



Train4best

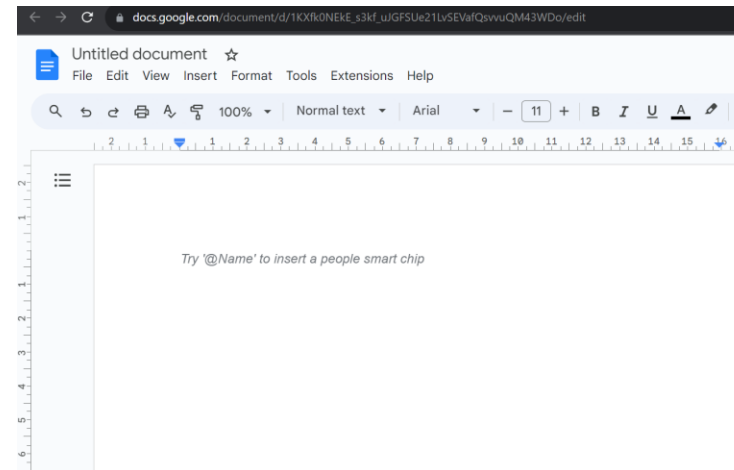
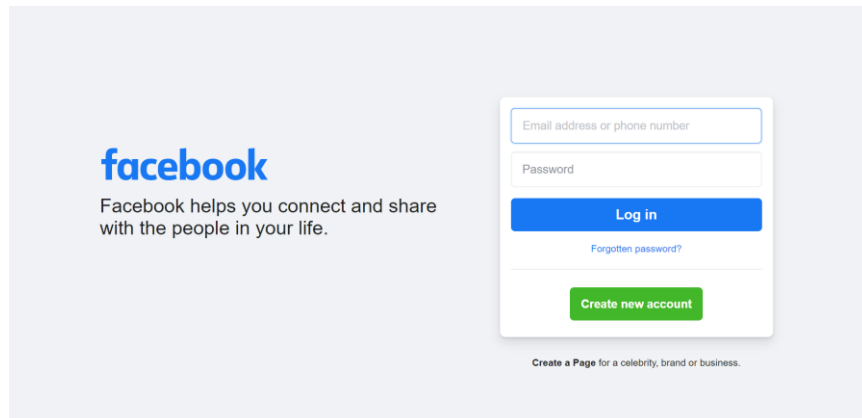
PEREKAYASA PLATFORM IOT

CHAPTER #2: PROTOKOL KOMUNIKASI IOT PADA APLIKASI BERBASIS WEB

APLIKASI BERBASIS WEB

APLIKASI BERBASIS WEB

- Aplikasi berbasis web adalah aplikasi yang dijalankan pada browser web.
- Aplikasi ini dapat berjalan di platform apa pun selama kita memiliki browser dan koneksi internet. Itu sebabnya disebut juga aplikasi berbasis browser.



KEUNTUNGAN WEB APP

- Platform-agnostik: Aplikasi web tidak dibangun untuk platform tertentu. Aplikasi web dapat dibangun untuk semua platform selama aplikasi web dapat dijalankan di browser web.
- Biaya awal rendah: Karena aplikasi web tidak spesifik untuk suatu platform, aplikasi web tidak memerlukan banyak waktu atau pengembang untuk dibangun. Ini berarti aplikasi web akan lebih murah biayanya pada awalnya dibandingkan dengan jenis aplikasi lainnya.
- Mudah untuk dikelola: Aplikasi web menggunakan bahasa pemrograman yang umum digunakan di berbagai platform. Ini membuat aplikasi web relatif mudah tidak hanya untuk dibangun tetapi juga untuk dikelola.

KEUNTUNGAN WEB APP

- Tidak ada pasar aplikasi: Karena aplikasi web tidak mengikuti protokol sistem operasi standar, aplikasi web tidak memerlukan persetujuan dari toko aplikasi. Ini berarti aplikasi web dapat dirilis dalam format dan waktu yang dipilih oleh para pengembang.
- Pembaruan otomatis: Karena aplikasi web tidak diunduh dari toko aplikasi, pengguna tidak perlu pergi ke toko aplikasi untuk mengelola pembaruan mereka. Pengguna selalu melihat versi yang paling mutakhir saat mereka membuka aplikasi web.

KEKURANGAN WEB APP

- Memerlukan koneksi internet: Karena aplikasi web diakses melalui browser, pengguna perlu terhubung ke internet untuk menggunakannya.
- Keterbatasan dalam ditemukan: Karena aplikasi web tidak terdaftar di toko aplikasi, aplikasi web tidak dipasarkan atau dipromosikan di pasar tersebut.
- Fungsionalitas yang lebih sedikit: Karena aplikasi web tidak dirancang untuk sistem operasi tertentu, aplikasi web tidak memiliki akses ke fitur-fitur sistem tertentu.

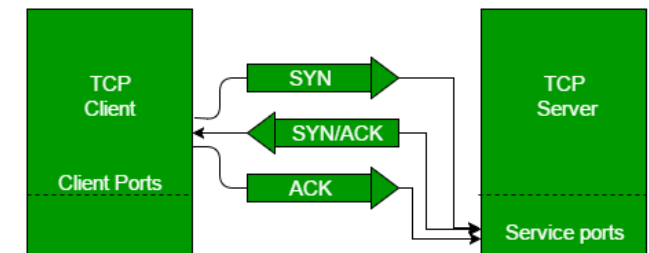
PROTOKOL

PROTOKOL JARINGAN

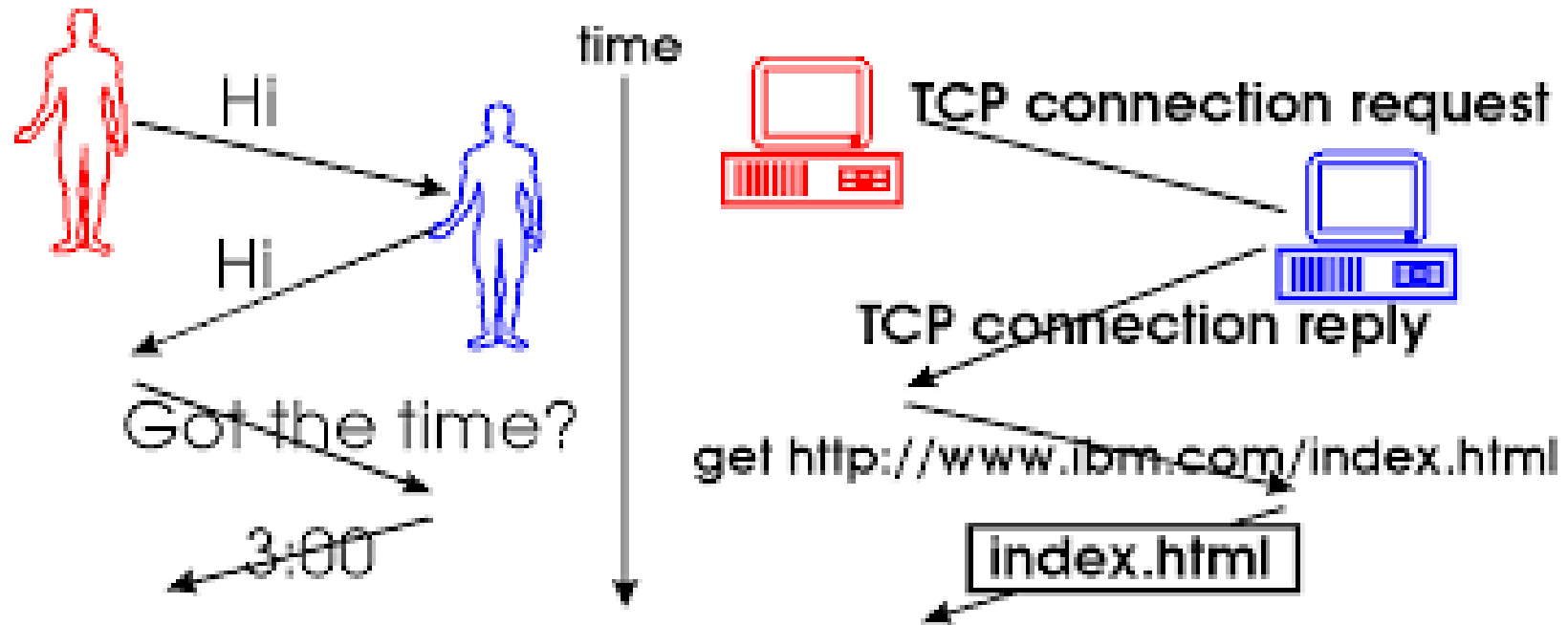
Protokol jaringan adalah sebuah aturan atau standar yang mengatur atau mengizinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih titik komputer.

Tugas yang biasanya dilakukan oleh sebuah protokol dalam sebuah jaringan diantaranya adalah :

- Melakukan deteksi adanya koneksi fisik atau ada tidaknya komputer / mesin lainnya.
- Melakukan metode “jabat-tangan” (handshaking).
- Negosiasi berbagai macam karakteristik hubungan.



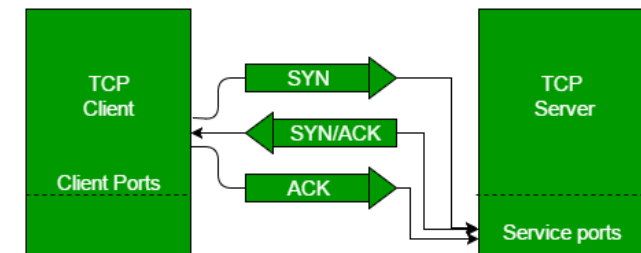
PROTOKOL JARINGAN



PROTOKOL JARINGAN

Tugas yang biasanya dilakukan oleh sebuah protokol dalam sebuah jaringan diantaranya adalah :

- Bagaimana mengawali dan mengakhiri suatu pesan.
- Bagaimana format pesan yang digunakan.
- Yang harus dilakukan saat terjadi kerusakan pesan atau pesan yang tidak sempurna.
- Mendeteksi rugi-rugi pada hubungan jaringan dan langkah-langkah yang dilakukan selanjutnya.
- Mengakhiri suatu koneksi.



OSI LAYER

OSI (Open Systems Interconnection) Layer adalah model referensi yang digunakan dalam dunia komputer dan jaringan untuk menggambarkan cara komunikasi antar perangkat dalam jaringan komputer.

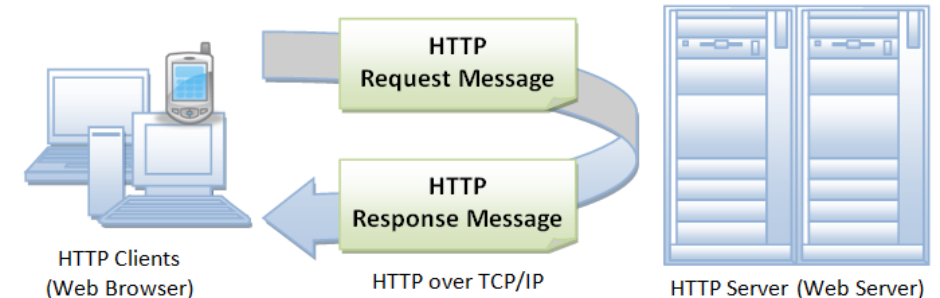
Layer	Name	Protocols
Layer 7	Application	SMTP, HTTP, FTP, POP3, SNMP
Layer 6	Presentation	MPEG, ASCH, SSL, TLS
Layer 5	Session	NetBIOS, SAP
Layer 4	Transport	TCP, UDP
Layer 3	Network	IPV5, IPV6, ICMP, IPSEC, ARP, MPLS.
Layer 2	Data Link	RAPA, PPP, Frame Relay, ATM, Fiber Cable, etc.
Layer 1	Physical	RS232, 100BaseTX, ISDN, 11.

Ref: <https://www.guru99.com/layers-of-osi-model.html>

HTTP

HTTP

- Hypertext Transfer Protocol (HTTP) dirancang untuk memungkinkan komunikasi antara klien dan server.
- HTTP berfungsi sebagai protokol permintaan-respon antara klien dan server.
- Contoh: Sebuah klien (browser) mengirimkan permintaan HTTP ke server; kemudian server mengirimkan respon ke klien. Respon berisi informasi status tentang permintaan tersebut dan juga dapat berisi konten yang diminta.

