



MODUL PRAKTIKUM SISTEM OPERASI DAN JARINGAN KOMPUTER

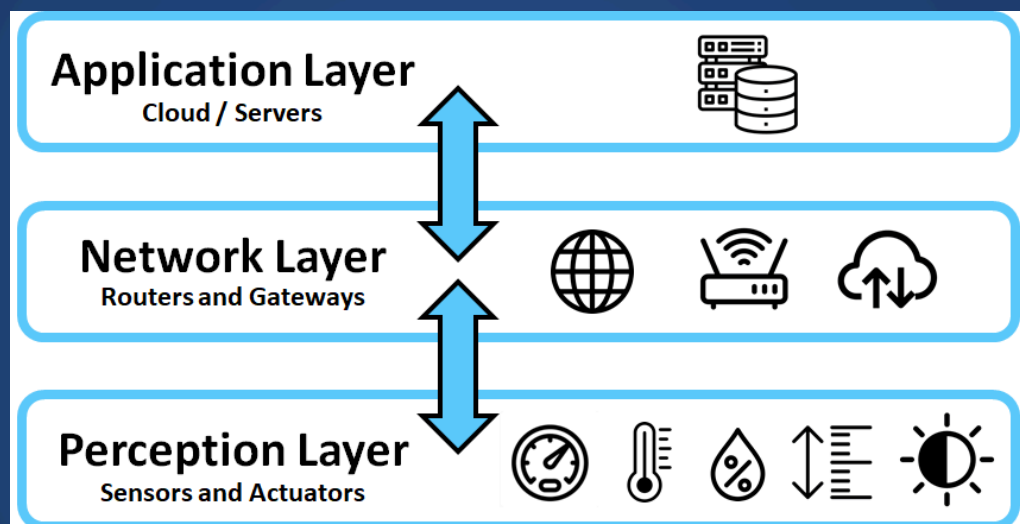
9. Internet of Things

Internet of Things

A. Definisi

Internet of Things adalah sebuah konsep yang bertujuan memperluas manfaat dari koneksi internet yang terhubung secara terus-menerus. Prinsip kerja perangkat IoT adalah memberikan objek di dunia nyata yang dapat dikalikan dan direpresentasikan pada sistem komputer dalam bentuk data. (Setiawan Aep, Arlitasari Erlin, Zuhri Mahfuddin, Hendriana Andri 2022).

B. Arsitektur Sederhana IoT



Sumber: www.iadiy.com

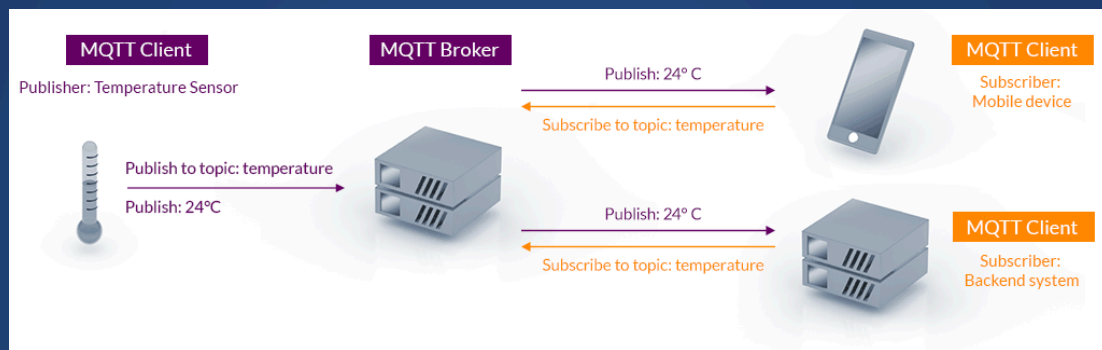
Layer	Contoh
Perception Layer	LED, DHT11, CCTV
Network Layer	Wifi, Bluetooth, Zigbee, LoRa
Application Layer	Aplikasi smart home, aplikasi monitoring suhu

C. Pengenalan Protokol pada IoT

Protokol	Penjelasan
----------	------------

MQTT	Protokol publish-subscribe yang ringan dan efisien untuk komunikasi antar perangkat dengan bandwidth rendah.
HTTP/HTTPS	Protokol berbasis HTTP yang umum digunakan untuk komunikasi dengan perangkat IoT yang memerlukan transfer data yang lebih besar.
LoRaWAN	Protokol komunikasi jarak jauh dengan daya rendah yang digunakan untuk koneksi perangkat IoT di area luas.
Zigbee	Protokol jaringan mesh nirkabel yang efisien untuk area lokal dengan daya rendah, terutama dalam aplikasi rumah pintar.

