows the 'Virtual Pin Datastream' configuration window in the Blynk application. The interface is dark-themed with white text. At the top, the title 'Virtual Pin Datastream' is displayed. Below it, there are several input fields and dropdown menus. The 'NAME' field contains 'Led', and the 'ALIAS' field also contains 'Led'. To the right of the 'ALIAS' field is a blue square button. Below these, the 'PIN' dropdown menu is set to 'V0', and the 'DATA TYPE' dropdown menu is set to 'Integer'. The 'UNITS' dropdown menu is set to 'None'. At the bottom, there are three input fields: 'MIN' with the value '0', 'MAX' with the value '1', and 'DEFAULT VALUE' with the value '0'. Below these fields is a link labeled 'ADVANCED SETTINGS' with a plus icon. At the very bottom, there are two buttons: 'Cancel' and 'Create'.

Virtual Pin Datastream

NAME: Led ALIAS: Led

PIN: V0 DATA TYPE: Integer

UNITS: None

MIN: 0 MAX: 1 DEFAULT VALUE: 0

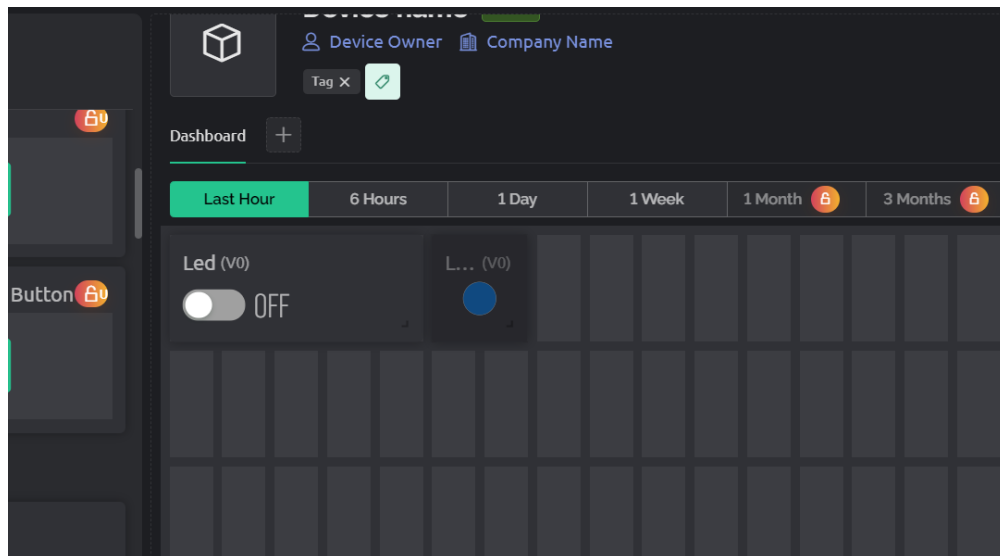
ADVANCED SETTINGS

Cancel Create



## ESP32 + Blynk LED

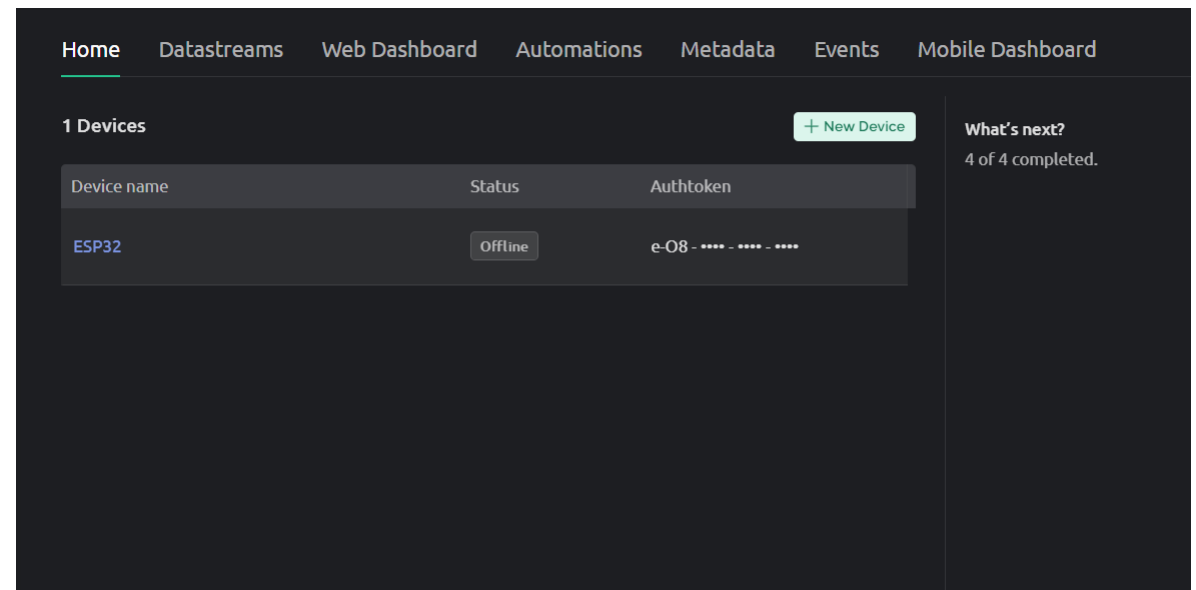
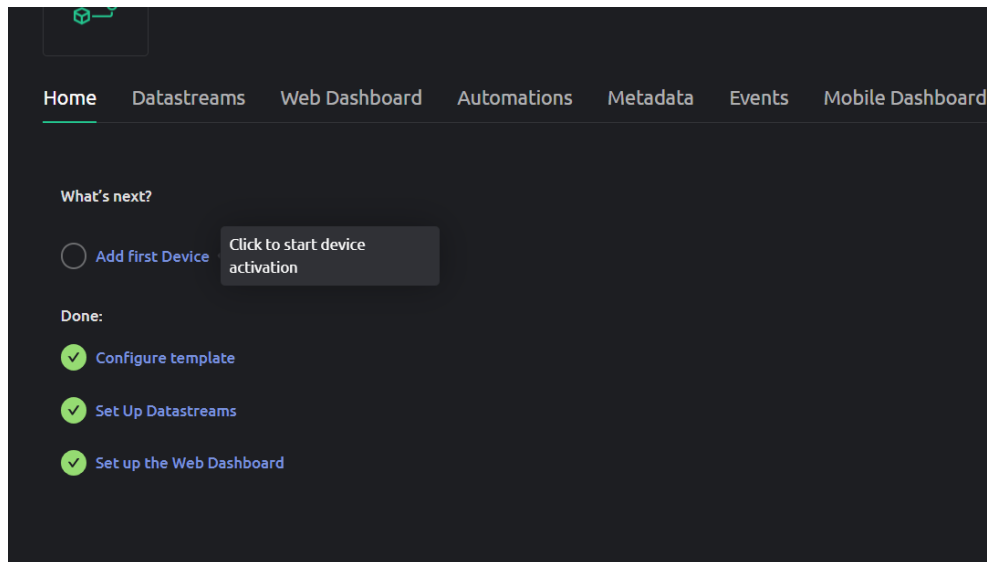
Buat dashboard