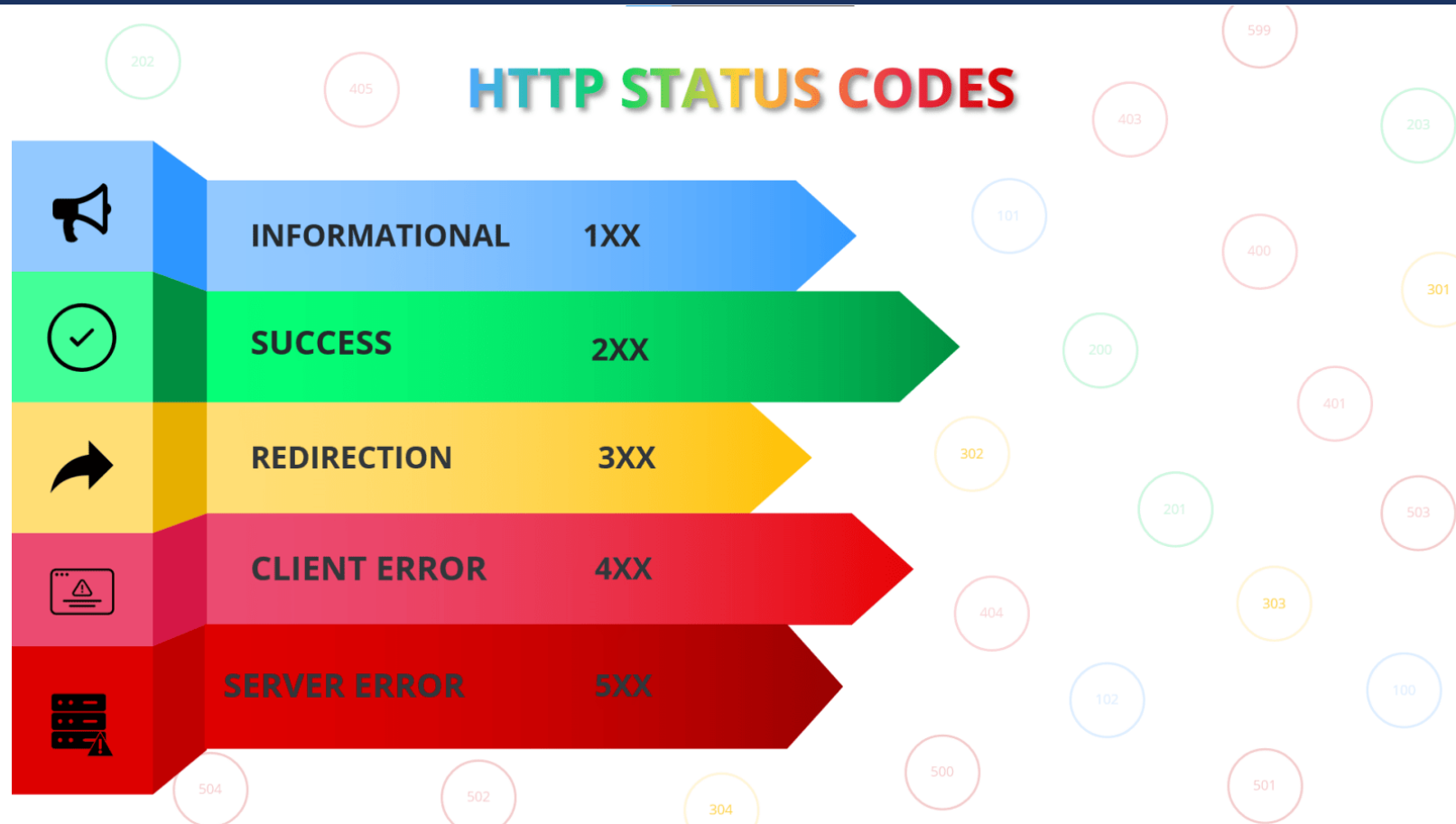


POINT-POINT PENTING HTTP

- Client-Server Communication
- Stateless
- Methods
- Uniform Resource Locators (URLs)
- Status Codes
- Ada versi aman dari HTTP yang disebut HTTPS (Hypertext Transfer Protocol Secure)

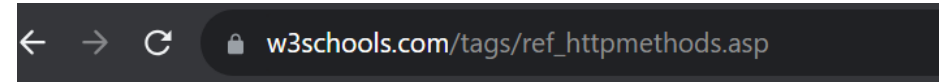


HTTP STATUS CODE



HTTP METHODS

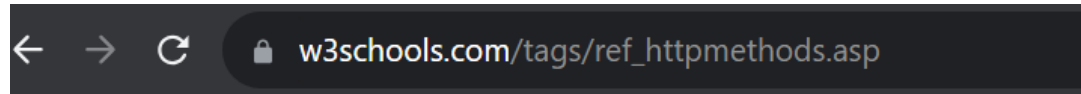
- GET
- POST
- PUT
- HEAD
- DELETE
- PATCH



Ref: https://www.w3schools.com/tags/ref_httpmethods.asp

HTTP METHODS

- OPTIONS
- CONNECT
- TRACE

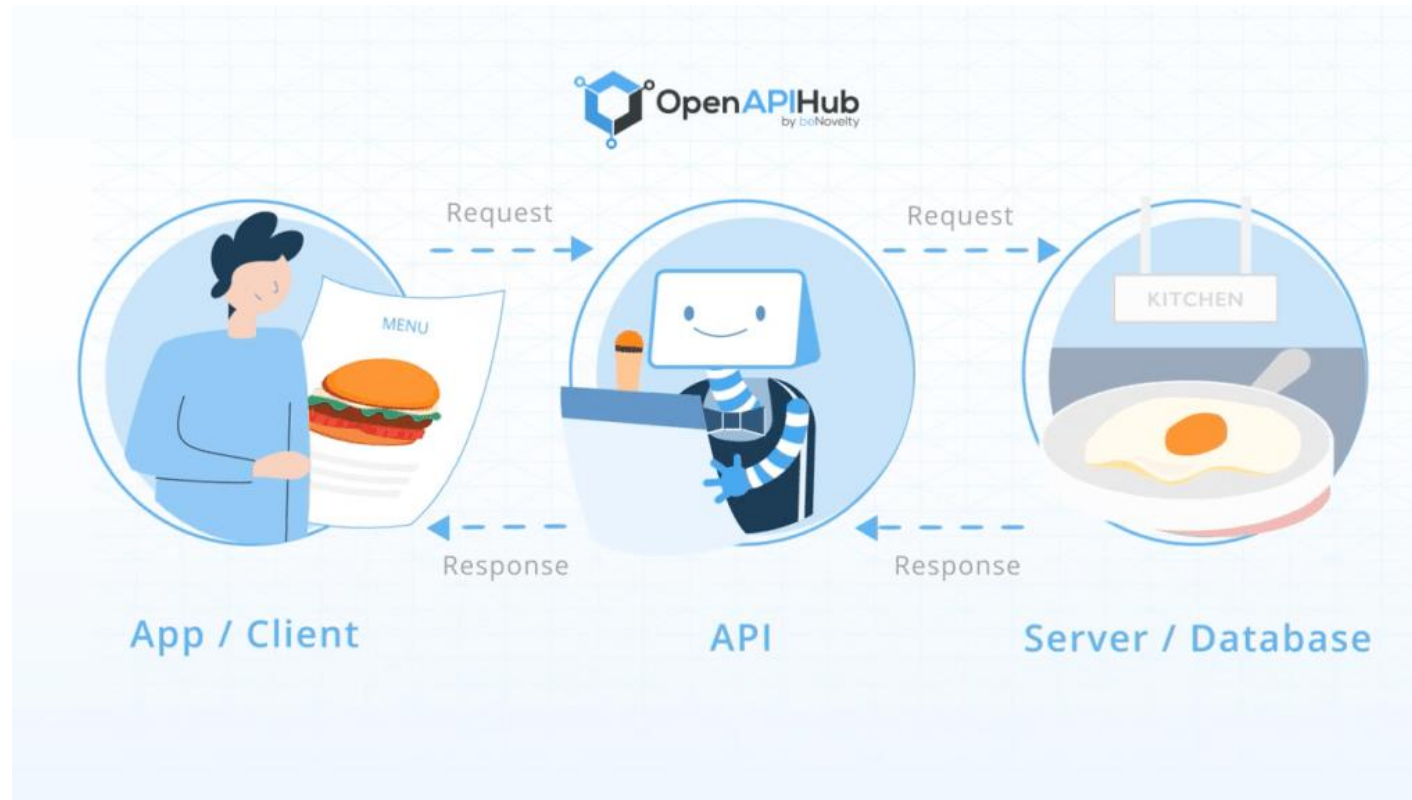


Dua metode HTTP yang paling umum adalah: GET dan POST. GET digunakan untuk meminta data dari sumber daya tertentu. POST digunakan untuk mengirim data ke server untuk membuat/memperbarui sumber daya.


API

- API (Application Programming Interface) adalah seperangkat aturan dan protokol yang memungkinkan berbagai aplikasi berkomunikasi dan berinteraksi satu sama lain.
- Dalam konteks API, HTTP sering digunakan sebagai protokol komunikasi untuk mengakses dan berinteraksi dengan berbagai layanan melalui permintaan HTTP ke endpoint API yang spesifik.

ANALOGI API



CONTOH API

 Overview Documentation API reference Examples

Log in

CTRL K

GET STARTED

- Introduction
- Overview
- Key concepts
- Next steps
- Quickstart
- Libraries
- Models
- Deprecations
- Tutorials
- Policies

GUIDES

- GPT
- GPT best practices
- Image generation
- Fine-tuning

Introduction

Overview

Looking for ChatGPT? Head to chat.openai.com.

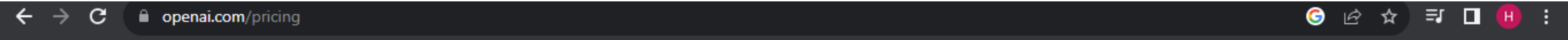
The OpenAI API can be applied to virtually any task that requires understanding or generating natural language and code. The OpenAI API can also be used to generate and edit images or convert speech into text. We offer a range of [models](#) with different capabilities and price points, as well as the ability to [fine-tune](#) custom models.

Resources

- Experiment in the [playground](#)
- Read the [API reference](#)
- Visit the [help center](#)
- View the current API [status](#)
- Check out the [OpenAI Developer Forum](#)
- Learn about our [usage policies](#)



CONTOH API



Research ▾ API ▾ ChatGPT ▾ Safety Company ▾

[Search](#) [Log in ↗](#)

[Get started ↗](#)

GPT-4

With broad general knowledge and domain expertise, GPT-4 can follow complex instructions in natural language and solve difficult problems with accuracy.

