LAPORAN PRAKTIKUM PEMANTAUAN

OTOMATIS SERVO MOTOR BERBASIS ESP32 DAN MQTT

Sugeng Aldi Widodo

Fakultas Vokasi, Universitas Brawijaya

sugengaldi330@student.ub.ac.id

ABSTRAK

Praktikum ini bertujuan untuk menerapkan konsep dasar Internet of Things (IoT) melalui sistem kontrol otomatis pada servo motor. Komponen utama yang digunakan adalah ESP32 sebagai mikrokontroler yang dapat terhubung ke jaringan internet dan berkomunikasi menggunakan protokol MQTT. Servo dikendalikan agar membuka dan menutup secara bergantian dalam interval waktu tertentu, lalu status pergerakannya dikirimkan ke server MQTT HiveMQ. Di sisi pengguna, sebuah antarmuka web dikembangkan untuk menampilkan status servo secara langsung menggunakan WebSocket. Melalui eksperimen ini, didapatkan bahwa ESP32 mampu mengirimkan informasi ke server dengan lancar dan antarmuka web berhasil menampilkan perubahan status servo tanpa hambatan berarti.

Kata kunci: ESP32, Servo Motor, MQTT, IoT, WebSocket

ABSTRACT

This practicum aims to apply the basic concept of the Internet of Things (IoT) through an automatic control system on a servo motor. The main component used is the ESP32 as a microcontroller that can be connected to the internet network and communicate using the MQTT protocol. The servo is controlled to open and close alternately at certain time intervals, then its movement status is sent to the HiveMQ MQTT server. On the user side, a web interface is developed to display the servo status directly using WebSocket. Through this experiment, it was found that the ESP32 was able to send information to the server smoothly and the web interface successfully displayed changes in servo status without significant obstacles.

Keywords*: ESP32, Servo Motor, MQTT, IoT, WebSocket*

Pendahuluan

Internet of Things (IoT) kini menjadi bagian penting dalam pengembangan teknologi digital, terutama dalam bidang otomasi. Perangkat seperti sensor, aktuator, dan mikrokontroler bisa saling terhubung dan bertukar data melalui internet. Salah satu contoh sederhana dari sistem IoT adalah pengendalian perangkat otomatis seperti palang parkir, yang bisa dibuka dan ditutup secara otomatis lalu dipantau dari jarak jauh.

Dalam kegiatan praktikum ini, sistem yang dibangun menggunakan ESP32 sebagai pengendali utama yang berkomunikasi dengan sebuah servo motor. Servo ini akan bergerak secara otomatis setiap beberapa detik untuk menunjukkan prinsip kerja sistem kontrol otomatis. Data status servo kemudian dikirimkan melalui protokol MQTT ke server broker publik. Di sisi lain, halaman web yang dibuat berfungsi untuk memantau status servo tersebut secara langsung.

Tujuan Praktikum

Praktikum ini memiliki tujuan:

1. Membuat sistem kendali otomatis berbasis mikrokontroler ESP32.
2. Menerapkan komunikasi antar perangkat menggunakan MQTT.
3. Menampilkan status perangkat (servo) dalam halaman web secara real-time.
4. Memahami alur komunikasi antara perangkat keras dengan perangkat lunak dalam sistem IoT.

Metodologi

1.Alat dan Bahan

Perangkat Keras:

- ESP32 Board
- Motor Servo SG90
- Kabel jumper

Perangkat Lunak:

- Wokwi (simulator mikrokontroler online)
- Arduino IDE (untuk pemrograman ESP32)
- HiveMQ (broker MQTT berbasis cloud)
- HTML dan JavaScript untuk tampilan web
- MQTT.js (library untuk koneksi MQTT dari web)

2 Langkah Pengerjaan

Langkah-langkah utama dalam pembuatan sistem ini dijelaskan sebagai berikut:

1. **Menyusun Rangkaian Simulasi di Wokwi**
   Simulasi ESP32 dilakukan melalui situs wokwi.com. Pin servo dihubungkan ke pin GPIO33, sementara pin power dan ground disesuaikan ke 5V dan GND ESP32.
2. **Menulis Program ESP32 di Arduino IDE**
   Program ditulis agar servo berganti posisi (antara 0° dan 90°) secara otomatis setiap lima detik. Untuk mengirim status gerakan servo ke MQTT, digunakan library PubSubClient.
3. **Membuat Halaman Web Pemantau**
   Web dibangun menggunakan HTML dan JavaScript sederhana. Melalui library MQTT.js, web terhubung ke broker HiveMQ dan mendengarkan pesan pada topik tertentu. Ketika status servo diterima, tampilan di web diperbarui sesuai data yang masuk.
4. **Pengujian dan Pemantauan**
   Sistem dijalankan penuh dalam simulasi, dan diperiksa apakah status servo berubah setiap lima detik, serta apakah tampilan web mampu menampilkan status tersebut dengan benar.

Hasil dan Pembahasan

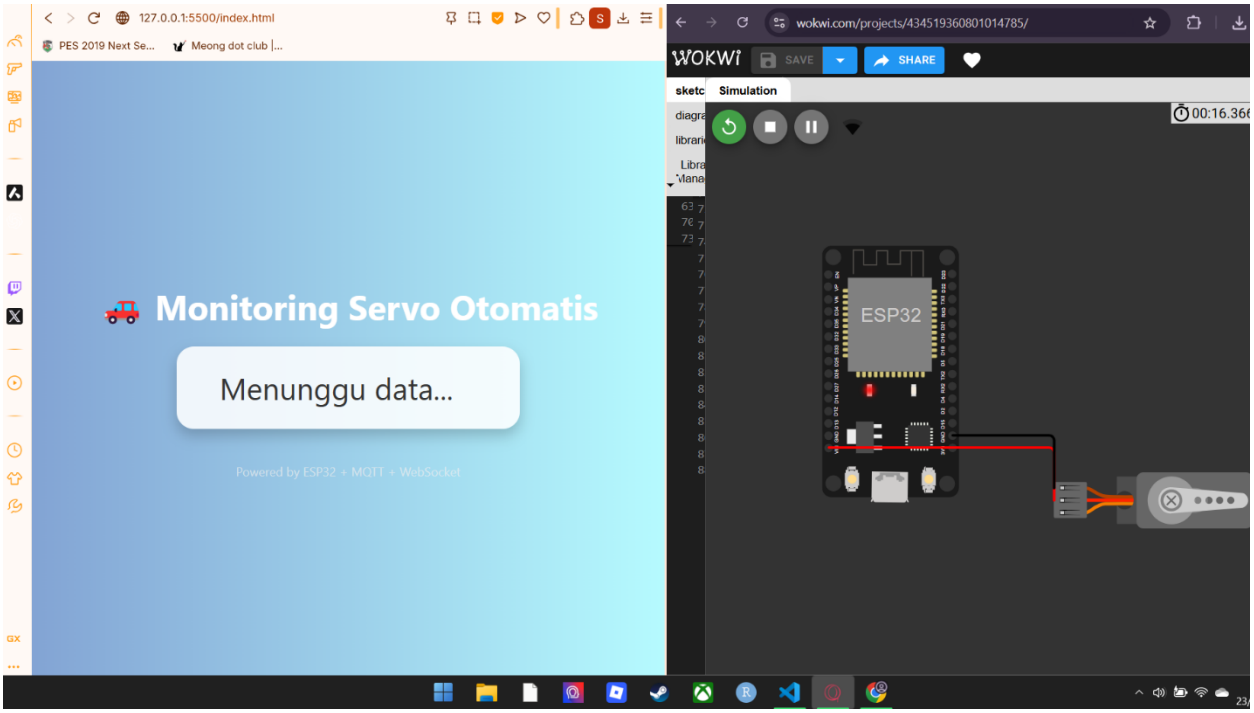
Selama proses pengujian, sistem berhasil berfungsi sesuai yang dirancang. Servo motor secara otomatis berganti posisi dari TUTUP ke BUKA setiap 5 detik. Perubahan posisi ini diikuti dengan pengiriman pesan ke broker MQTT yang kemudian diteruskan ke halaman web pemantau.

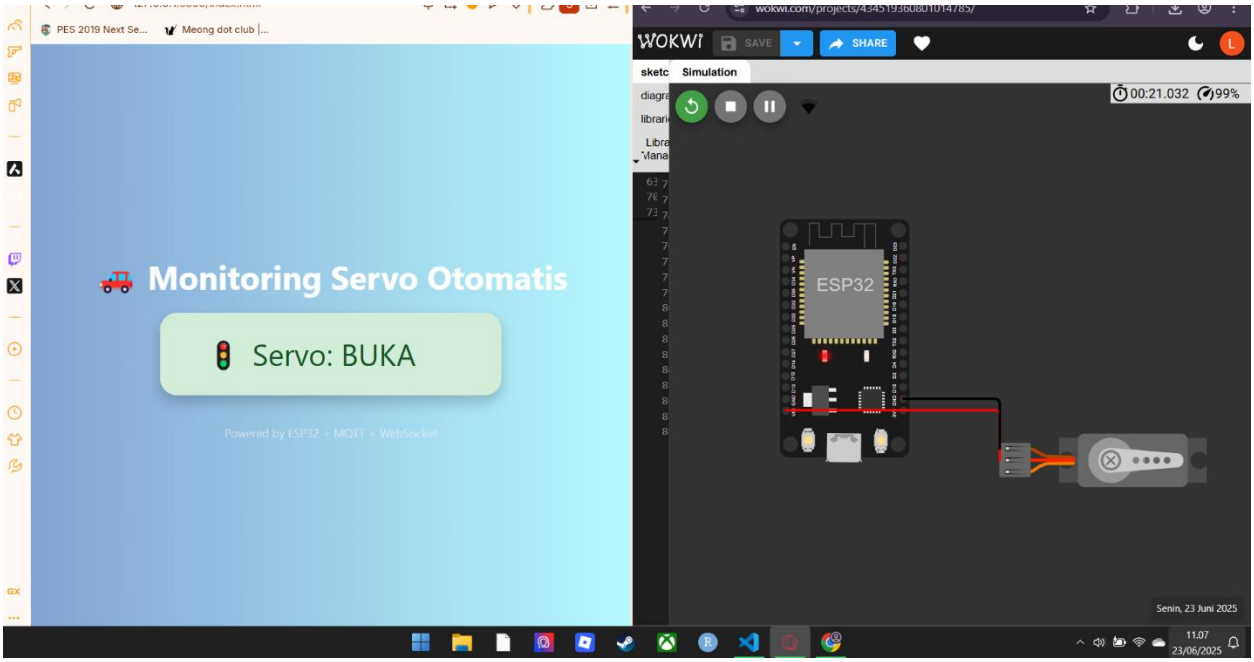
Ketika servo dalam posisi 90°, halaman web menampilkan tulisan “Servo: BUKA” dengan latar hijau. Sedangkan saat servo kembali ke posisi 0°, web memperlihatkan tulisan “Servo: TUTUP” dengan latar merah muda.

