



# MQTT

# Protocol

[Short Bootcamp]

# IoT?



**IoT** merupakan singkatan dari "**Internet of Things**". IoT merujuk pada **konsep di mana berbagai jenis perangkat** fisik, seperti sensor, perangkat elektronik, kendaraan, peralatan rumah tangga, dan lain sebagainya, **dapat saling terhubung** melalui jaringan **internet dan berkomunikasi satu sama lain**. Tujuan utama dari IoT adalah **menghubungkan dunia fisik dengan dunia digital**, sehingga objek-objek tersebut dapat mengumpulkan, mengirimkan, dan menerima data serta berinteraksi secara otomatis.

**The physical world meets the digital world**

# Komponen Dasar [IoT]



Microcontroller



Sensor & Actuator



Communication  
Module



Power source

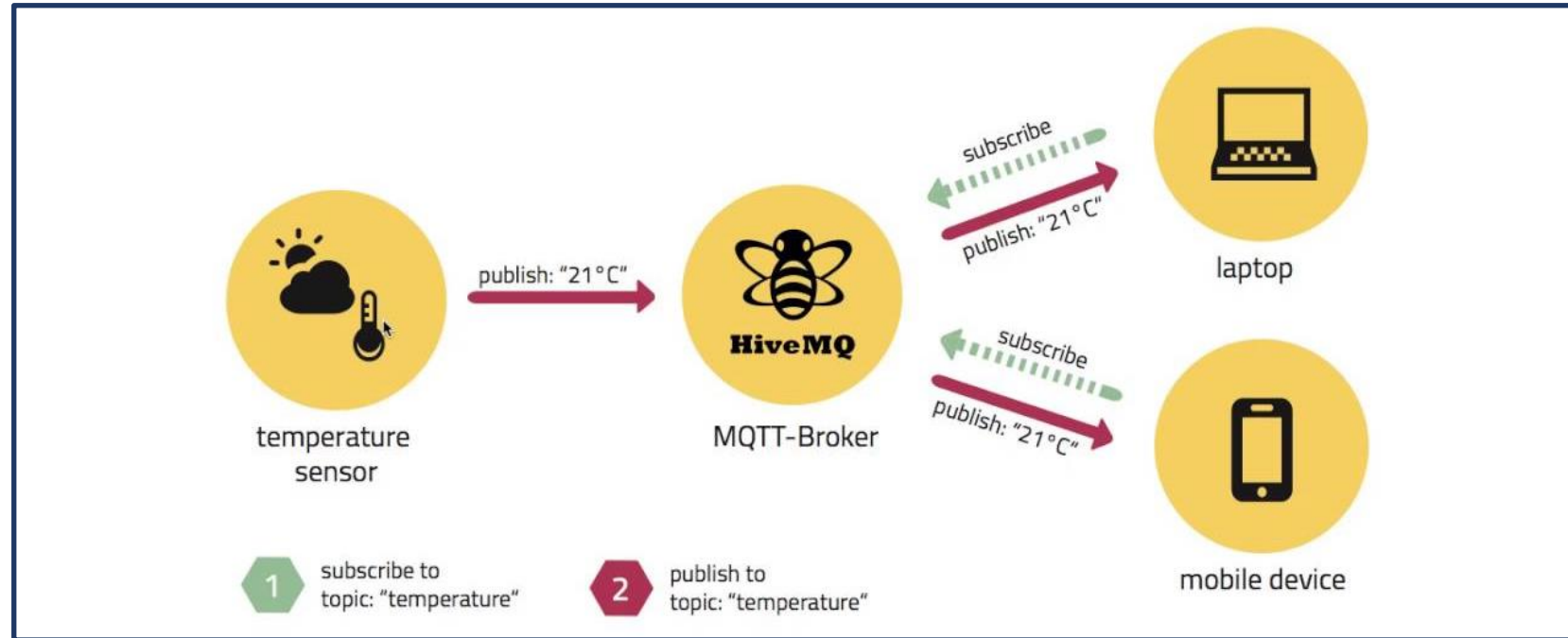


# MQTT Protocol

# IoT Protocols



# MQTT?



Source: <https://medium.com/@ghostlulzhacks/>

Message Queuing Telemetry Transport. merupakan protokol messaging yang menggunakan model **Publish-Subscribe**.