## ESP32 + Blynk LED

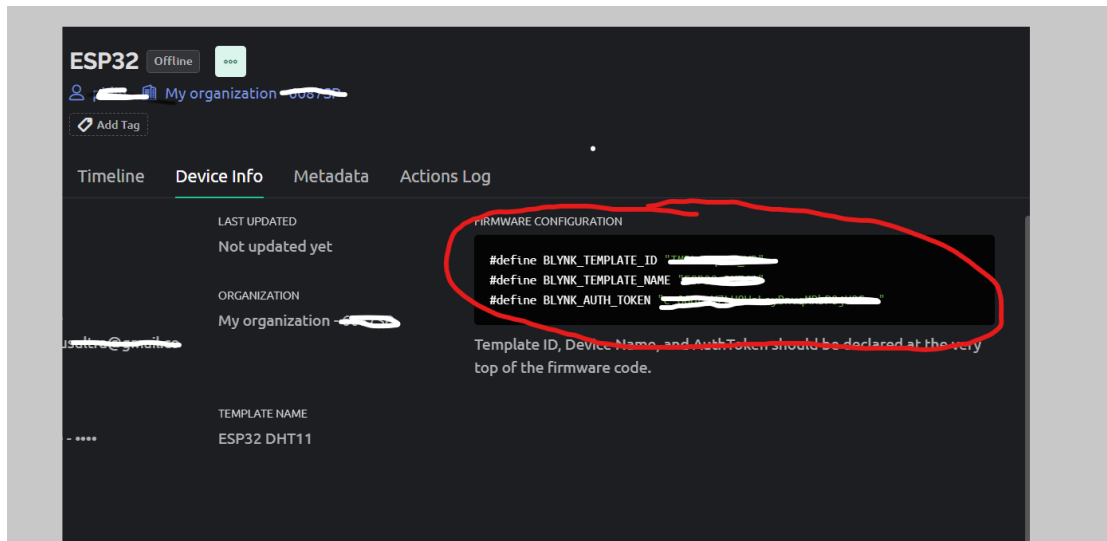
Add first device

Setelah menambahkan device, klik device tersebut di blynk kita



## ESP32 + Blynk LED

Lalu klik device info, dan copy firmware configuration kita



# COBA ESP32 + BLYNK LED

- Ke [https://bit.ly/t4b\\_iot\\_code](https://bit.ly/t4b_iot_code)
- Klik folder day 4
- Download file test\_blynk\_led.ino
- Paste firmware configuration sesuai baris di kode
- Ubah hal2 yang diperlukan
- Upload kodenya

```
15
16     /* Fill-in information from Blynk Device Info here */
17     #define BLYNK_TEMPLATE_ID "ganti"
18     #define BLYNK_TEMPLATE_NAME "ganti"
19     #define BLYNK_AUTH_TOKEN "ganti"
30     // Your WiFi credentials.
31     // Set password to "" for open networks.
32     char ssid[] = "ganti";
33     char pass[] = "ganti";
34
```

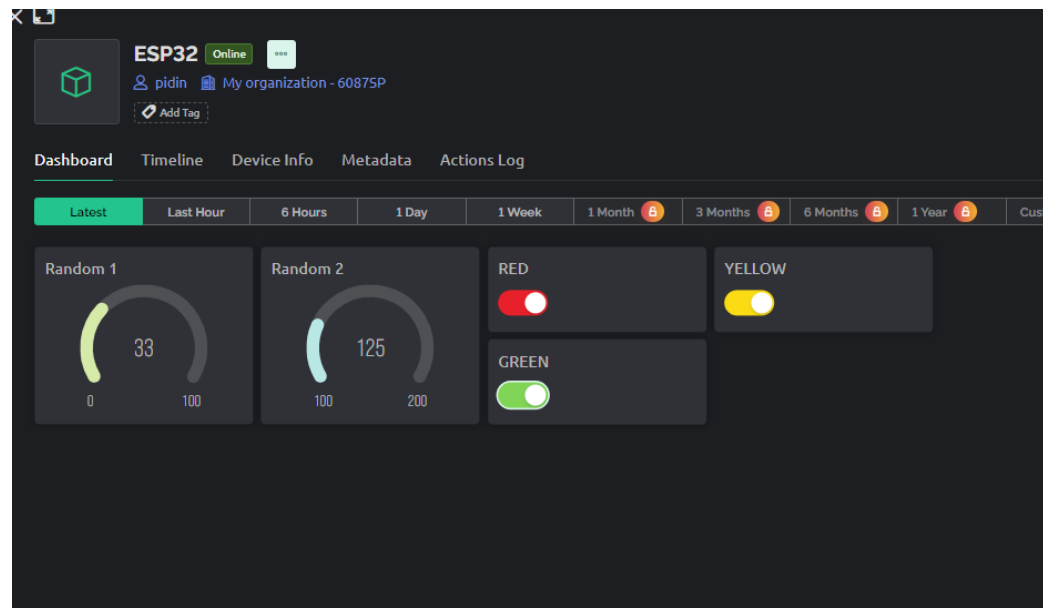
Ref: <https://docs.blynk.io/en/>

# PEMBAHASAN KODE ESP32 + BLYNK LED

- [https://github.com/pidinplusplusultra/train4best\\_PENS\\_September/blob/main/Day%204/test\\_blynk\\_led.ino](https://github.com/pidinplusplusultra/train4best_PENS_September/blob/main/Day%204/test_blynk_led.ino)

# TUGAS BLYNK

- Modifikasi template blynk dan kode esp32, agar dapat mengendalikan modul lampu traffic light.
- Tips: untuk menyalakan lampu traffic light adalah dengan memberikan tegangan di pin yang sesuai dengan warna lampu yang ingin dinyalakan.



## EXAMPLE CODE BLYNK

- Untuk contoh” kode blynk lainnya dapat kita temukan di <https://examples.blynk.cc/>
- Atau dapat kita cari di dokumentasi blynk <https://docs.blynk.io/en/>

