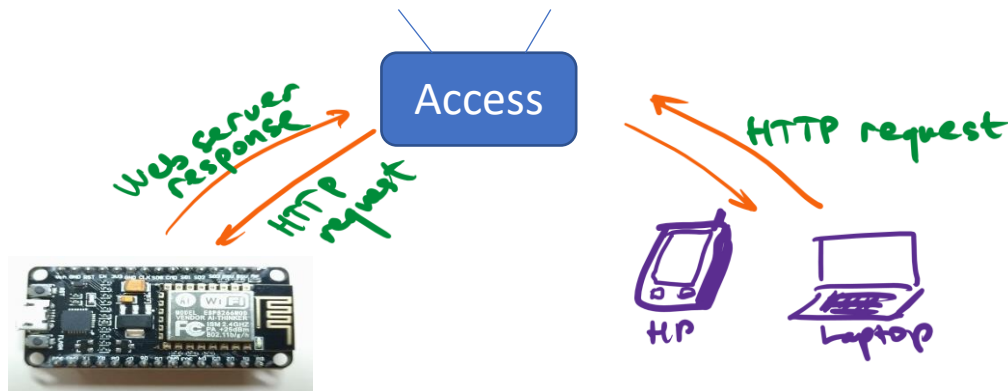


IoT-Based Access Control System Using ESP8266 and Web Server

Raven Daniel Martin

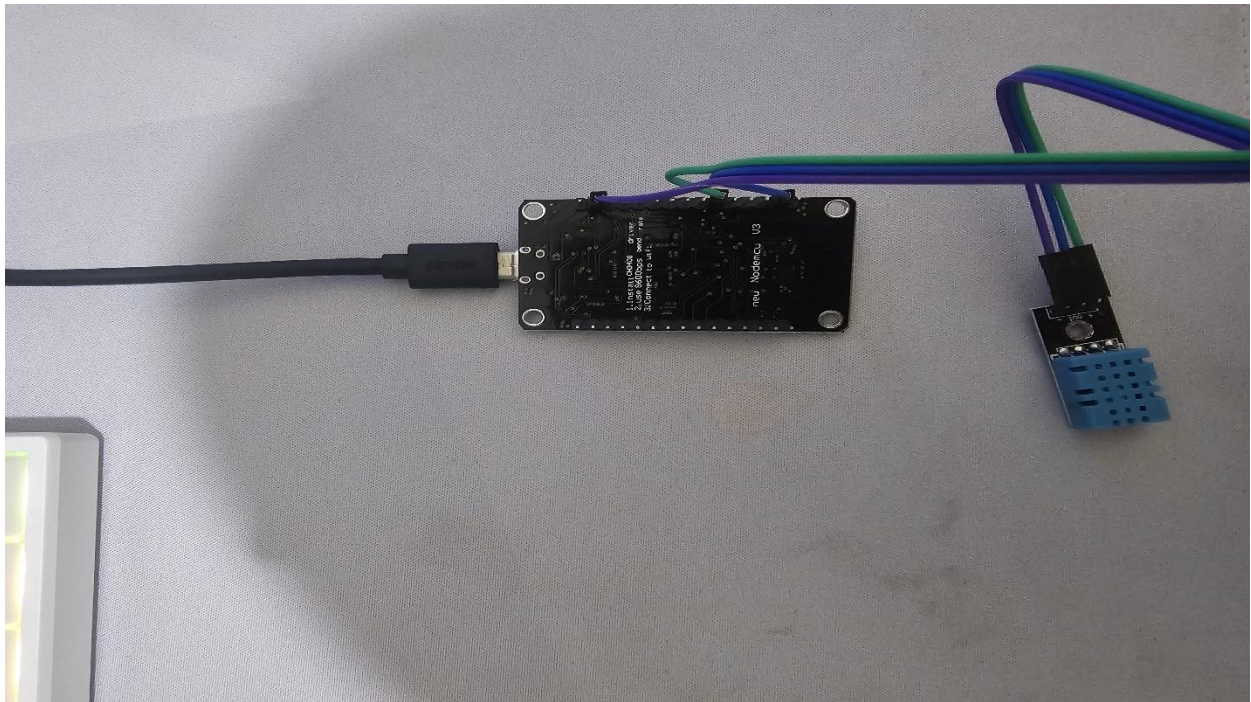


Lakukan langkah-langkah Percobaan