- Publishers and Subscribers.
- Connect to broker
- Create specific topic



- Filtering
- Receiving
- Determining
- Sending



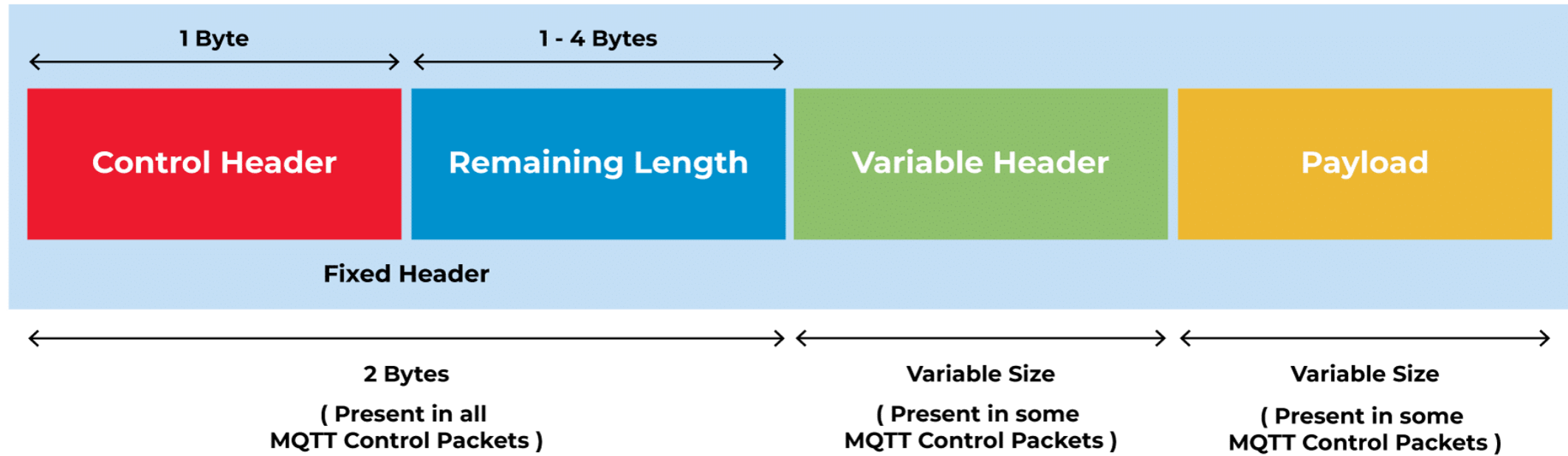
- Based on TCP/IP
- Send CONNECT
- Client never connect to each other directly

# QoS (Quality of Services)

Tingkat kualitas layanan pengiriman atau **Quality of Service (QoS)** pada MQTT terdapat Tiga level :

1. QoS 0 (at most once delivery) : **pesan hanya dikirim sekali**, setelah pesan dikirim **tidak ada respon apakah pesan berhasil dikirim**, pesan bisa saja tidak terkirim, atau tidak diterima *subscriber*.
2. QoS 1 (at least once delivery) : pesan paling sedikit dikirim sekali, jika *subscriber* tidak menerima pesan maka broker akan mengirim respon terhadap *publisher* bahwa pesan gagal dikirim. Kemungkinan duplikasi pesan dapat terjadi.
3. QoS 2 (exactly once delivery) : pesan dikirim sekali dan hanya sekali. QoS 2 memastikan bahwa pesan diterima *subscriber*. *Mencegah adanya duplikasi pesan*.