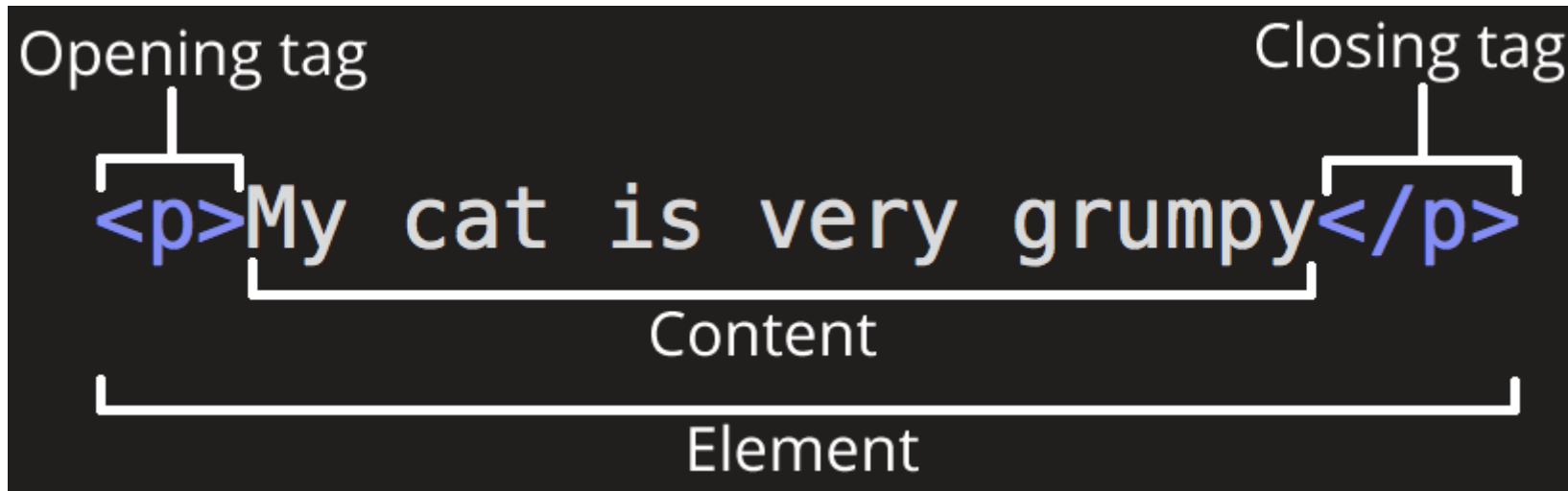
# DASAR HTML



## HTML

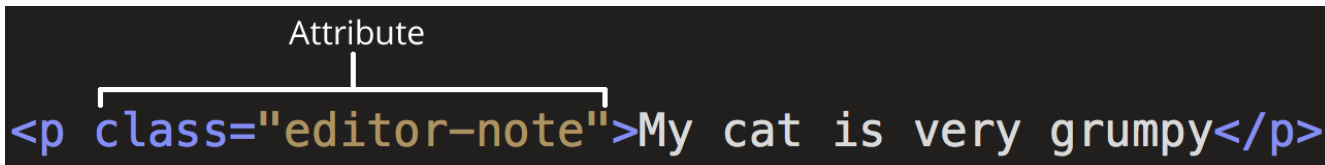
- HTML (HyperText Markup Language) adalah kode yang digunakan untuk menyusun halaman web dan kontennya.
- Sebagai contoh, konten dapat disusun dalam serangkaian paragraf, daftar poin-poin, atau menggunakan gambar dan tabel data.
- HTML adalah bahasa markup yang mendefinisikan struktur konten kita.
- HTML terdiri dari serangkaian elemen, yang kita gunakan untuk mengapit, atau membungkus, bagian konten yang berbeda untuk membuatnya tampil dengan cara tertentu, atau bertindak dengan cara tertentu.
- Tag-tag yang melingkupi dapat membuat sebuah kata atau gambar menjadi hyperlink ke tempat lain, dapat memiringkan kata, dapat membuat font lebih besar atau lebih kecil, dan seterusnya.

## Anatomi Elemen HTML



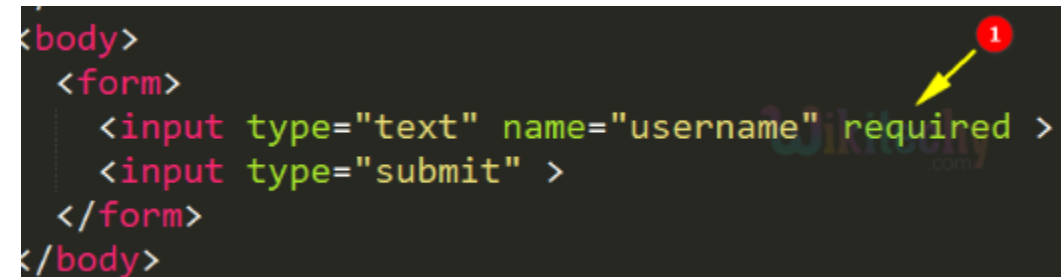
## Atribut Elemen HTML

- Atribut berisi informasi tambahan tentang elemen yang tidak ingin kita tampilkan di konten yang sebenarnya.
- Di sini, class adalah nama atribut dan editor-note adalah nilai atribut
- Beberapa atribut tidak memiliki nilai, seperti required.



A diagram illustrating the structure of an HTML element. A bracket labeled "Attribute" points to the `class="editor-note"` part of the code snippet below.

```
<p class="editor-note">My cat is very grumpy</p>
```



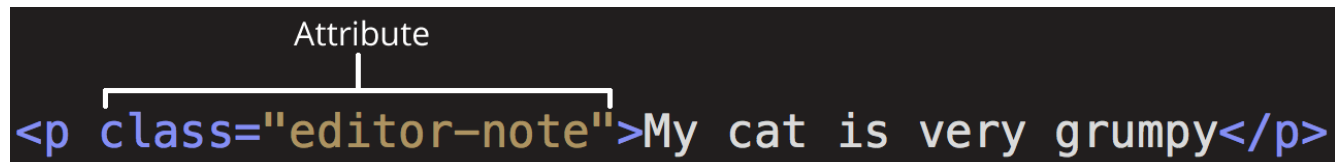
A code snippet showing HTML code. A yellow arrow points to the `required` attribute in the `<input type="text" name="username" required >` line, which is marked with a red circle containing the number 1.

```
<body>
  <form>
    <input type="text" name="username" required >
    <input type="submit" >
  </form>
</body>
```

## Atribut Elemen HTML

Atribut yang selalu menetapkan nilai memiliki:

- Spasi di antara atribut tersebut dan nama elemen (atau atribut sebelumnya, jika elemen tersebut sudah memiliki satu atribut atau lebih).
- Nama atribut yang diikuti dengan tanda sama dengan.
- Nilai atribut yang dibungkus dengan tanda petik pembuka dan penutup.



```
<p class="editor-note">My cat is very grumpy</p>
```

## Nesting elements

- Kita juga dapat meletakkan elemen di dalam elemen lain - ini disebut nesting.
- My cat is **very** grumpy

HTML



```
<p>My cat is <strong>very</strong> grumpy.</p>
```

## Void elements

- Beberapa elemen tidak memiliki konten dan disebut elemen kosong.
- Contoh berikut berisi dua atribut, tetapi tidak ada tag `</img>` penutup dan tidak ada konten bagian dalam. Ini karena elemen gambar tidak membungkus konten.
- Tujuannya adalah untuk menyematkan gambar pada halaman HTML di tempat gambar itu muncul.

```
HTML

```

## Anatomi dokumen HTML

Sekarang kita akan melihat bagaimana masing-masing elemen digabungkan untuk membentuk keseluruhan halaman HTML.

### HTML

```
<!doctype html>
<html lang="en-US">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>My test page</title>
  </head>
  <body>
    
  </body>
</html>
```

## Anatomi dokumen HTML

- `<!DOCTYPE html>` - doctype. Ini adalah pembukaan yang diperlukan.
  - Deklarasi `<!DOCTYPE html>` adalah cara untuk menentukan bahwa dokumen HTML yang akan diuraikan adalah versi HTML5.
- `<html></html>` - elemen `<html>`. Elemen ini membungkus semua konten di seluruh halaman dan kadang-kadang dikenal sebagai elemen root.
  - Elemen ini juga menyertakan atribut `lang`, yang mengatur bahasa utama dokumen.



## Anatomi dokumen HTML

- `<head></head>` - elemen `<head>`. Elemen ini berfungsi sebagai wadah untuk semua hal yang ingin kita sertakan pada halaman HTML yang bukan merupakan konten yang kita tampilkan kepada pemirsa halaman kita.
  - Ini termasuk hal-hal seperti kata kunci dan deskripsi halaman yang ingin kita tampilkan di hasil pencarian, CSS untuk menata konten kita, deklarasi set karakter, dan banyak lagi.