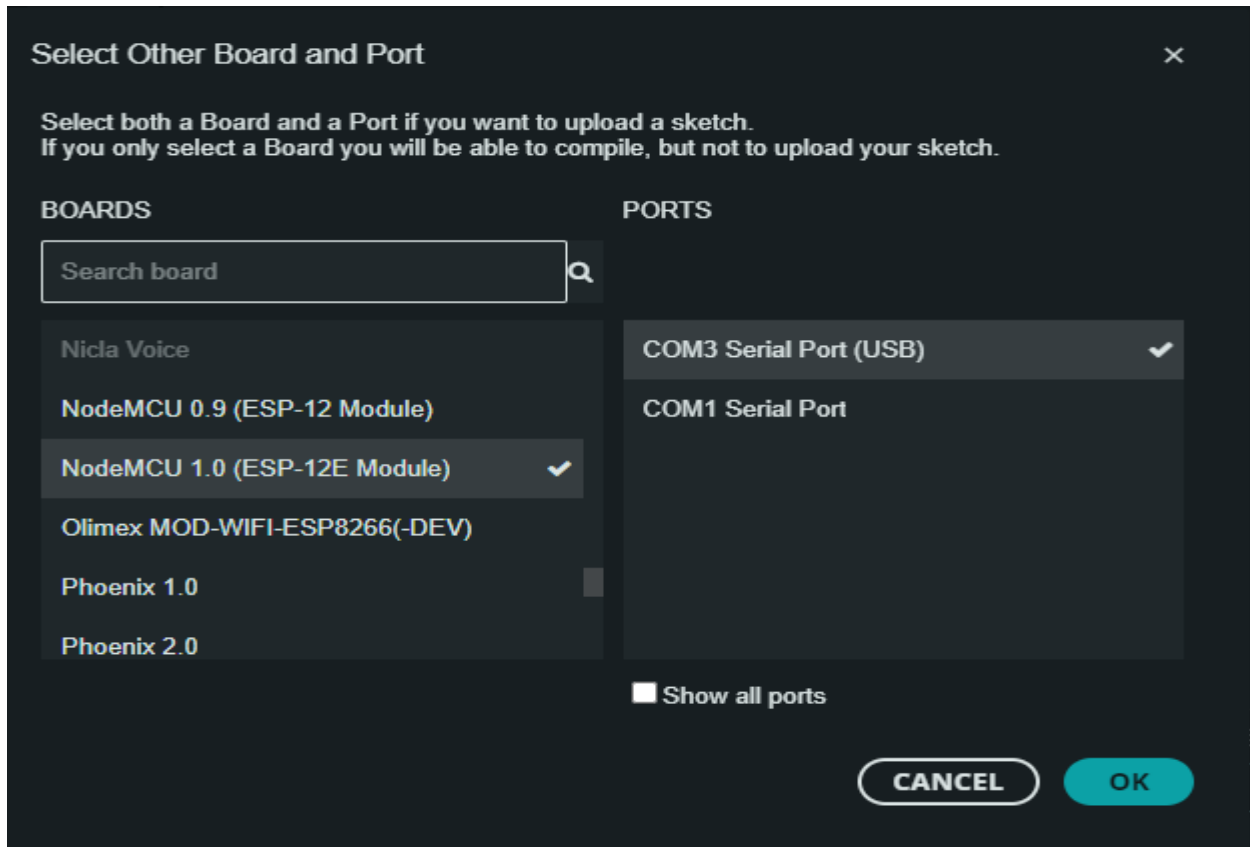
- Merangkai IoT device sederhana menggunakan ESP8266 atau ESP32.
- Perangkat diprogram dengan Arduino IDE.
- Sensor yang digunakan adalah sensor suhu dengan keluaran tegangan analog, LM-35.

