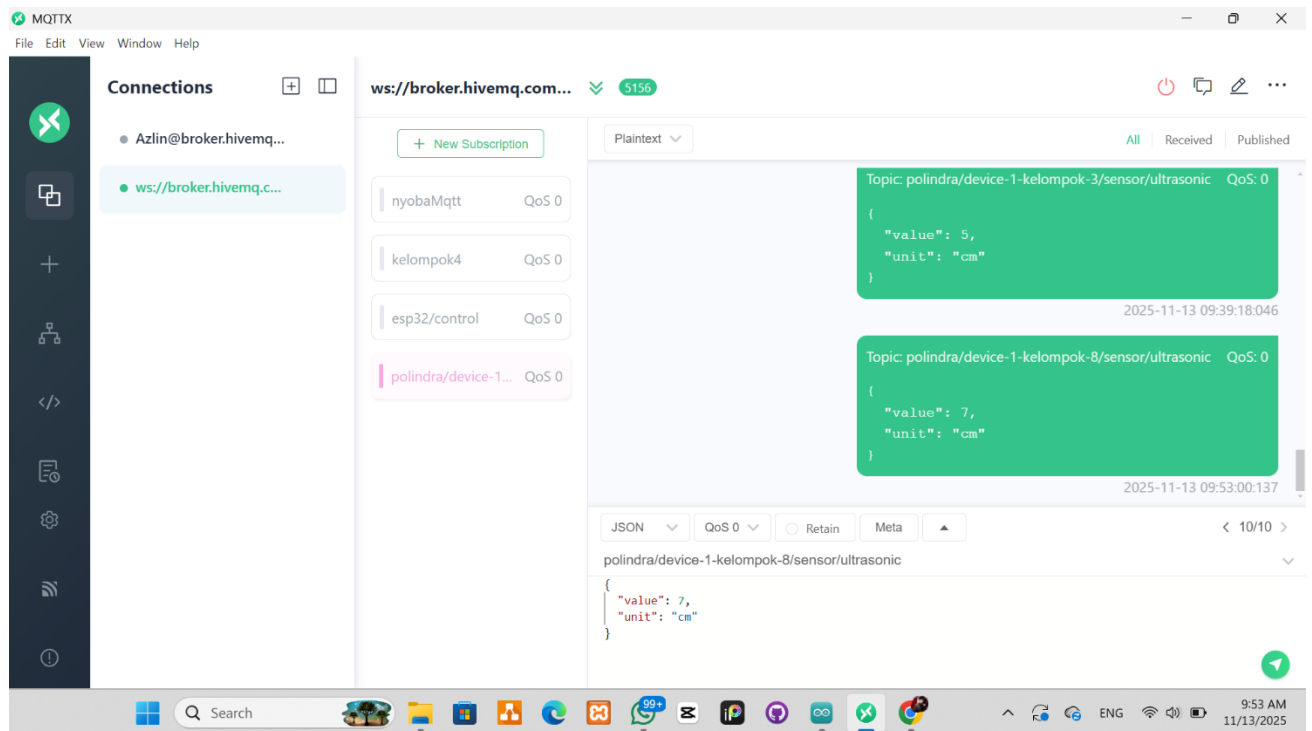




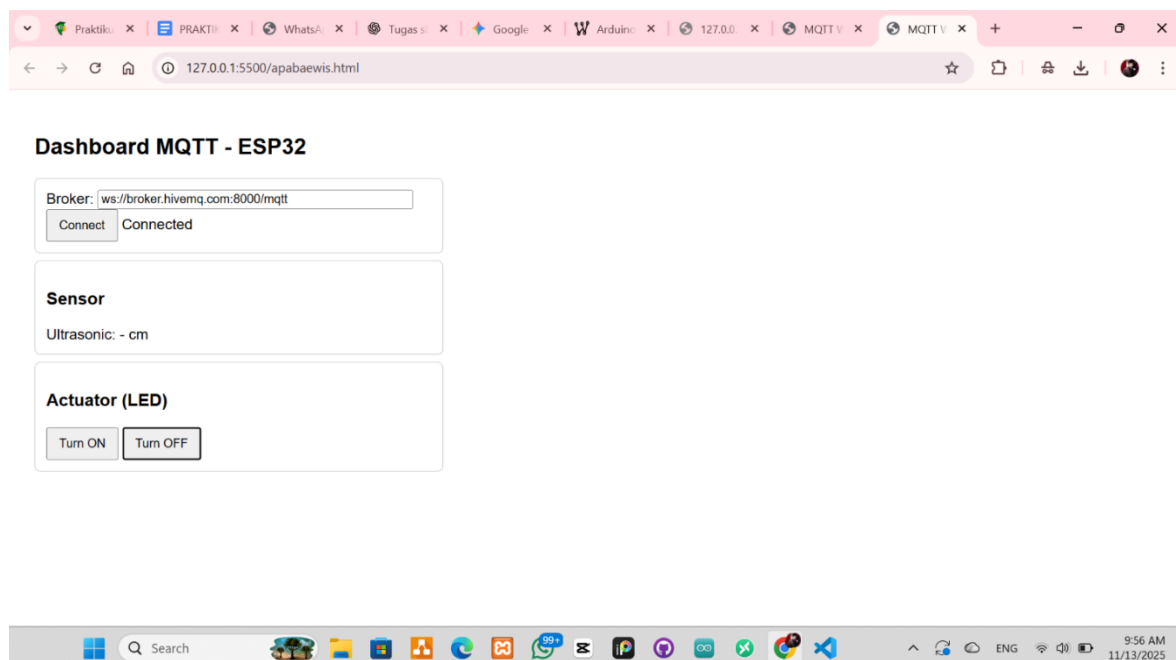
Nama : Siti Sa'adah
NIM 2403001
Kelas : D3 TI 2A