## Anatomi dokumen HTML

- `<meta charset="utf-8">` - Elemen ini menetapkan set karakter yang harus digunakan dokumen kita ke UTF-8 yang mencakup sebagian besar karakter dari sebagian besar bahasa tertulis.
  - Pada dasarnya, elemen ini sekarang dapat menangani konten tekstual apa pun yang kita masukkan ke dalamnya.
  - Tidak ada alasan untuk tidak mengaturnya, dan ini dapat membantu menghindari beberapa masalah di kemudian hari.
- `<meta name="viewport" content="width=device-width">` - Elemen viewport ini memastikan halaman dirender sesuai dengan lebar viewport, sehingga browser seluler tidak merender halaman yang lebih lebar daripada viewport dan kemudian mengecilkannya.

## Anatomi dokumen HTML

- `<title></title>` - Elemen `<title>`. Elemen ini menetapkan judul halaman kita, yaitu judul yang muncul di tab browser tempat halaman dimuat.
  - Elemen ini juga digunakan untuk mendeskripsikan halaman ketika kita menandai/memfavoritkannya.
- `<body></body>` - elemen `<body>`. Elemen ini berisi semua konten yang ingin kita tampilkan kepada pengguna web ketika mereka mengunjungi halaman kita, baik itu teks, gambar, video, game, trek audio yang dapat diputar, atau apa pun.

## DASAR CSS

## CSS

- CSS (Cascading Style Sheets) adalah kode yang menata style konten web.
- Seperti HTML, CSS bukanlah bahasa pemrograman. CSS juga bukan bahasa markup.
- CSS adalah bahasa style sheet.
- CSS adalah apa yang kita gunakan untuk mem-style elemen HTML secara selektif.
- Sebagai contoh, CSS ini memilih teks paragraf, menetapkan warna menjadi merah:

CSS



```
p {  
  color: red;  
}
```

## Anatomi CSS

Keseluruhan struktur ini disebut dengan ruleset. (Istilah ruleset sering disebut sebagai rule saja.) Nama-nama bagiannya:

- Selector
  - Ini adalah nama elemen HTML pada awal ruleset.
  - Ini mendefinisikan elemen yang akan diberi gaya (dalam contoh ini, elemen <p>).
  - Untuk menata elemen yang berbeda, ubah selektor.
- Deklarasi
  - Ini adalah aturan tunggal seperti color: red;.
  - Aturan ini menentukan properti elemen mana yang ingin kita beri gaya.



## Anatomi CSS

- Properti
  - Ini adalah cara-cara yang dapat digunakan untuk menata elemen HTML. (Dalam contoh ini, color adalah properti dari elemen <p>.)
  - Dalam CSS, kita memilih properti mana yang ingin kita pengaruhi dalam aturan.
- Property value
  - Di sebelah kanan properti-setelah titik dua-terdapat nilai properti.
  - Ini memilih satu dari sekian banyak kemungkinan tampilan untuk properti tertentu.
  - (Sebagai contoh, ada banyak nilai warna selain merah).



## Anatomi CSS

Perhatikan bagian penting lainnya dari sintaks:

- Selain pemilih, setiap kumpulan aturan harus diapit dengan tanda kurung kurawal. ({})
- Dalam setiap deklarasi, kita harus menggunakan tanda titik dua (:) untuk memisahkan properti dari nilai atau nilai-nilainya.
- Dalam setiap ruleset, kita harus menggunakan tanda titik koma (;) untuk memisahkan setiap deklarasi dari deklarasi berikutnya.

Untuk memodifikasi beberapa nilai properti dalam satu ruleset, tuliskan nilai tersebut dengan dipisahkan oleh titik koma.



```
p {  
  color: red;  
  width: 500px;  
  border: 1px solid black;  
}
```



## Anatomi CSS

Kita juga dapat memilih beberapa elemen dan menerapkan satu set aturan ke semuanya. Pisahkan beberapa selektor dengan koma.

### CSS

```
p,  
li,  
h1 {  
  color: red;  
}
```

# Selektor CSS

Ada banyak jenis selektor yang berbeda. Berikut ini beberapa jenis selektor yang lebih umum:

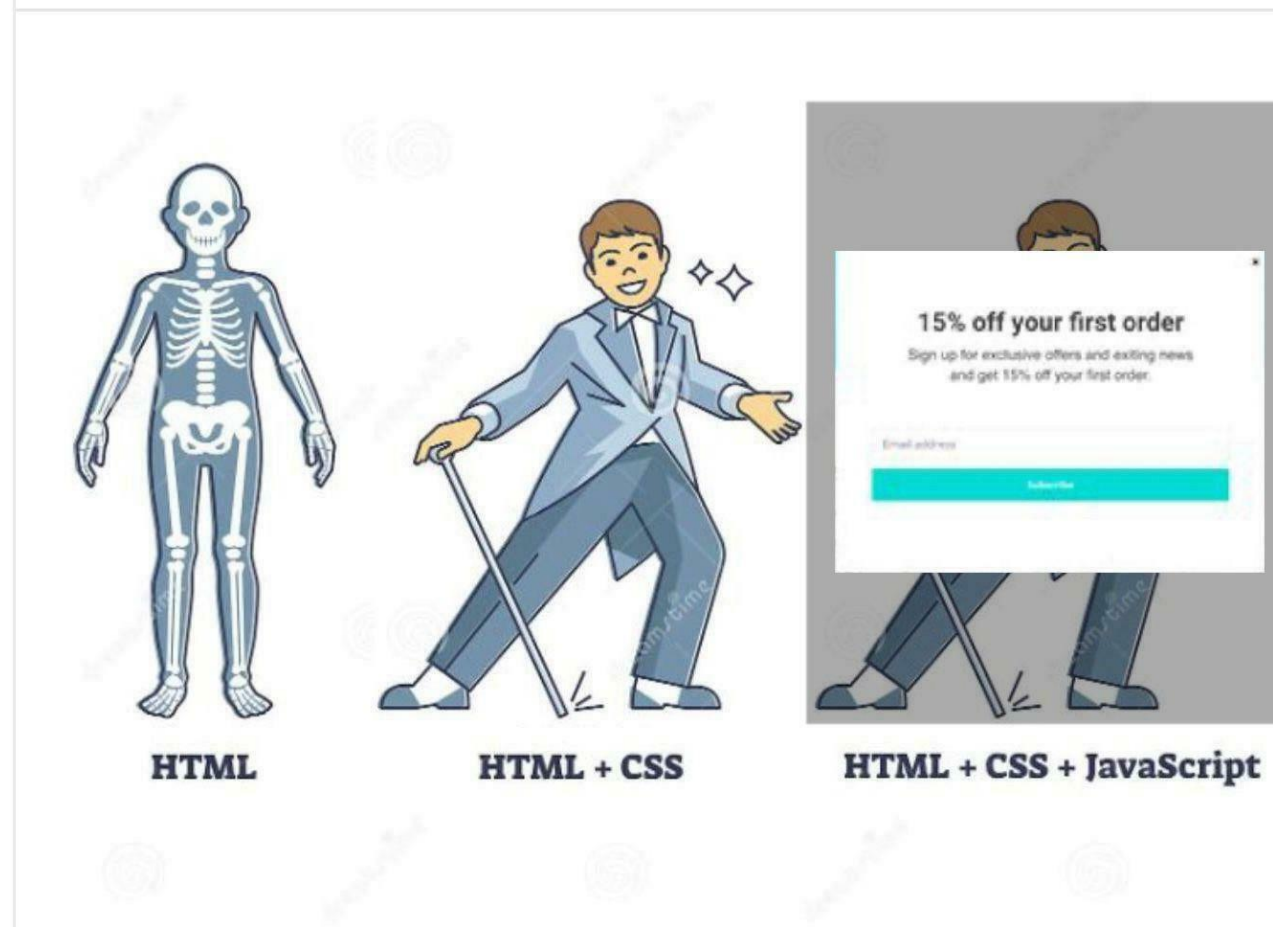
Selector name	What does it select	Example
<b>Element selector (sometimes called a tag or type selector)</b>	All HTML elements of the specified type.	p selects <p>
<b>ID selector</b>	The element on the page with the specified ID. On a given HTML page, each id value should be unique.	#my-id selects <p id="my-id"> or <a id="my-id">
<b>Class selector</b>	The element(s) on the page with the specified class. Multiple instances of the same class can appear on a page.	.my-class selects <p class="my-class"> and <a class="my-class">
<b>Attribute selector</b>	The element(s) on the page with the specified attribute.	img[src] selects  but not <img>
<b>Pseudo-class selector</b>	The specified element(s), but only when in the specified state. (For example, when a cursor hovers over a link.)	a:hover selects <a>, but only when the mouse pointer is hovering over the link.