[Learn about GPT-4](#)

Model	Input	Output
8K context	\$0.03 / 1K tokens	\$0.06 / 1K tokens
32K context	\$0.06 / 1K tokens	\$0.12 / 1K tokens

CONTOH API

← → ↻ 🔒 shopee.co.id/robots.txt

```
Disallow: /mall/just-for-you/
Disallow: /mall/*-cat.
Disallow: /from_same_shop/
Disallow: /you_may_also_like/
Disallow: *-i.*/similar?from=flash_sale
Disallow: /find_similar_products/
Disallow: /top_products
Disallow: /search*searchPrefill
Disallow: /index.html
Disallow: /search*hashtag=
Disallow: /search?shop=
Disallow: /buyer/*/rating/
Disallow: *-cat.*?category=
Disallow: /shop/*/search
Disallow: /shop/*/details
Disallow: /search/?
Disallow: /search?brands=
Disallow: /search?originalCategoryId=
Disallow: /addon-deal-selection/
Disallow: /bundle-deal/
Disallow: /*?sp_atk=
Disallow: *_classic_=1
Disallow: /search_user
Disallow: /inspirasi-shopee/?s=
```

Agent:*

Crawl-delay:1

```
Disallow: /cart/
Disallow: /checkout/
Disallow: /buyer/login/otp
Disallow: /user/
Disallow: /me/
Disallow: /order/
Disallow: /daily_discover/
Disallow: /mall/just-for-you/
Disallow: /mall/*-cat.
Disallow: /from_same_shop/
Disallow: /you_may_also_like/
Disallow: *-i.*/similar
Disallow: /find_similar_products/
Disallow: /top_products
Disallow: /search*searchPrefill
Disallow: /index.html
```

 Shopee Open Platform

Documentation ▾

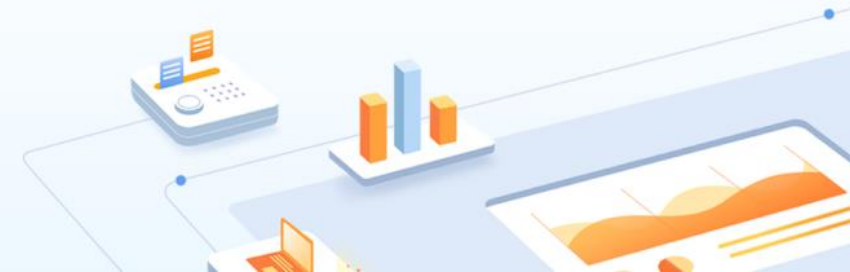
Support Center ▾

Announcement

Consol

Empowering Merchants To U
New Opportunities

Get Access (Now)



PORT

- Dalam jaringan, "port" mengacu pada titik akhir logis atau virtual endpoint untuk komunikasi dalam jaringan komputer.
- Port digunakan untuk membedakan antara berbagai layanan atau proses yang berjalan pada perangkat atau host yang sama, sehingga memungkinkan beberapa layanan untuk hidup berdampingan dan berkomunikasi melalui satu alamat IP.
- Port yang terkenal (0-1023): Dicalangkan untuk layanan yang umum digunakan seperti HTTP (port 80), FTP (port 21), dan SMTP (port 25).
- Port yang terdaftar (1024-49151): Digunakan untuk aplikasi yang tidak umum tetapi masih mengikuti standar. Port ini dapat didaftarkan pada Internet Assigned Numbers Authority (IANA).
- Port dinamis atau sementara (49152-65535): Digunakan untuk sambungan sementara dan dipilih oleh sistem operasi ketika klien memulai sambungan.



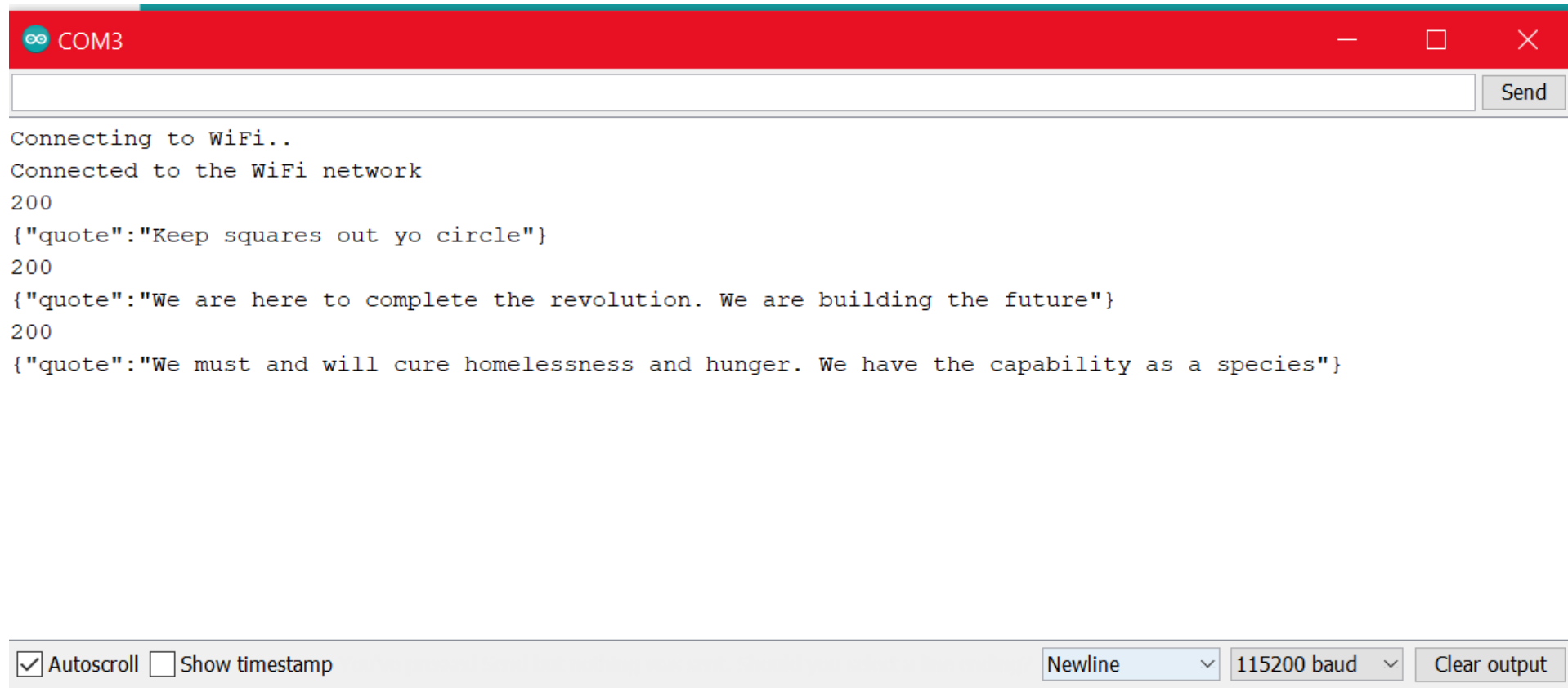
COBA ESP32 HTTP GET

- Ke https://bit.ly/t4b_iot_code
- Klik folder day 2
- Download file `http_get_tutorialx.ino`
- Ubah hal-hal yang diperlukan
- Upload filenya

```
4    const char* ssid = "ganti ssid wifi";  
5    const char* password = "ganti password wifi";
```

```
28    http.begin("https://www.google.com/"); //Specify the URL
```


ESP32 HTTP GET



```
COM3
Connecting to WiFi..
Connected to the WiFi network
200
{"quote":"Keep squares out yo circle"}
200
{"quote":"We are here to complete the revolution. We are building the future"}
200
{"quote":"We must and will cure homelessness and hunger. We have the capability as a species"}
```

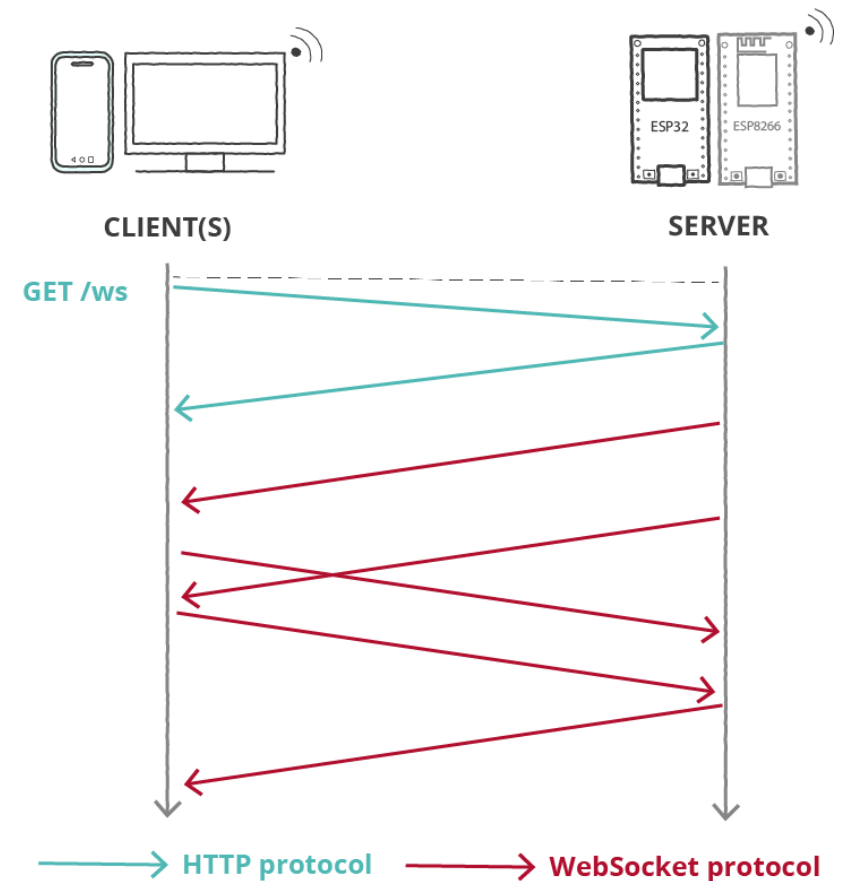
☒ Autoscroll ☐ Show timestamp Newline 115200 baud Clear output

Ref: <https://techtutorialsx.com/2017/05/19/esp32-http-get-requests/>

WEBSOCKET

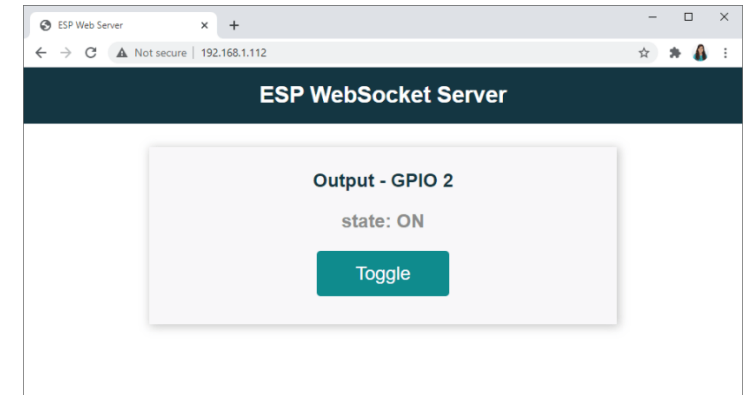
WEBSOCKET

- WebSocket adalah koneksi persisten antara klien dan server yang memungkinkan komunikasi dua arah antara kedua pihak menggunakan koneksi TCP.
- Kita dapat mengirim data dari klien ke server dan dari server ke klien kapan saja.
- Klien membuat koneksi WebSocket dengan server melalui proses yang dikenal handshake WebSocket.
- Handshake dimulai dengan permintaan/respons HTTP, memungkinkan server menangani koneksi HTTP serta koneksi WebSocket pada port yang sama.
- Setelah koneksi dibuat, klien dan server dapat mengirim data WebSocket dalam mode dupleks penuh.