D. Protokol MQTT



Sumber: mqtt.org

Komponen MQTT:

Komponen	Penjelasan
MQTT Client	MQTT Client adalah setiap perangkat dari server ke mikrokontroler yang menjalankan <i>library</i> MQTT. Jika klien mengirim pesan, ia bertindak sebagai publisher , dan jika klien menerima pesan, ia bertindak sebagai subscriber .
MQTT Broker	MQTT Broker adalah sistem backend yang mengkoordinasikan pesan antara klien yang berbeda. Tanggung jawab broker meliputi menerima dan memfilter pesan, mengidentifikasi klien yang berlangganan setiap pesan, serta mengirimkan pesan kepada klien.
MQTT Connect	Klien memulai koneksi dengan mengirimkan pesan

CONNECT ke MQTT Broker. Broker mengonfirmasi bahwa koneksi telah dibuat dengan merespons menggunakan pesan CONNACK. Baik klien MQTT maupun broker memerlukan layer TCP/IP untuk berkomunikasi.

Udah paham cara kerja PDU? HTTP? Uehehe....

Cara kerja:

1. MQTT client membuat koneksi dengan broker.
2. Jika terhubung, klien melakukan *publish*, *subscribe*, atau keduanya terhadap suatu pesan dari **topic** tertentu.
3. Broker menerima pesan dari client dan diteruskan ke client lain yang menjadi *subscriber* dari topic tersebut.

Hivemq menjadi server yang umum dipakai untuk keperluan MQTT dalam tahap development. Selain itu, untuk aplikasi yang umum dipakai adalah mosquitto.

E. Persyaratan Aplikasi

Hivemq menjadi server yang umum dipakai untuk keperluan MQTT dalam tahap development. Selain itu, untuk aplikasi yang umum dipakai adalah mosquitto.

Berikut prosedur penginstalan hingga penggunaan mosquitto.

1. Install mosquitto

Windows	https://mosquitto.org/files/binary/win64/mosquitto-2.0.20-install-windows-x64.exe https://mosquitto.org/files/binary/win32/mosquitto-2.0.20-install-windows-x86.exe
Linux	snap install mosquitto sudo apt install mosquitto && sudo apt install mosquitto-clients <i>^ franklin rasis gaada tutor bwt arch</i>
Mac	brew install mosquitto

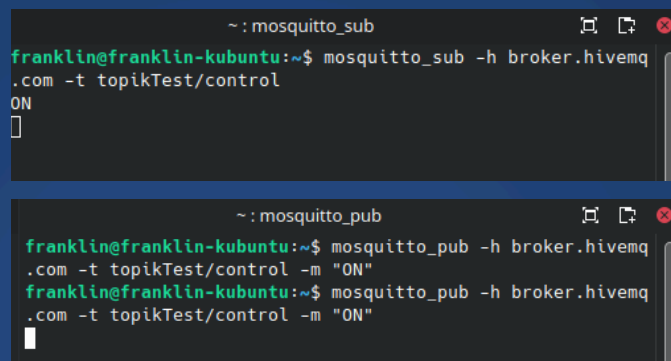
2. Jalankan command untuk melakukan subscribe ke topic *topikTest/control*.

```
mosquitto_sub -h broker.hivemq.com -t topikTest/control
```

3. Jalankan command untuk melakukan publish.

```
mosquitto_pub -h broker.hivemq.com -t topikTest/control -m "ON"
```

4. Lihat hasilnya menjadi seperti berikut

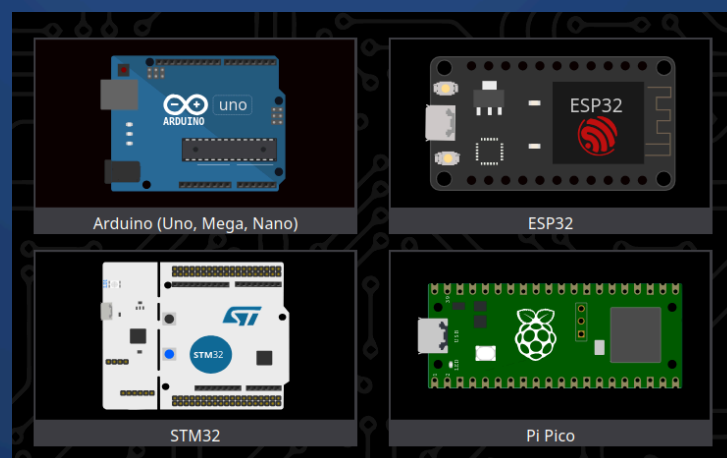


The image shows two terminal windows. The top window is titled '~: mosquitto_sub' and shows the command 'mosquitto_sub -h broker.hivemq.com -t topikTest/control' being executed, with the output 'ON' displayed below. The bottom window is titled '~: mosquitto_pub' and shows the command 'mosquitto_pub -h broker.hivemq.com -t topikTest/control -m "ON"' being executed twice, with the output 'ON' displayed below each execution.