# DASAR JAVASCRIPT

## Javascript

- JavaScript adalah sebuah bahasa pemrograman yang digunakan untuk mengembangkan aplikasi web interaktif.
- Ini adalah salah satu komponen utama dari teknologi web bersama dengan HTML (Hypertext Markup Language) dan CSS (Cascading Style Sheets).
- JavaScript adalah bahasa pemrograman yang digunakan untuk mengendalikan perilaku halaman web.

# HTML + CSS + Javascript



# ESP32 FILESYSTEM UPLOADER

## ESP32 Filesystem Uploader

- ESP32 berisi Serial Peripheral Interface Flash File System (SPIFFS).
- SPIFFS adalah sistem berkas ringan yang dibuat untuk mikrokontroler dengan chip flash, yang dihubungkan dengan protokol SPI, seperti memori flash ESP32.
- SPIFFS memungkinkan kita mengakses memori flash seperti yang kita lakukan pada sistem berkas normal di komputer, tetapi lebih sederhana dan lebih terbatas.
- Kita dapat membaca, menulis, menutup, dan menghapus file.

## ESP32 Filesystem Uploader

Menggunakan SPIFFS dengan papan ESP32 sangat berguna untuk:

- Membuat file konfigurasi dengan pengaturan;
- Menyimpan data secara permanen;
- Membuat file untuk menyimpan data dalam jumlah kecil alih-alih menggunakan kartu microSD;
- Menyimpan file HTML dan CSS untuk membangun server web;
- Menyimpan gambar, gambar, dan ikon;
- Dan masih banyak lagi.

Dengan SPIFFS, kita dapat menulis HTML dan CSS dalam file terpisah dan menyimpannya di sistem berkas ESP32.



## ESP32 Filesystem Uploader

- Kita dapat membuat, menyimpan, dan menulis file ke sistem berkas ESP32 dengan menulis kode sendiri di Arduino IDE.
- Ini tidak terlalu berguna, karena kita harus mengetikkan konten file kita di sketch Arduino.
- Ada sebuah plugin untuk Arduino IDE yang memungkinkan kita untuk mengunggah file secara langsung ke filesystem ESP32 dari sebuah folder di komputer kita.

## Install ESP32 Filesystem Uploader (windows)

I. Buka halaman <https://github.com/me-no-dev/arduino-esp32fs-plugin/releases/> untuk mendownload zip filenya



Latest release

1.0

278ffa0

Verified

### Release for esptool\_py

me-no-dev released this on Jan 15

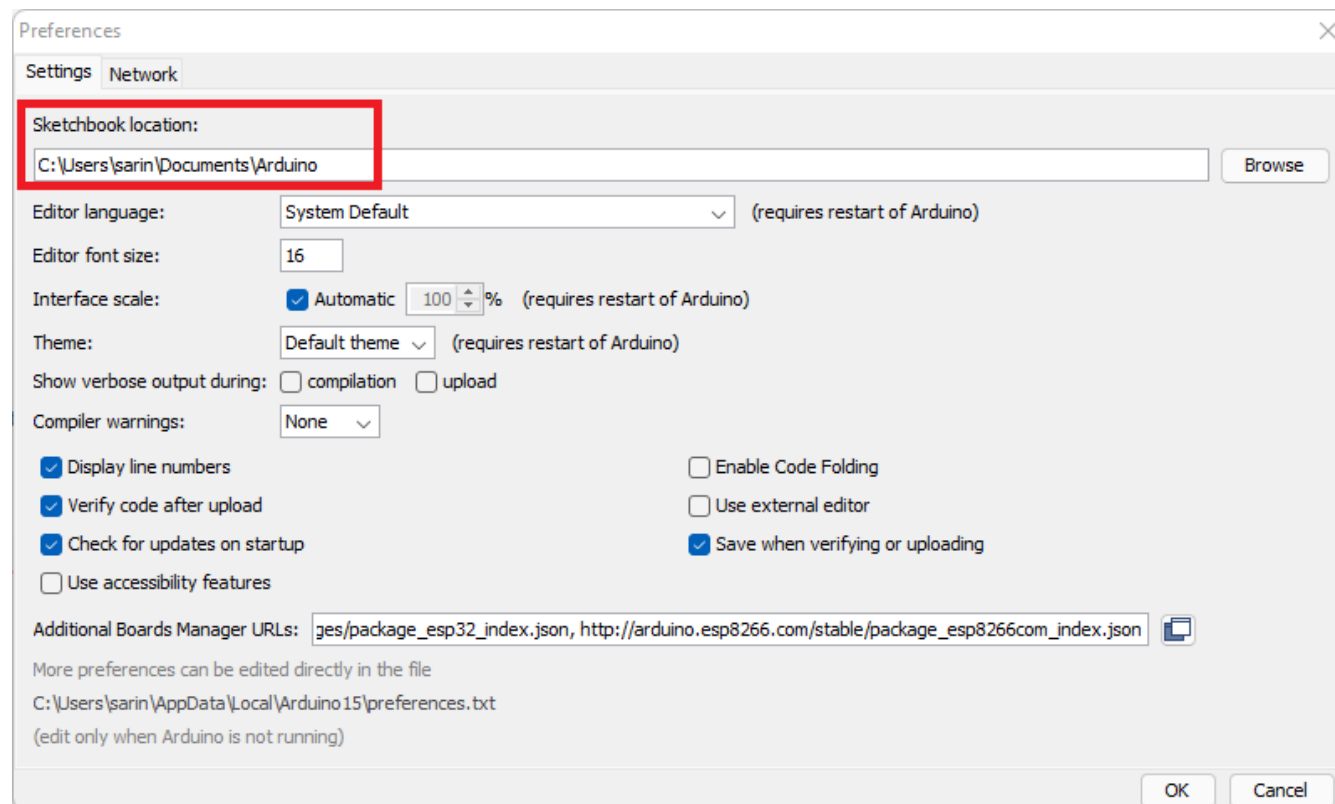
Updates the path to esptool to work with the latest Arduino for ESP32

