



Internet of Things

Aplikasi Kontrol dan Pemantauan dengan MQTT dan ESP32

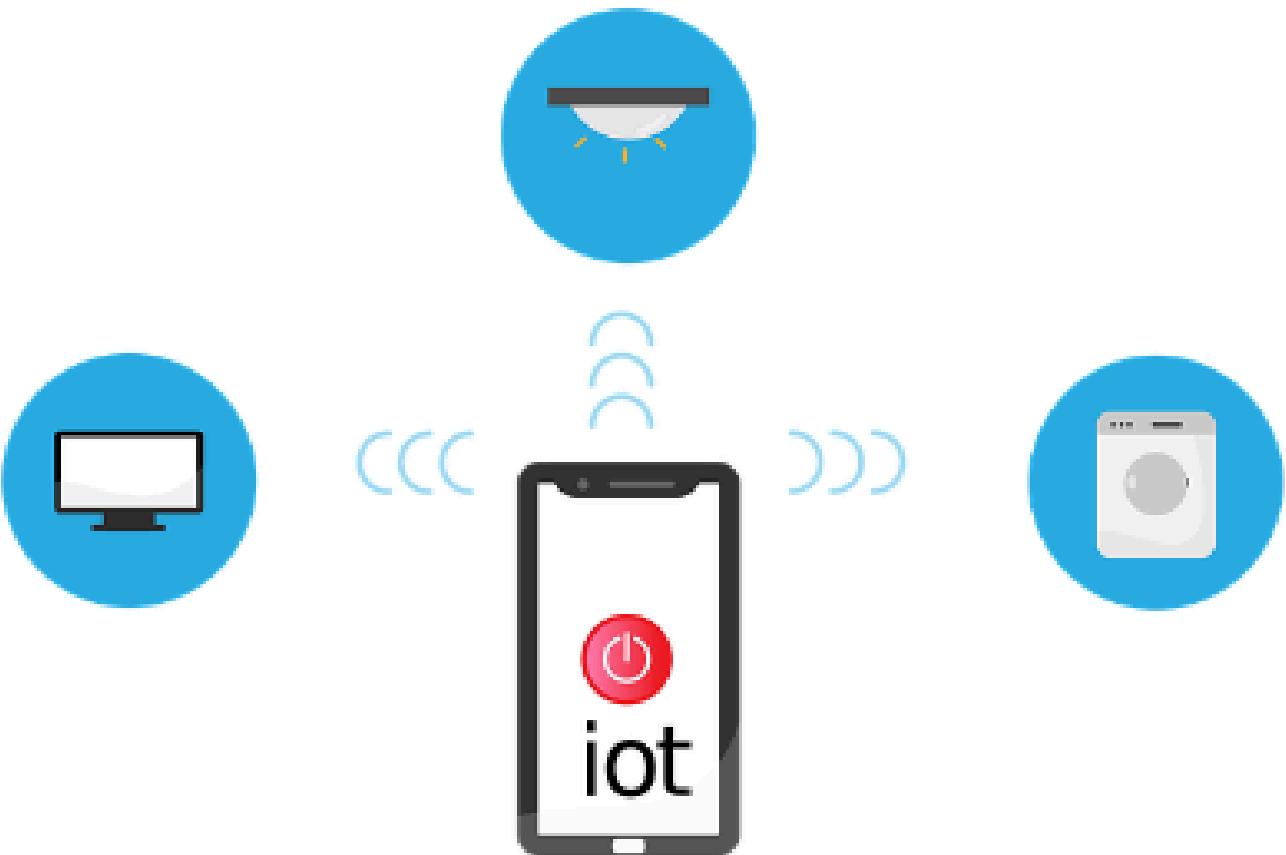


Muhammad Ikhwan Fathulloh



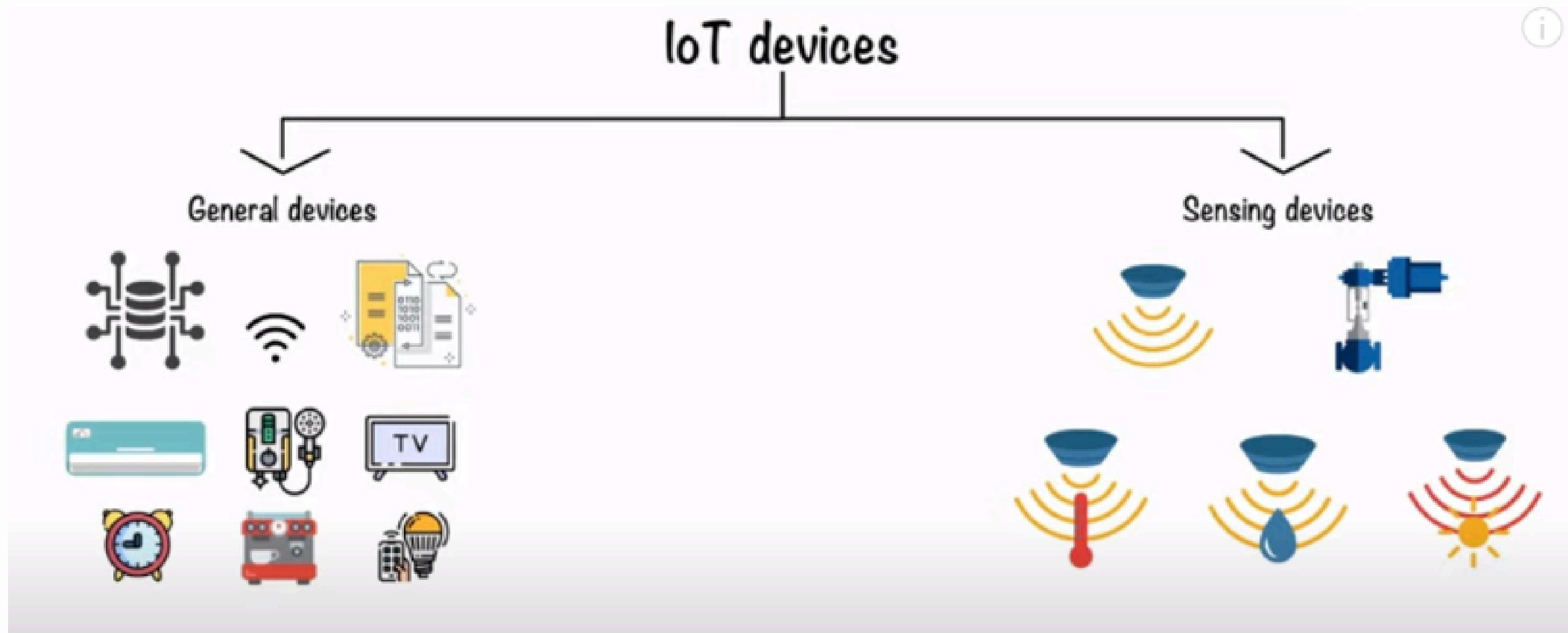
Pengenalan Internet of Things

IoT atau Internet of Things adalah konsep di mana berbagai perangkat bersensor saling terhubung melalui internet untuk mengumpulkan dan mentransfer data.