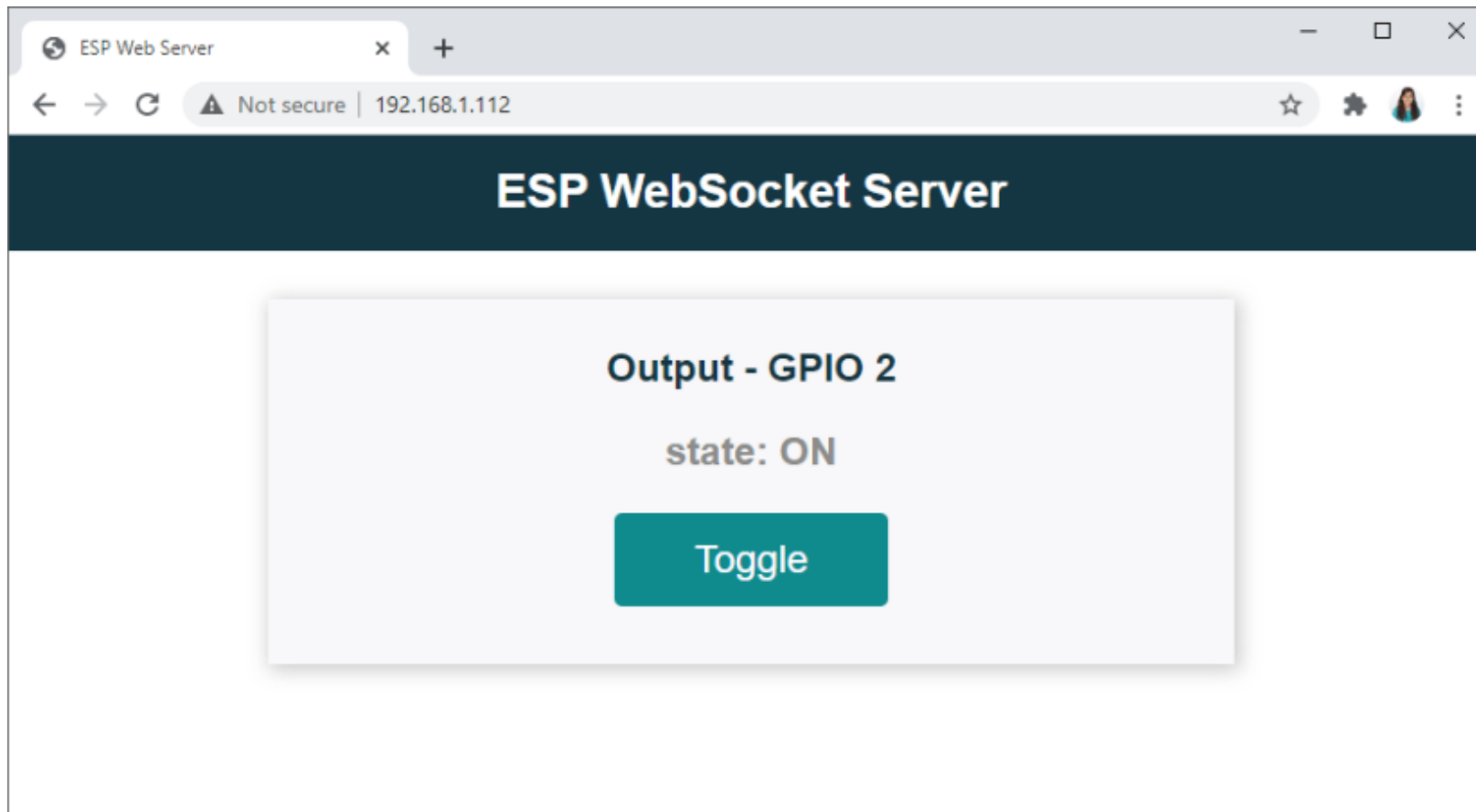
WEBSOCKET + ESP32

- Server web ESP32 menampilkan halaman web dengan tombol untuk mengubah status GPIO 2;
- Untuk mempermudah, kita mengontrol GPIO 2 – LED terpasang. Kita dapat menggunakan contoh ini untuk mengontrol GPIO lainnya;
- Antarmuka menunjukkan status GPIO saat ini. Setiap kali terjadi perubahan pada status GPIO, antarmuka diperbarui secara instan;
- Jika kita membuka beberapa tab browser web di perangkat yang sama atau di perangkat berbeda, semuanya akan diperbarui secara bersamaan.



Ref: <https://randomnerdtutorials.com/esp32-websocket-server-arduino/>

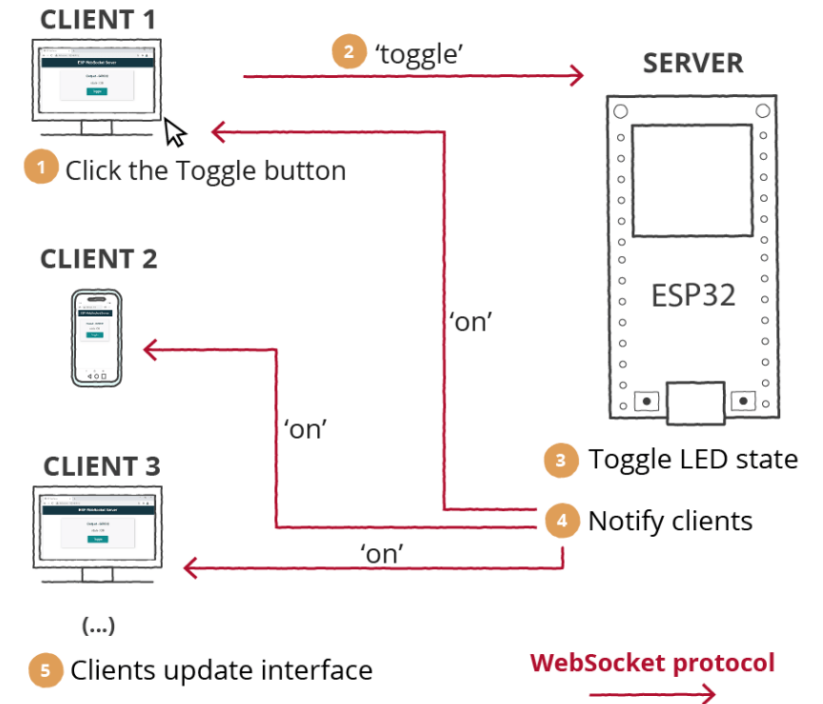
WEBSOCKET + ESP32



WEBSOCKET + ESP32

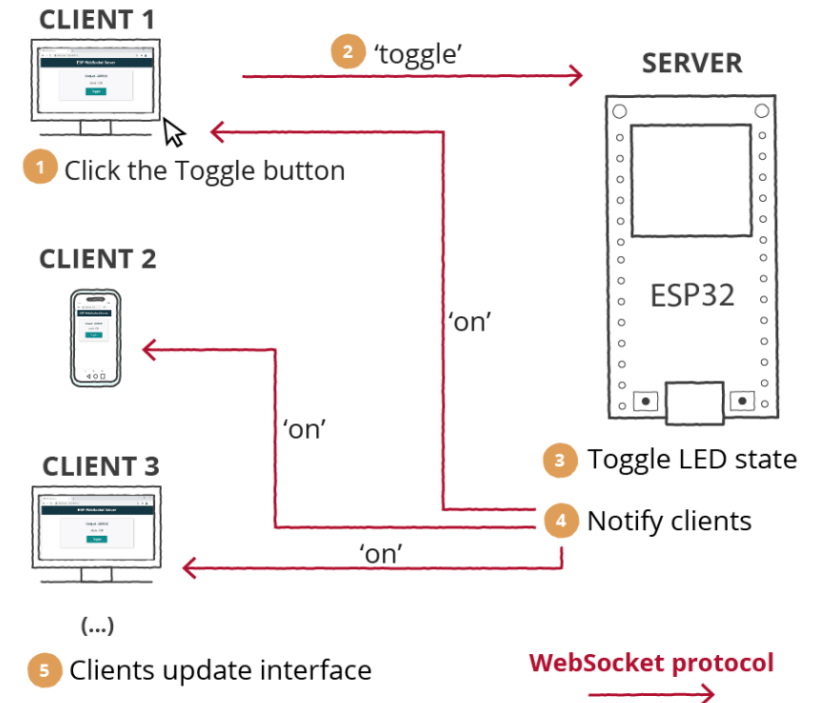
Berikut adalah apa yang terjadi ketika kita mengklik tombol "Toggle":

- Klik tombol "Toggle";
- Klien (browser kita) mengirimkan data melalui protokol WebSocket dengan pesan "toggle";
- ESP32 (server) menerima pesan ini, sehingga ia tahu bahwa ia harus mengalihkan keadaan LED. Jika LED sebelumnya mati, nyalakan;
- Kemudian, server mengirimkan data dengan keadaan LED baru ke semua klien juga melalui protokol WebSocket;
- Klien-klien menerima pesan dan memperbarui keadaan LED pada halaman web sesuai dengan itu. Ini memungkinkan kita untuk memperbarui semua klien hampir seketika ketika terjadi perubahan.



WEBSOCKET + ESP32

- Kemudian, server mengirimkan data dengan keadaan LED baru ke semua klien juga melalui protokol WebSocket;
- Klien-klien menerima pesan dan memperbarui keadaan LED pada halaman web sesuai dengan itu. Ini memungkinkan kita untuk memperbarui semua klien hampir seketika ketika terjadi perubahan.



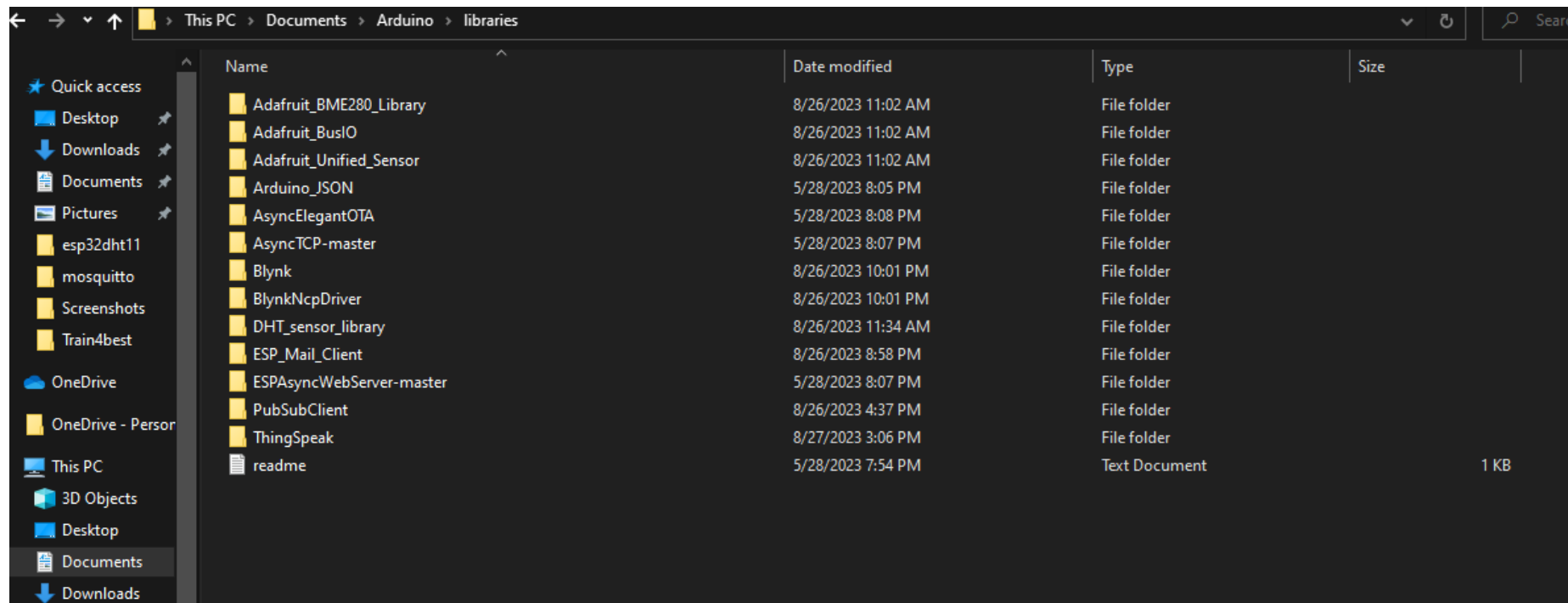
COBA WEBSOCKET + ESP32

- Untuk membangun web server, kita akan menggunakan ESPAsyncWebServer library. Library ini memerlukan library AsyncTCP agar dapat berfungsi dengan baik.
- Download library dengan membuka link
<https://github.com/me-no-dev/ESPAsyncWebServer/archive/master.zip>
- Download library dengan membuka link
<https://github.com/me-no-dev/AsyncTCP/archive/master.zip>
- Library ini tidak tersedia untuk diinstal melalui Arduino Library Manager, jadi kita perlu menyalin file library ke folder Arduino Installation Libraries.
- Atau, di Arduino IDE kita, kita dapat pergi Sketch > Include Library > Add .zip Library dan pilih Library yang baru saja kita unduh.

