

# Bab 1

## Simulasi Internet of Things dan MQTT

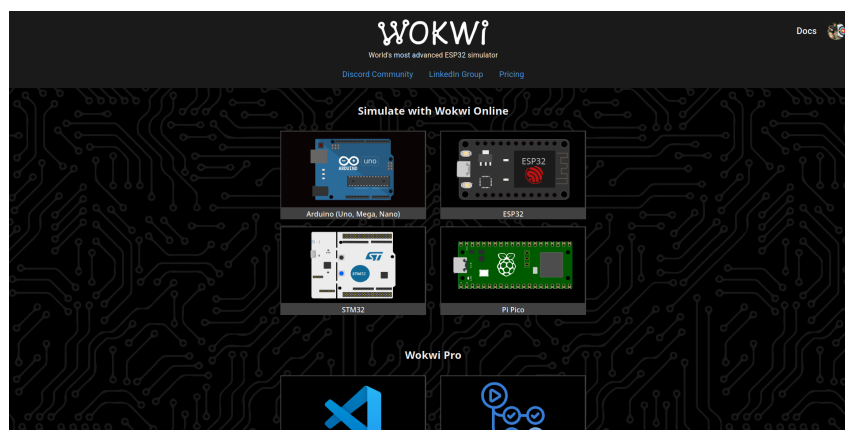
### 1.1 Tutorial

1. Mahasiswa membuat website [wokwi.com](http://wokwi.com) dan melakukan registrasi akun di sana



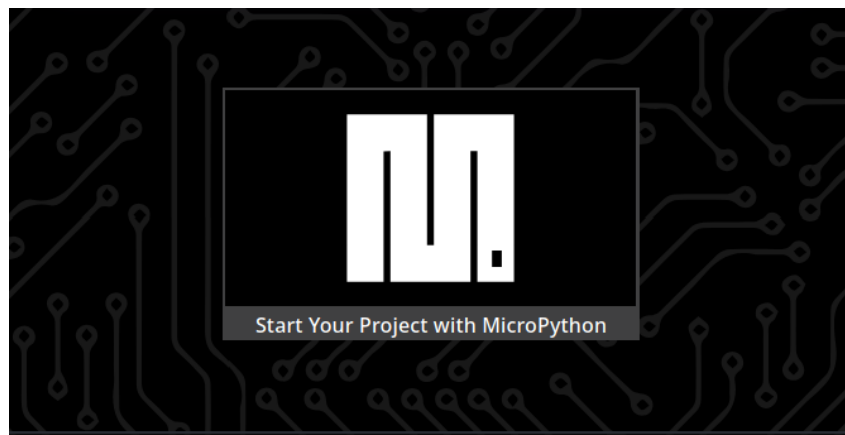
Gambar 1.1: Website Wokwi

2. Setelah teregistrasi, klik halaman depan untuk melihat template yang ada



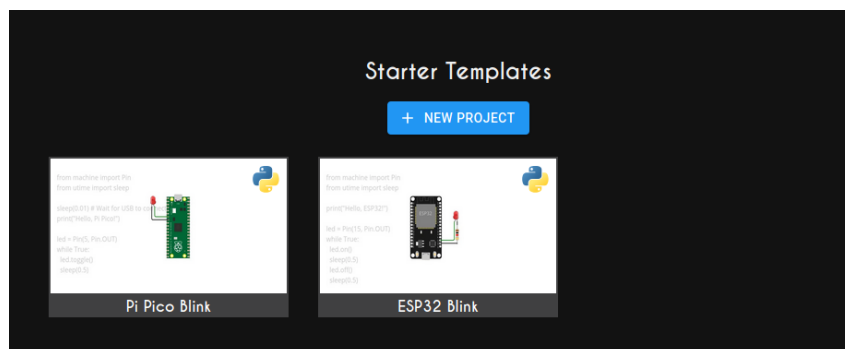
Gambar 1.2: Akses Dashboard

3. Scroll ke bawah dan cari **Start Your Project with MicroPython**



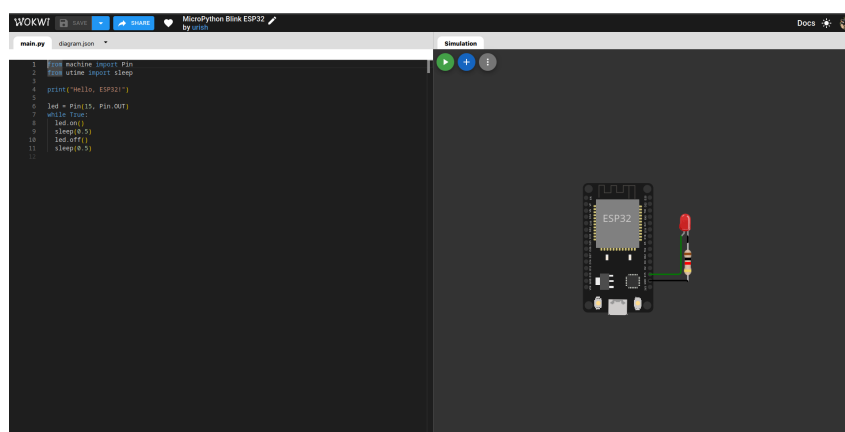
Gambar 1.3: Projek MicroPython

4. Cari **ESP32 Blink**

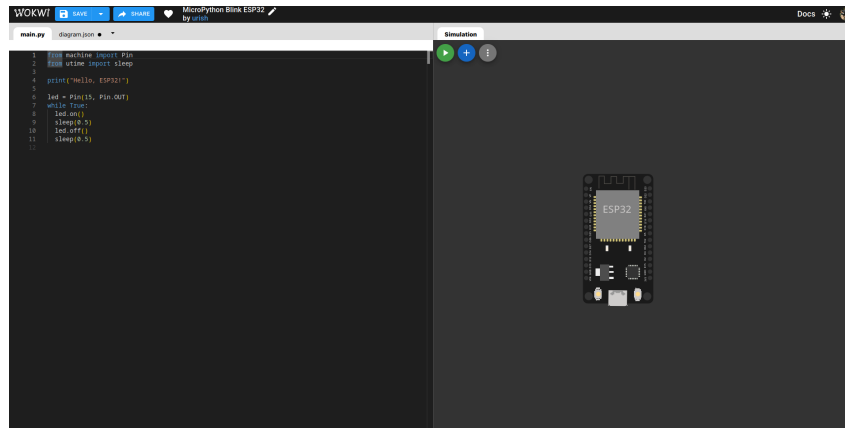


Gambar 1.4: ESP32-Blink

5. Tunggu halaman selesai load, lalu hapus komponen **LED** dan **Resistor**

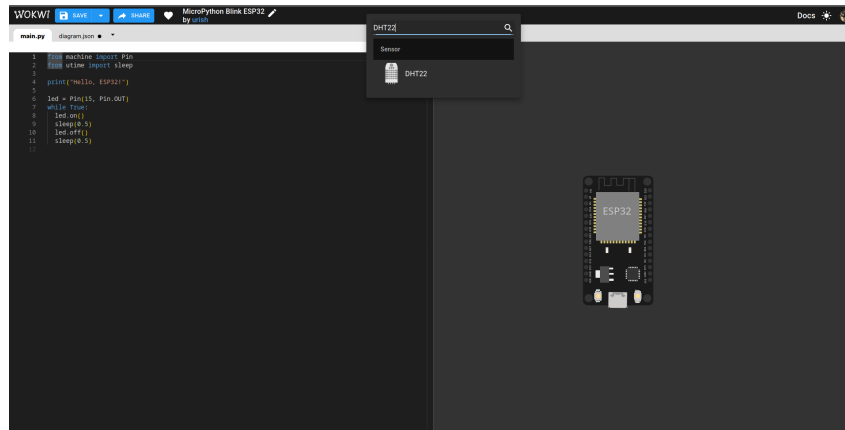


Gambar 1.5: ESP32-Blink



Gambar 1.6: ESP32-Blink

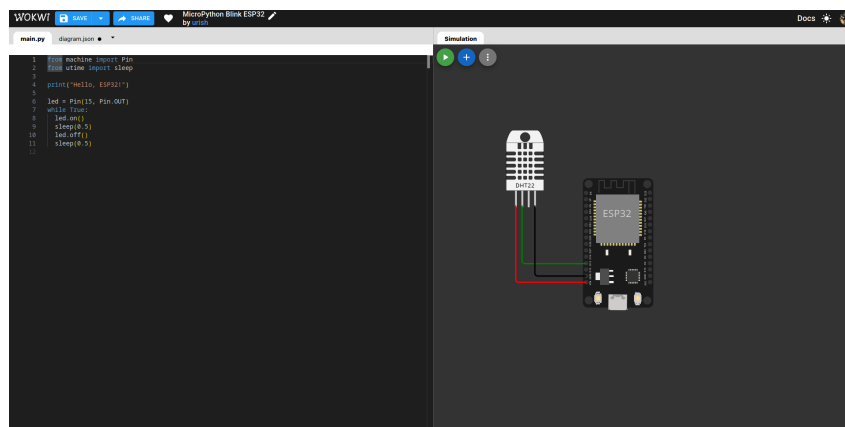
6. Di panel kanan **Simulation**, klik tombol + **Plus** dan cari DHT22, klik sensor itu