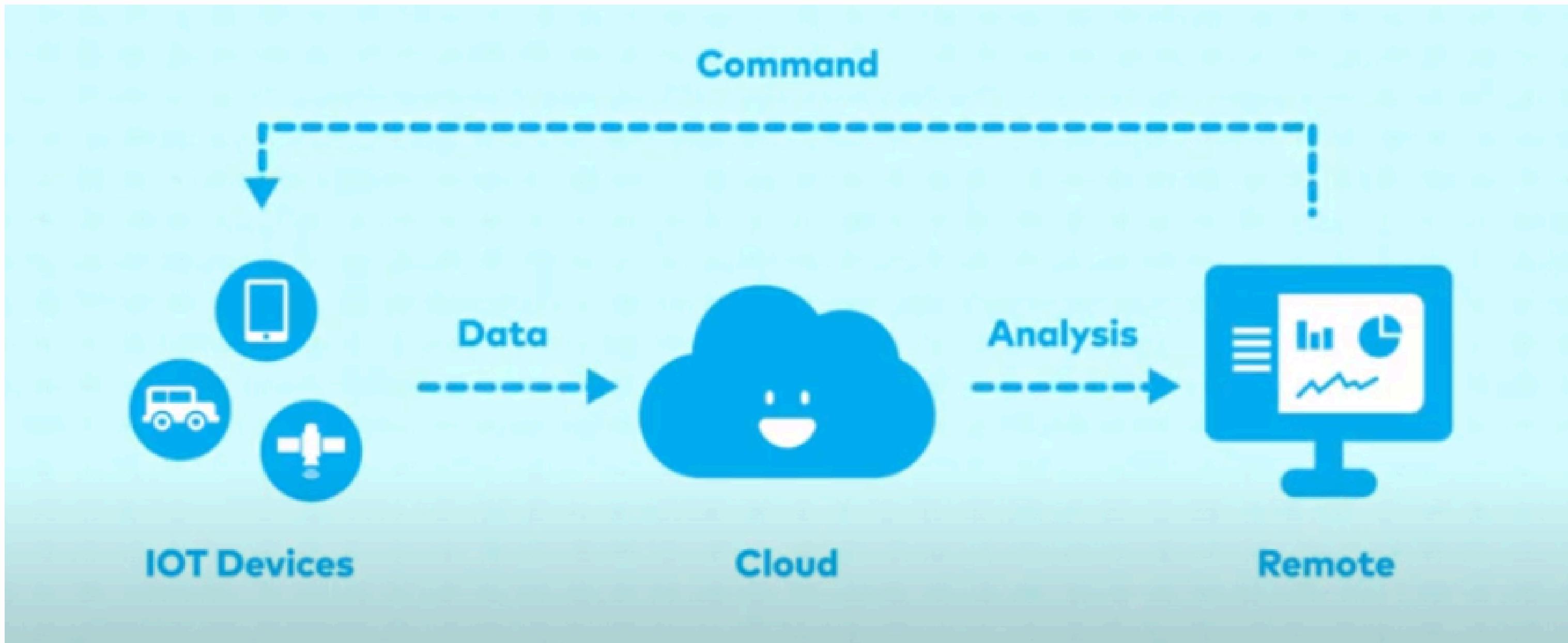


Pengenalan Internet of Things





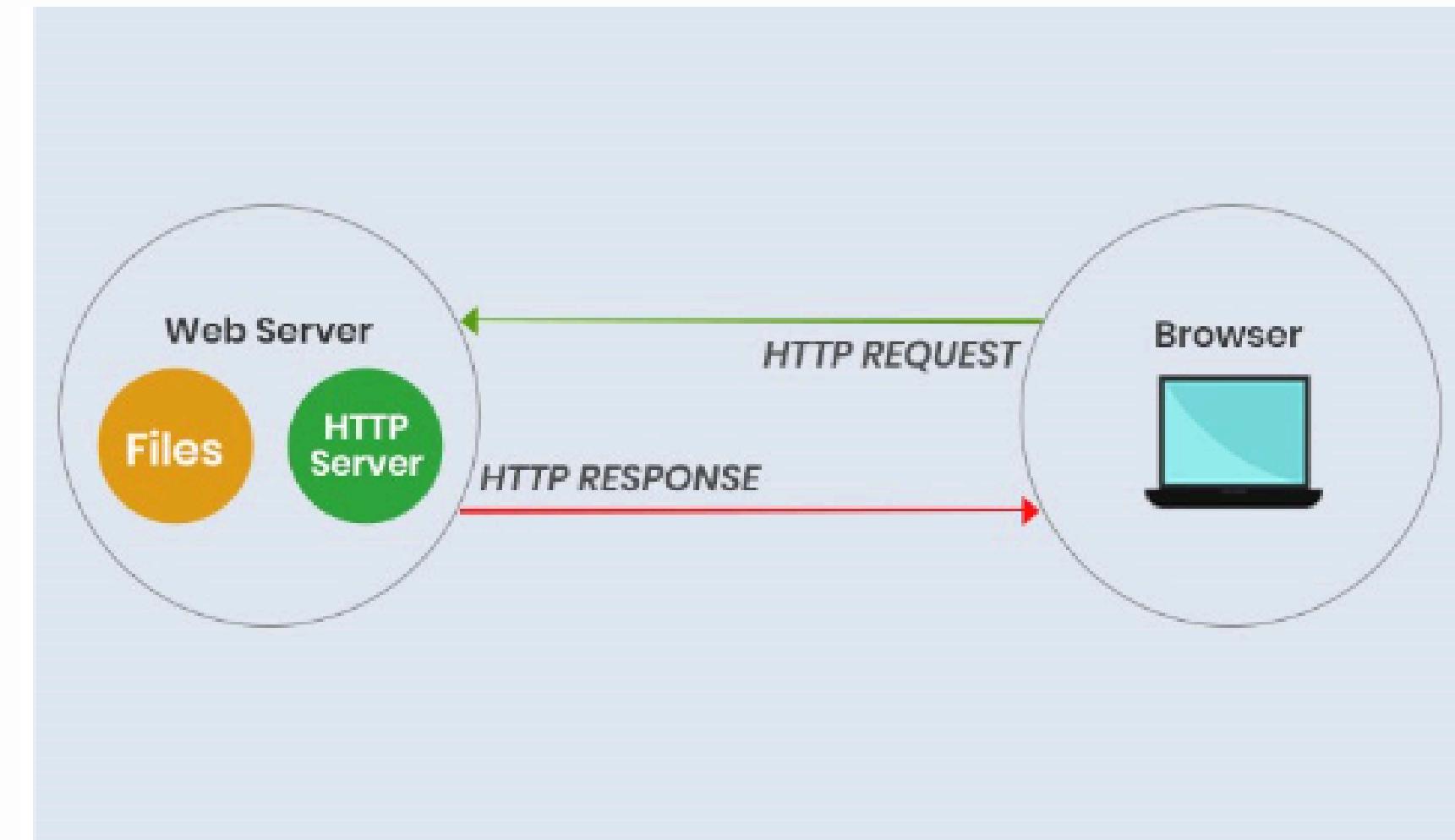
Protokol Internet of Things





1. HTTP(HyperText Transfer Protocol)

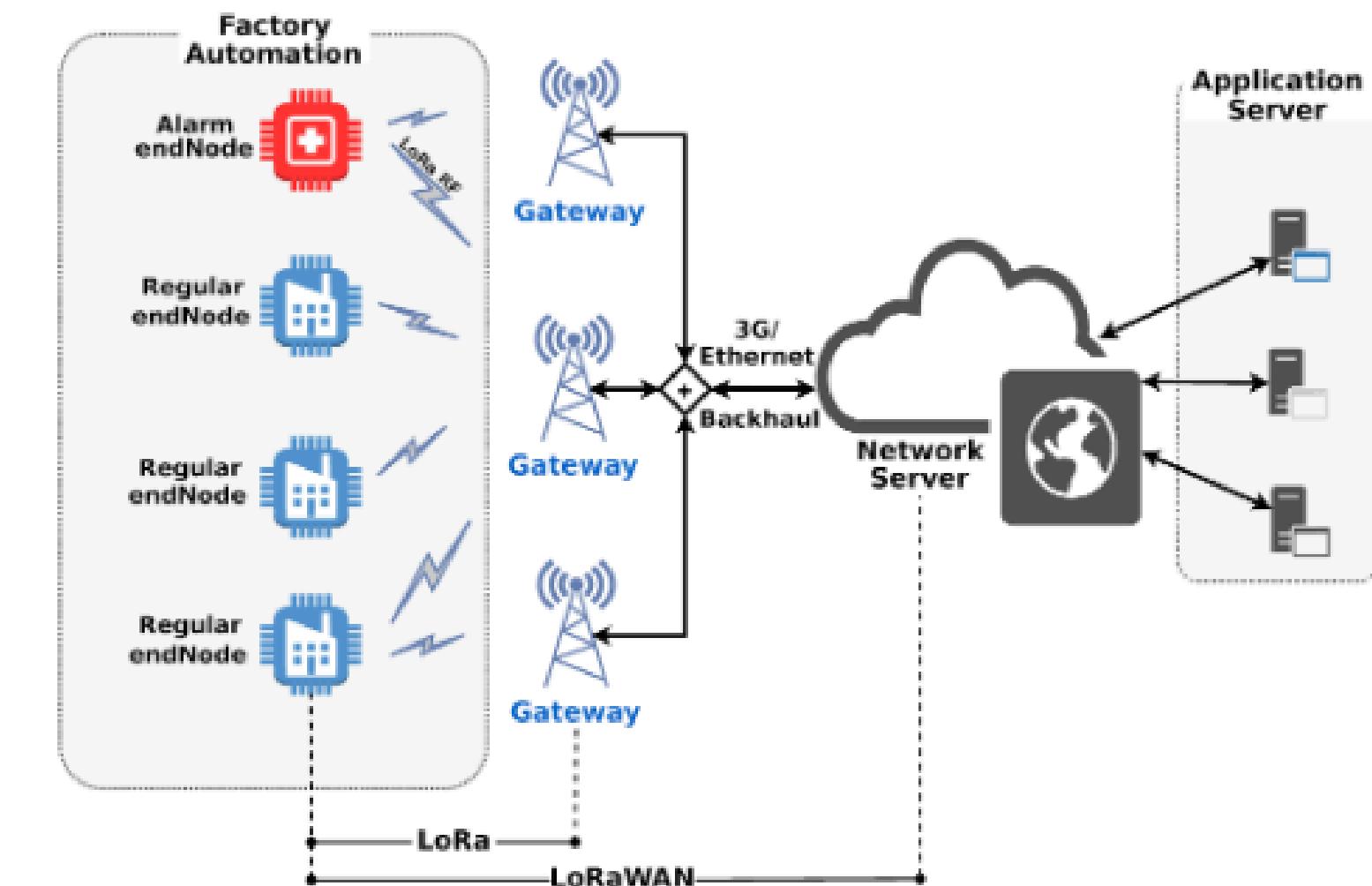
HyperText Transfer Protocol adalah contoh dari IoT network protocols. HTTP dapat mengirimkan berbagai jenis data melalui web. HTTP menjamin data yang dikirim antar perangkat tidak akan rusak karena dibangun di atas TCP (Transmission Control Protocol).