▼ Assets 3

- ESP32FS-1.0.zip
- Source code (zip)
- Source code (tar.gz)

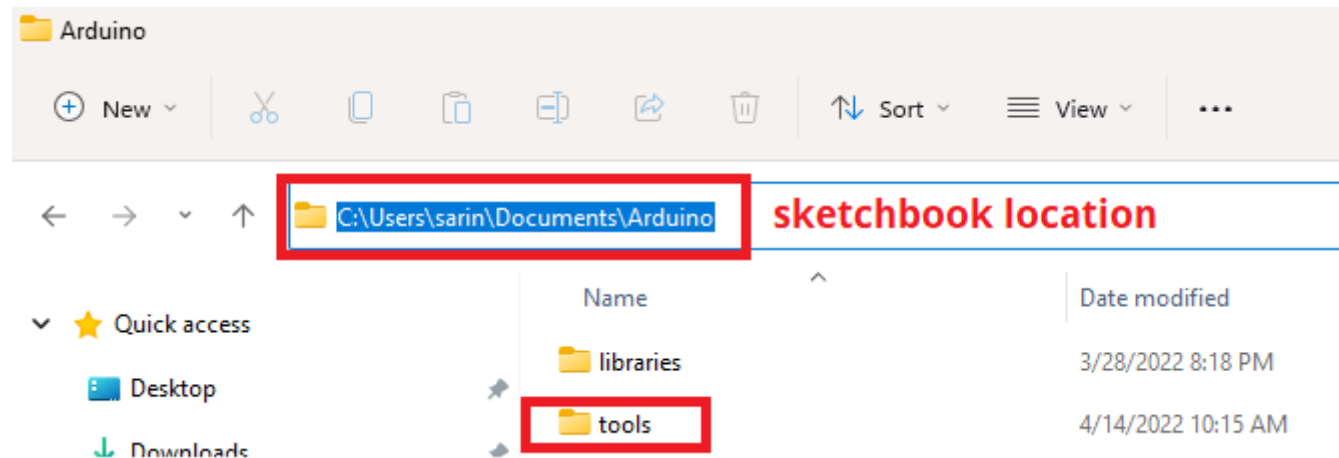
## Install ESP32 Filesystem Uploader (windows)

2. Temukan lokasi Sketchbook kita. Di Arduino IDE kita, buka File > Preferences dan periksa lokasi Sketchbook kita.



## Install ESP32 Filesystem Uploader (windows)

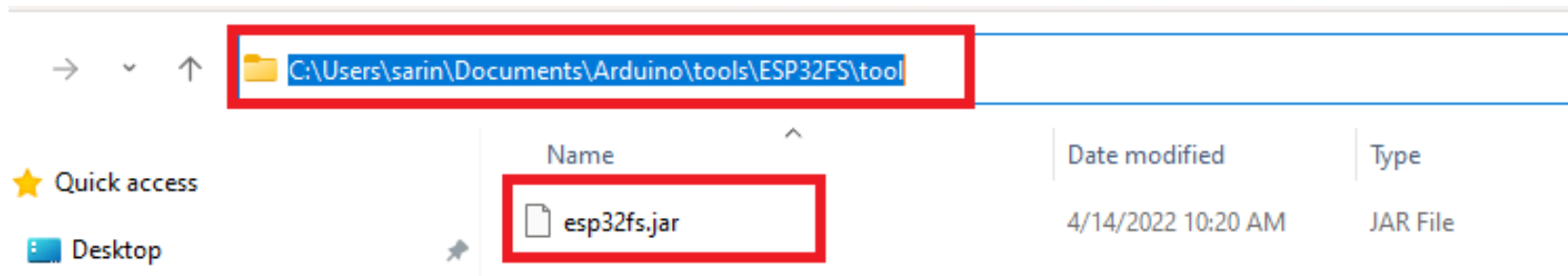
3. Buka lokasi sketchbook, dan buat folder tools.



## Install ESP32 Filesystem Uploader (windows)

4. Unzip folder .zip yang telah diunduh. Buka dan salin folder ESP32FS ke folder tools yang telah kita buat pada langkah sebelumnya. kita seharusnya memiliki struktur folder yang serupa:.

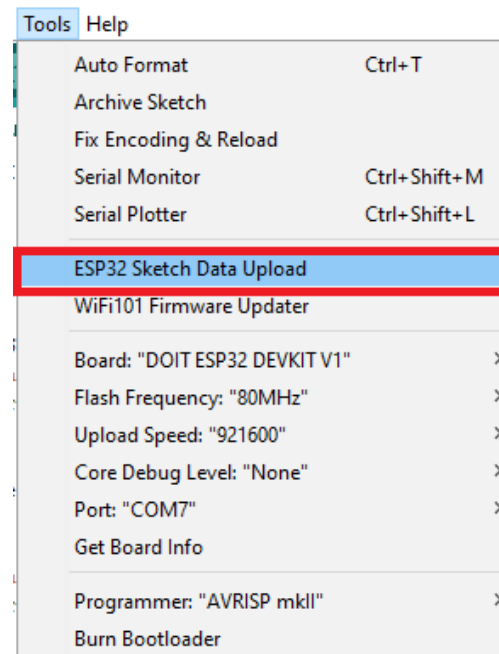
<Sketchbook-location>/tools/ESP32FS/tool/esp32fs.jar



## Install ESP32 Filesystem Uploader (windows)

5) Terakhir, restart Arduino IDE kita.

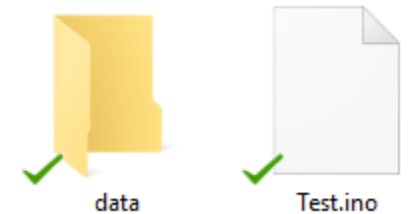
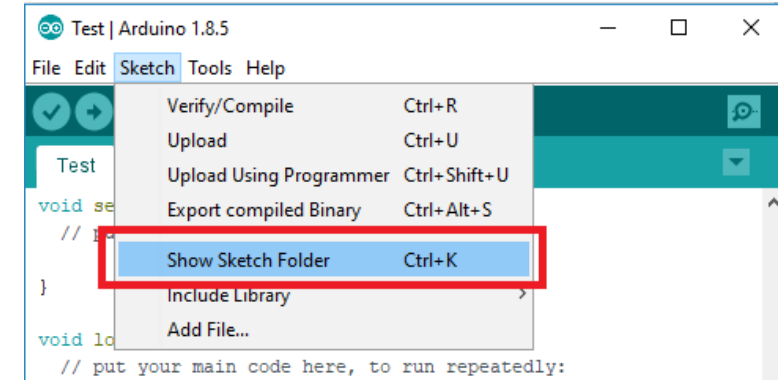
Untuk memeriksa apakah plugin telah berhasil diinstal, buka Arduino IDE kita. Pilih board ESP32 kita, buka Tools dan periksa apakah kita memiliki opsi "ESP32 Sketch Data Upload".