Sebagai proof-of-concept bahwa IoT device kita dapat terhubung ke jaringan TCP/IP, skenario yang diambil adalah skenario sederhana menjadikan ESP8266 atau ESP32 sebagai webserver di dalam jaringan adhoc lokal

DHT11	NODEMCU	KABEL JUMPER
pin Kutub negatif (-)	pin Ground	Ungu
pin Kutub positif (+) / VCC	pin 3V	Hijau
pin Output	pin D1 (Digital)	Biru



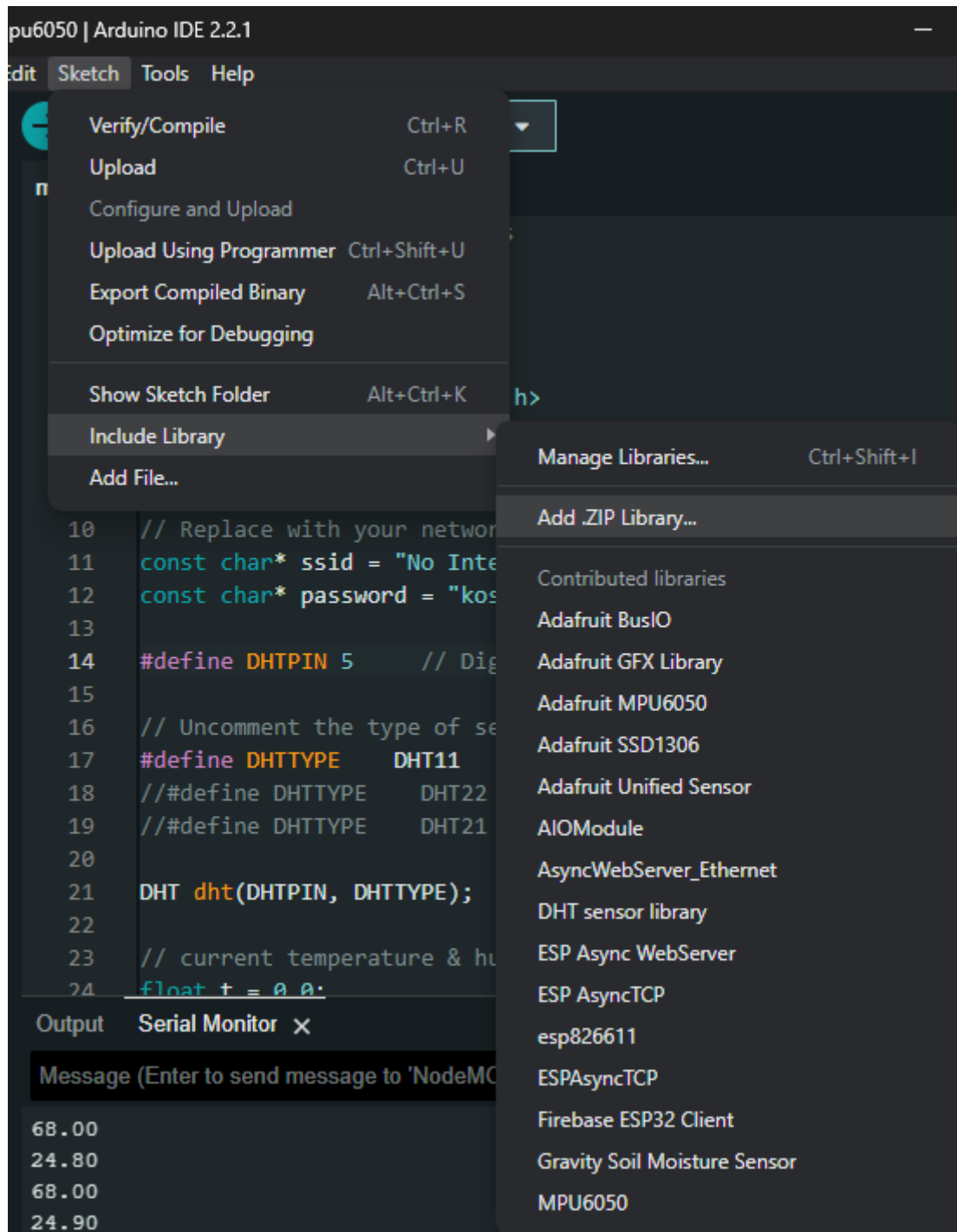
Hal yang utama, Arduino IDE perlu ter-install library DHT dan Adafruit Unified Sensor seperti pada Tugas Personal 2-IOT. Selanjutnya pada aplikasi Arduino IDE pilih Board dan Port seperti pada gambar dibawah. Karena saya menggunakan ESP8266, maka board yang dipilih adalah NodeMCU 1.0 (ESP-12E Module), sedangkan untuk Port disesuaikan dengan komputer masing-masing.