2. LoRaWan(Long Range Wide Area Network)

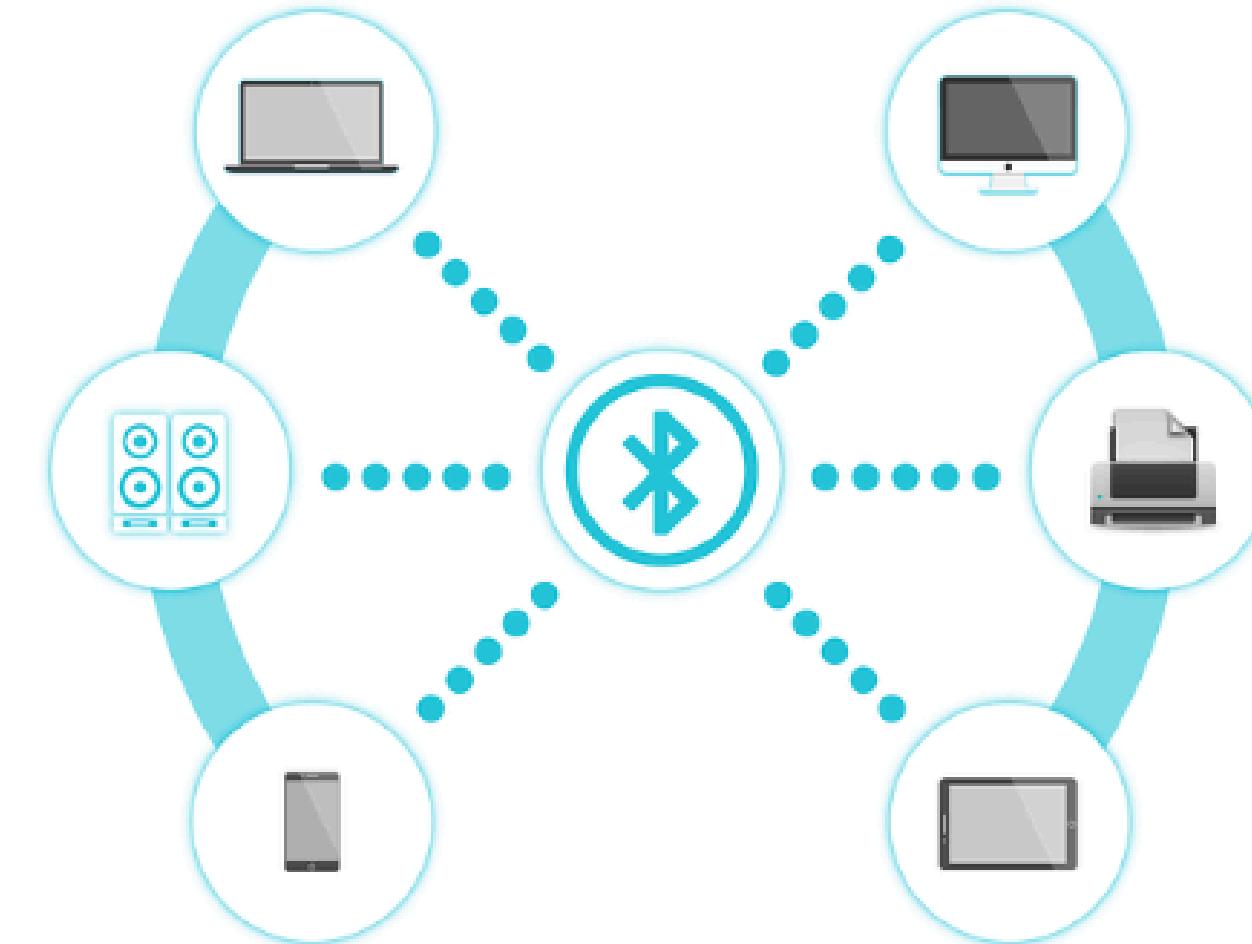
LoRaWAN adalah protokol media access control (MAC) yang didesain sebagai jaringan publik skala besar dengan satu operator. LoRaWan menghubungkan apapun yang beroperasi melalui baterai secara nirkabel menuju Internet baik secara private network maupun global network.





3. Bluetooth

Bluetooth adalah salah satu teknologi nirkabel jarak pendek yang paling banyak dipakai dalam IoT. Protokol ini sangat aman kegiatan komunikasi dan bagus pada transmisi jarak pendek, menghasilkan daya yang rendah, biaya rendah, dan nirkabel antara perangkat elektronik.