Gambar 1.7: ESP32-Blink

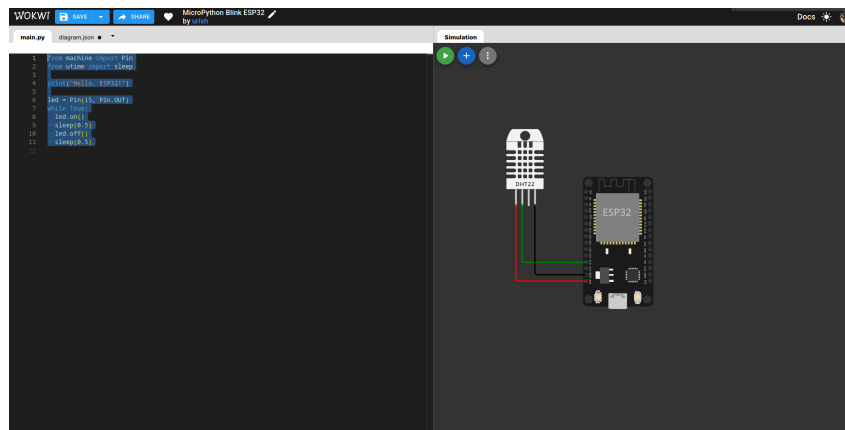
7. Sambungkan kaki ke ESP32 dengan ketentuan

- dht1:VCC → esp:VIN
- dht1:SDA → esp:D12
- dht1:GND → esp:GND.2



Gambar 1.8: ESP32-Blink

8. Di sisi kiri, hapus semua kode yang ada



Gambar 1.9: ESP32-Blink

9. Lalu masukkan potongan kode berikut

- (a) **Parameter dan Library**, ubah **nim** menjadi **nim anda tanpa titik (G211240012)** agar tidak tabrakan

**Potongan Kode**

```
import time
import json
import network
import ubinascii
from umqtt.simple import MQTTClient
import dht
import machine
from machine import Pin

MQTT_BROKER = "broker.hivemq.com"
MQTT_PORT = 1883
MQTT_USER = ""
MQTT_PASSWORD = ""

MQTT_TOPIC = b"nim/sensor"

WIFI_SSID = "Wokwi-GUEST"
WIFI_PASSWORD = ""

DHT_PIN = 12
PUBLISH_INTERVAL = 10
```

```

1 import time
2 import json
3 import network
4 import ubinascii
5 from umqtt.simple import MQTTClient
6 import dht
7 import machine
8 from machine import Pin