Mata Kuliah : Internet of Things

TUGAS PRAKTIKUM

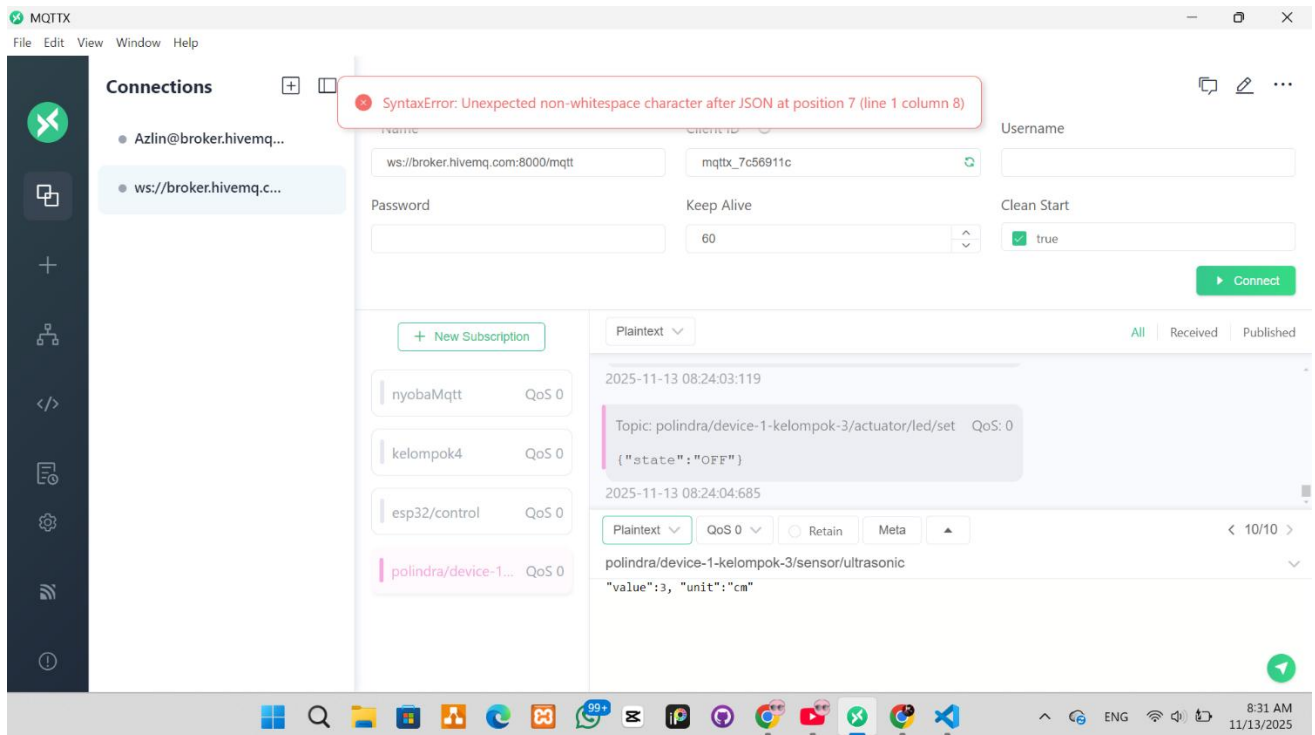
Pada praktikum kali ini mencoba menghubungkan antara MQTT dengan web iot.