4. Zigbee

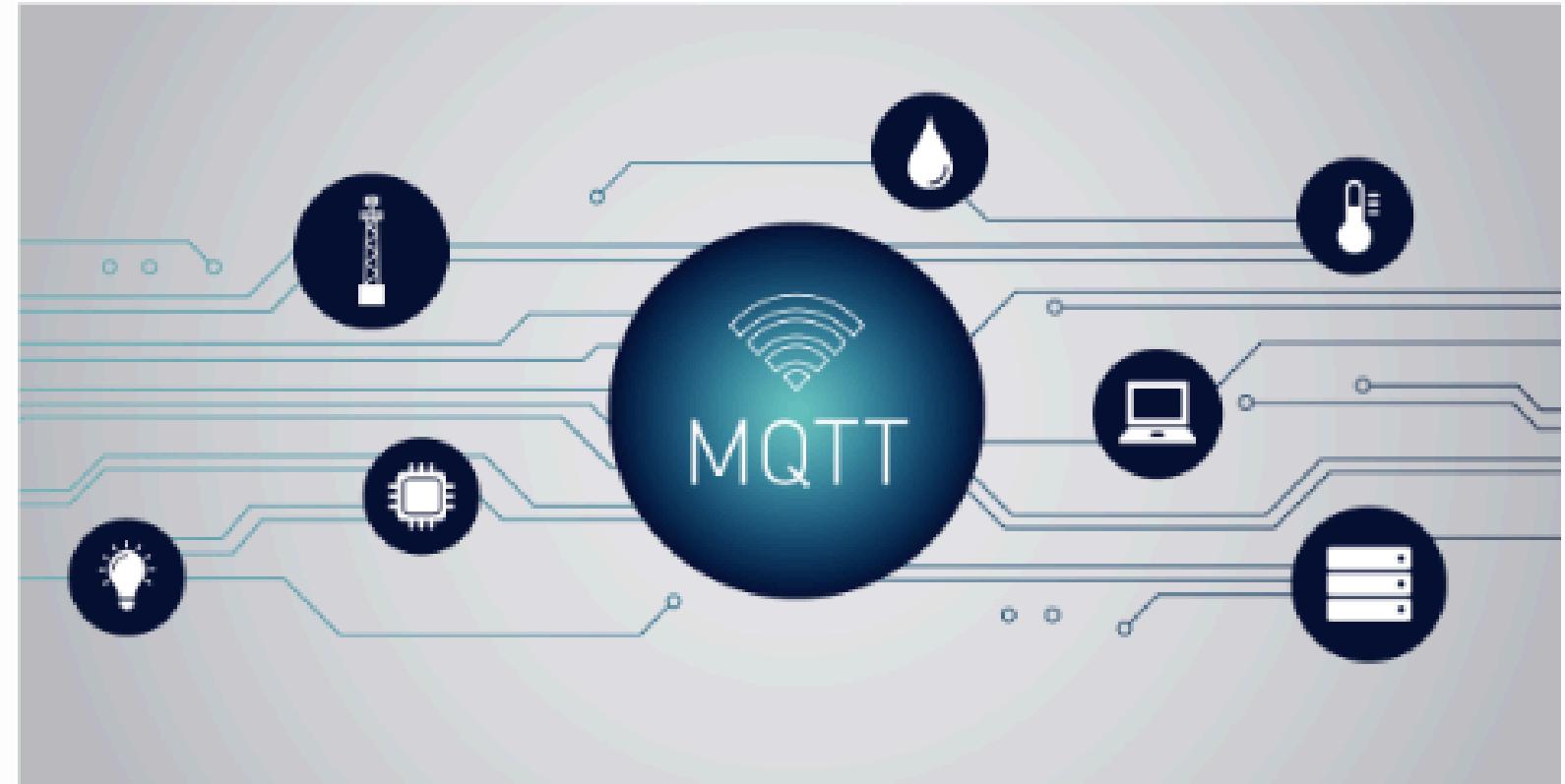
ZigBee merupakan protokol IoT (Internet of Things) yang dikhususkan kepada pelaku industri. Oleh karena itu, ZigBee digunakan untuk transfer data kecepatan rendah dalam jarak pendek.





5. MQTT

Terdapat protokol yang berjalan di atas stack TCP/IP yang bernama MQTT (Message Queuing Telemetry Transport). Dan dirancang untuk machine to machine (M2M) yang tidak memiliki alamat khusus. Contohnya bisa ditemukan pada arduino, raspi atau device lain yang tidak memiliki alamat khusus.





Sejarah MQTT



Andy Stanford-Clark

MQTT awalnya ditemukan oleh Andy Stanford-Clark berkolaborasi dengan Arlen Nipper (sekarang CirrusLink) dalam merancang sebuah protokol untuk berkomunikasi dengan sistem SCADA di industri minyak dan gas.

Sejak Internet of Things menjadi populer, MQTT juga melonjak.

- 2008 – Broker MQTT open-source pertama, Mosquitto diciptakan.
- 2012 – Projek Eclipse Paho menyediakan library klien MQTT open-source untuk Java, C, JavaScript, dan Python dan dari sana daftar library klien MQTT telah bertambah.
- 2014 – MQTT versi 3.1.1 menjadi standar OASIS dan membuka jalan untuk MQTT untuk menjadi standar ISO.
- 2017 – MQTT banyak digunakan oleh perusahaan teknologi terkemuka seperti Amazon, Microsoft, IBM, dll.

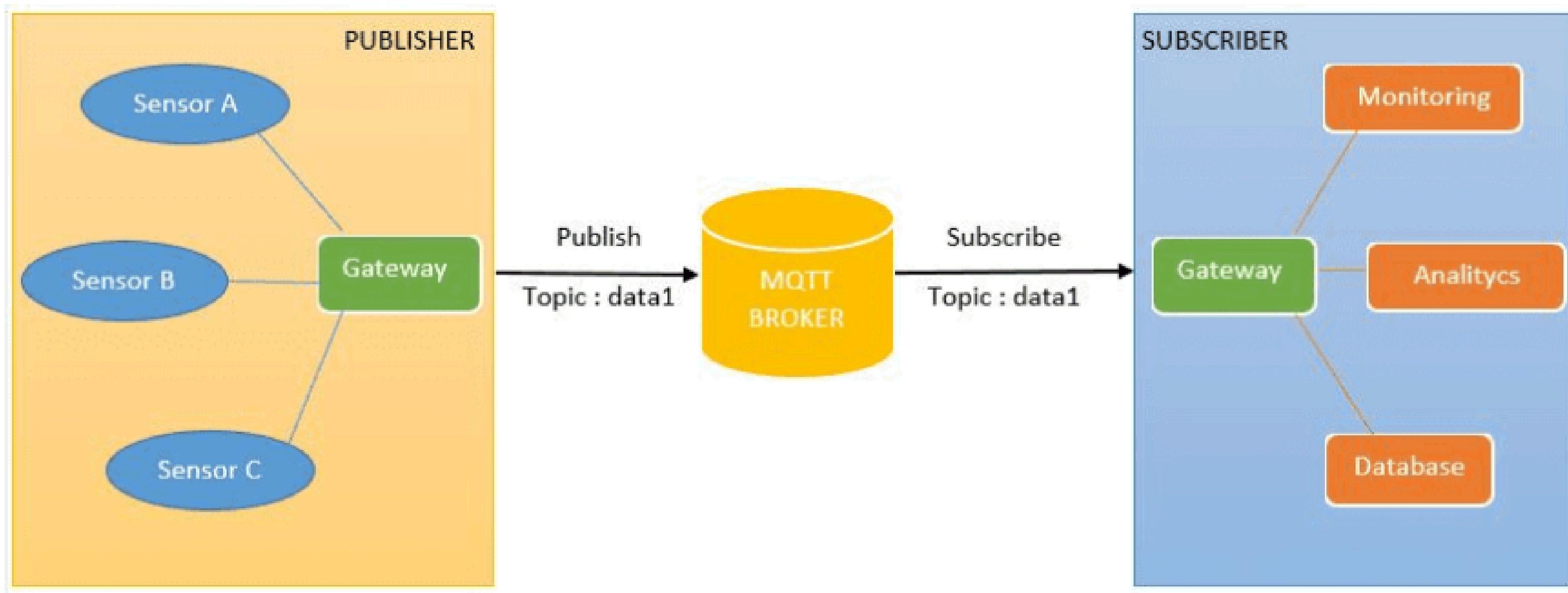


Karakteristik

1. MQTT merupakan sebuah protocol komunikasi yang diperuntukkan untuk komunikasi “machine to machine”
2. MQTT berjalan pada TCP/IP layer.
3. MQTT menggunakan model publish/subscribe/
4. MQTT hanya membutuhkan daya dan bandwidth yang sedikit untuk mengirim data.
5. MQTT hanya dapat mengirim data dalam ukuran yang kecil.
6. MQTT dapat mengirim data dalam beberapa format seperti JSON, XML, Text.