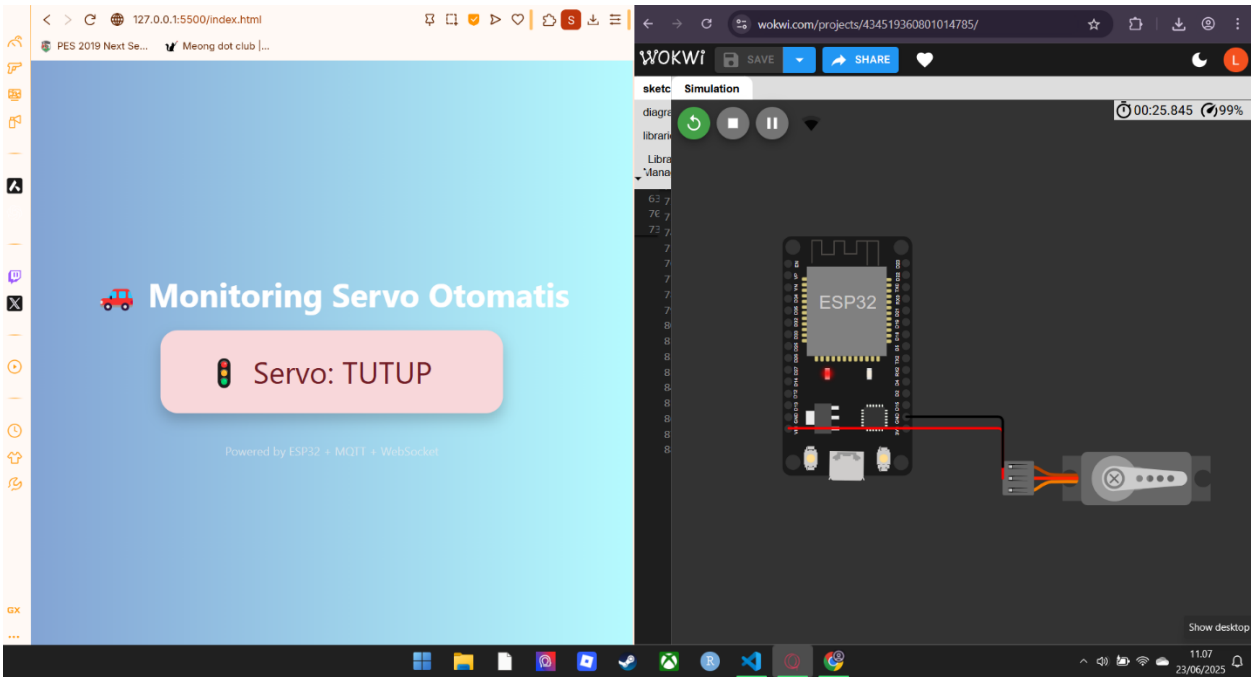
Pesan MQTT yang dikirimkan juga berhasil diterima oleh web secara real-time tanpa jeda yang berarti. Ini menunjukkan bahwa komunikasi melalui WebSocket berjalan dengan stabil. Jika halaman web dibuka sebelum ESP32 mengirim pesan pertama, maka akan muncul tulisan “Menunggu data...”.

Gambar berikut menunjukkan tampilan web dan posisi servo saat status BUKA maupun TUTUP:

- Tampilan “BUKA” saat servo terbuka.
- Tampilan “TUTUP” saat servo tertutup.
- Tampilan awal saat menunggu koneksi.
- Rangkaian lengkap ESP32 dan servo di Wokwi.




Monitoring Servo Otomatis
Servo: BUKA
Powered by ESP32 + MQTT + WebSocket
WOKWI
SAVE
SHARE
Simulation
00:21.032
99%
ESP32
Senin, 23 Juni 2025
23/06/2025


127.0.0.1:5500/index.html
wokwi.com/projects/434519360801014785/
Monitoring Servo Otomatis
Servo: TUTUP
Powered by ESP32 + MQTT + WebSocket
WOKWI
SAVE
SHARE
Simulation
00:25.845
99%
ESP32
Show desktop
23/06/2025

Implementasi Kode:

sketch.ino

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ESP32Servo.h>

// WiFi Wokwi
const char* ssid = "Wokwi-GUEST";
const char* password = "";

// MQTT HiveMQ
const char* mqtt_server = "broker.hivemq.com";
const char* topic_servo = "iot/parking/servo";

WiFiClient espClient;
PubSubClient client(espClient);
Servo myServo;

const int servoPin = 33;
bool isOpen = false;
unsigned long lastToggleTime = 0;
unsigned long toggleInterval = 5000;  // 5 detik

void reconnect() {
  while (!client.connected()) {
    Serial.print("🔄 Menghubungkan MQTT...");
    if (client.connect("esp32ClientAuto")) {
      Serial.println("✅ MQTT Terhubung!");
    } else {
      Serial.print("❌ Gagal, rc=");
      Serial.println(client.state());
      delay(2000);
    }
  }
}

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

```
  // Servo Setup
  myServo.attach(servoPin);
  myServo.write(0);  // mulai dari TERTUTUP
  Serial.println("✅ Servo terpasang di pin 33");

  // WiFi Setup
  WiFi.begin(ssid, password);
  Serial.print("🌐 Menghubungkan WiFi");
  unsigned long wifiStart = millis();
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    if (millis() - wifiStart > 10000) {
      Serial.println("\n⚠️ Gagal konek WiFi. Lanjut tanpa internet.");
      break;
    }
  }
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\n✅ WiFi Tersambung");
  }

  // MQTT Setup
  client.setServer(mqtt_server, 1883);
}

void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    if (!client.connected()) reconnect();
    client.loop();
  }

  unsigned long currentTime = millis();
  if (currentTime - lastToggleTime >= toggleInterval) {
    lastToggleTime = currentTime;

    if (isOpen) {
      myServo.write(0); // TUTUP
      if (client.connected()) client.publish(topic_servo, "TUTUP");
      Serial.println("🔒 Servo Tertutup");
    } else {
      myServo.write(90); // BUKA
```

```
      if (client.connected()) client.publish(topic_servo, "BUKA");
      Serial.println("🔓 Servo Terbuka");
    }

    isOpen = !isOpen; // toggle status
  }

  delay(50);
}
```