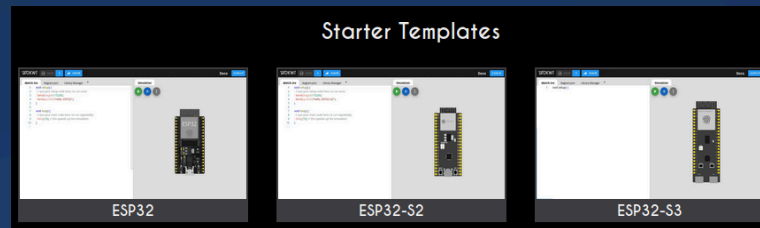
F. Penggunaan Wokwie

Membuat program sederhana (Belum memenuhi IoT concept)

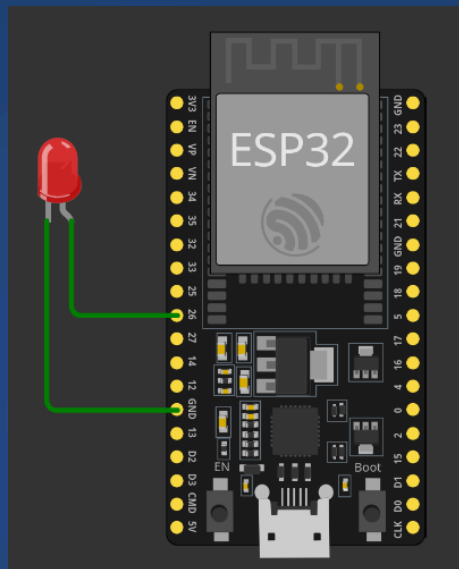
1. Buka <https://wokwi.com/>
2. Pilih ESP32 pada Simulate with Wokwi Online



3. Scroll ke bawah hingga section Starter Templates. Pilih ESP32 pada pojok kiri atas.



4. Buat arsitektur sederhana seperti berikut. Hubungkan kaki negatif LED ke pin GND(Ground) dan kaki positif ke pin 26.



5. Buat program sederhana seperti berikut.

```
#define ledPin 26

void setup() {
  Serial.begin(115200);
  Serial.println("SETUP ESP32...");
  pinMode(ledPin, OUTPUT);
}

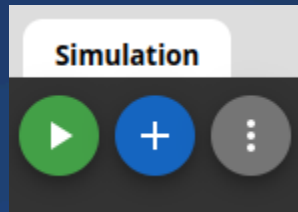
void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);

  digitalWrite(ledPin, LOW);
}
```

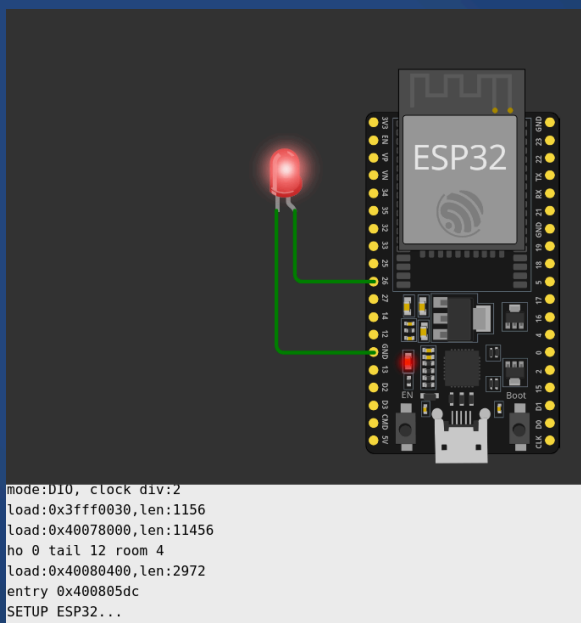


```
delay(1000);  
}
```

6. Jalankan program dengan menekan tombol play pada menu simulation di pojok kiri atas.

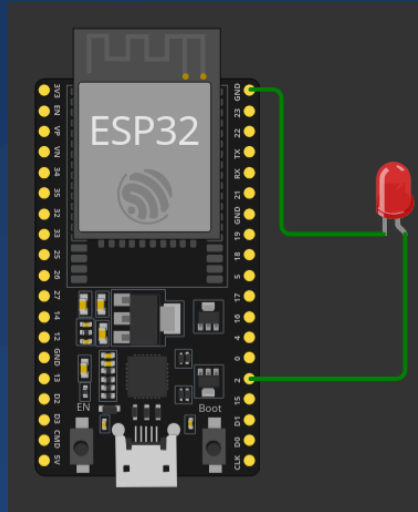


7. Tunggu dan dapatkan hasil seperti berikut. LED berubah kondisi cahaya dan serial monitor menampilkan hasil seperti berikut.