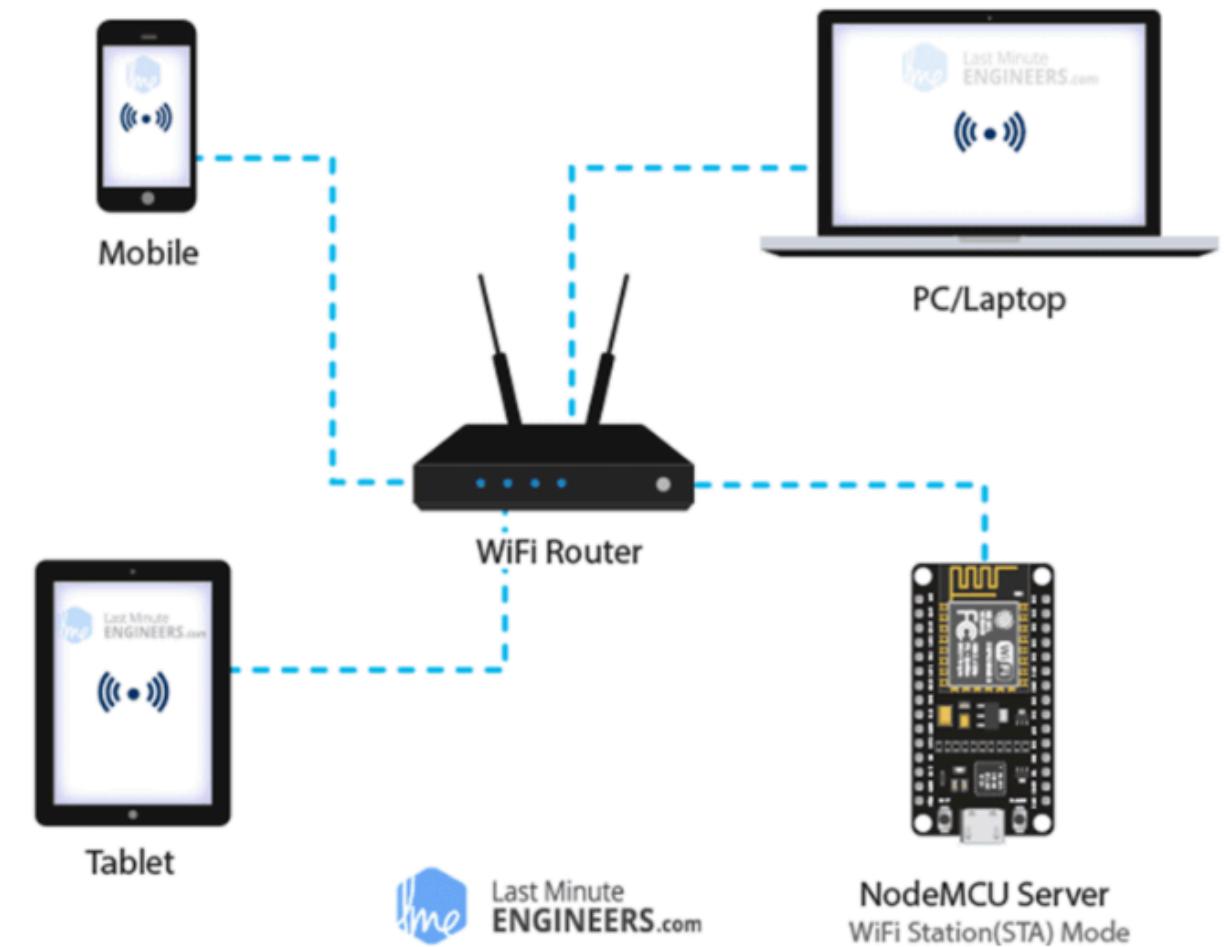
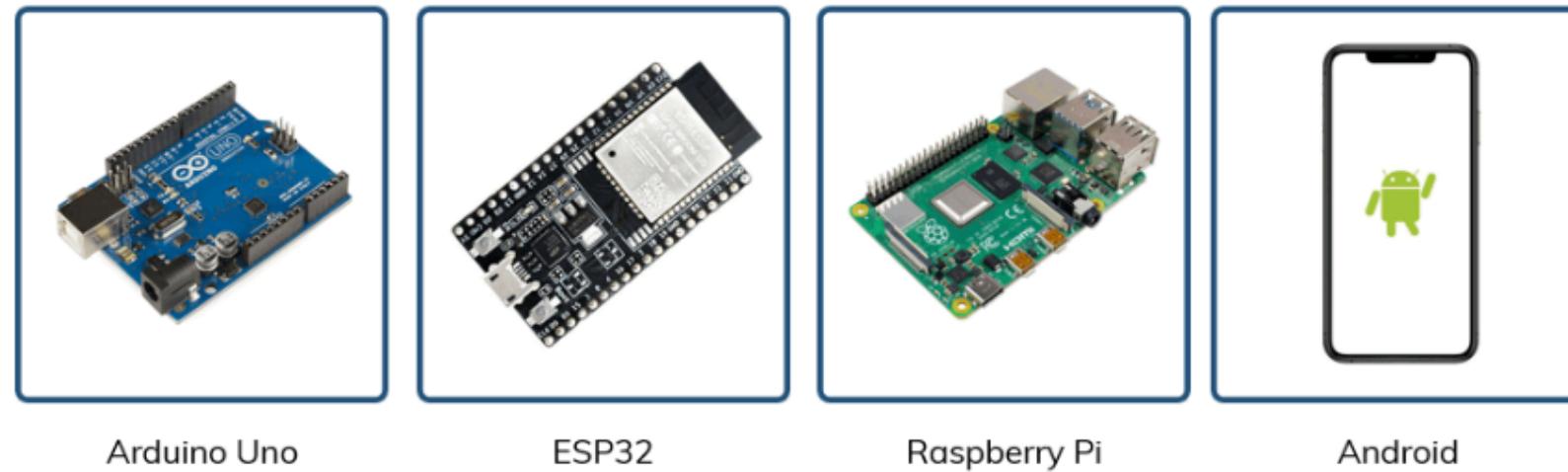