# Packet Format





# MQTT Topic

sensors/temperature/out



Topic level      Topic level      Topic level

## MQTT Topic Wildcard

### Multi-level

multi-level  
wildcard  
↓  
myhome / groundfloor / #



only at the end  
multiple topic levels

- ✓ myhome / groundfloor / livingroom / temperature
- ✓ myhome / groundfloor / kitchen / temperature
- ✓ myhome / groundfloor / kitchen / brightness
- ✗ myhome / firstfloor / kitchen / temperature

### Single-level

single-level  
wildcard  
↓  
myhome / groundfloor / + / temperature



only one level

- ✓ myhome / groundfloor / livingroom / temperature
- ✓ myhome / groundfloor / kitchen / temperature
- ✗ myhome / groundfloor / kitchen / brightness
- ✗ myhome / firstfloor / kitchen / temperature
- ✗ myhome / groundfloor / kitchen / fridge / temperature

# ESP 32 Intro



## MCU

802.11 b/g/n Wi-Fi

Bluetooth

Typical Frequency

SRAM

Flash

GPIO

Hardware /Software PWM

SPI/I2C/I2S/UART

ADC

CAN

Ethernet MAC Interface

Touch Sensor

Temperature Sensor

Hall effect sensor

Working Temperature

## ESP32

Xtensa Dual-Core 32-bit LX6 with 600 DMIPS

HT40

Bluetooth 4.2 and BLE

160 MHz

Yes

Yes

36

None / 16 channels

4/2/2/2

12-bit

Yes

Yes

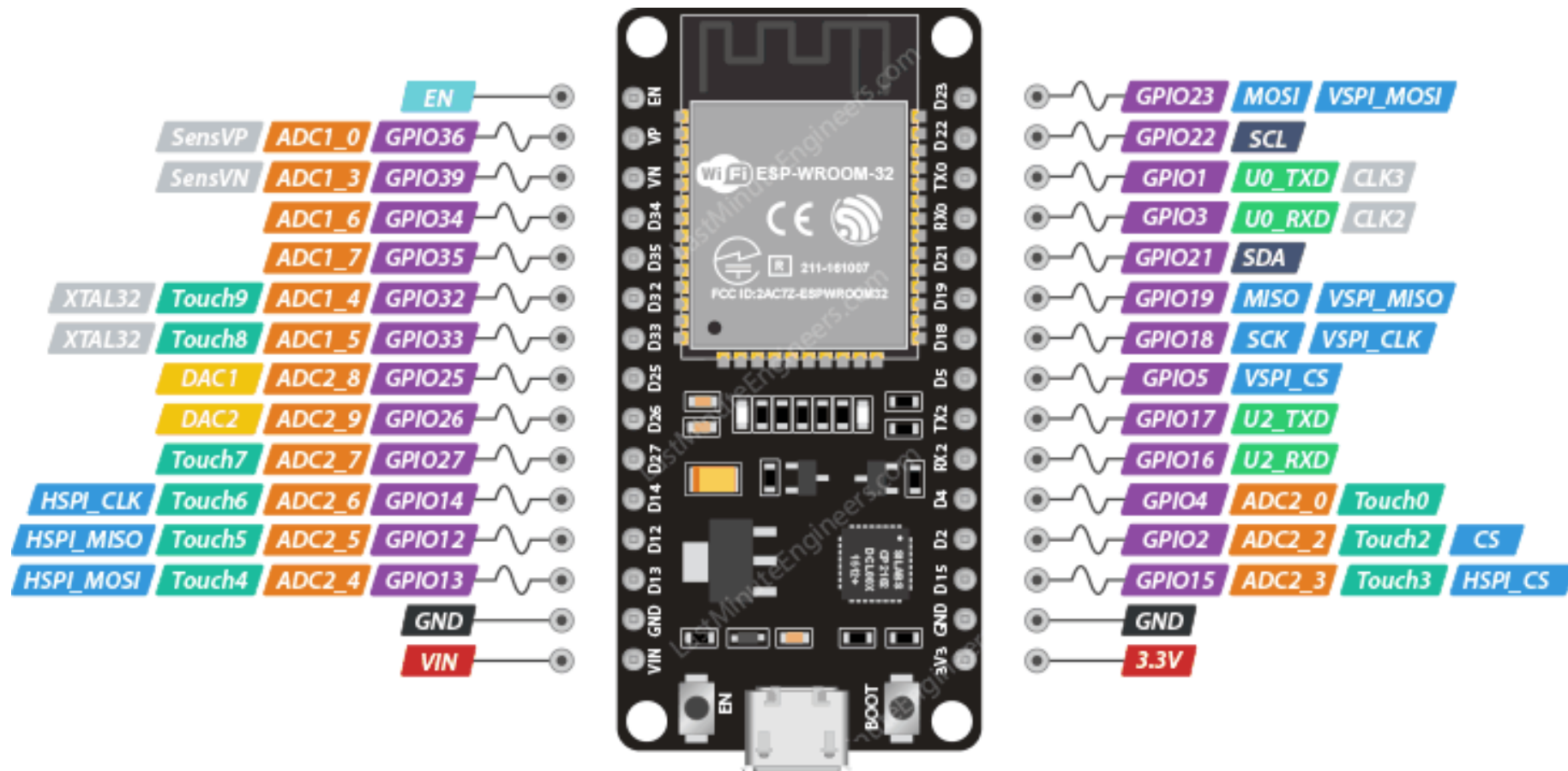
Yes

Yes

Yes

-40°C to 125°C

# Pin I/O

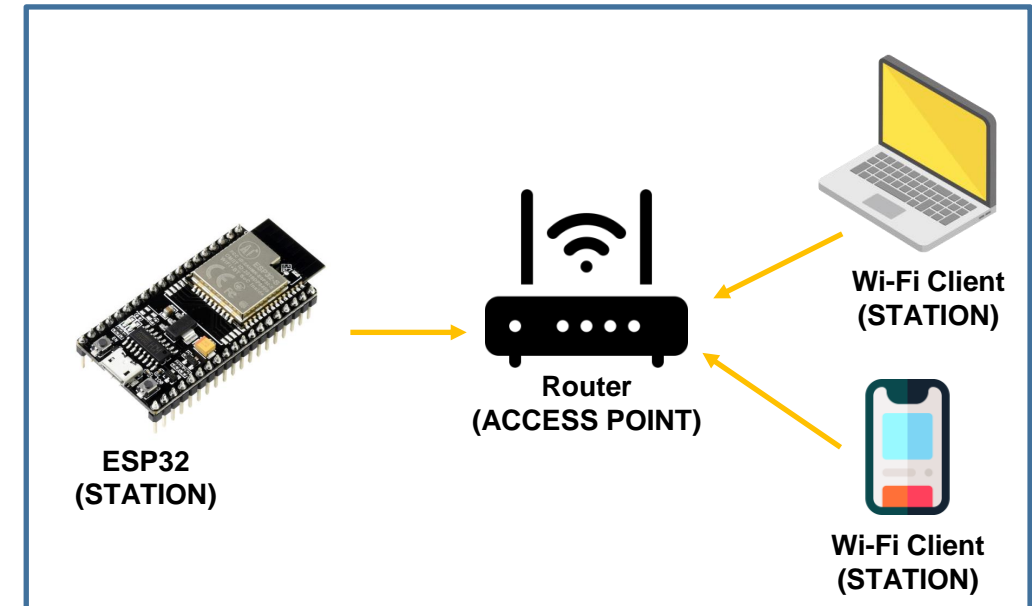
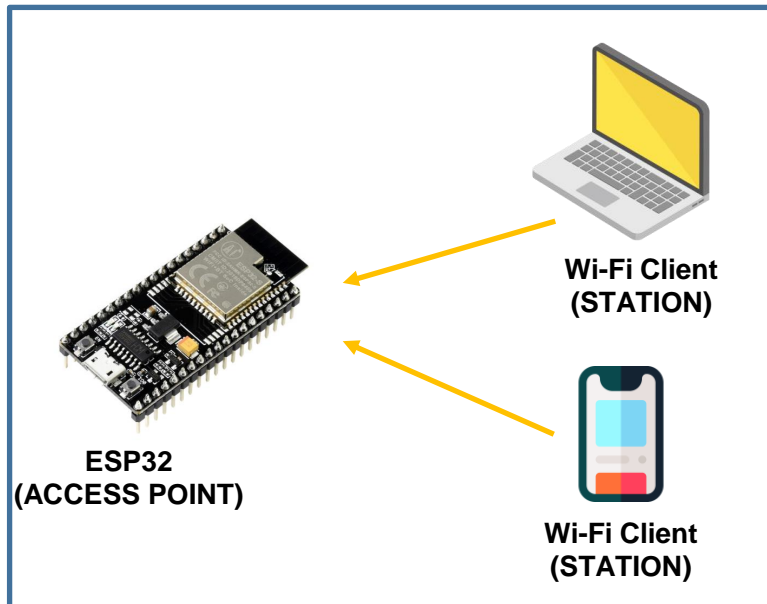