9
10 MQTT_BROKER = "broker.hivemq.com"
11 MQTT_PORT = 1883
12 MQTT_USER = ""
13 MQTT_PASSWORD = ""
14
15 MQTT_TOPIC = b"dht22/sensor"
16
17 WIFI_SSID = "Wokwi-GUEST"
18 WIFI_PASSWORD = ""
19
20 DHT_PIN = 12
21 PUBLISH_INTERVAL = 10

```

Gambar 1.10: Kode Inisialisasi

(b) Koneksi Wi-Fi

Potongan Kode

```

def connect_wifi():
    print("Connecting to WiFi...")
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(WIFI_SSID, WIFI_PASSWORD)

    while not wlan.isconnected():
        time.sleep(0.5)
        print(".", end="")

    print("\nWiFi connected!")
    print("IP:", wlan.ifconfig()[0])

```

```

23 def connect_wifi():
24     print("Connecting to WiFi...")
25     wlan = network.WLAN(network.STA_IF)
26     wlan.active(True)
27     wlan.connect(WIFI_SSID, WIFI_PASSWORD)
28
29     while not wlan.isconnected():
30         time.sleep(0.5)
31         print(".", end="")
32
33     print("\nWiFi connected!")
34     print("IP:", wlan.ifconfig()[0])
35