Cara Kerja MQTT



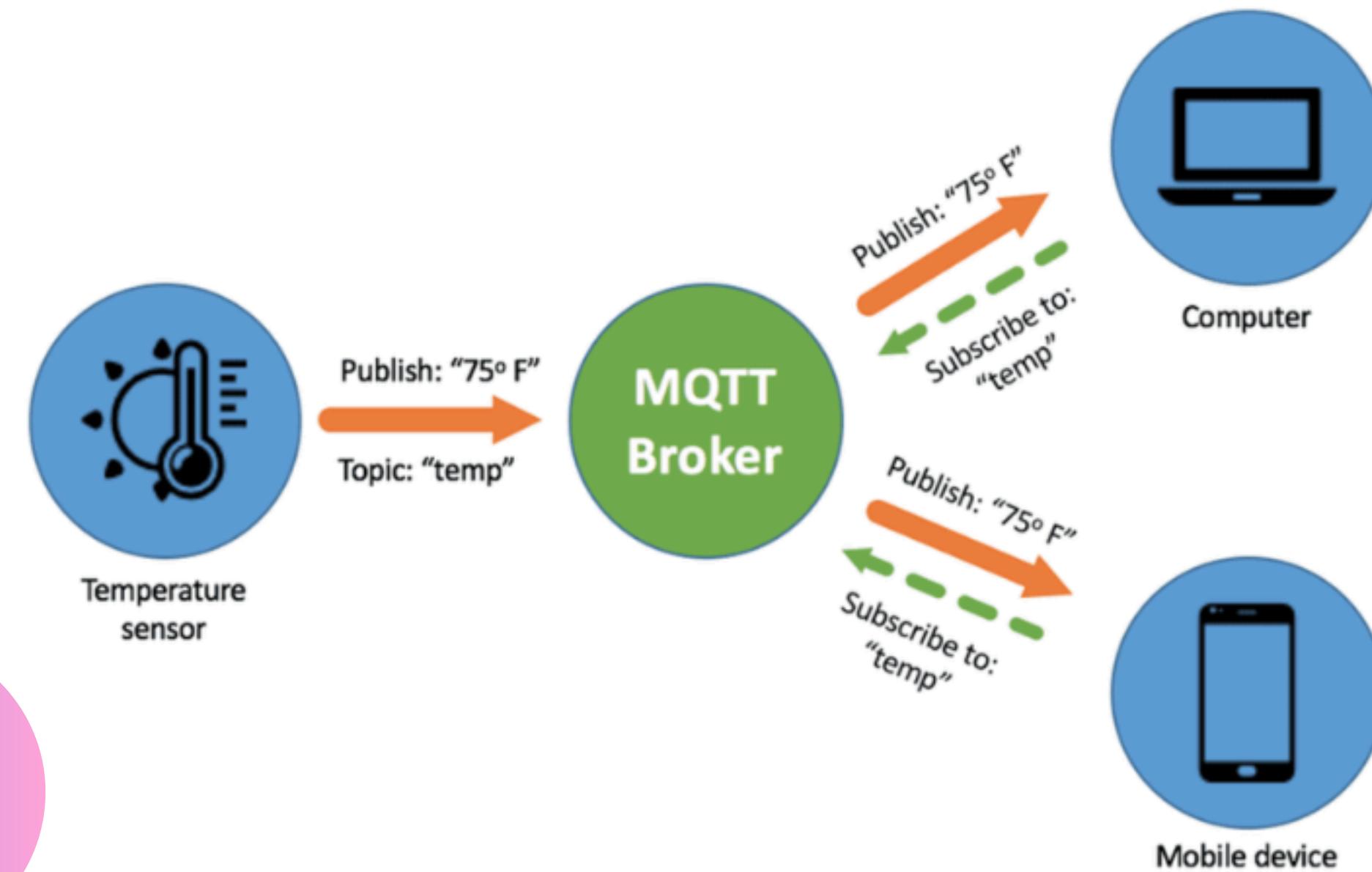


Cara Kerja MQTT



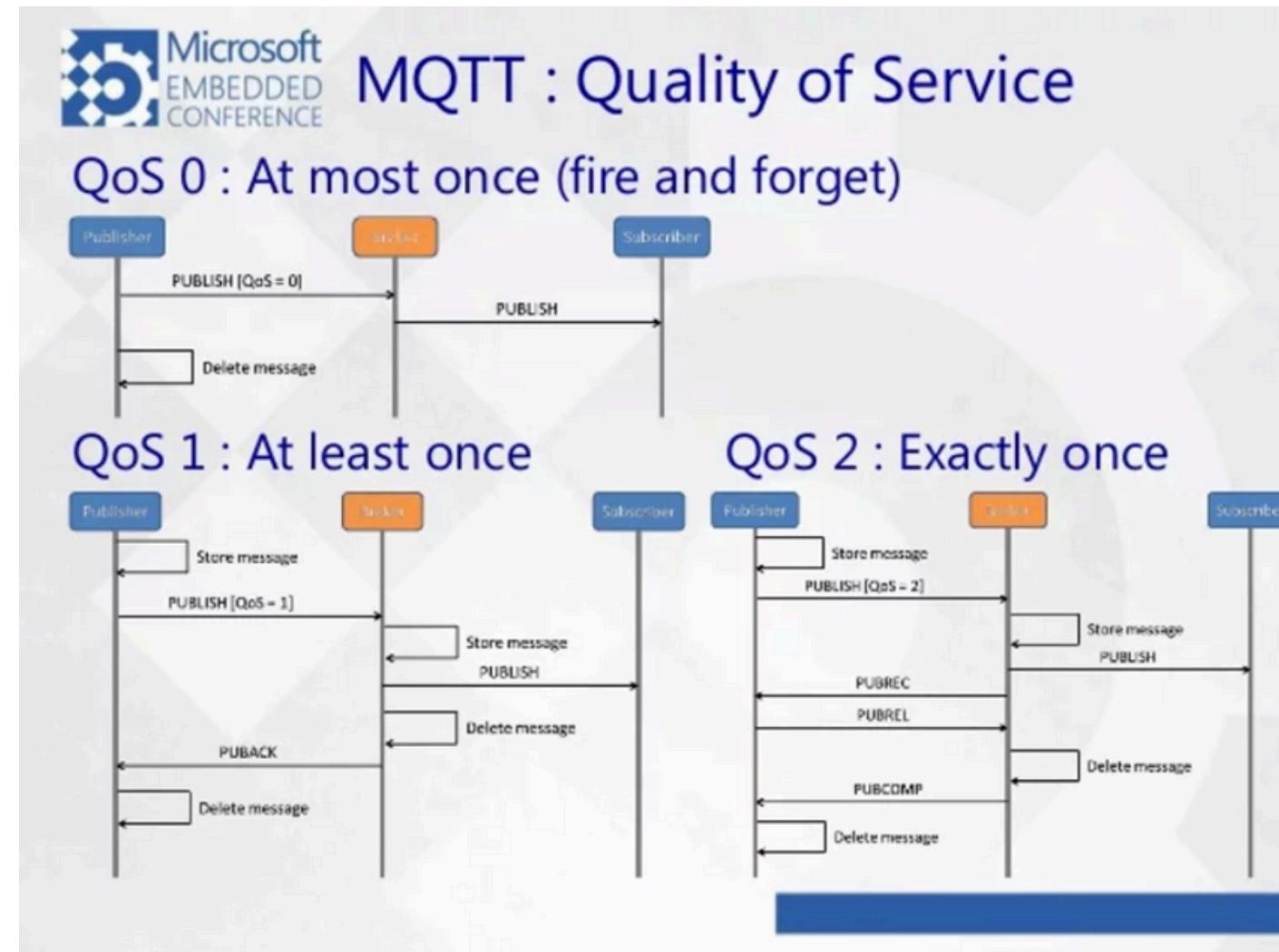


Cara Kerja MQTT





Quality of Service MQTT



Satu lagi yang membuat MQTT berbeda adalah adanya level kualitas dari layanan atau QoS. Jadi pesan yang di publish pasti memiliki satu dari 3 level QoS. Level-level ini memberikan garansi akan konsistensi (reliability) dari pengiriman pesan. Klien dan broker menyediakan mekanisme penyimpanan dan pengiriman kembali dari pesan sehingga meningkatkan konsistensi data akibat kegagalan network, restart dari aplikasi dan sebab-sebab lainnya.

QoS pada MQTT merupakan tingkat kualitas layanan pengiriman pesan pada MQTT. Terdapat 3 level QoS

- QoS level 0 (at most once delivery) : pesan hanya dikirim sekali, setelah pesan dikirim tidak terdapat respon seperti apakah pesan berhasil dikirim.
- QoS level 1 (at least once delivery) : pesan paling sedikit dikirim sekali, jika subscriber tidak menerima pesan maka broker akan mengirimkan respon terhadap publisher bahwa pesan gagal dikirim. Terjadi duplikasi pesan karena pengiriman pesan yang berulang-ulang.
- QoS level 2 (exactly once delivery) : pesan dikirim sekali dan hanya sekali, memastikan ulang bahwa pesan telah diterima oleh subscriber, mencegah terjadinya duplikasi pesan.



MQTT vs HTTP

Perbandingan MQTT dengan HTTP

	MQTT	HTTP
Full Form	Message Queue Telemetry Transport	Hyper Text Transfer Protocol
Design Methodology	Data centric	Document centric
Architecture	Publish/Subscribe	Request/Response
Complexity	Simple	More complex
Message Size	Small, with a compact binary header just two bytes in size	Larger, partly because status detail is text-based
Service Levels	3 quality of service settings	All messages get the same level of service
Data Distribution	1 to 0, 1 to 1, and 1 to n	1 to 1 only
Data Security	Yes	No
		 seeed <small>The IoT Hardware Enabler</small>



MQTT Broker shiftr.io

The page features a prominent green header with the shiftr.io logo on the left, followed by navigation links for Cloud, Desktop, Try, Docs, and Blog, and Sign In/Sign Up buttons on the right. Below the header is a large, bold title: "The IoT Platform for Interconnected Projects". A subtitle below it reads: "Rapidly connect hardware and software using our cloud service and desktop app." At the bottom of the green area is a large, dark green rectangular button with the text "Cloud Service" and "Desktop App" in white.

The IoT Platform for
Interconnected Projects

Rapidly connect hardware and software
using our cloud service and desktop app.

Cloud Service Desktop App

MQTT Documentation



MQTT Client

Need a fully managed MQTT broker?
Get your own Cloud broker and connect up to 100 devices for free. [Get your free account](#)

Connection

Host: mqtt-dashboard.com | Port: 8884 | ClientID: clientId-vfns7FPlas | [Connect](#)

Username: Password: Keep Alive: 60 | SSL: Clean Session:

Last-Will Topic: Last-Will QoS: 0 | Last-Will Retain:

Last-Will Message:

Publish

Subscriptions

Messages



ESP32

Koneksi MQTT Broker

WOKWi
Simulate IoT Projects in Your Browser

Docs 

Discord Community Facebook Group

We're adding new stuff all the time, want us to tell You?