Diagram.json

```
{
  "version": 1,
  "author": "ChatGPT",
  "editor": "wokwi",
  "parts": [
    {
      "type": "wokwi-esp32-devkit-v1",
      "id": "esp",
      "top": 40,
      "left": 20,
      "attrs": {}
    },
    {
      "type": "wokwi-ir-sensor",
      "id": "ir1",
      "top": 180,
      "left": 20,
      "attrs": {}
    },
    {
      "type": "wokwi-servo",
      "id": "servo1",
      "top": 180,
      "left": 200,
      "attrs": {}
    }
  ],
  "connections": [
    [ "ir1:VCC", "esp:3V3", "red", [ "v0" ] ],
```

```
    [ "ir1:GND", "esp:GND.2", "black", [ "v0" ] ],
    [ "ir1:OUT", "esp:25", "green", [ "v0" ] ],

    [ "servo1:PWM", "esp:33", "orange", [ "v0" ] ],
    [ "servo1:V+", "esp:VIN", "red", [ "v0" ] ],
    [ "servo1:GND", "esp:GND.1", "black", [ "v0" ] ]
  ]
}
```

libraries.txt

```
# Wokwi Library List
# See https://docs.wokwi.com/guides/libraries

PubSubClient
ESP32Servo
```

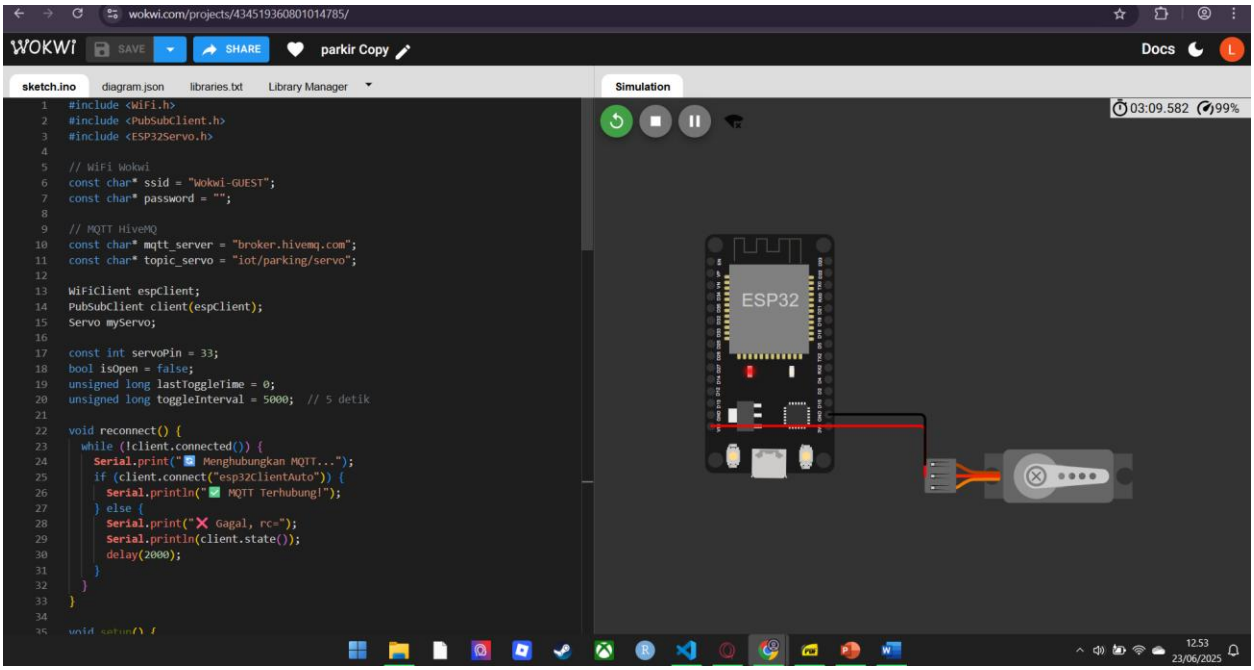
Kode Program Tampilan Web:

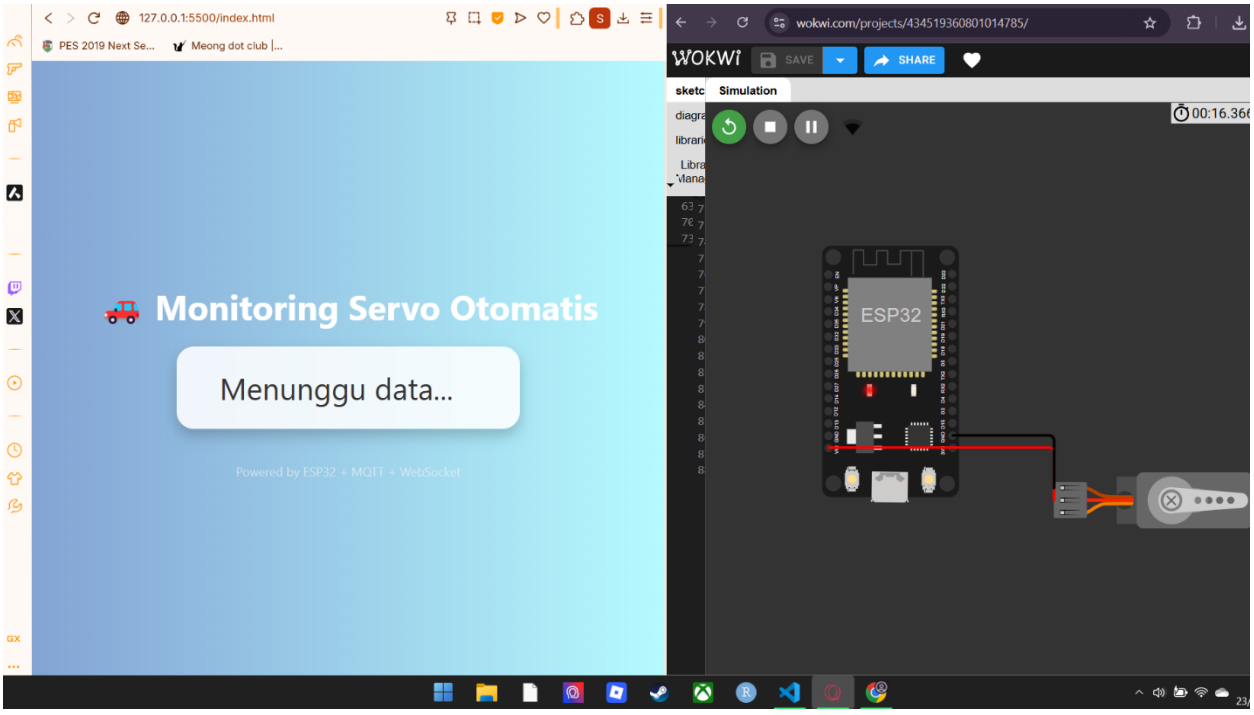
```
<!DOCTYPE html>
<html lang="id">
  <head>
    <meta charset="UTF-8" />
    <title>Status Servo Otomatis</title>
    <script src="https://unpkg.com/mqtt/dist/mqtt.min.js"></script>
    <style>
      body {
        margin: 0;
        padding: 0;
        font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
        background: linear-gradient(to right, #83a4d4, #b6fbff);
        display: flex;
        flex-direction: column;
        align-items: center;
        justify-content: center;
        height: 100vh;
      }

      h1 {
        color: #fff;
        margin-bottom: 20px;
        font-size: 2.5rem;
```

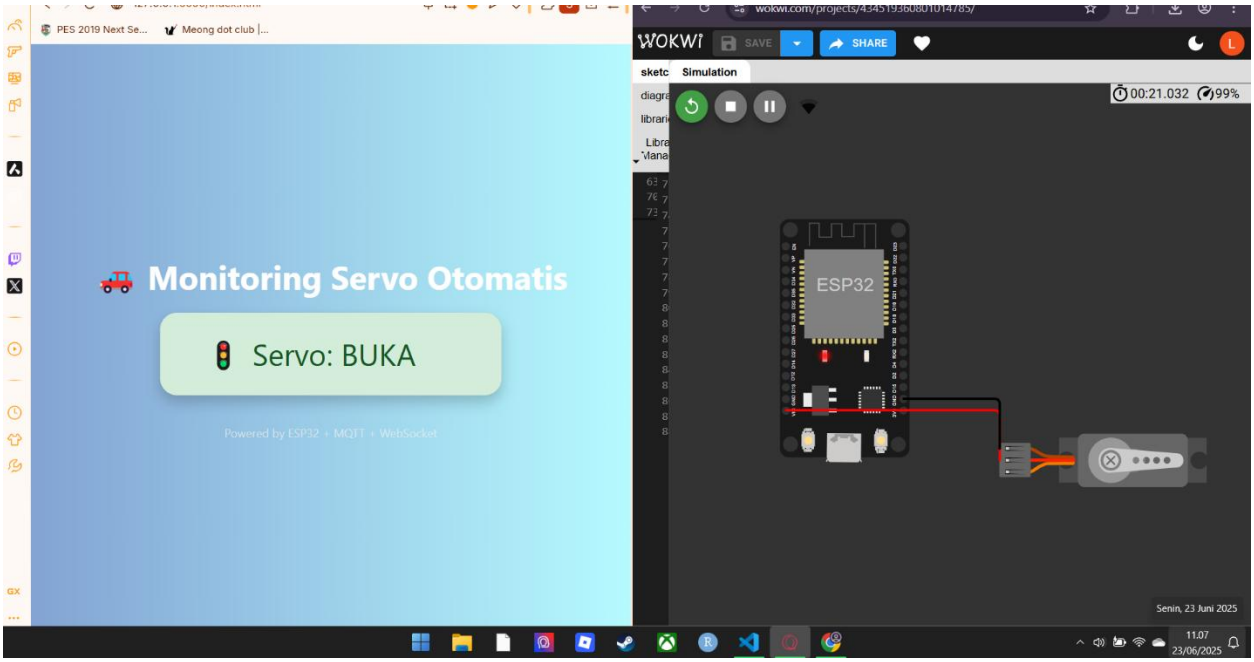
```
    }

    #status {
      font-size: 2.2rem;
      padding: 25px 50px;
      background-color: #ffffffdd;
      border-radius: 20px;
      box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
      transition: background-color 0.5s, transform 0.3s;
      color: #333;
      min-width: 300px;
    }

    #status.buka {
      background-color: #d4edda;
      color: #155724;
      transform: scale(1.05);
    }

    #status.tutup {
      background-color: #f8d7da;
      color: #721c24;
      transform: scale(1.05);
    }

    footer {
      margin-top: 40px;
      font-size: 0.9rem;
      color: #f0f0f0aa;
    }
  </style>
</head>
<body>
  <h1>🚗 Monitoring Servo Otomatis</h1>
  <div id="status">Menunggu data...</div>
  <footer>Powered by ESP32 + MQTT + WebSocket</footer>

  <script>
    const statusEl = document.getElementById("status");

    const client = mqtt.connect("wss://broker.hivemq.com:8884/mqtt");

    client.on("connect", function () {
      console.log("✅ Terhubung ke MQTT WebSocket");
      client.subscribe("iot/parking/servo", function (err) {
        if (!err) {
          console.log("📡 Subscribe ke topik berhasil");
        }
      });
```

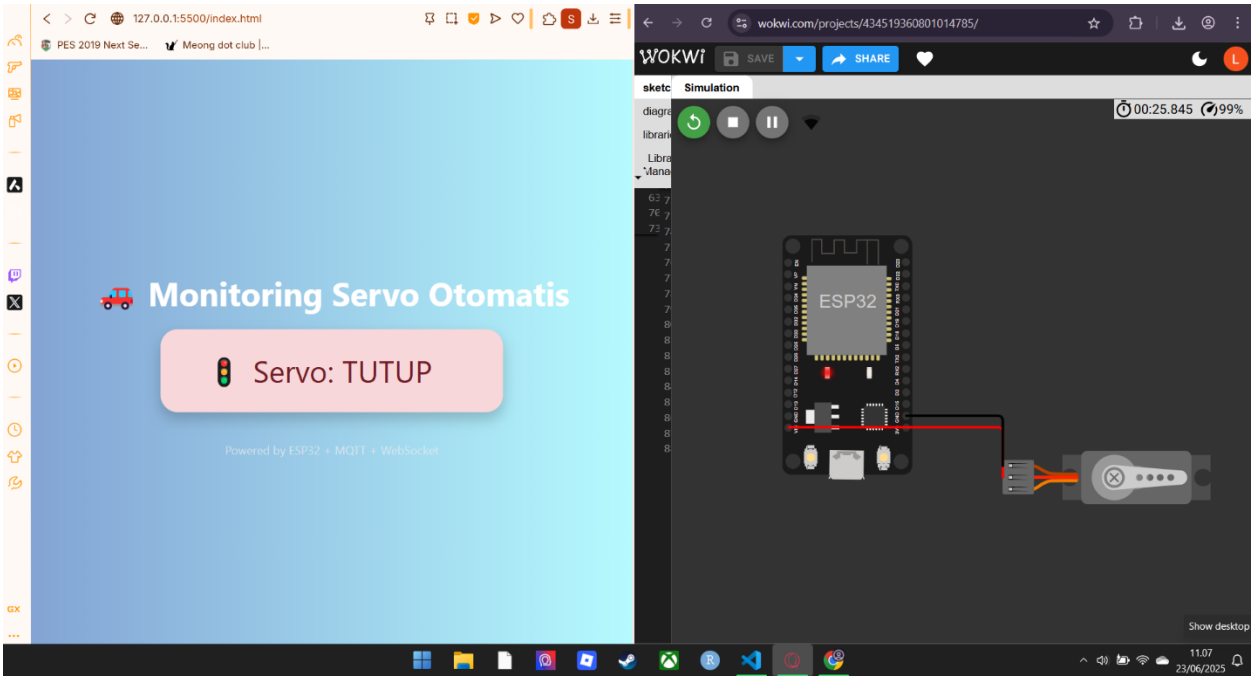
```
    });

    client.on("message", function (topic, message) {
      const status = message.toString().trim().toUpperCase();
      console.log(" 🔔 Status diterima:", status);

      statusEl.textContent = ` 🚦 Servo: ${status}`;

      // Tambahkan class visual
      if (status === "BUKA" || status === "AUTO BUKA") {
        statusEl.classList.add("buka");
        statusEl.classList.remove("tutup");
      } else if (status === "TUTUP" || status === "AUTO TUTUP") {
        statusEl.classList.add("tutup");
        statusEl.classList.remove("buka");
      } else {
        statusEl.classList.remove("buka", "tutup");
      }
    });
  </script>
 </body>
</html>
```

Lampiran:


wokwi.com/projects/434519360801014785/
WOKWI
SAVE
SHARE
parkir Copy
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
03:09.582
99%
ESP32
#include <WiFi.h>
#include <PubSubClient.h>
#include <ESP32Servo.h>
// WiFi Wokwi
const char* ssid = "Wokwi-GUEST";
const char* password = "";
// MQTT HiveMQ
const char* mqtt_server = "broker.hivemq.com";
const char* topic_servo = "iot/parking/servo";
WiFiClient espClient;
PubSubClient client(espClient);
Servo myServo;
const int servoPin = 33;
bool isOpen = false;
unsigned long lastToggleTime = 0;
unsigned long toggleInterval = 5000; // 5 detik
void reconnect() {
while (!client.connected()) {
Serial.print("🔄 Menghubungkan MQTT...");
if (client.connect("esp32ClientAuto")) {
Serial.println("✅ MQTT Terhubung!");
} else {
Serial.print("❌ Gagal, rc=");
Serial.println(client.state());
delay(2000);
}
}
}
12.53
23/06/2025


127.0.0.1:5500/index.html
PES 2019 Next Se...
Meong dot club |...
Monitoring Servo Otomatis
Menunggu data...
Powered by ESP32 + MQTT + WebSocket
wokwi.com/projects/434519360801014785/
WOKWI
SAVE
SHARE
Simulation
00:16.366
ESP32





Link Wokwi :
https://wokwi.com/projects/434519360801014785