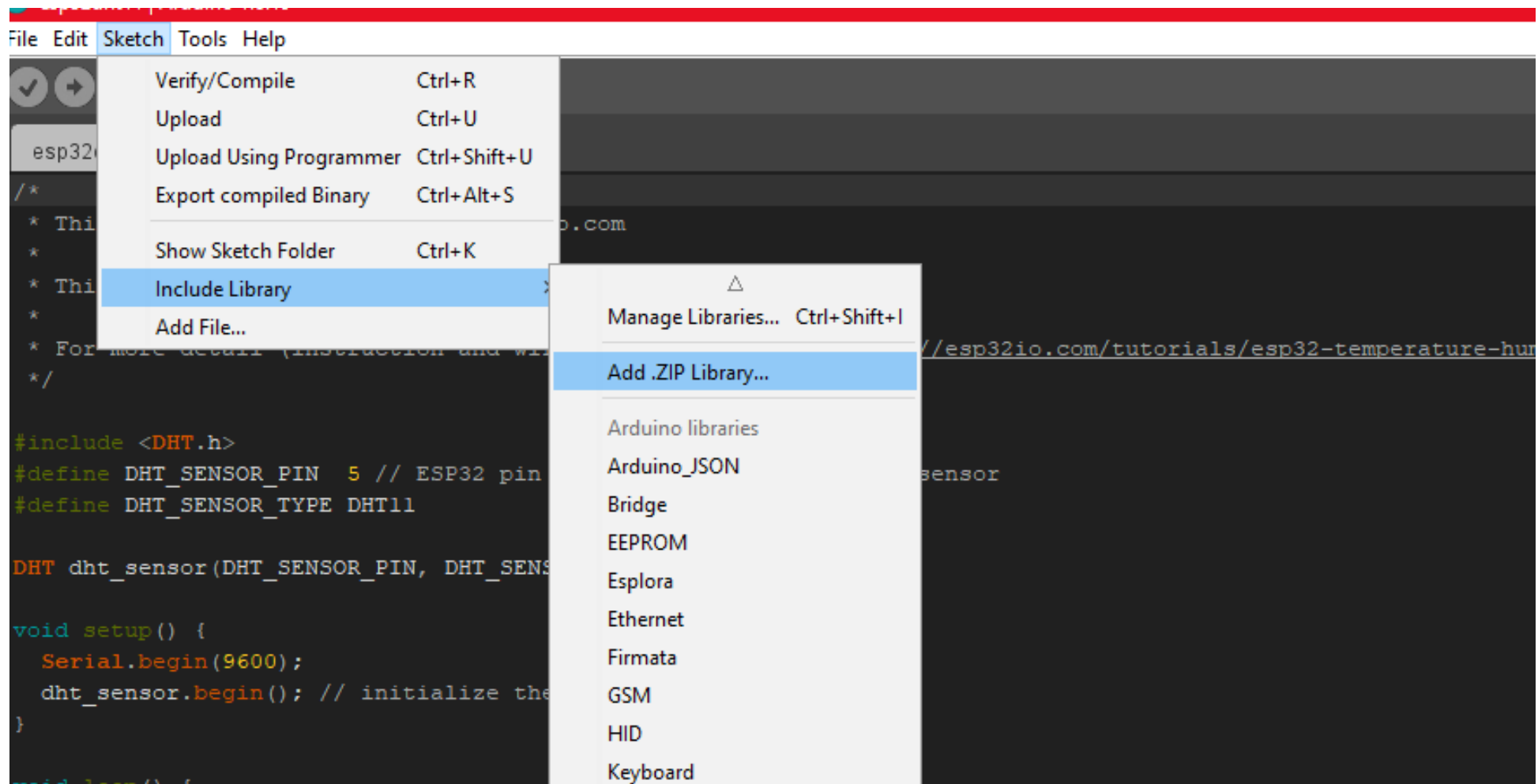
Ref: <https://randomnerdtutorials.com/esp32-websocket-server-arduino/>