Your email address Stay Updated

Featured Simulation Projects

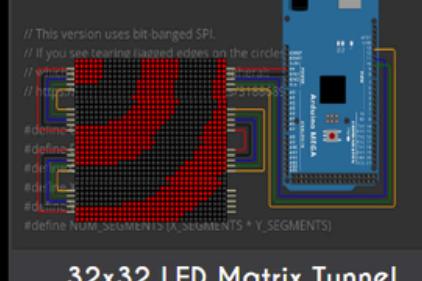
Simon Game with Score

/*
 * Simon Game for WOKWi
 * Copyright (C) 2022 Uri Shani
 * Released under the MIT License
 */
#include "pitches.h"

/* Constants - define pin numbers for LEDs, buttons and speaker, and also the game tones: */

Nano Pong

/*
 * A simple Pong game.
 * https://notabug.org/MaverickW/nano_pong
 * Based on Arduino Pong by mkrueger
 * https://github.com/mkrueger
 */
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

32x32 LED Matrix Tunnel

// This version uses bit-banged SPI.
// If you see tearing/hashed edges on the circles
// try this:
#define SPI_MOSI 10
#define SPI_MISO 11
#define SPI_SCK 12
#define SPI_CS 13
#define Neo_Segment_X 14 // SEGMENTS + Y_SEGMENTS)

#include <MD_MAX72xx.h>


/**
 * Arduino Calculator
 * https://notabug.org/robbert/Calculator
 */

/**
 * Arduino Digital Alarm Clock
 * https://notabug.org/robbert/DigitalClock
 */



ESP32

Koneksi MQTT Broker

```
#include <WiFi.h>
#include <MQTT.h>

const char ssid[] = "ssid";
const char pass[] = "pass";

WiFiClient net;
MQTTClient client;

unsigned long lastMillis = 0;
```

```
void connect() {
    Serial.print("checking wifi...");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(1000);
    }

    Serial.print("\nconnecting...");
    while (!client.connect("arduino", "public", "public")) {
        Serial.print(".");
        delay(1000);
    }

    Serial.println("\nconnected!");
    client.subscribe("hello");
}

void messageReceived(String &topic, String &payload) {
    Serial.println(topic + ": " + payload);
}
```



ESP32

Koneksi MQTT Broker

```
void setup() {
    Serial.begin(115200);

    // start wifi and mqtt
    WiFi.begin(ssid, pass);
    client.begin("public.cloud.shiftr.io", net);
    client.onMessage(messageReceived);

    connect();
}
```

```
void loop() {
    client.loop();
    delay(10);

    // check if connected
    if (!client.connected()) {
        connect();
    }

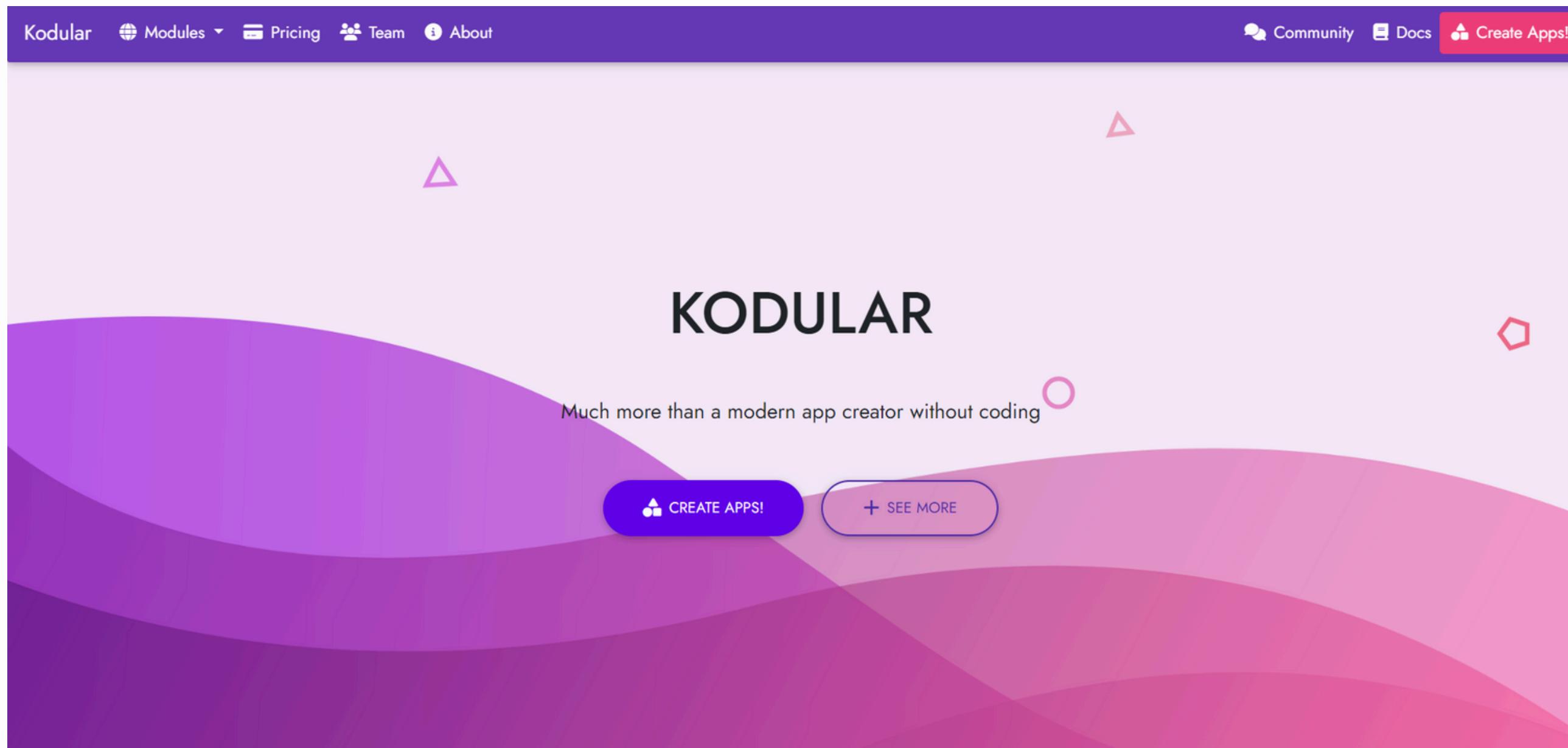
    // publish a message roughly every second.
    if (millis() - lastMillis > 1000) {
        lastMillis = millis();
        client.publish("/hello", "world");
    }
}
```