```

Gambar 1.11: Kode Koneksi WiFi

(c) Koneksi ke MQTT

Potongan Kode

```

def connect_mqtt():
    client_id = ubinascii.hexlify(machine.unique_id())
    client = MQTTClient(client_id, MQTT_BROKER, port=MQTT_PORT,
                        user=MQTT_USER, password=MQTT_PASSWORD, keepalive=60)
    client.connect()
    print("Connected to MQTT broker:", MQTT_BROKER)
    return client

```

```

35
36 def connect_mqtt():
37     client_id = ubinascii.hexlify(machine.unique_id())
38     client = MQTTClient(client_id, MQTT_BROKER, port=MQTT_PORT,
39                        user=MQTT_USER, password=MQTT_PASSWORD, keepalive=60)
40     client.connect()
41     print("Connected to MQTT broker:", MQTT_BROKER)
42     return client

```

Gambar 1.12: Kode Koneksi MQTT

(d) Inisialisasi Sensor dan Wi-Fi

Potongan Kode

```

# Initialize DHT22
sensor = dht.DHT22(Pin(DHT_PIN))

# Connect to WiFi and MQTT
connect_wifi()
client = connect_mqtt()

print("DHT22 MQTT Publisher (JSON) started!")

last_time = 0

```

```

43
44 # Initialize DHT22
45 sensor = dht.DHT22(Pin(DHT_PIN))
46
47 # Connect to WiFi and MQTT
48 connect_wifi()
49 client = connect_mqtt()
50
51 print("DHT22 MQTT Publisher (JSON) started!")
52
53 last_time = 0

```

Gambar 1.13: Kode Inisialisasi Sensor dan WiFi

(e) Kode Eksekusi Pengiriman Data

### Potongan Kode

```
while True:
    if time.time() - last_time >= PUBLISH_INTERVAL:
        try:
            sensor.measure()
            temp = sensor.temperature()
            hum = sensor.humidity()

            # Create JSON payload
            data =

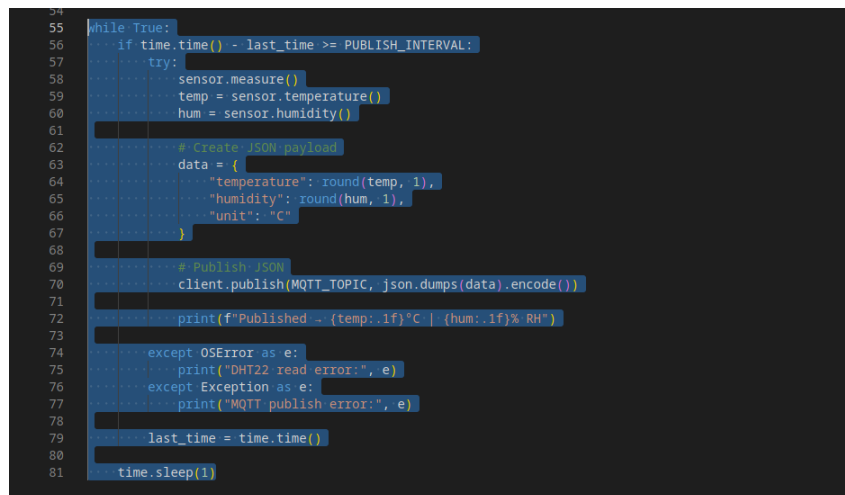
            # Publish JSON
            client.publish(MQTT_TOPIC, json.dumps(data).encode())

            print(f"Published → {temp:.1f}°C | {hum:.1f}% RH")

        except OSError as e:
            print("DHT22 read error:", e)
        except Exception as e:
            print("MQTT publish error:", e)

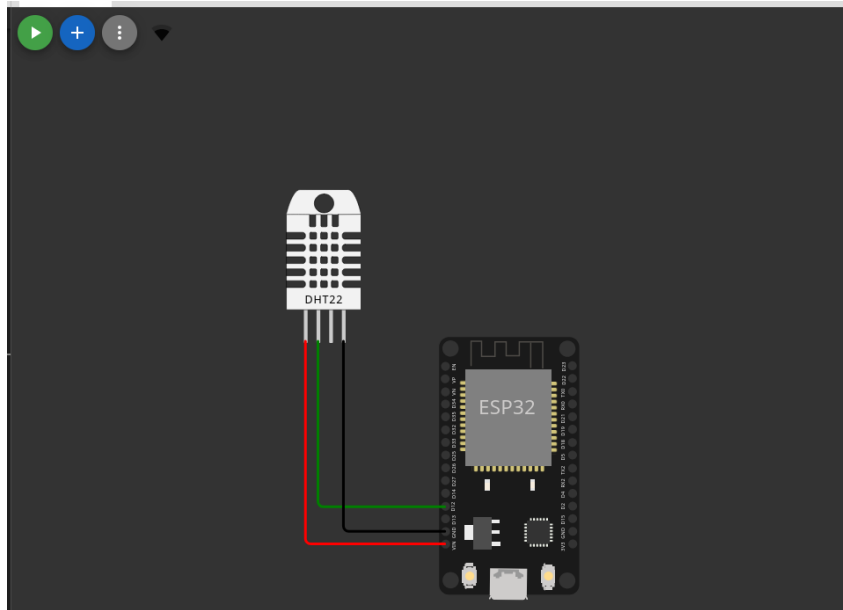
        last_time = time.time()

time.sleep(1)
```