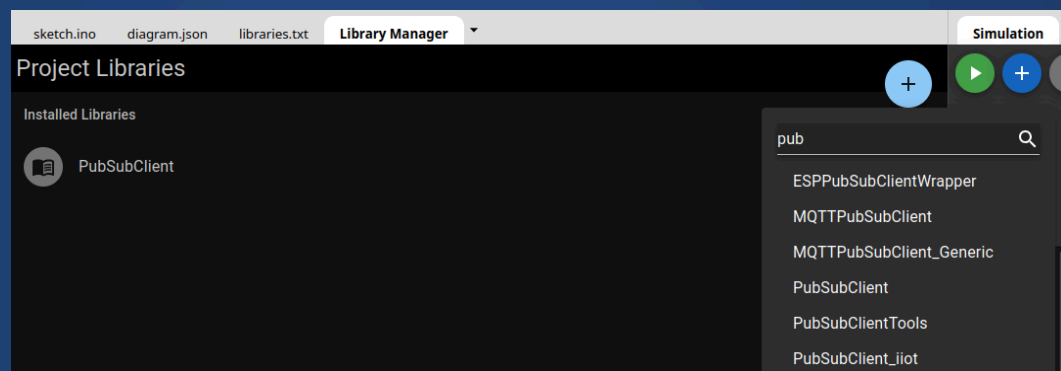


Membuat program menghidup matikan LED menggunakan MQTT

1. Buat arsitektur sederhana seperti berikut. Hubungkan kaki negatif LED ke GND dan kaki positif ke pin 2



2. Tambah library MQTT dengan nama PubSubClient.



3. Buat program dengan flow dasar seperti berikut.

```
#include <WiFi.h>
#include "PubSubClient.h"

// PIN and MQTT definitions
const int ledPin = 2;
const char* ssid = "Wokwi-GUEST";
const char* password = "";
const char* mqttServer = "broker.hivemq.com";
const int mqttPort = 1883;

// State and client declarations
bool ledState = false;
WiFiClient espClient;
```



```

PubSubClient client(espClient);

void setup() {
    Serial.begin(115200);
    pinMode(ledPin, OUTPUT);
    connectWiFi();
    setupMQTT();
}

void loop() {
    if (!client.connected()) reconnectMQTT();
    client.loop();
    delay(2000); // Publish every 2 seconds
}

```

4. Buat function *connectWifi()* untuk membuat koneksi dengan wifi Wokwi.

```

void connectWiFi() {
    Serial.printf("Connecting to %s", ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nWiFi connected");
}

```

5. Buat function *setupMQTT()* untuk menyiapkan koneksi dengan MQTT.

```

void setupMQTT() {
    client.setServer(mqttServer, mqttPort);
    client.setCallback(handleMqttMessage);
}

```

6. Buat function *reconnectMQTT()* untuk membuat koneksi dengan MQTT.

```

void reconnectMQTT() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        String clientId = "clientId-" + String(random(1000));

```

```

if (client.connect(clientId.c_str())) {
    Serial.println(" connected");
    client.subscribe("topikTest/control");
} else {
    Serial.print("failed, rc=");
    Serial.println(client.state());
    delay(5000);
}
}
}

```

7. Buat function *handleMqttMessage()* untuk menangani mqtt message.

```

void handleMqttMessage(char* topic, byte* payload, unsigned
int length) {
    String message;
    for (unsigned int i = 0; i < length; i++) message +=
(char)payload[i];
    Serial.printf("Message arrived [%s] %s\n", topic,
message.c_str());

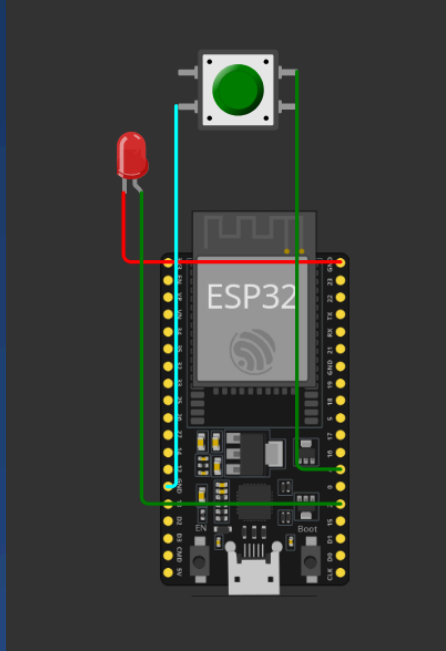
    ledState = (message == "ON");
    digitalWrite(ledPin, ledState ? HIGH : LOW);
    Serial.println(ledState ? "LED turned ON" : "LED turned
OFF");
}

```

8. Run program dan uji coba menggunakan program dibawah atau menggunakan **mosquitto**.

Membuat program pengontrol LED jarak jauh menggunakan MQTT (Skip aja g sehh? hehe..)

1. Buat arsitektur sederhana seperti berikut. Hubungkan LED (opsional) dan button dengan ESP32. Untuk button, hubungkan L.2 dengan GND dan R.1 dengan 4.