# ESP Mode

## #include <WiFi.h>

Librari ini adalah libarray untuk menggunakan Modul WiFi yang tersedia pada board ESP, library ini secara otomatis akan terinstal ketika anda menambahkan ESP32 ke Arduino IDE Anda

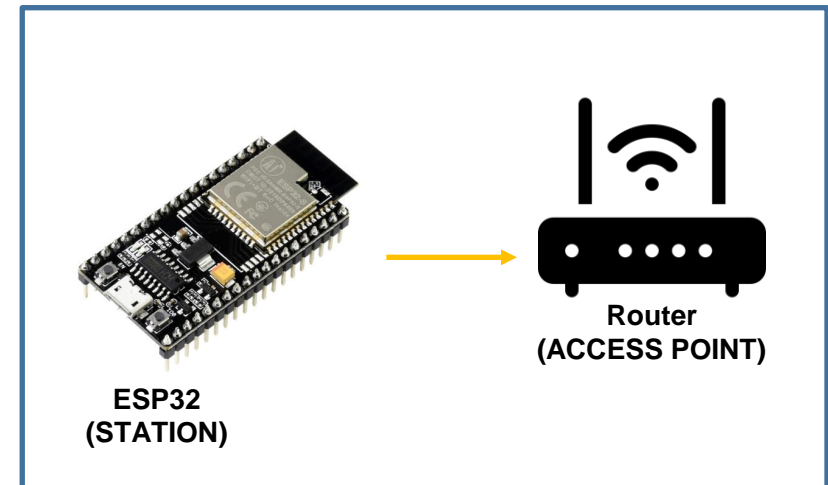
1. Station Mode (STA)
2. Access Point Mode (AP)
3. STA & AP Mode