A screenshot of a code editor with a dark background and light-colored text. The code is Python and is identical to the code block above. Line numbers 54 through 81 are visible on the left side of the editor. The code includes comments for each step: 'sensor.measure()', '# Create JSON payload', '# Publish JSON', and error handling for 'OSError' and 'Exception'. The output of the print statement is visible as a comment: 'Published → {temp:.1f}°C | {hum:.1f}% RH'.

Gambar 1.14: Kode Pengiriman Data

10. Lalu jalankan simulasi dengan klik **Play** di panel kanan tengah



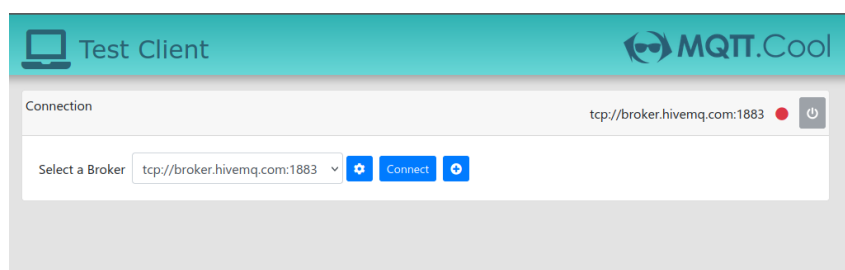
Gambar 1.15: Menjalankan Simulasi

11. Tunggu hingga muncul publish sukses

```
load:0x40080400,len:4124
entry 0x40080680
Connecting to WiFi...
.....
WiFi connected!
IP: 10.10.0.2
Connected to MQTT broker: broker.hivemq.com
DHT22 MQTT Publisher (JSON) started!
Published -> 24.0°C | 40.0% RH
Published -> 24.0°C | 40.0% RH
```

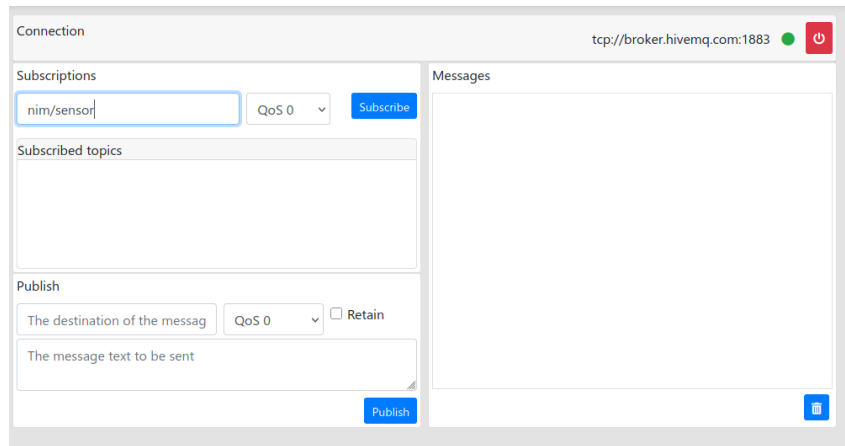
Gambar 1.16: Hasil Publish

12. Buka website [https://mqtt.aarsoftwareserver.com:444/test\\_client/](https://mqtt.aarsoftwareserver.com:444/test_client/) dan pilih broker milik **hivemq**. Lalu klik **Connect**



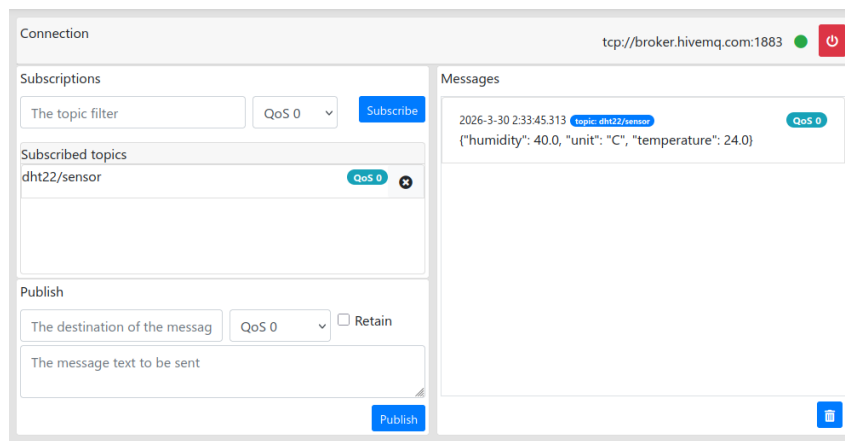
Gambar 1.17: Koneksi ke HiveMQ Broker

13. Di bagian **Subscription**, masukkan **nim/sensor** ganti **nim** dengan nim mahasiswa tanpa titik dan pastikan kapital sesuai. lalu klik **Subscribe**



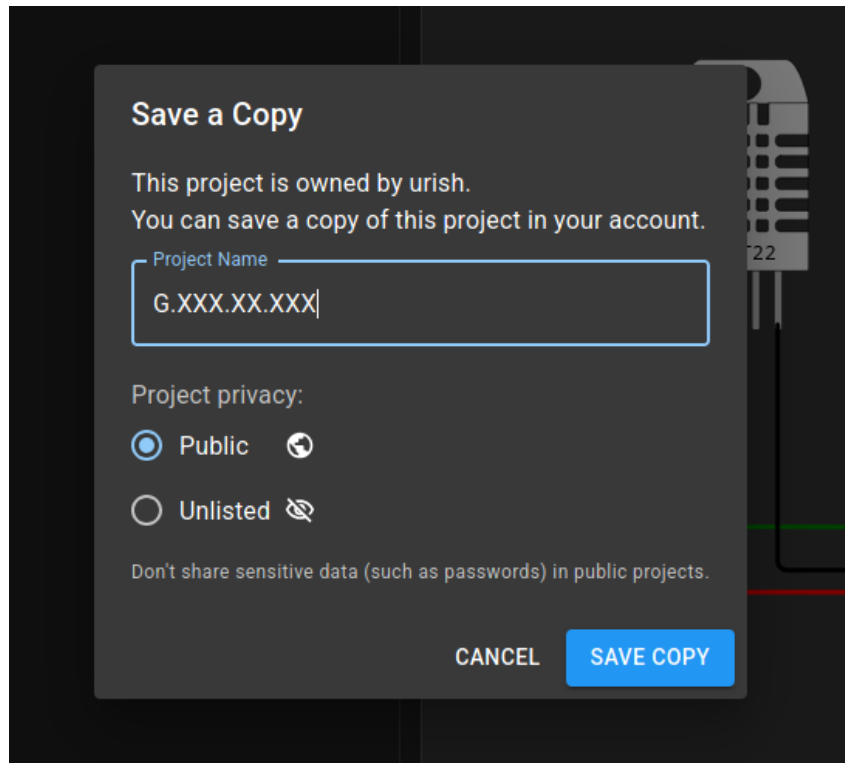
Gambar 1.18: Mengatur Subscription

14. Data nantinya akan dikirimkan dan ditampilkan di sisi kanan



Gambar 1.19: Hasil Tampilan Data

15. Data ini adalah asli dan bisa diakses di manapun melalui protokol MQTT.
16. Kembali ke Wokwi, Klik **Save**, masukkan NIM dan pilih **Save Copy**. Projek bisa diakses kembali melalui **Profile** → **My Projects**



Gambar 1.20: Hasil Tampilan Data