2. Buat program dengan flow dasar seperti berikut.

```
// Sesuaikan bagian ini
// . . .

void setup() {
  Serial.begin(115200);
  Serial.println("Hello, ESP32!");
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW); // Start with LED off
  connectWiFi();
  client.setServer(mqttServer, mqttPort);
}

void loop() {
  if (!client.connected()) reconnectMQTT();
  client.loop();

  int buttonState = digitalRead(buttonPin);
  Serial.print("Button Pin Output: ");
  Serial.println(buttonState);
}
```

```

if (buttonState == LOW) {
    toggleLEDAndSendMessage();
}

delay(200); // Check button state every 200 ms
}

```

3. Buat program *toggleLEDAndSendMessage()*.

```

void toggleLEDAndSendMessage() {
    digitalWrite(ledPin, HIGH); // Turn on LED
    client.publish("topikTest/control", "ON"); // Send "ON"
    message
    Serial.println("LED turned ON, message sent: ON");
    delay(1000); // Keep LED on for 1 second
    digitalWrite(ledPin, LOW); // Turn off LED
    client.publish("topikTest/control", "OFF"); // Send "OFF"
    message
    Serial.println("LED turned OFF, message sent: OFF");
}

```

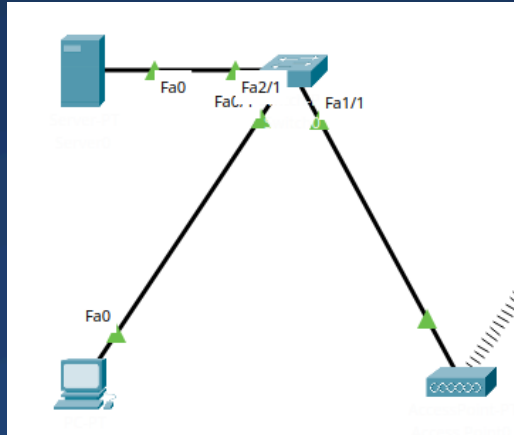
Source code:

1. Program LED Sederhana:
<https://wokwi.com/projects/413733879445638145>
2. Program LED menggunakan MQTT:
<https://wokwi.com/projects/413725325980032001>
3. Program Button Menggunakan MQTT:
<https://wokwi.com/projects/413725230001751041>

G. Penggunaan CPT

Membuat program sederhana dengan control melalui web

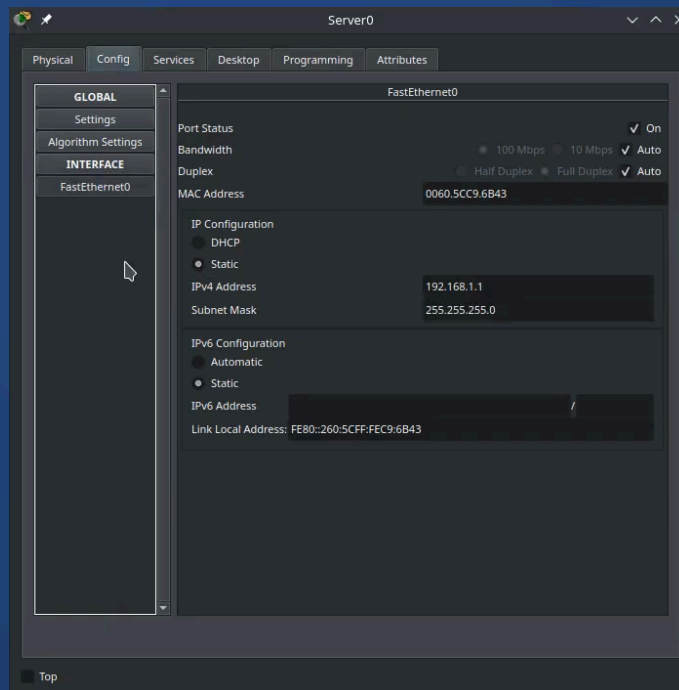
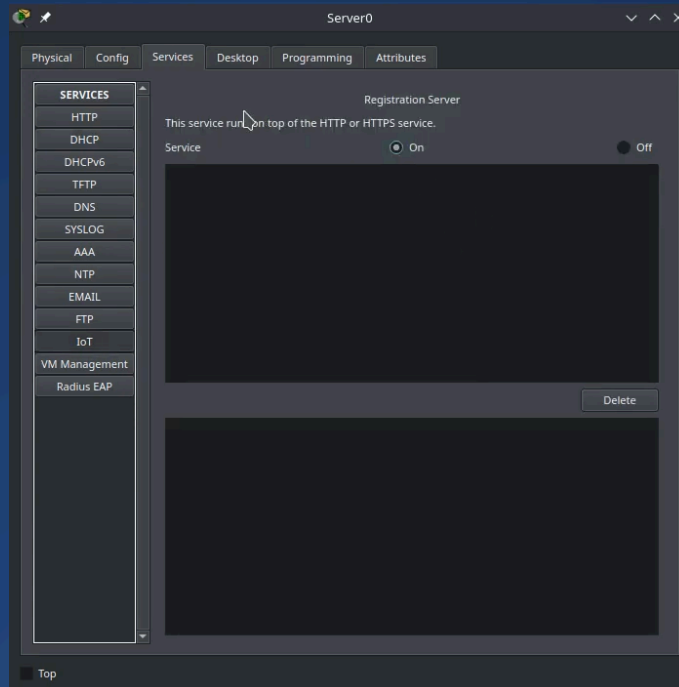
1. Buat infrastruktur seperti berikut. Dapat diunduh pada [halaman ini](#).