IRON



Kodular

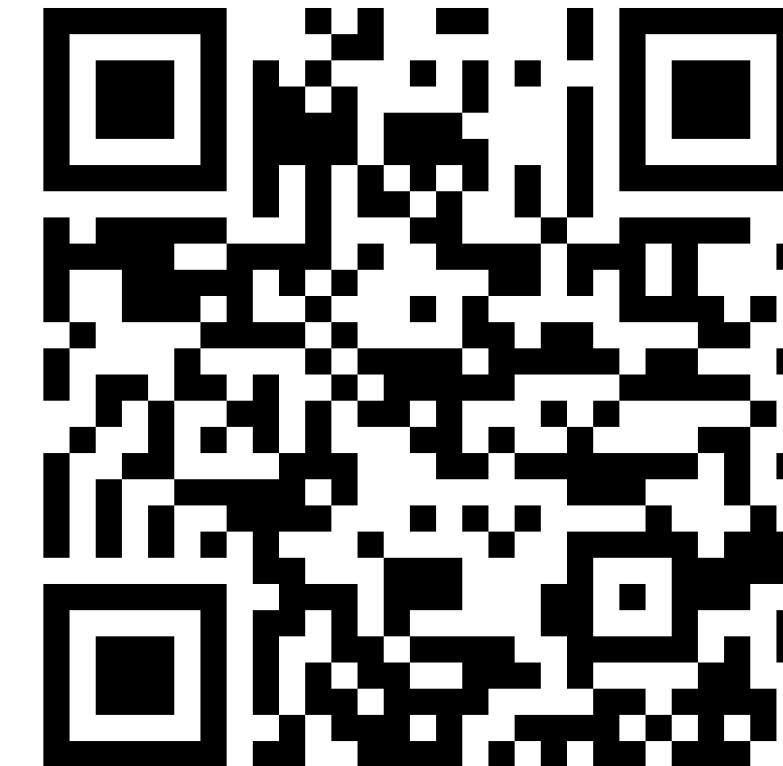
Koneksi MQTT Broker





Kodular

Koneksi MQTT Broker

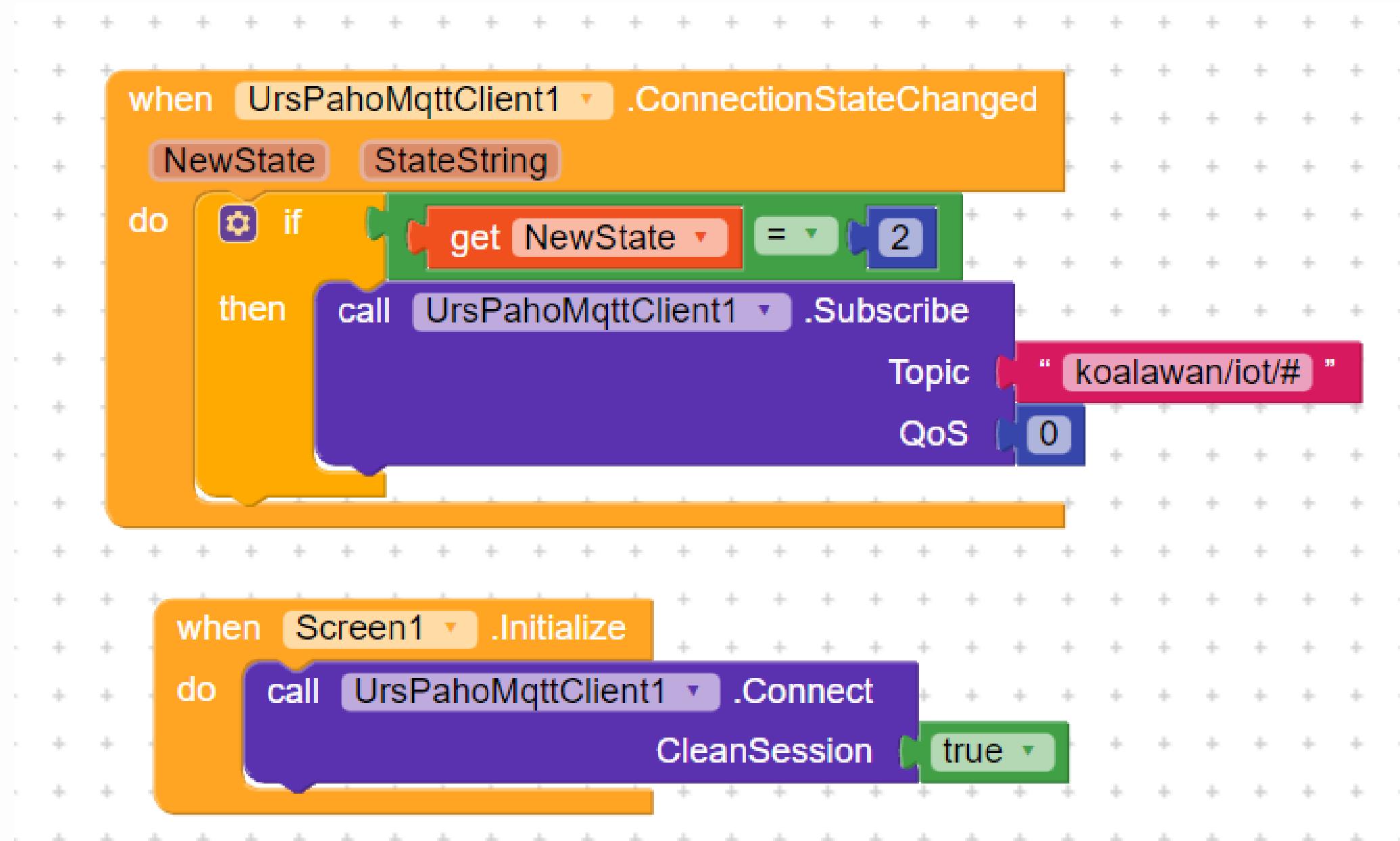


<https://s.id/LibraryMQTTKodular>



Kodular

Koneksi Subscribe Topic





```
when UrsPahoMqttClient1 .Message Received
    Topic Payload Message RetainFlag DupFlag
do
    if "koalawan/iot/message" = get Topic
        then set message .Text to get Message
    if "koalawan/iot/temperature" = get Topic
        then set temp .Text to join get Message
            "°C"
    if "koalawan/iot/humidity" = get Topic
        then set humd .Text to join get Message
            "%"
    if "koalawan/iot/distance" = get Topic
        then set distance .Text to join get Message
            "CM"
    if "koalawan/iot/status_temp" = get Topic
        then set status_temp .Text to get Message
    if "koalawan/iot/status_humd" = get Topic
        then set status_humd .Text to get Message
```

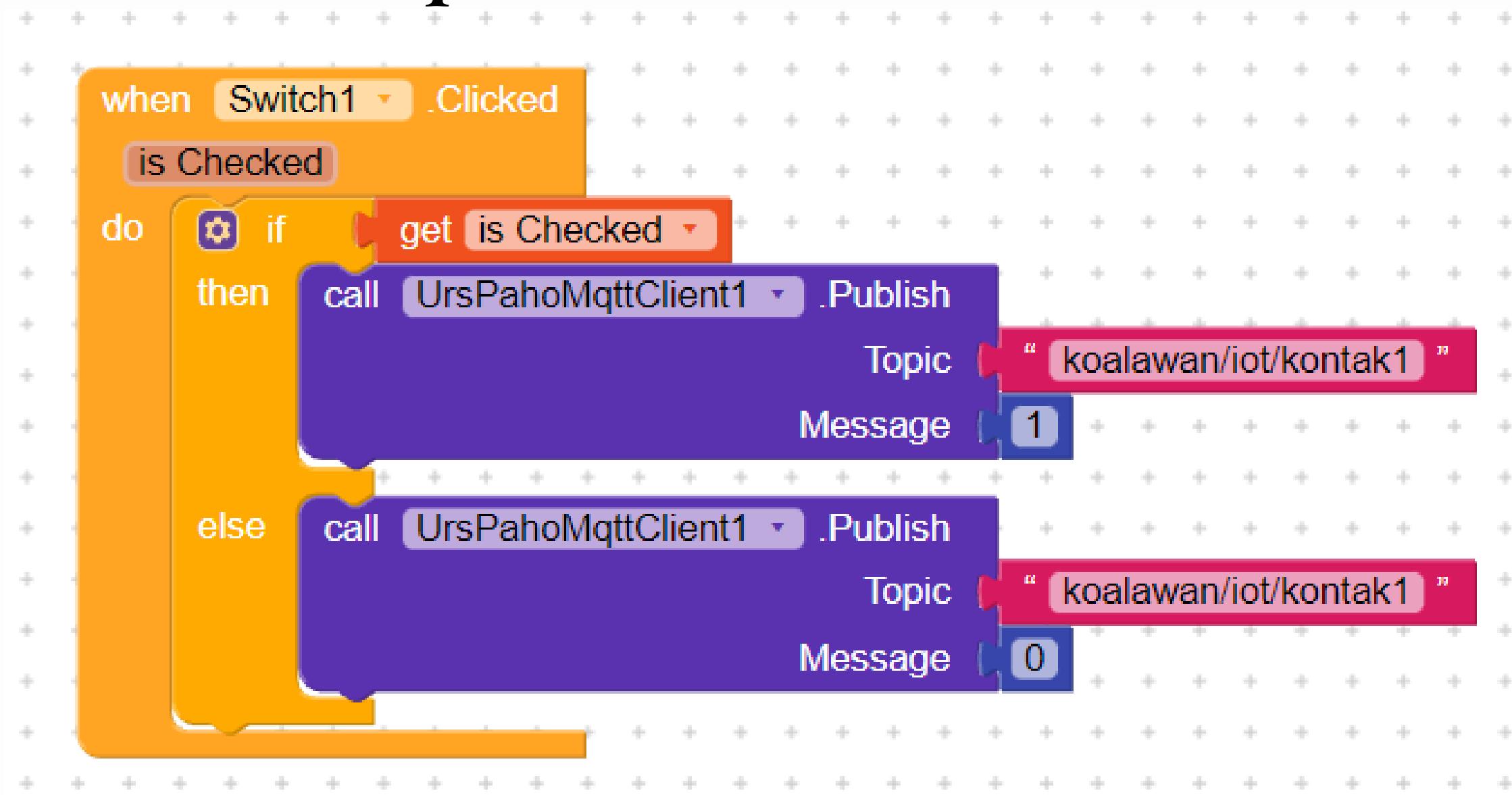
Kodular

Get Topic Broker



Kodular

Publish Topic Broker



IRON



Thank You