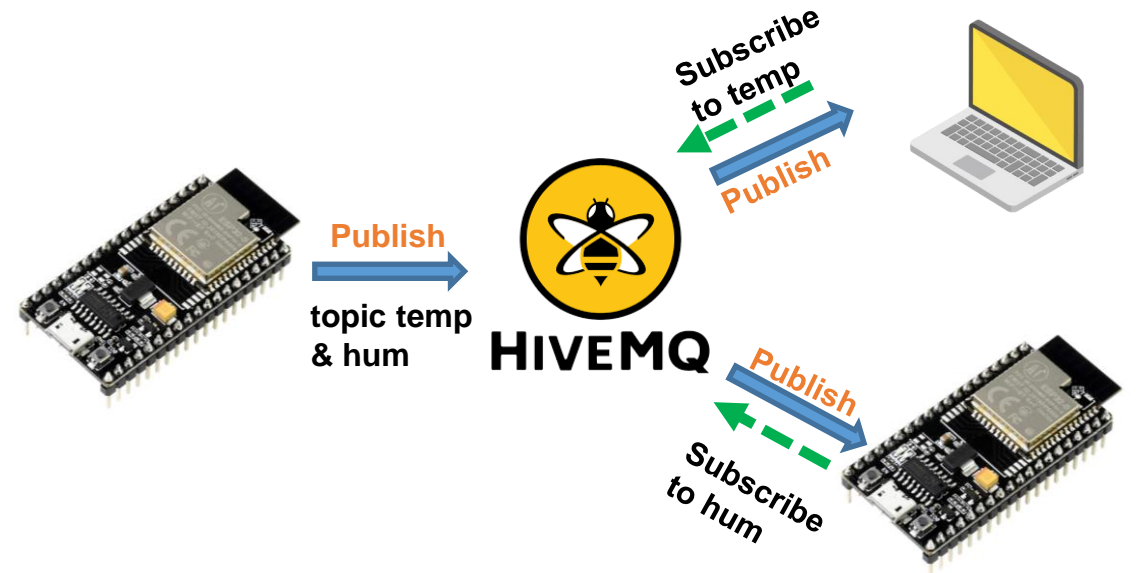
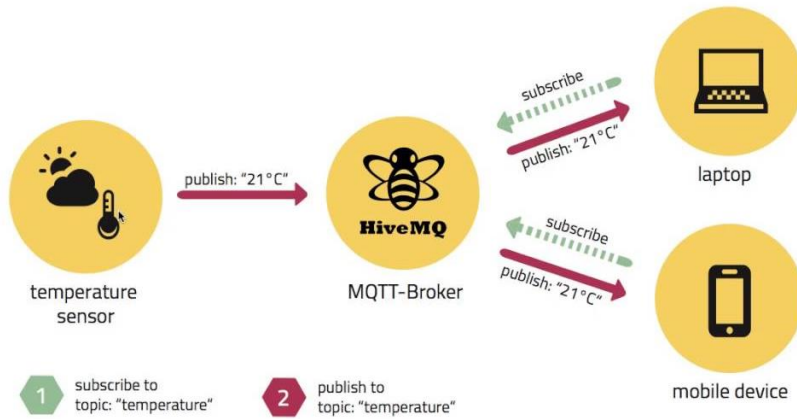
	Fungsinya
WiFi.mode(WIFI_STA)	Station Mode : ESP32 dapat connect ke AP
WiFi.mode(WIFI_AP)	Access Point mode: stations dapat terhubung ke ESP32
WiFi.mode(WIFI_STA_AP)	ESP32 dapat diatur sebagai stasiun Wi-Fi dan titik akses secara bersamaan

# Connecting to Wi-Fi Network

Function	Keterangan
WiFi.begin(ssid, password)	Untuk terhubung ke jaringan
WiFi.status()	Mendapatkan nama SSID Wifi Network
WL_CONNECTED	Constanta/nilai ketika ESP telah terhubung ke Wi-Fi
WiFi.localIP()	Untuk mengetahui IP yang di berikan oleh router ke ESP32
WiFi.scanNetworks()	Akan memberikan nilai balikan jumlah network yang ditemukan
WiFi.SSID(i)	Mendapatkan nama SSID Wifi Network
WiFi.RSSI(i)	Mendapatkan nilai RSSI Wifi Network

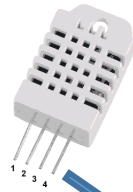


# Showcase

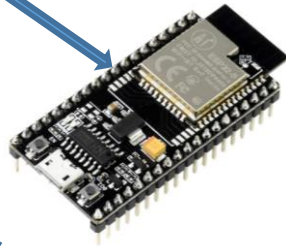


# Showcase

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



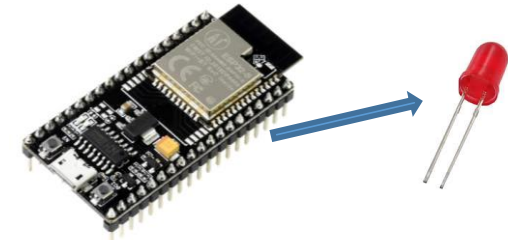
Sensor Cahaya



Publisher



Broker



Subscriber

# MQTT Broker

The screenshot shows the HiveMQ MQTT Dashboard. The top navigation bar includes the HiveMQ logo, the text 'HiveMQ | Public Broker | MQTT Dashboard', and a status indicator 'HiveMQ cluster ( 3 nodes )'. The left sidebar contains links for 'Home' and 'Legal'. The main content area is divided into several sections:

- Broker:** A section explaining the MQTT Dashboard's use of the HiveMQ MQTT broker and providing information about testing, usage, and the HiveMQ Cloud platform.
- Getting Started:** A section providing links to MQTT Essentials, MQTT Websocket client, and MQTT Client Library Encyclopedia.
- Outgoing Messages:** A counter showing 1012615295 messages.
- Incoming Messages:** A counter showing 1660893880 messages.
- Subscriptions:** A counter showing 194011 subscriptions.
- Retained Messages:** A counter showing 32374 retained messages.
- Clients:** A section with a semi-circular progress indicator.
- MQTT connection settings:** A section with a red border containing the following information:
  - Host: broker.hivemq.com
  - TCP Port: 1883
  - Websocket Port: 8000

At the bottom of the dashboard, there is a 'Learn more' button and a link to the MQTT Dashboard website.

<http://www.mqtt-dashboard.com/>



## DHT

### DHT22

**#include <DHT.h>**

Salah satu alternatif library yang digunakan untuk mengakses DHT Module

**DHT**  
**dhtku(DHTPIN,DHTTYPE);**

digunakan untuk membuat objek dari kelas DHT

**dhtku.begin();**

menginisialisasi sensor DHT dan menyiapkan koneksi antara sensor DHT dan board Arduino

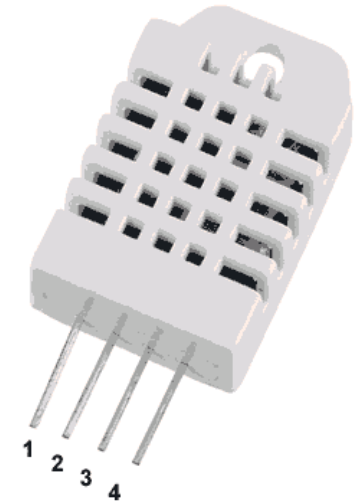
**dhtku.readHumidity();**

Digunakan untuk mengakses fungsi pembacaan nilai humidity (%)

**dhtku.readTemperature();**

Digunakan untuk mengakses fungsi pembacaan nilai temperature dalam celcius

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



# MQTT Syntax

Function	Keterangan
<code>#include "PubSubClient.h"</code>	Librari untuk mengkoneksikan ESP32 ke MQTT broker
<code>PubSubClient mqttClient()</code>	Membuat object "mqttClient"
<code>mqttClient.setServer(mqttServer, mqttPort)</code>	Setting alamat target dan port broker
<code>mqttClient.setCallback()</code>	Untuk dapat menerima pesan balikan dari server
<code>mqttClient.connected()</code>	Mengembalikan nilai True jika berhasil terkoneksi dan False jika belum
<code>mqttClient.connect(clientID)</code>	Untuk dapat terhubung dengan broker (ClientID harus unique) dan memiliki keluaran Boolean
<code>mqttClient.publish(topic, payload)</code>	Untuk melakukan publish ke suatu topic (Payload berupa String)
<code>mqttClient.subscribe(topik)</code>	Untuk berlangganan suatu topik
<code>client.loop()</code>	untuk memproses semua penerimaan dan pengiriman pesan yang tersedia untuk klien (client)

## More info about IoT

- **Blog:** <https://medium.com/@sonyalfathani>
- **YTube :** <https://youtu.be/R3v8kGaafHM>

Sub : <https://wokwi.com/projects/356666910782912513>

Pub : <https://wokwi.com/projects/373043061822247937>



THANK  
YOU