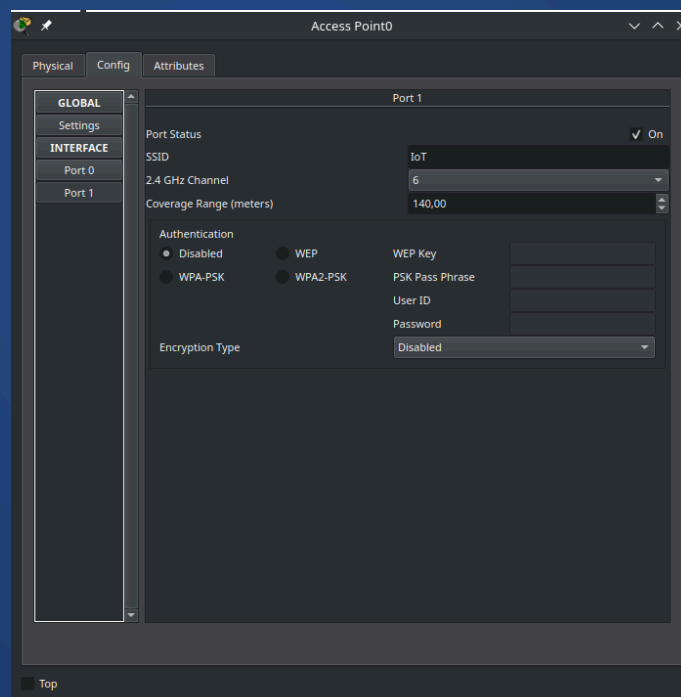
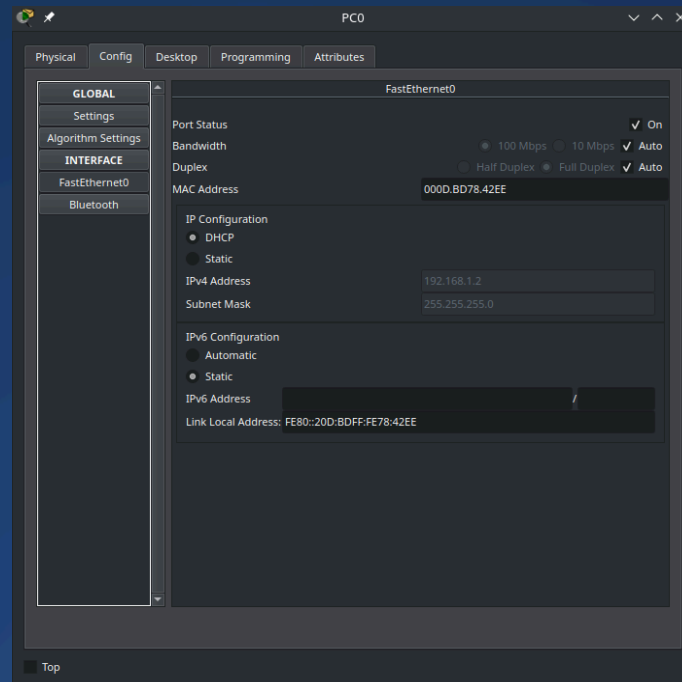


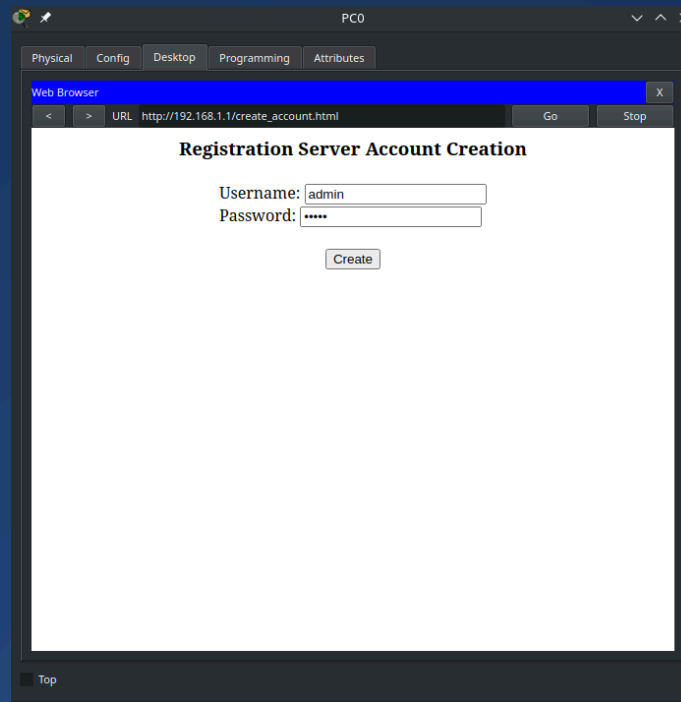
Spesifikasi:

1. Server menggunakan Service DHCP dan IoT registration Server
2. Network 192.168.1.0/24
3. IP Address Server PT adalah 192.168.1.1
4. Perangkat **PC** dan SBC(lihat langkah 2 untuk SBC) mengaktifkan DHCP pada confignya

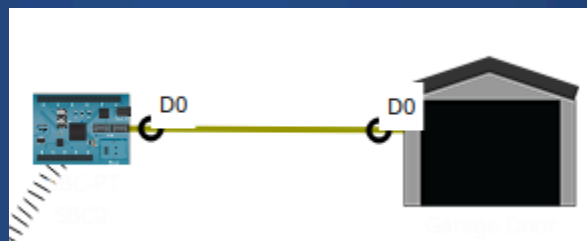




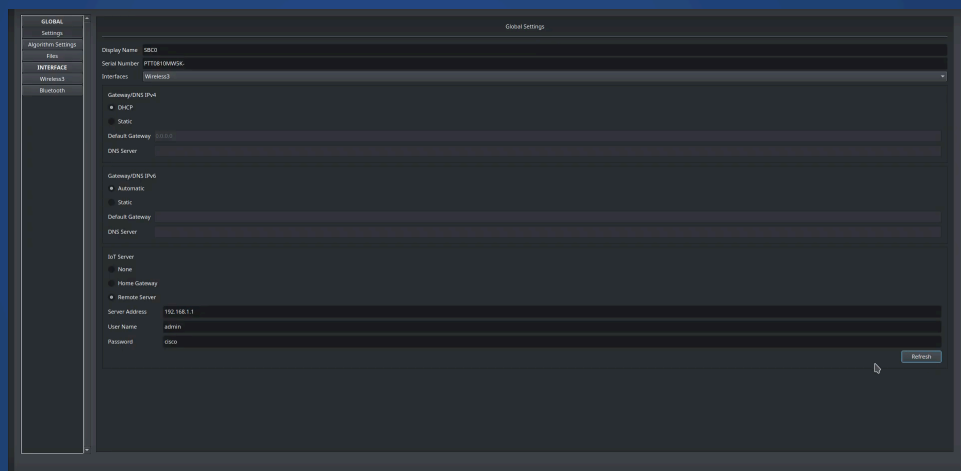
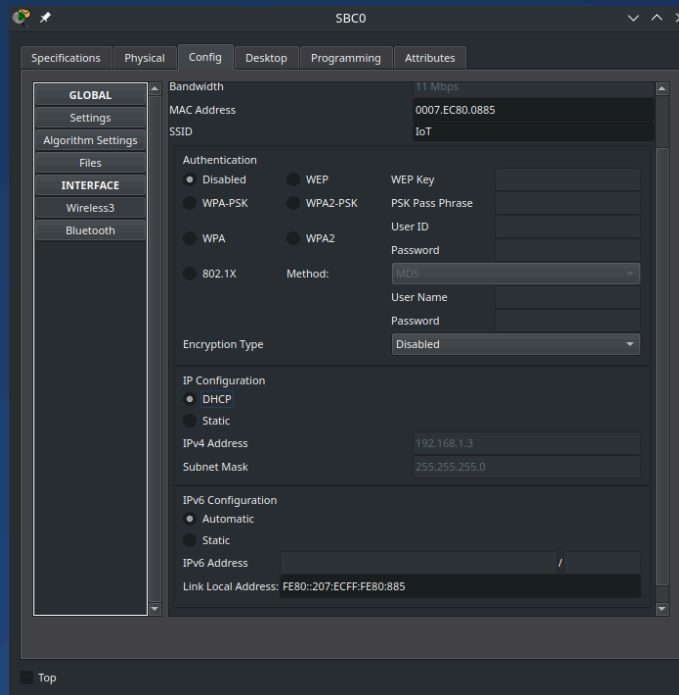