Untuk membuat server lokal dan tampilan web menggunakan html, langkah pertama adalah melakukan instalasi library ESPAsyncWebServer dan ESPAsyncTCP di Arduino IDE secara manual dengan link dibawah.

<https://github.com/me-no-dev/ESPAsyncWebServer/archive/master.zip>
<https://github.com/me-no-dev/ESPAsyncTCP/archive/master.zip>

Setelah mendownload file library tersebut, kita perlu menambahkan file library tersebut kedalam Arduino IDE secara manual dengan cara berikut. Sketch > Include Library > add .ZIP library. Setelah berhasil lakukan restart pada Arduino IDE.



Setelah melakukan install seluruh kebutuhan library, selanjutnya masuk ke tahap pemrograman skenario menjadikan ESP8266 sebagai webserver di dalam jaringan adhoc lokal. Hal pertama yang dilakukan adalah import library yang dibutuhkan (library yang diinstal sebelumnya pada langkah awal).

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Hash.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
```

Selanjutnya membuat program agar ESP8266 dapat terkoneksi kepada wifi lokal yang akan digunakan. Informasi dibawah merupakan SSID dan Password yang saya gunakan, dapat disesuaikan dengan informasi jaringan hotspot lain.

```
// Replace with your network credentials
const char* ssid = "No Internet";
const char* password = "kosonginaja";
```

Langkah selanjutnya adalah menentukan variable yang akan digunakan. Seperti gambar dibawah: Saya menggunakan pin D1/GPIO5 pada ESP8266 maka saya program DHTPIN menjadi 5, dan DHTTYPE yang digunakan DHT11 serta output data (temperatur dan humidity) akan berupa angka float dengan dua desimal.

```
#define DHTPIN 5 // Digital pin connected to the DHT sensor

// Uncomment the type of sensor in use:
#define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT22 // DHT 22 (AM2302)
// #define DHTTYPE DHT21 // DHT 21 (AM2301)

DHT dht(DHTPIN, DHTTYPE);

// current temperature & humidity, updated in loop()
float t = 0.0;
float h = 0.0;
```

Kemudian hal yang perlu dilakukan adalah mempersiapkan webserver (menggunakan port 80), dengan DHT akan membaca serta memberikan data setiap 10 detik sekali. Interval penginderaan sensor dapat disesuaikan dengan kebutuhan.

```
// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0; // will store last time DHT was updated

// Updates DHT readings every 10 seconds
const long interval = 10000;
```

Langkah selanjutnya adalah membuat tampilan web berbasis html. Tentunya pada bagian ini dapat dibuat sesuai kreativitas individu masing-masing dalam menampilkan datanya pada sebuah web.