## Mengunggah File menggunakan Filesystem Uploader

Untuk mengunggah file ke filesystem ESP32,:

- 1) Buat sketch Arduino dan simpan. Untuk tujuan demonstrasi, kita bisa menyimpan sketch kosong.
- 2) Kemudian, buka folder sketch. kita bisa pergi ke Sketch > Show Sketch Folder. Folder tempat sketsa kita disimpan akan terbuka.
- 3) Di dalam folder tersebut, buatlah sebuah folder baru bernama data.
- 4) Di dalam folder data adalah tempat kita harus meletakkan file yang ingin kita simpan ke dalam filesystem ESP32.
- 5) Kemudian, untuk mengunggah file, di Arduino IDE, kita hanya perlu masuk ke Tools > ESP32 Sketch Data Upload.



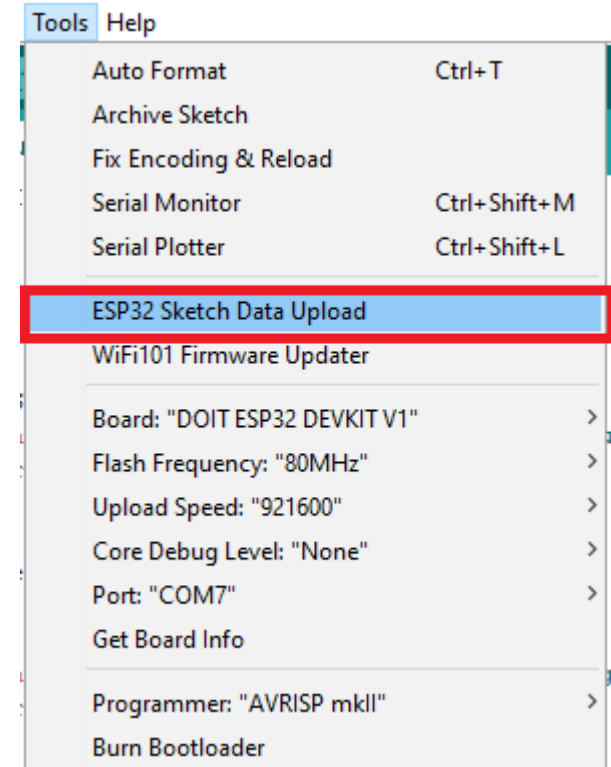
## Mengunggah File menggunakan Filesystem Uploader

5) Kemudian, untuk mengunggah file, di Arduino IDE, kita hanya perlu masuk ke Tools > ESP32 Sketch Data Upload.

6) File berhasil diunggah ke filesystem ESP32 ketika kita melihat pesan "SPIFFS Image Uploaded".



The screenshot shows the Arduino IDE serial monitor interface. At the top, a teal status bar displays "SPIFFS Image Uploaded" in white text, which is highlighted with a red rectangular border. Below this, the main serial output area has a black background with white text showing the upload process: "Hash of data verified.", "Leaving...", and "Hard resetting...". At the bottom of the window, a teal footer bar contains the text "DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on COM7".





# ESP32 WEB SERVER USING SPIFFS (SPI FLASH FILE SYSTEM)

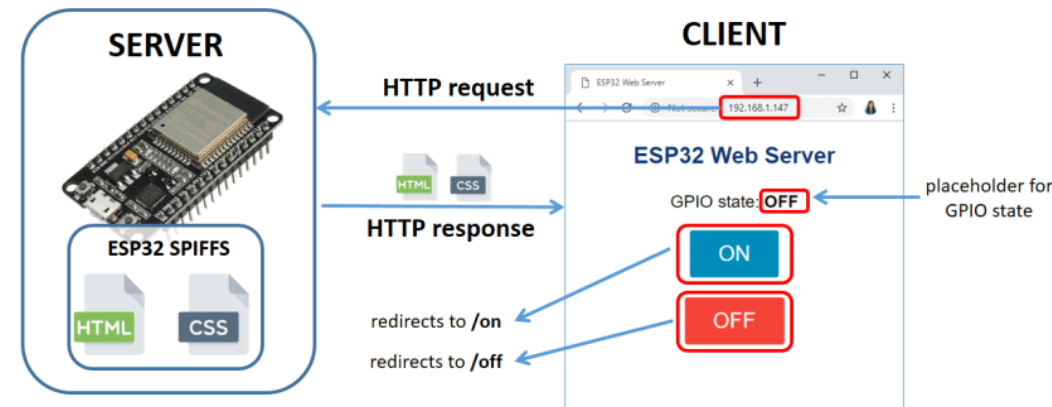
# ESP32 WEB SERVER USING SPIFFS

- Server web yang akan kita buat mengontrol LED yang terhubung ke ESP32 GPIO 2. Ini adalah LED on-board ESP32. kita dapat mengontrol GPIO lainnya;
- Halaman server web menampilkan dua tombol: ON dan OFF - untuk menghidupkan dan mematikan GPIO 2;
- Halaman server web juga menunjukkan status GPIO saat ini.
- Gambar berikut ini menunjukkan diagram yang disederhanakan untuk menunjukkan cara kerja semuanya.



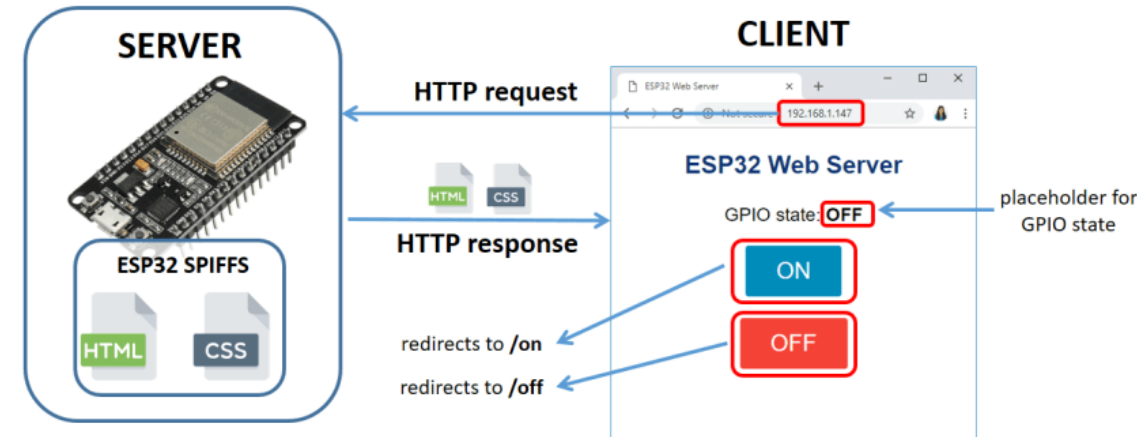
# ESP32 WEB SERVER USING SPIFFS

- ESP32 menjalankan kode server web berdasarkan library ESPAsyncWebServer;
- File HTML dan CSS disimpan di ESP32 SPIFFS ( Serial Peripheral Interface Flash File System);
- Ketika kita membuat permintaan pada URL tertentu menggunakan browser kita, ESP32 merespons dengan file yang diminta;
- Ketika kita mengklik tombol ON, kita akan diarahkan ke URL root yang diikuti dengan /on dan LED akan menyala;



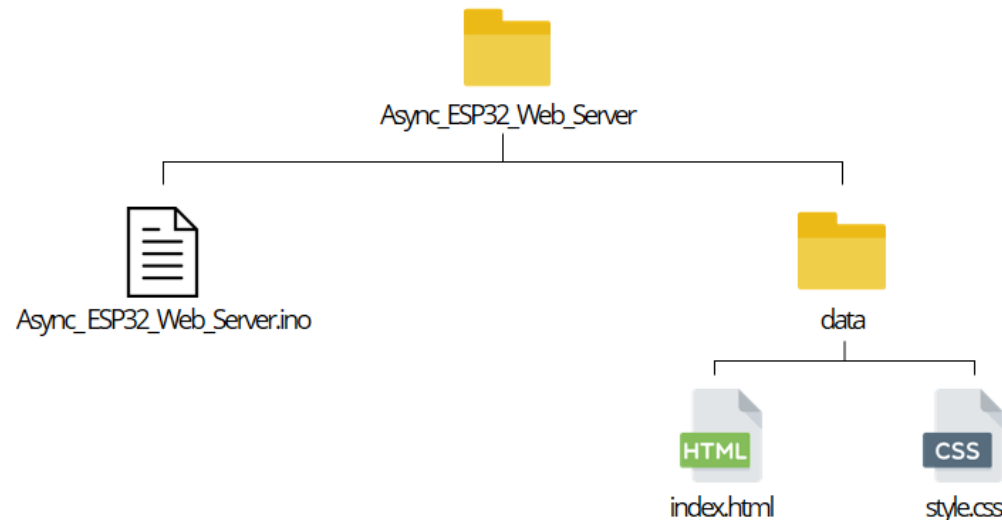
# ESP32 WEB SERVER USING SPIFFS

- Ketika kita mengklik tombol OFF, kita akan diarahkan ke URL root yang diikuti dengan /off dan LED dimatikan;
- Pada halaman web, terdapat placeholder untuk status GPIO. Placeholder untuk status GPIO ditulis langsung di file HTML di antara tanda %, misalnya %STATE%.



# MENGATUR FILE-FILE

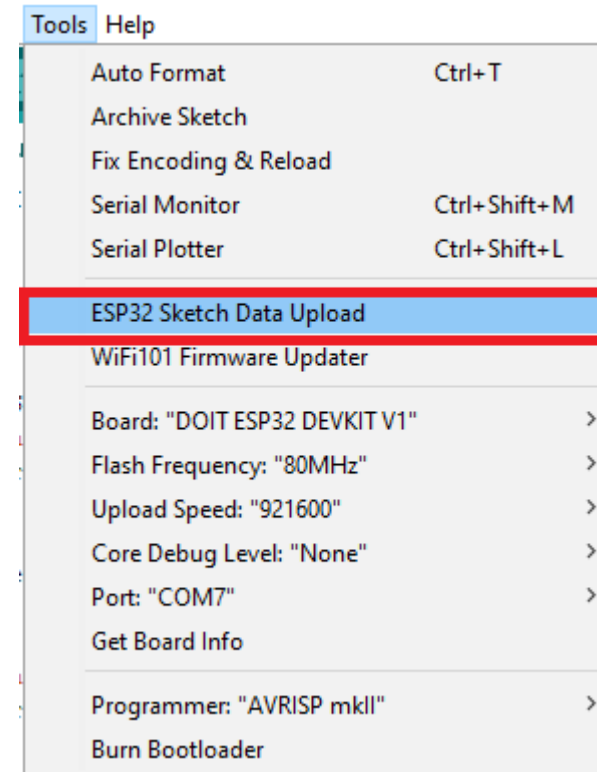
- Untuk membangun server web, kita memerlukan tiga file yang berbeda. Sketsa Arduino, file HTML, dan file CSS.
- File HTML dan CSS harus disimpan di dalam folder bernama data di dalam folder sketsa Arduino, seperti yang ditunjukkan di bawah ini:



# COBA ESP32 WEB SERVER USING SPIFFS

- Ke [https://bit.ly/t4b\\_iot\\_code](https://bit.ly/t4b_iot_code)
- Klik folder day 4
- Download folder ESP32\_Async\_Web\_Server
- Ubah hal2 yang diperlukan

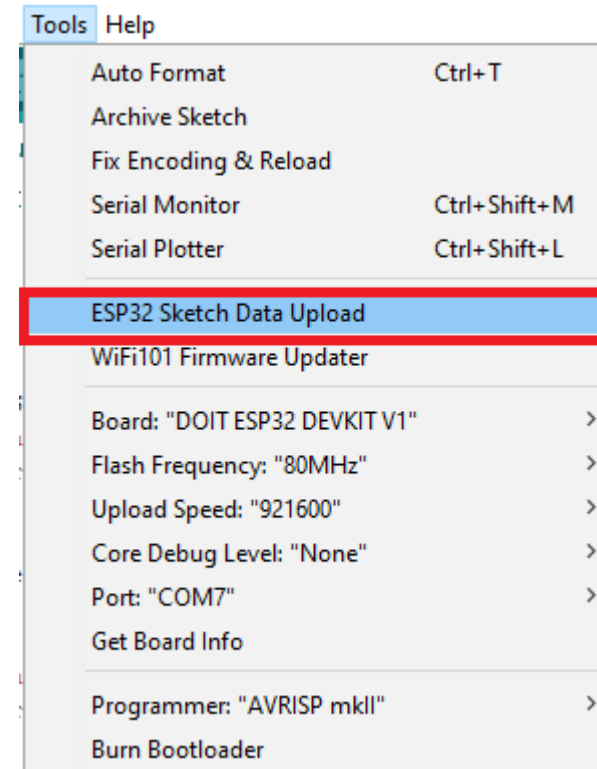
```
11 // Replace with your network credentials
12 const char* ssid = "ganti";
13 const char* password = "ganti";
14
```



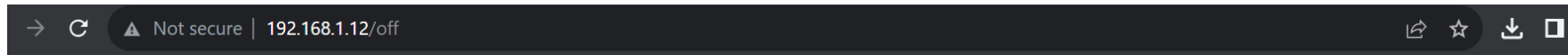
# COBA ESP32 WEB SERVER USING SPIFFS

- Upload kodenya
- Setelah mengunggah kode, kita perlu mengunggah file. Buka Tools > ESP32 Data Sketch Upload dan tunggu hingga file diunggah.
- Klik reset/en pada esp32, lalu buka serial monitor dengan baud rate 115200
- Setelah connect wifi, akan terdapat ip address dari esp32 tersebut, buka ip tersebut melalui browser

```
11 // Replace with your network credentials
12 const char* ssid = "ganti";
13 const char* password = "ganti";
14
```



# HASIL COBA ESP32 WEB SERVER USING SPIFFS



## ESP32 Web Server

GPIO state: **OFF**

ON

OFF



# PEMBAHASAN KODE ESP32 WEB SERVER USING SPIFFS

[https://github.com/pidinplusplusultra/train4best\\_PENS\\_September/tree/main/Day%204/ESP32\\_Async\\_Web\\_Server](https://github.com/pidinplusplusultra/train4best_PENS_September/tree/main/Day%204/ESP32_Async_Web_Server)

# TUGAS ESP32 WEB SERVER USING SPIFFS + DHT11

Tambahkan pembacaan sensor dht11 pada tampilan web, seperti berikut

## ESP32 Web Server

GPIO state: **OFF**

Temperature: **23.80 C**

Humidity: **66.00 %**

ON

OFF



*Train4best*

THANK YOU