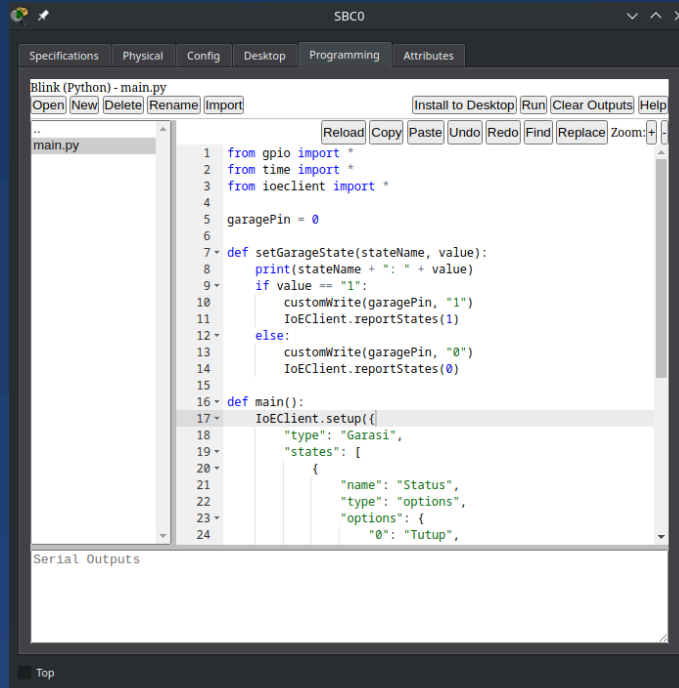
2. Tambahkan SBC dan garage door. Hubungkan keduanya pada pin digital.



3. Hubungkan SBC dan access point dengan konfigurasi sebagai berikut.

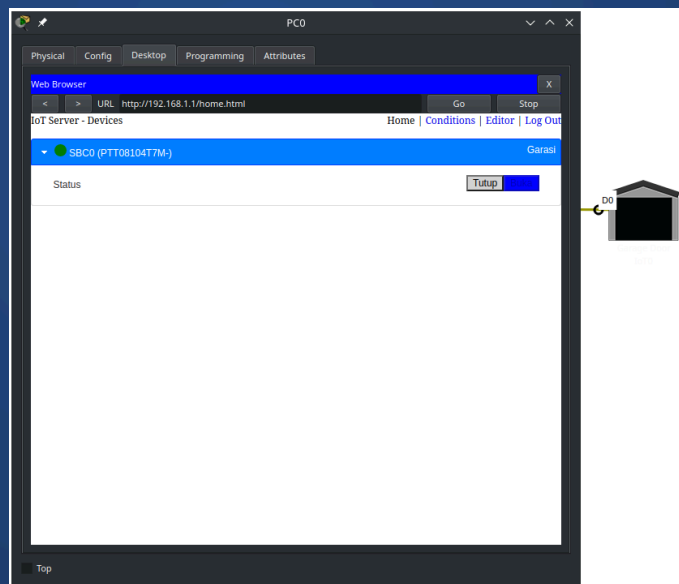


4. Masukkan program berikut ke SBC dan running. Unduh pada [halaman ini](#).



```
1 from gpio import *
2 from time import *
3 from ioeclient import *
4
5 garagePin = 0
6
7 def setGarageState(stateName, value):
8     print(stateName + ": " + value)
9     if value == "1":
10         customWrite(garagePin, "1")
11         IoEClient.reportStates(1)
12     else:
13         customWrite(garagePin, "0")
14         IoEClient.reportStates(0)
15
16 def main():
17     IoEClient.setup({
18         "type": "Garasi",
19         "states": [
20             {
21                 "name": "Status",
22                 "type": "options",
23                 "options": {
24                     "0": "Tutup",
```

5. Garage Door kini dapat dibuka tutup melalui web.



Ini pake protokol MQTT bang? G tau bang, g nemu dokumentasinya.. Hooohoo..

Dokumentasi IoEClient:

https://tutorials.ptnetacad.net/help/default/iot_python_api.htm

Playlist Youtube: <https://youtu.be/WvQC2cVxwds?feature=shared>

Referensi

- SA. Setiawan, E. Arlitasari, M. Zuhri, and A. Hendriana, "Monitoring Pemberian Pakan Ikan Otomatis Menggunakan Iot di Laboratorium Perikanan Sekolah Vokasi Ipb," Jurnal Informatika Teknologi dan Sains, vol. 4, no. 3, pp. 108-116, 2022, doi: 10.51401/jinteks.v4i3.1901.
- Laghari, A. A., He, H., & Laghari, R. A. (2020). "Internet of Things (IoT) Protocols and Standards." Future Generation Computer Systems, vol. 107, pp. 231-244.
- Kovatsch, M., Shelby, Z., & Bormann, C. (2014). "CoAP (Constrained Application Protocol) Protocol Specification." Internet Engineering Task Force (IETF).
- Banks, A., & Gupta, R. (2014). "MQTT Version 3.1.1." OASIS Standard.
- Palattella, M. R., Accettura, N., Vilajosana, X., & Alonso, M. (2013). "Standardized Protocol Stack for the Internet of (Important) Things." IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1389-1406.
- Buratti, C., Conti, A., Dardari, D., & Verdone, R. (2009). "An Overview on Wireless Sensor Networks Technology and Evolution." Sensors, vol. 9, pp. 6869-6896.
- Modul 7 IoT Praktikum Jaringan Komputer. Asisten Praktikum Jaringan Komputer 2023.
- <https://aws.amazon.com/id/what-is/mqtt/>
- <https://mosquitto.org/download/>