

## 7.3 Praktikum Membuat Web Server Monitoring dan Kendali



Indobot Academy 21 November 2022

### 1. Dasar Teori

#### 1.1. Frontend VS Backend

**Frontend** maupun **Backend** merupakan istilah yang sangat populer untuk mendefinisikan bagian dari pengembangan website. Dengan adanya istilah tadi, proses pengembangan web menjadi lebih mudah diatur sehingga dapat mencapai hasil yang maksimal.

**Frontend** adalah bagian website yang mengatur tentang berbagai macam hal yang berhubungan dengan desain visual (bagian depan layar). Frontend ini pada umumnya digunakan sebagai media interaksi antara sistem dengan pengguna. Bagian frontend ini meliputi tata letak, warna, font, menu, navigasi, dan lain sebagainya. Bahasa yang digunakan untuk keperluan frontend adalah HTML, CSS, dan Javascript.

**Backend** adalah bagian website yang mengatur tentang berbagai macam hal yang berhubungan dengan logika pemrograman (bagian balik layar). Backend ini pada umumnya digunakan sebagai media sisi server (server-side). Bagian backend ini meliputi database, security, dan lain sebagainya. Bahasa yang kerap digunakan untuk keperluan backend diantaranya yaitu SQL dan PHP.



Gambar 1. Frontend VS Backend (www.konstaninfo.com)

## 1.2. Fullstack

Sebelumnya anda telah mengenal apa itu frontend dan backend, nah pada kesempatan kali ini anda akan belajar tentang apa itu fullstack. **Fullstack** adalah gabungan dari **Frontend** dan **Backend**, yang pasti akan mengatur banyak hal dari sisi pengembangan. Disini programmer dituntut harus memahami ranah frontend dan backend sekaligus.

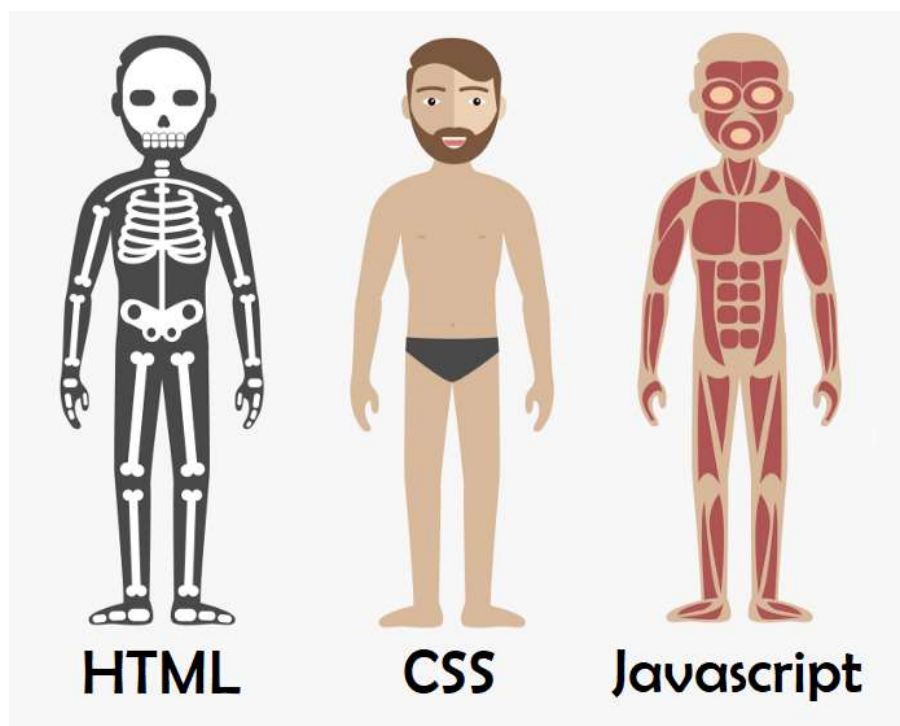


Gambar 2. Fullstack (www.weblineindia.com)