```
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-
fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
  <style>
    html {
      font-family: Arial;
      display: inline-block;
      margin: 0px auto;
      text-align: center;
    }
    h2 { font-size: 3.0rem; }
    p { font-size: 3.0rem; }
    .units { font-size: 1.2rem; }
    .dht-labels{
      font-size: 1.5rem;
      vertical-align:middle;
      padding-bottom: 15px;
    }
  </style>
</head>
<body>
  <h2>ESP8266 DHT Server</h2>
  <p>
    <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
    <span class="dht-labels">Temperature</span>
    <span id="temperature">%TEMPERATURE%</span>
    <sup class="units">&deg;C</sup>
  </p>
  <p>
    <i class="fas fa-tint" style="color:#00add6;"></i>
    <span class="dht-labels">Humidity</span>
    <span id="humidity">%HUMIDITY%</span>
    <sup class="units">%</sup>
  </p>
</body>
```

Dari program diatas saya menggunakan tag <meta> agar membuat halaman web lebih responsif di setiap aplikasi browser yang akan digunakan. Tag <link> digunakan untuk

menambahkan icon temperatur dan humidity dari web fontawesome. Tag <style> digunakan untuk membuat gaya tampilan pada halaman web berbasis CSS. Dan tag lainnya untuk menyesuaikan text serta posisi tampilan (text, label, icon, warna) berbasis html. Hal ini tentunya dapat dibuat sesuai keinginan masing-masing.

Langkah selanjutnya menambahkan beberapa script berbasis program JavaScript agar halaman web dapat menampilkan perubahan data temperatur dan humidity secara otomatis setiap 10 detik. Tag script yang digunakan dalam html berupa <script></script>

```
<script>
setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("temperature").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/temperature", true);
    xhttp.send();
}, 10000 ) ;

setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("humidity").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/humidity", true);
    xhttp.send();
}, 10000 ) ;
</script>
</html>>rawliteral";
```

Selanjutnya kita perlu membuat processor agar text berubah sesuai data yang telah diatur (interval 10 detik) pada html.

```
// Replaces placeholder with DHT values
String processor(const String& var){
    //Serial.println(var);
    if(var == "TEMPERATURE"){
        return String(t);
    }
    else if(var == "HUMIDITY"){
        return String(h);
    }
    return String();
}
```

Langkah selanjutnya adalah melakukan setup untuk proses debugging. Serial monitor disetup dengan kecepatan 115200 baud dan dilanjutkan dengan inisiasi sensor DHT untuk melakukan penginderaan. Setelah itu menghubungkan ESP8266 ke jaringan lokal yang telah disediakan. Dan terakhir menambahkan program untuk meng-handle webserver, ketika kita request root URL, kita akan mengirimkan html text yang akan disimpan pada index_html dan akan dihubungkan ke fungsi processor yang akan meng-update text ke nilai yang sebenarnya dengan program server.on. serta memulai server dengan server.begin();

```
void setup(){
  // Serial port for debugging purposes
  Serial.begin(115200);
  dht.begin();

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  Serial.println("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println(".");
  }

  // Print ESP8266 Local IP Address
  Serial.println(WiFi.localIP());

  // Route for root / web page
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html, processor);
  });
  server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", String(t).c_str());
  });
  server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", String(h).c_str());
  });

  // Start server
  server.begin();
}
```

Langkah terakhir adalah menggunakan loop agar kita dapat melakukan perintah penginderaan sensor DHT11 setiap 10 detik. Setelah itu kita lakukan Upload pada program Arduino IDE dan kemudian pada serial monitor akan menampilkan IP address webserver yang telah kita buat.


```
void loop(){
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    // save the last time you updated the DHT values
    previousMillis = currentMillis;
    // Read temperature as Celsius (the default)
    float newT = dht.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit = true)
    //float newT = dht.readTemperature(true);
    // if temperature read failed, don't change t value
    if (isnan(newT)) {
      Serial.println("Failed to read from DHT sensor!");
    }
    else {
      t = newT;
      Serial.println(t);
    }
    // Read Humidity
    float newH = dht.readHumidity();
    // if humidity read failed, don't change h value
    if (isnan(newH)) {
      Serial.println("Failed to read from DHT sensor!");
    }
    else {
      h = newH;
      Serial.println(h);
    }
  }
}
```

Berikut adalah tampilan webserver yang telah kita buat menggunakan ESP8266 dan DHT11 berbasis html.:

