

MODUL 9

ESP8266 STANDALONE WEB SERVER

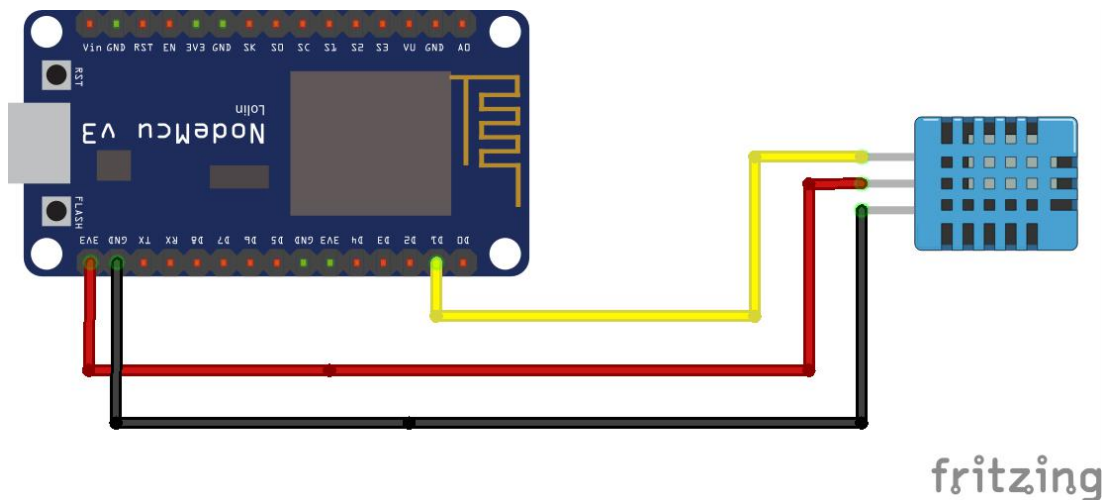
ESP8266 memiliki seluruh bagian/layer dari TCP/IP, sehingga secara independen ESP8266 mampu menjalankan fungsi sebagai application layer sekaligus transport layer. Kita dapat memprogram ESP8266 untuk menjalankan sebuah aplikasi web sederhana (dalam contoh ini adalah sebuah web server) sekaligus menjadikan ESP8266 sebagai media komunikasi antara sensor (things) dengan layer network.

Web server sederhana yang dibuat akan digunakan untuk mengambil data sensor DHT11 (temperatur dan kelembaban udara) untuk kemudian disajikan dalam sebuah tampilan web yang dapat diakses melalui jaringan nirkabel (WiFi).

ALAT DAN BAHAN

1. Modul ESP8266 Wemos Lolin (V3)
2. Kabel jumper (male-female x 4)
3. Breadboard
4. Sensor DHT11
5. Jaringan WiFi dan atau internet

RANGKAIAN DAN LANGKAH PRAKTIKUM



Gambar 9.1. Rangkaian ESP8266 Standalone Web Server

Langkah-langkah praktikum dijabarkan sebagai berikut.

1. Buat rangkaian seperti pada gambar di atas. Kemudian tulis kode berikut pada Arduino IDE. Perhatikan baik-baik **bagian kode yang dicetak tebal**.

```
#include <ESP8266WiFi.h>
#include "DHT.h"
#define DHTTYPE DHT11

const char* ssid = "<b><ganti dengan nama AP yang digunakan></b>";
const char* password = "<b><ganti dengan password AP yang digunakan (bila ada)></b>";

WiFiServer server(80);

const int DHTPin = 5;
DHT dht(DHTPin, DHTTYPE);

static char celsiusTemp[7];
static char fahrenheitTemp[7];
static char humidityTemp[7];

void setup() {
    Serial.begin(115200);
    delay(10);
    dht.begin();

    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    server.begin();
    Serial.println("Web server running. Waiting for the ESP IP...");
    delay(1000);
    Serial.println(WiFi.localIP());
}

void loop() {
    WiFiClient client = server.available();

    if (client) {
        Serial.println("New client");
        boolean blank_line = true;
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();
```

```

        if (c == '\n' && blank_line) {
            float h = dht.readHumidity();
            float t = dht.readTemperature();
            float f = dht.readTemperature(true);
            if (isnan(h) || isnan(t) || isnan(f)) {
                Serial.println("Failed to read from DHT sensor!");
                strcpy(celsiusTemp, "Failed");
                strcpy(fahrenheitTemp, "Failed");
                strcpy(humidityTemp, "Failed");
            }
            else{
                float hic = dht.computeHeatIndex(t, h, false);
                dtostrf(hic, 6, 2, celsiusTemp);
                float hif = dht.computeHeatIndex(f, h);
                dtostrf(hif, 6, 2, fahrenheitTemp);
                dtostrf(h, 6, 2, humidityTemp);
            }
            client.println("HTTP/1.1 200 OK");
            client.println("Content-Type: text/html");
            client.println("Connection: close");
            client.println();
            client.println("<!DOCTYPE HTML>");
            client.println("<html>");
            client.println("<head></head><body><h1>ESP8266 -
Temperature and Humidity</h1><h3>Temperature in Celsius: ");
            client.println(celsiusTemp);
            client.println("<C</h3><h3>Temperature in Fahrenheit: ");
            client.println(fahrenheitTemp);
            client.println("<F</h3><h3>Humidity: ");
            client.println(humidityTemp);
            client.println("<%</h3><h3>");
            client.println("</body></html>");
            break;
        }
        if (c == '\n') {
            blank_line = true;
        }
        else if (c != '\r') {
            blank_line = false;
        }
    }
}
delay(1);
client.stop();
Serial.println("Client disconnected.");
}
}

```

2. Verify/compile dan bila tidak ada error, buka Serial Monitor, kemudian upload kode tersebut ke modul ESP8266.

3. Tujuan membuka Serial Monitor sebelum melakukan upload program ke modul ESP8266 adalah supaya kita bisa langsung mendapatkan informasi alamat IP dari modul ESP8266 (tanpa harus melakukan reset pada modul).
4. Akses web server sederhana dengan cara membuka alamat IP dari modul ESP8266 di browser yang terhubung pada AP yang sama.
5. Amati hasilnya dan catat serta cermati apabila ditemukan error.

TUGAS PRAKTIKUM

1. Buatlah Standalone Web Server dengan ES8266 dengan menggunakan sensor lain selain DHT11!