## 1.3. Bahasa Pemrograman Website Dasar

### 1.3.1. JavaScript

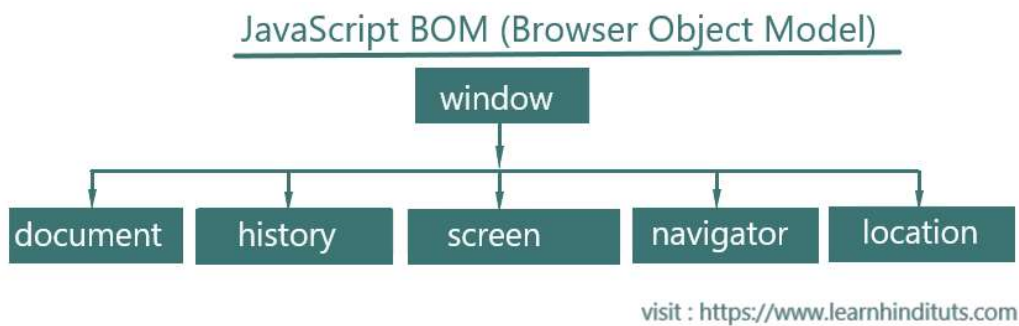
**JavaScript (JS)** adalah bahasa scripting yang menjadi elemen penting dalam pengembangan suatu website. Saat ini, hampir semua situs web menggunakan JS. **JavaScript ES2015 (ES6)** adalah versi terbaru JS, yang tergolong populer karena lebih efisien daripada versi sebelumnya. Dalam pemrograman JavaScript, anda juga akan mengenal tentang istilah **BOM** dan **DOM**. JavaScript ini dapat diilustrasikan sebagai saraf atau otot pada tubuh manusia.



Gambar 3. Ilustrasi JavaScript (velog.io)

- **BOM (Browser Object Model)**

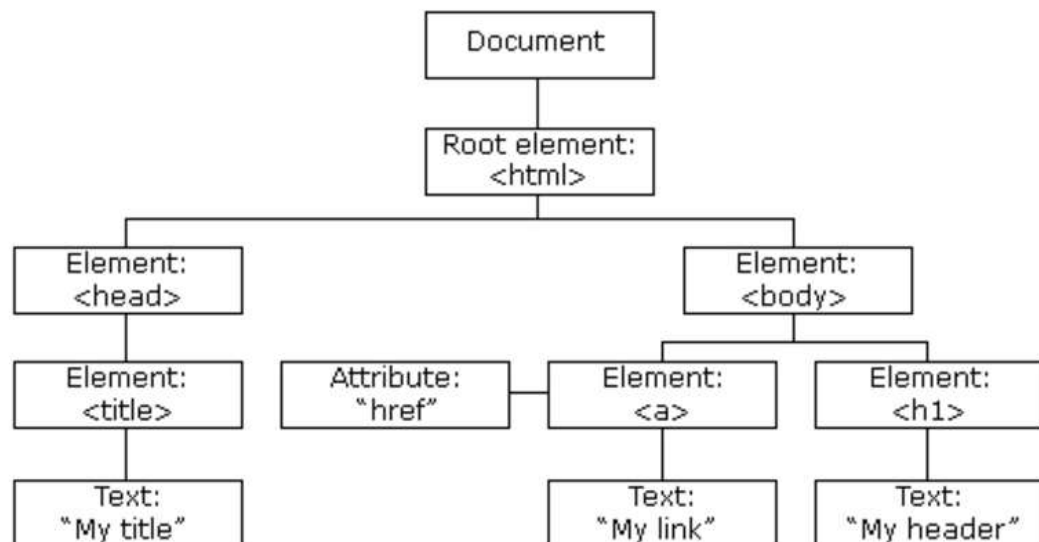
adalah sebuah cara bagi bahasa pemrograman untuk dapat melakukan kegiatan manipulasi pada browser secara langsung. BOM mempunyai 5 objek yaitu meliputi **Document**, **Location**, **Navigator**, **Screen**, dan **History**. Semua objek tersebut juga merupakan bagian dari global variable yang bernama **window object**. Dengan bahasa pemrograman javascript, terdapat cara untuk mengakses BOM, yaitu dengan menggunakan perintah window.