CARA I

- Extract folder ke This PC > Documents > Arduino > Libraries



CARA 2



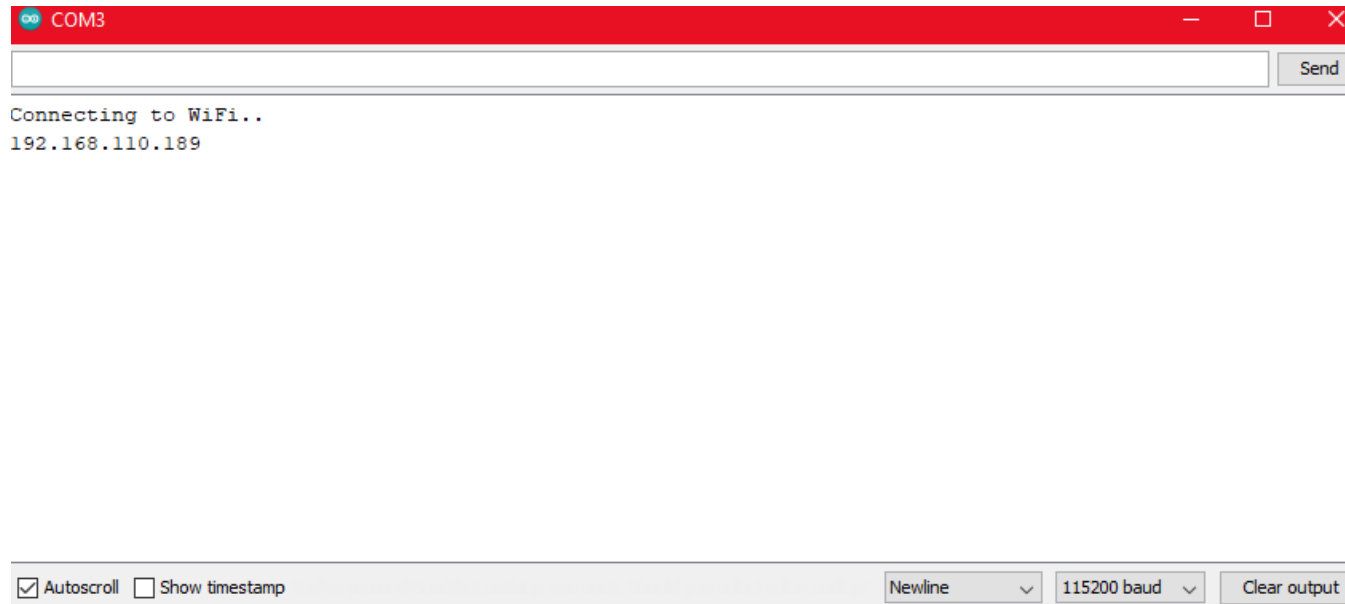
COBA WEBSOCKET + ESP32

- Ke https://bit.ly/t4b_iot_code
- Klik folder day 2
- Download file esp32_websocket.ino
- Ubah hal-hal yang diperlukan
- Upload filenya

```
// Replace with your network credentials  
const char* ssid = "REPLACE_WITH_YOUR_SSID";  
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

COBA WEBSOCKET + ESP32

- Setelah mengunggah kode, buka Serial Monitor pada kecepatan baud 115200 dan tekan tombol EN/RST on-board. Alamat IP ESP sudah tercetak.



Ref: <https://randomnerdtutorials.com/esp32-websocket-server-arduino/>

COBA WEBSOCKET + ESP32

- Buka browser di jaringan lokal kita dan masukkan alamat IP ESP32. Kita akan mendapatkan akses ke halaman web untuk mengontrol output.



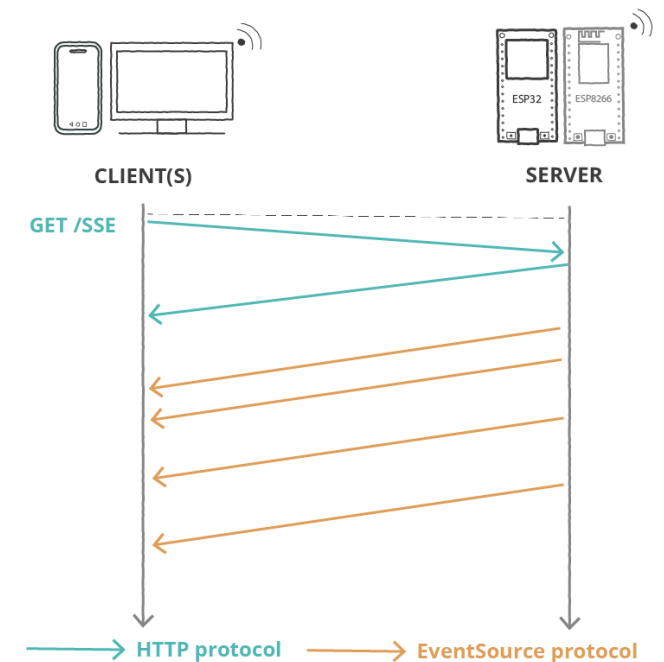
COBA WEBSOCKET + ESP32

- Klik pada tombol untuk mengalihkan LED.
- Kita dapat membuka beberapa tab browser web secara bersamaan atau mengakses server web dari perangkat yang berbeda dan status LED akan diperbarui secara otomatis di semua klien setiap kali ada perubahan.

SERVER-SENT EVENTS (SSE)

SERVER-SENT EVENTS (SSE)

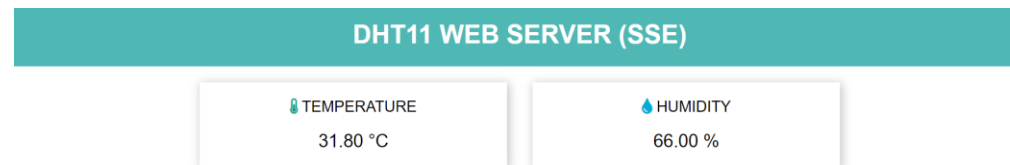
- Server-Sent Events (SSE) memungkinkan klien menerima pembaruan otomatis dari server melalui koneksi HTTP.
- Klien memulai koneksi SSE dan server menggunakan protokol event source untuk mengirim pembaruan ke klien.
- Klien akan menerima pembaruan dari server, tetapi tidak dapat mengirim data apa pun ke server setelah handshake awal.
- Hal ini berguna untuk mengirim pembacaan sensor yang diperbarui ke browser.
- Kapan pun pembacaan baru tersedia, ESP32 mengirimkannya ke klien dan halaman web dapat diperbarui secara otomatis tanpa perlu request lebih lanjut.



ESP32 + SSE

Web yang akan kita coba buat

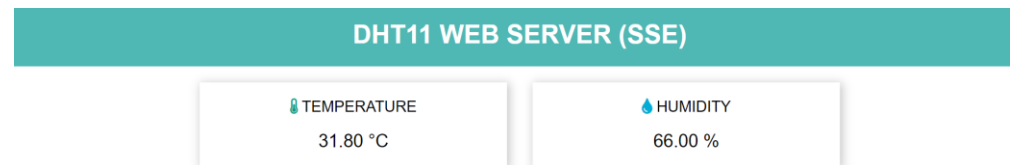
- Server web ESP32 menampilkan 2 kartu dengan pembacaan suhu, dan kelembapan DHT11;
- ESP32 mendapatkan pembacaan baru dari sensor setiap 5 detik;
- Setiap kali pembacaan baru tersedia, board (server) mengirimkannya ke klien menggunakan event yang dikirim server;



ESP32 + SSE

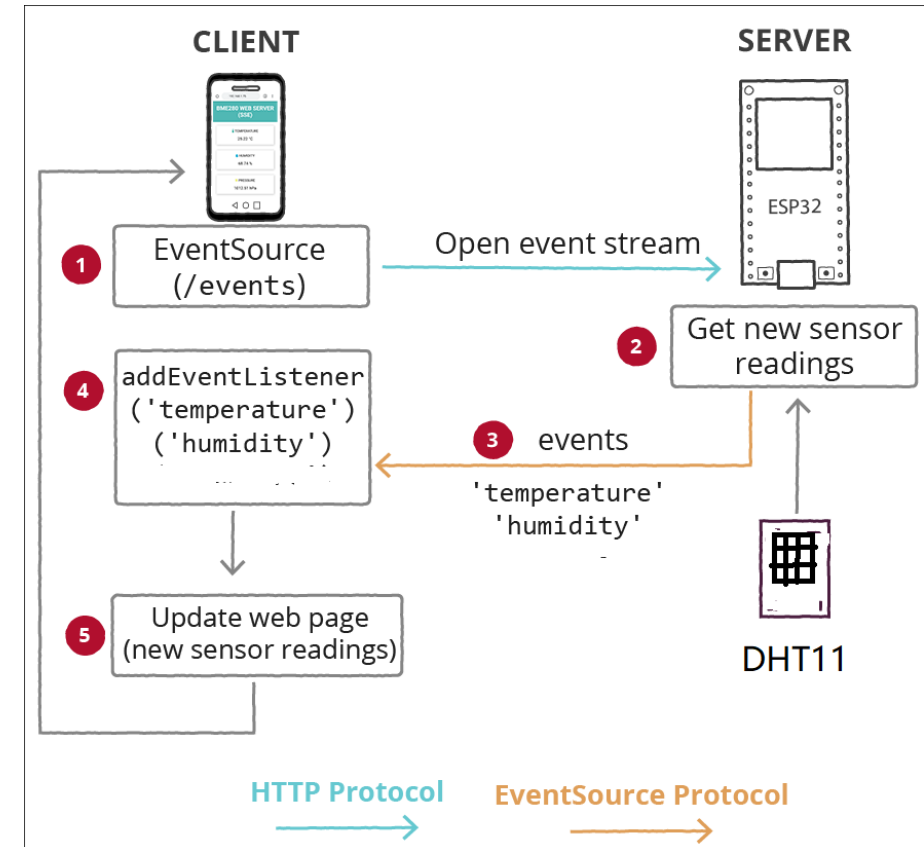
Web yang akan kita coba buat

- Klien menerima data dan memperbarui halaman web yang sesuai;
- Hal ini memungkinkan halaman web diperbarui secara otomatis setiap kali pembacaan baru tersedia.



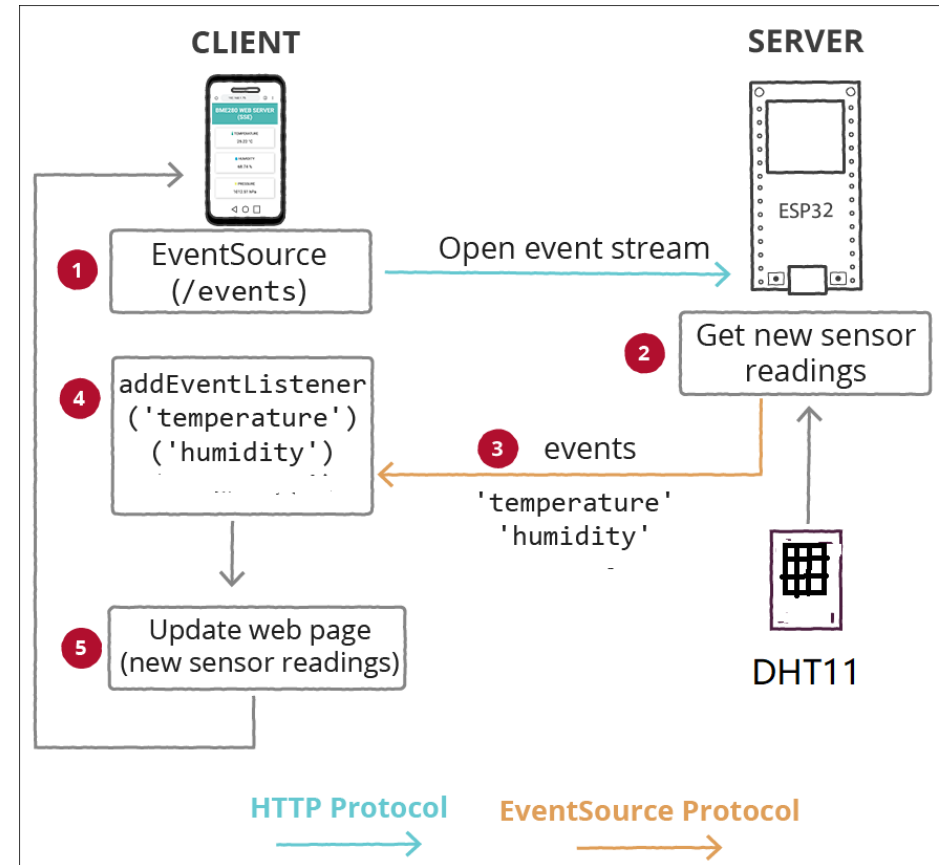
CARA KERJA ESP32 + SSE

- Klien memulai koneksi SSE dan server menggunakan protokol EventSource pada URL /events untuk mengirim pembaruan ke klien;
- ESP32 mendapatkan pembacaan sensor baru;
- ESP32 mengirimkan pembacaan sebagai Event dengan nama berikut kepada klien 'temperature', 'humidity' and 'pressure';.

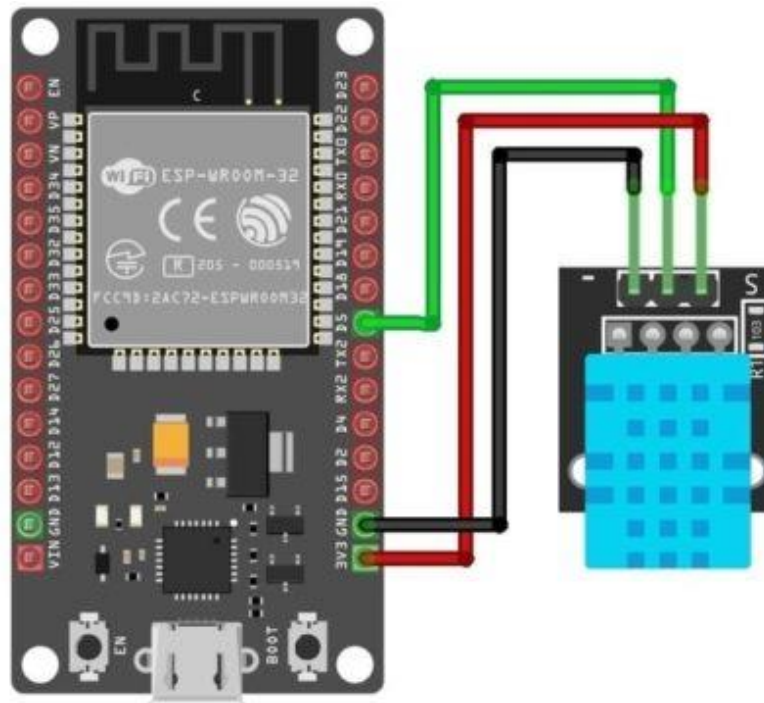


CARA KERJA ESP32 + SSE

- Klien memiliki EventListener untuk Event yang dikirim oleh server dan menerima pembacaan sensor yang diperbarui pada Event tersebut;
- Ini memperbarui halaman web dengan pembacaan terbaru.



COBA ESP32 + SSE



Vcc – 3.3v
Gnd – gnd
Data – d5

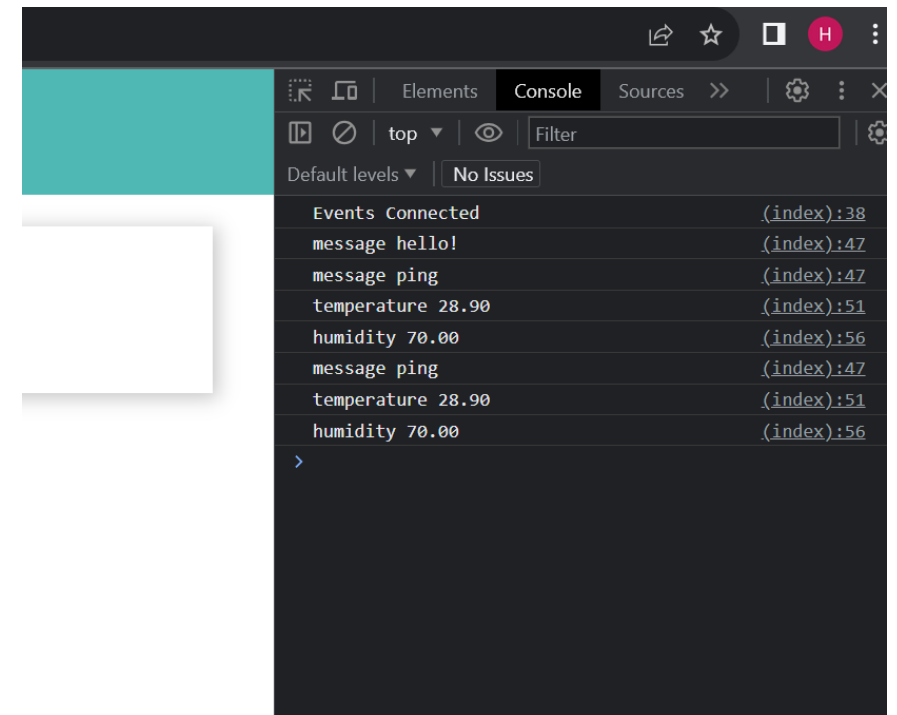
COBA ESP32 + SSE

- Ke https://bit.ly/t4b_iot_code
- Klik folder day 2
- Download file esp32_websocket.ino
- Ubah hal-hal yang diperlukan
- Upload filenya
- Setelah mengunggah kode, buka Serial Monitor pada kecepatan baud 115200 dan tekan tombol EN/RST on-board. Alamat IP ESP sudah tercetak.

```
// Replace with your network credentials
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

COBA ESP32 + SSE

- Buka browser di jaringan lokal kita dan masukkan alamat IP ESP32.
- Kita akan mendapatkan akses ke halaman web untuk memantau pembacaan sensor.
- Pembacaan diperbarui secara otomatis setiap 5 detik.
- Pada saat yang sama, kita akan mendapatkan pembacaan sensor baru pada Serial Monitor seperti yang ditunjukkan pada layar cetak sebelumnya.
- Selain itu, kita dapat memeriksa apakah klien menerima Event. Pada browser kita, buka konsol dengan menekan Ctrl+Shift+J.



MQTT

Apa itu MQTT?

- MQTT adalah singkatan dari Message Queuing Telemetry Transport.
- MQTT adalah protokol perpesanan sederhana, yang dirancang untuk perangkat terbatas dengan bandwidth rendah.
- Komunikasi MQTT bekerja sebagai sistem publish dan subscribe.
- Perangkat mempublikasikan pesan tentang topik tertentu. Semua perangkat yang subscribe pada topik tersebut menerima pesan tersebut.
- Aplikasi utamanya termasuk mengirim pesan untuk mengontrol output, membaca dan mempublikasikan data dari node sensor dan banyak lagi.



Apa itu MQTT?

Dalam MQTT ada beberapa konsep dasar yang perlu dipahami:

- Publish/Subscribe
- Messages
- Topics
- Broker

MQTT – Publish/Subscribe

Dalam sistem publish dan subscribe, perangkat dapat mempublikasikan pesan tentang suatu topik, atau dapat berlangganan ke topik tertentu untuk menerima pesan

- Misalnya Perangkat 1 menerbitkan tentang suatu topik.
- Perangkat 2 berlangganan topik yang sama dengan topik yang diterbitkan oleh perangkat 1.
- Jadi, perangkat 2 menerima pesan tersebut.

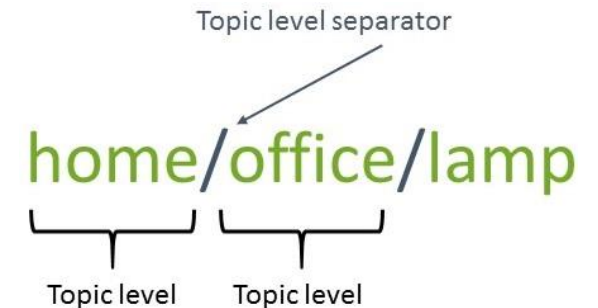


MQTT – Messages

- Message adalah informasi yang ingin kita tukarkan di antara perangkat kita.
- Pesan dapat berupa pesan seperti perintah atau data seperti pembacaan sensor, misalnya.