MQTT mengirimkan data ke web iot



Tampilan awal web iot



The screenshot displays the MQTTX application interface. On the left is a dark sidebar with navigation icons. The main window is divided into three sections:

- Connections:** A list of active connections. The first connection is "Azlin@broker.hivemq.com..." and the second, which is selected, is "ws://broker.hivemq.com...".
- Subscriptions:** A list of subscriptions for the selected connection. It includes "nyobaMqtt", "kelompok4", "esp32/control", and "polindra/device-1...". Each subscription shows a "QoS 0" status.
- Message History:** A view showing received messages. It displays two messages with the topic "polindra/device-1-kelompok-3/sensor/ultrasonic". Each message is a JSON object:


```
{
  "value": 6,
  "unit": "cm"
}
```

 The messages are timestamped "2025-11-13 09:00:33:074" and "2025-11-13 09:00:44:659".

At the bottom of the screen, there is a Windows taskbar with various application icons and a system clock showing "9:07 AM 11/13/2025".

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8" />
  <title>MQTT Web Dashboard</title>
  <script src="https://unpkg.com/mqtt/dist/mqtt.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      padding: 20px;
```

```

    }
    .card {
        border: 1px solid #ccc;
        padding: 12px;
        margin: 8px 0;
        border-radius: 6px;
        max-width: 420px;
    }
    button {
        padding: 8px 12px;
    }
</style>
</head>
<body>
    <h2>Dashboard MQTT - ESP32</h2>
    <div class="card">
        <div>Broker: <input id="broker" value="ws://broker.hivemq.com:8000/mqtt"
style="width:80%"></div>
        <div><button id="connectBtn">Connect</button> <span
id="connStatus">Disconnected</span></div>
    </div>
    <div class="card">
        <h3>Sensor</h3>
        <div>Ultrasonic: <span id="ultrasonic">-</span> cm</div>
    </div>
    <div class="card">
        <h3>Actuator (LED)</h3>
        <button id="ledOnBtn">Turn ON</button>
        <button id="ledOffBtn">Turn OFF</button>
    </div>
    <script>
        let client = null;
        const deviceId = "device-1-kelompok-3";
        const topics = {
            ultrasonic: `polindra/${deviceId}/sensor/ultrasonic`,
            ledSet: `polindra/${deviceId}/actuator/led/set`
        };
        function connect(brokerUrl) {
            client = mqtt.connect(brokerUrl);
            document.getElementById('connStatus').innerText = 'Connecting...';
            client.on('connect', () => {
                document.getElementById('connStatus').innerText = 'Connected';
                client.subscribe(topics.ultrasonic);
            });
            client.on('message', (topic, payload) => {
                try {
                    const msg = JSON.parse(payload.toString());
                    if (topic === topics.ultrasonic) {
                        document.getElementById('ultrasonic').innerText =
msg.value;
                    } else if (topic === topics.lainnya) {
                        // silakan tambahkan logika untuk topic lainnya
                    }
                }
            });
        }
    </script>

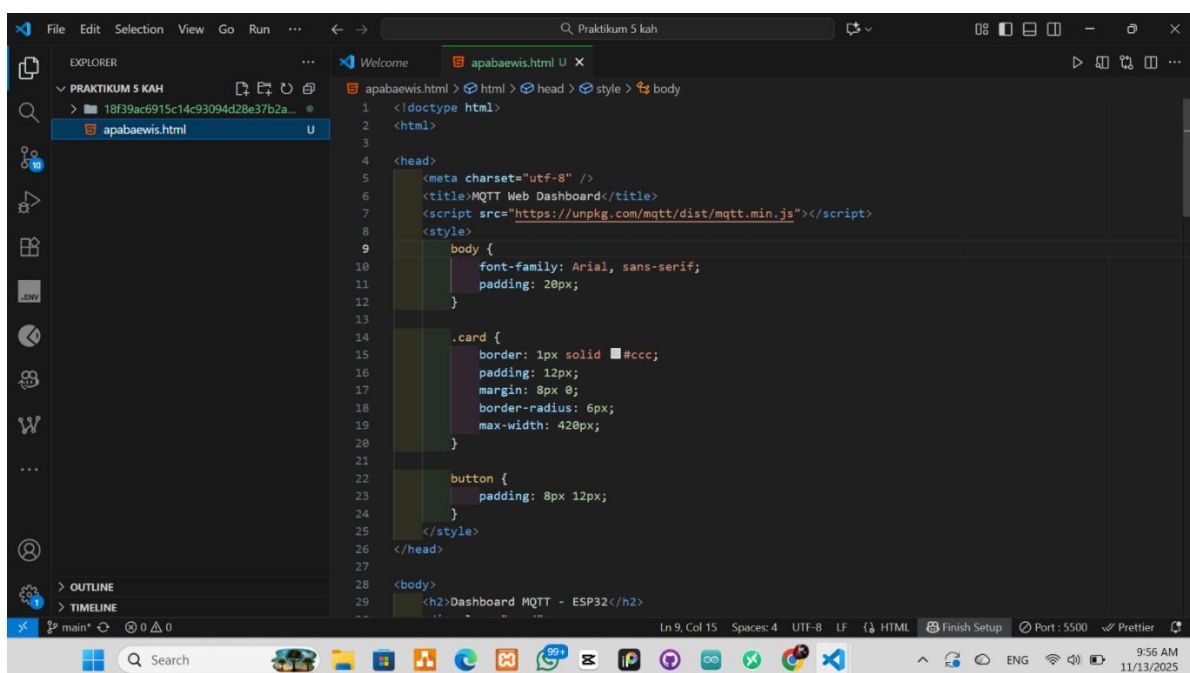
```

```

    } catch (e) {
        console.log('Invalid JSON', payload.toString());
    }
});
client.on('error', (err) => {
    console.error(err);
    document.getElementById('connStatus').innerText = 'Error';
});
client.on('close', () => {
    document.getElementById('connStatus').innerText = 'Disconnected';
});
}
document.getElementById('connectBtn').addEventListener('click', () => {
    const url = document.getElementById('broker').value;
    connect(url);
});
document.getElementById('ledOnBtn').addEventListener('click', () => {
    if (client && client.connected) {
        client.publish(topics.ledSet, JSON.stringify({ state: "ON" }));
    }
});
document.getElementById('ledOffBtn').addEventListener('click', () => {
    if (client && client.connected) {
        client.publish(topics.ledSet, JSON.stringify({ state: "OFF" }));
    }
});
</script>
</body>
</html>

```

Kode program web iot



Kode program di VSCode



127.0.0.1:5500/apabaewis.html

Dashboard MQTT - ESP32

Broker:

Connect

Connected

Sensor

Ultrasonic: 3 cm

Actuator (LED)

Turn ON

Turn OFF

Tampilan web iot ketika berhasil menerima data dari MQTT



IOT_PRAKTEK13NOV | Arduino IDE 2.3.6

File Edit Sketch Tools Help

ESP32 Dev Module

```
IOT_PRAKTEK13NOV.ino
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 // Konfigurasi WiFi
5 const char* ssid = "Mahasiswa";
6 const char* password = "@KeretaCepat2023";
7
8 // Broker MQTT publik
9 const char* mqtt_server = "broker.hivemq.com";
10
11 // Topic MQTT
12 const char* topicSensor = "polindra/device-1-kelompok-8/sensor/ultrasonic";
13 const char* topicLed1 = "polindra/device-1-kelompok-8/actuator/led1/set";
14 const char* topicLed2 = "polindra/device-1-kelompok-8/actuator/led2/set";
15 const char* topicLed3 = "polindra/device-1-kelompok-8/actuator/led3/set";
16 const char* topicLed4 = "polindra/device-1-kelompok-8/actuator/led4/set";
17
18 WiFiClient espClient;
19 PubSubClient client(espClient);
20
21 // Pin konfigurasi sensor & LED
```

Output Serial Monitor X

Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')

New Line 115200 baud

```
load:0x3fff0030,len:4980
load:0x40078000,len:16612
load:0x40080400,len:3480
entry 0x400805b4
E (53) flash_parts: partition 3 invalid - offset 0x340000 size 0x330000 exceeds flash chip size 0x400000
E (53) boot: Failed to verify partition table
E (56) boot: load partition table error!
ets Jul 29 2019 12:21:46

rst:
```

Ln 22, Col 24 ESP32 Dev Module on COM3 2

9:52 AM 11/13/2025

Mencoba menghubungkan perangkat iot dengan web iot, namun masih gagal.