MQTT - Topik

- Topik adalah cara kita mendaftarkan minat untuk pesan yang masuk atau cara kita menentukan di mana kita ingin mempublikasikan pesan tersebut.
- Topik direpresentasikan dengan string yang dipisahkan oleh garis miring ke depan. Setiap garis miring menunjukkan tingkat topik.
- Catatan: topik sensitif terhadap huruf besar/kecil



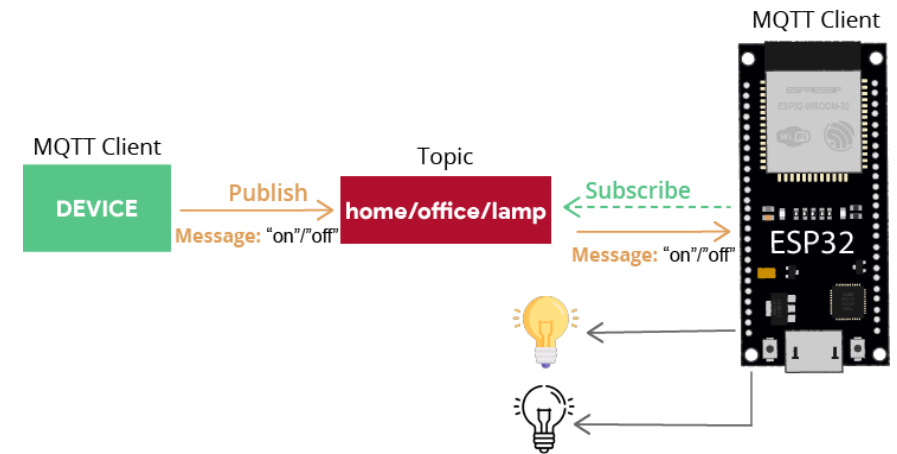
home/office/lamp

≠

home/office/LAmp

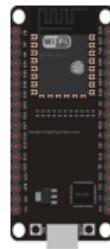
MQTT - Topik

- Perangkat menerbitkan pesan " on" dan "off" pada topik rumah/kantor/lampu.
- Kita memiliki perangkat yang mengontrol lampu (dapat berupa ESP32, ESP8266, atau board atau perangkat lain).
- ESP32 yang mengendalikan lampu kita, berlangganan topik yang sama: rumah/kantor/lampu.
- Jadi, ketika ada pesan baru yang diterbitkan pada topik tersebut, ESP32 menerima pesan " on " atau " off " dan menyalakan atau mematikan lampu.

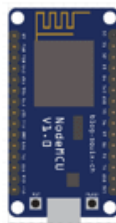


MQTT - Topik

Perangkat yang menerbitkan pesan dapat berupa ESP32, ESP8266, atau platform pengontrol Otomasi Rumah dengan dukungan MQTT seperti Node-RED, Home Assistant, Domoticz, atau OpenHAB, misalnya



ESP32

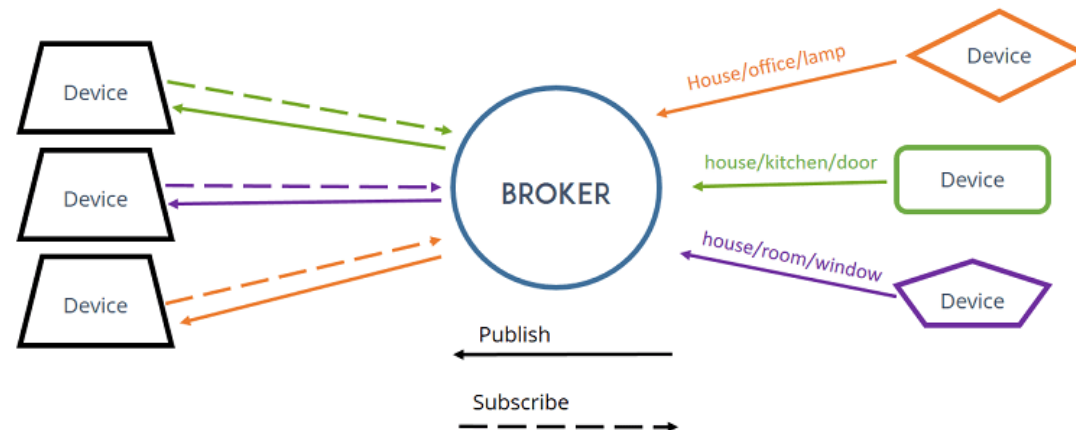


ESP8266



MQTT – Broker

Broker MQTT bertanggung jawab untuk menerima semua pesan, menyaring pesan, memutuskan siapa yang tertarik dengan pesan tersebut, dan kemudian mempublikasikan pesan tersebut ke semua klien yang berlangganan.



COBA ESP32 + MQTT

- Install library pubsubclient by nick o'liery di library manager
- Ke https://bit.ly/t4b_iot_code
- Klik folder day 2
- Download file mqtt_esp32.ino
- Ubah ssid dan password wifi, serta topic_publish dan topic_subscribe



```
26  const char* ssid = "ganti";  
27  const char* password = "ganti";
```

```
30  char* topic_subscribe = "LampuOnAtauOff";  
31  char* topic_publish = "kirimHelloWorld";  
32
```

COBA ESP32 + MQTT

- Buka web <https://www.hivemq.com/demos/websocket-client/>
- Klik connect
- Untuk publish, buka bagian publish, lalu masukkan topic dan messagenya lalu klik publish
- Bila kita message yang diawali dengan angka 1 maka lampu biru di ESP32 akan menyala. Bila kita message yang tidak diawali dengan angka 1 maka lampu biru tersebut akan mati

The screenshot displays the HiveMQ Websocket Client interface. On the left, the 'Connect' section includes a 'ClientID' field with the value 'clientId-rTxXnIsJB', a 'Connect' button, and checkboxes for 'Keep Alive' (checked), 'SSL' (unchecked), and 'Clean Session' (unchecked). The 'Publish' section in the center features a 'Topic' field with 'LampuOnAtauOff', a 'QoS' dropdown set to '0', a 'Retain' checkbox (unchecked), and a 'Publish' button. Below these is a 'Message' input field containing the number '1'. On the right, the 'Subscriptions' section shows a 'connected' status with a green dot, an 'Add New Topic Subscription' button, and a list of subscriptions including 'Qos: 2' and 'kirimHelloWorld'.

COBA ESP32 + MQTT

- Untuk subscribe, buka bagian subscribe, klik add new topic subscription, masukkan topicnya, lalu klik subscribe
- kita akan melihat pesan dari topic yang kita subscribe



ClientID

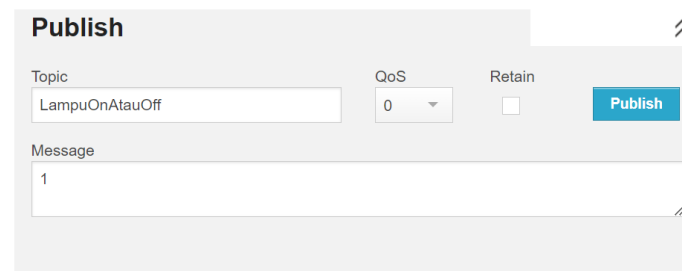
clientId-rTnxXnlsJB

Connect

Keep Alive 60

SSL ☒

Clean Session ☒



Publish

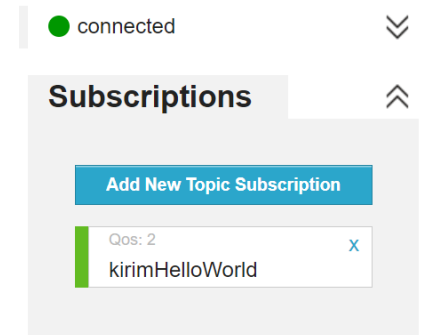
Topic LampuOnAtauOff

QoS 0

Retain ☐

Publish

Message 1



connected

Subscriptions

Add New Topic Subscription

Qos: 2

irimHelloWorld

HASIL COBA ESP32 + MQTT

LampuOnAtauOff

Message

00

Publish message: hello world #111
Publish message: hello world #112
Publish message: hello world #113
Publish message: hello world #114
Publish message: hello world #115
Publish message: hello world #116
Publish message: hello world #117
Publish message: hello world #118
Publish message: hello world #119
Publish message: hello world #120

Messages

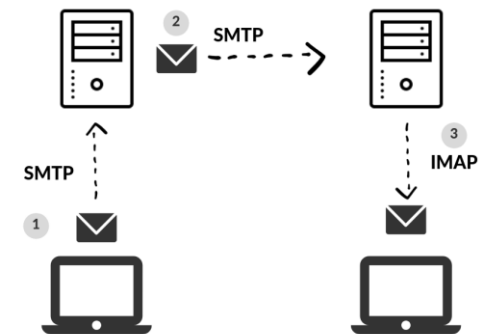
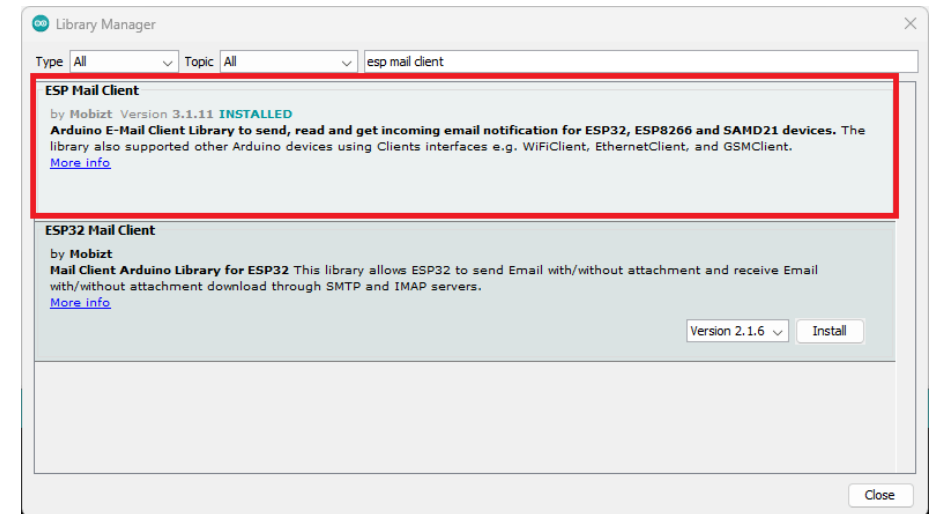
☒ Autoscrol ☐ Show timestamp

2023-08-26 18:41:29	Topic: kirimHelloWorld	Qos: 0
hello world #120		
2023-08-26 18:41:27	Topic: kirimHelloWorld	Qos: 0
hello world #119		
2023-08-26 18:41:25	Topic: kirimHelloWorld	Qos: 0
hello world #118		
2023-08-26 18:41:24	Topic: kirimHelloWorld	Qos: 0
hello world #117		

SMTP

ESP32 + SMTP

- SMTP berarti Simple Mail Transfer Protocol dan merupakan standar internet untuk transmisi email.
- Untuk mengirim email dengan ESP32, kita akan menggunakan library ESP-Mail-Client.
- Library ini memungkinkan ESP32 untuk mengirim dan menerima email dengan atau tanpa attachment melalui server SMTP dan IMAP.
- Buka Sketch > Include Library > Manage Libraries dan cari ESP Mail Client. Instal library ESP Mail Client dari Mobitz.



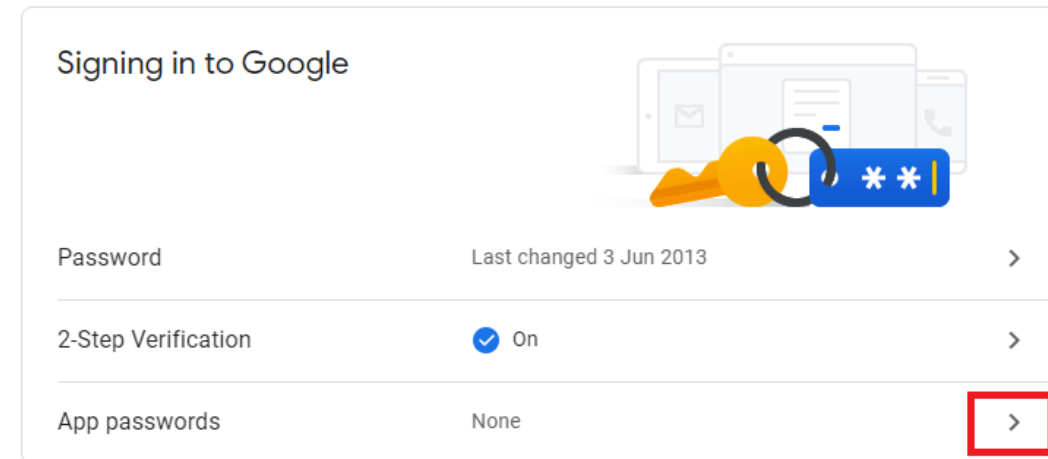
Ref: <https://randomnerdtutorials.com/esp32-send-email-smtp-server-arduino-ide/>

ESP32 + SMTP

- Jangan gunakan email pribadi utama kita untuk mengirim email melalui ESP32.
- Jika terjadi kesalahan pada kode kita atau jika secara tidak sengaja kita membuat terlalu banyak permintaan, kita dapat diblokir atau akun kita dinonaktifkan sementara.
- Email penerima dapat berupa email pribadi kita tanpa masalah.

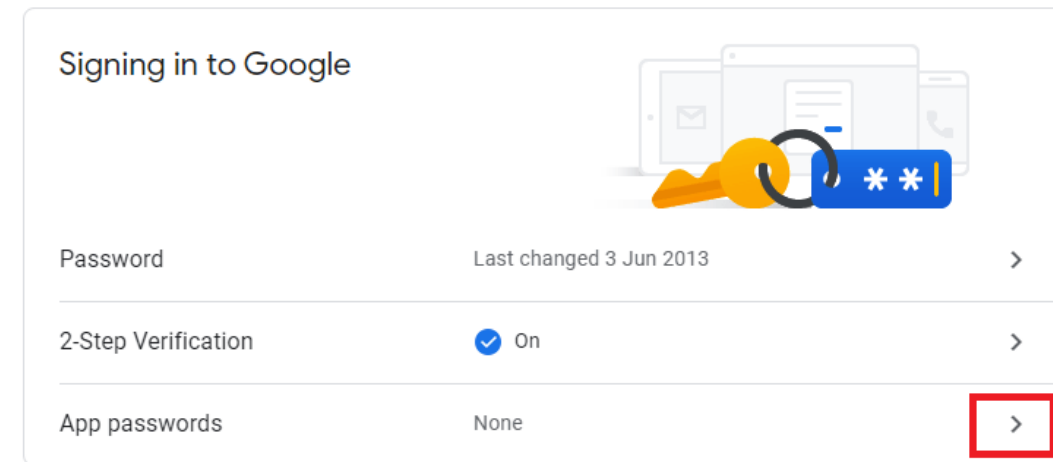
MENGATUR EMAIL UNTUK ESP32 + SMTP

- Kita perlu membuat app password agar ESP32 dapat mengirim email menggunakan akun Gmail kita.
- App Password adalah kode sandi 16 digit yang memberikan izin kepada aplikasi atau perangkat yang kurang aman untuk mengakses Akun Google kita.
- " app password" hanya dapat digunakan dengan akun yang mengaktifkan verifikasi 2 langkah.



MENGATUR EMAIL UNTUK ESP32 + SMTP

- Buka Akun Google kita.
- Di panel navigasi, pilih Security (Keamanan).
- Di bawah "Sign in to Google," pilih 2-Step Verification > Get started.
- Ikuti langkah-langkah di layar.
- Setelah mengaktifkan verifikasi 2 langkah, kita dapat membuat "app password".



Ref: <https://randomnerdtutorials.com/esp32-send-email-smtp-server-arduino-ide/>

MENGATUR EMAIL UNTUK ESP32 + SMTP

- Buka Akun Google kita.
- Di panel navigasi, pilih Security (Keamanan).
- Di bawah "Sign in to Google," pilih App Passwords.
- Di bidang "Select app", pilih mail (Surat). Untuk perangkat, pilih Other (Lainnya) dan beri nama, misalnya ESP32.
- Kemudian, klik Generate.

The screenshot shows the Google Account Security settings page. On the left, there are input fields for Email (securesally@gmail.com) and Password (masked with dots). The main heading is "App passwords". Below it, a text block explains that app passwords allow signing in from apps on devices that don't support 2-Step Verification. A table titled "Your app passwords" is shown, with columns for Name, Created, and Last used. The table contains one entry: "ESP32" created at "11:16" with a trash icon for deletion. Below the table, there are dropdown menus for "Select app" and "Select device", and a "GENERATE" button.

Generated app password

Your app password for your device

How to use it

Go to the settings for your Google Account in the application or device you are trying to set

Email
securesally@gmail.com

Password
.....

← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

Your app passwords

Name	Created	Last used
ESP32	11:16	-

Select the app and device for which you want to generate the app password.

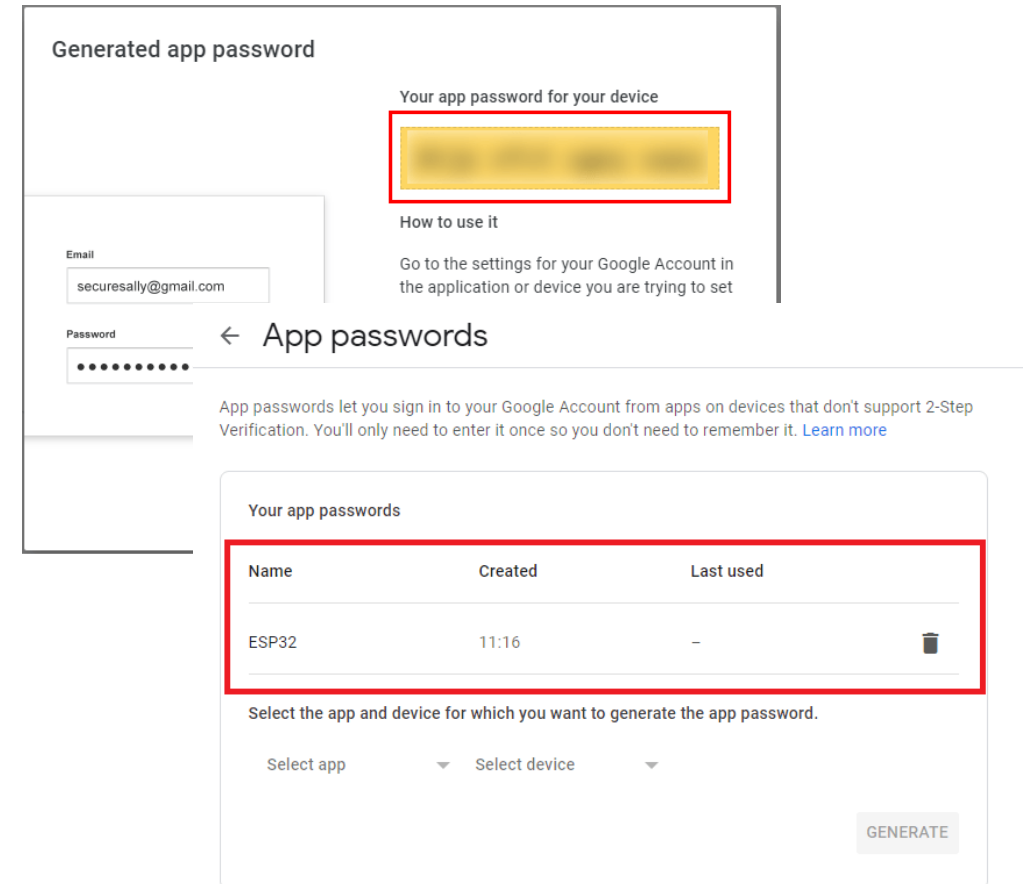
Select app ▼ Select device ▼

GENERATE

Ref: <https://randomnerdtutorials.com/esp32-send-email-smtp-server-arduino-ide/>

MENGATUR EMAIL UNTUK ESP32 + SMTP

- Simpan kata sandi tersebut (meskipun dikatakan bahwa kita tidak perlu mengingatnya) karena kita akan membutuhkannya nanti.
- Jika kita menggunakan penyedia email lain, periksa cara membuat kata sandi aplikasi.
- Kita seharusnya dapat menemukan petunjuknya dengan Google "your_email_provider + create app password".



The screenshot shows the Google App Passwords management interface. At the top, a 'Generated app password' is displayed in a yellow box, highlighted with a red rectangle. Below this, the 'How to use it' section explains that app passwords are used for signing in to Google Accounts from apps on devices that don't support 2-Step Verification. On the left, there are input fields for 'Email' (showing 'securesally@gmail.com') and 'Password' (masked with dots). Below the password field is a back arrow and the text 'App passwords'. At the bottom, there is a table titled 'Your app passwords' with columns 'Name', 'Created', and 'Last used'. The table contains one entry: 'ESP32' with a creation time of '11:16' and a last used time of '-'. A trash icon is visible next to the entry. Below the table, there is a section for generating a new app password, with dropdown menus for 'Select app' and 'Select device', and a 'GENERATE' button.

Name	Created	Last used
ESP32	11:16	-

MENGATUR EMAIL UNTUK ESP32 + SMTP

Jika kita menggunakan akun Gmail, berikut ini adalah rincian Server SMTP:

- Server SMTP: smtp.gmail.com
- Username SMTP: Alamat Gmail lengkap
- Password SMTP: Password Gmail kita
- Port SMTP (TLS): 587
- Port SMTP (SSL): 465
- Required SMTP TLS/SSL: yes

Jika kita menggunakan penyedia email lain, kita perlu mencari pengaturan SMTP Server-nya.

COBA ESP32 + SMTP

- Ke https://bit.ly/t4b_iot_code
- Klik folder day 2
- Download file smtp_esp32.ino
- Ubah hal2 yang diperlukan
- Upload kodenya

```
#define WIFI_SSID "REPLACE_WITH_YOUR_SSID"  
#define WIFI_PASSWORD "REPLACE_WITH_YOUR_PASSWORD"
```

```
#define SMTP_HOST "smtp.gmail.com"  
#define SMTP_PORT 465
```

```
#define AUTHOR_EMAIL "YOUR_EMAIL@XXXX.com"  
#define AUTHOR_PASSWORD "YOUR_EMAIL_PASS"
```

```
#define RECIPIENT_EMAIL "RECIPIENTE_EMAIL@XXXX.com"
```

```
/* Set the message headers */  
message.sender.name = F("ESP");  
message.sender.email = AUTHOR_EMAIL;  
message.subject = F("ESP Test Email");  
message.addRecipient(F("Sara"), RECIPIENT_EMAIL);
```

```
//Send raw text message  
String textMsg = "Hello World - Sent from ESP board";  
message.text.content = textMsg.c_str();  
message.text.charSet = "us-ascii";  
message.text.transfer_encoding = Content_Transfer_Encoding::enc_7bit;
```

Ref: <https://randomnerdtutorials.com/esp32-send-email-smtp-server-arduino-ide/>

HASIL COBA ESP32 + SMTP

```
COM3
Send
< S: 250 2.0.0 OK 1687452501 k18-20020adfe8d2000000b0030ae3a6be4asm7564860wrn.72

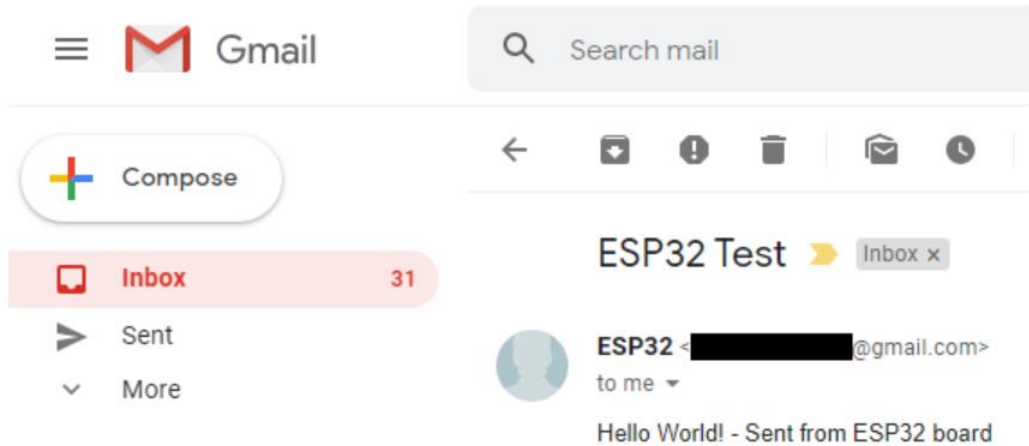
#### Closing the session...
> C: terminate the SMTP session

#### Message sent successfully
> C: message sent successfully

-----
Message sent success: 1
Message sent failed: 0
-----

Message No: 1
Status: success
Date/Time: June 22, 2023 19:48:18
Recipient:      :@gmail.com
Subject: ESP Test Email
-----

☒ Autoscroll ☐ Show timestamp
Newline ▼ 115200 baud ▼ Clear output
```



Ref: <https://randomnerdtutorials.com/esp32-send-email-smtp-server-arduino-ide/>



Train4best

THANK YOU