Gambar 4. Konsep BOM ([www.learnhindituts.com](https://www.learnhindituts.com))

- **DOM (Document Object Model)**

adalah kumpulan dari beberapa objek yang mencakup sebagian besar struktur, style, dan isi dari suatu dokumen yang ada di dalam sebuah website. DOM dalam JS berkaitan erat dengan halaman web (**HTML atau XML**). Lalu, DOM ini ada di dalam window object, dimana window object adalah bagian dari BOM. Dengan adanya DOM ini, anda dapat memanipulasi berbagai macam dokumen. Dengan bahasa pemrograman javascript, terdapat beberapa cara untuk mengakses elemen DOM, yaitu dengan menggunakan **id**, **class**, atau bahkan dengan **tag html** itu sendiri.



Gambar 5. Konsep DOM (Henrique M. Gaspar - ResearchGate)

Baik **BOM** maupun **DOM** dapat digunakan secara bersamaan karena keduanya saling terkait. Jika yang dibutuhkan adalah melakukan perubahan pada dokumen atau element HTML seperti isi **konten**, **struktur**, atau **style** maka konsep DOM yang akan digunakan. Akan tetapi, jika ternyata yang diperlukan adalah manipulasi atau pengambilan informasi pada browser seperti **tinggi window**, **lokasi**, **history**, dan lain sebagainya maka konsep BOM yang akan digunakan.

### 1.3.2. SQL

**SQL** merupakan bahasa pemrograman database untuk mengatur dan menganalisis data. SQL merupakan bahasa baku (**ANSI/SQL**), non procedural, dan berorientasi pada himpunan (**set-oriented language**). SQL dapat digunakan baik secara **interaktif** atau ditempelkan (**embedded**) pada sebuah program aplikasi.

Secara umum, bahasa pemrograman SQL memiliki beberapa bagian penting didalamnya, yaitu diantaranya:

- **Data Definition Language (DDL)**

DDL menyediakan perintah-perintah untuk mendefinisikan skema relasi, menghapus relasi, serta memodifikasi skema relasi. Perintah SQL pada kelompok ini terdiri dari **CREATE**, **ALTER**, **DROP**, dan lain sebagainya.

```
25
26  -- CREATE
27  CREATE TABLE Person(
28      PersonID INT IDENTITY (1,1) CONSTRAINT PK_PersonID PRIMARY KEY,
29      FirstName NVARCHAR(20),
30      LastName NVARCHAR(25)
31  );
32
33  -- ALTER
34  ALTER TABLE Person
35  ADD BirthDate DATETIME
36
37  -- DROP
38  DROP TABLE Person
39
```

Gambar 6. Contoh DDL (almirvuk.blogspot.com)

- **Data Manipulation Language (DML)**

DML menyediakan perintah-perintah untuk menyisipkan rekaman pada relasi, menghapus rekaman pada relasi, serta memodifikasi rekaman pada relasi. Perintah SQL pada kelompok ini terdiri dari **INSERT**, **SELECT**, **UPDATE**, **DELETE**, dan lain sebagainya.

```

4  -- SELECT
5  SELECT *
6  FROM employee
7
8  -- INSERT
9  INSERT INTO employee(emp_id, fname, minit, lname,
10                      job_id, job_lvl, pub_id, hire_date)
11                      VALUES('0000000', 'Almir', 'M', 'Vuk',
12                              7, 12, 1207, 2009-05-09)
13
14  -- UPDATE
15  UPDATE employee
16  SET fname = 'ALMIR'
17  WHERE emp_id = '0000000'
18
19  -- DELETE
20  DELETE
21  FROM employee
22  WHERE emp_id = '0000000'

```

Gambar 7. Contoh DML (almirvuk.blogspot.com)

- **Data Control Language (DCL)**

DCL menyediakan perintah-perintah untuk melakukan pengontrolan data dan server, seperti manipulasi user dan hak akses (priviledges). Perintah SQL pada kelompok ini terdiri dari **GRANT**, **REVOKE**, dan lain sebagainya.

```

40
41  -- GRANT
42  GRANT SELECT, INSERT, UPDATE, DELETE ON Employees TO almir
43
44  -- REVOKE
45  REVOKE INSERT ON Employees TO almir
46
47  -- DENY
48  DENY UPDATE ON Employees TO almir
49

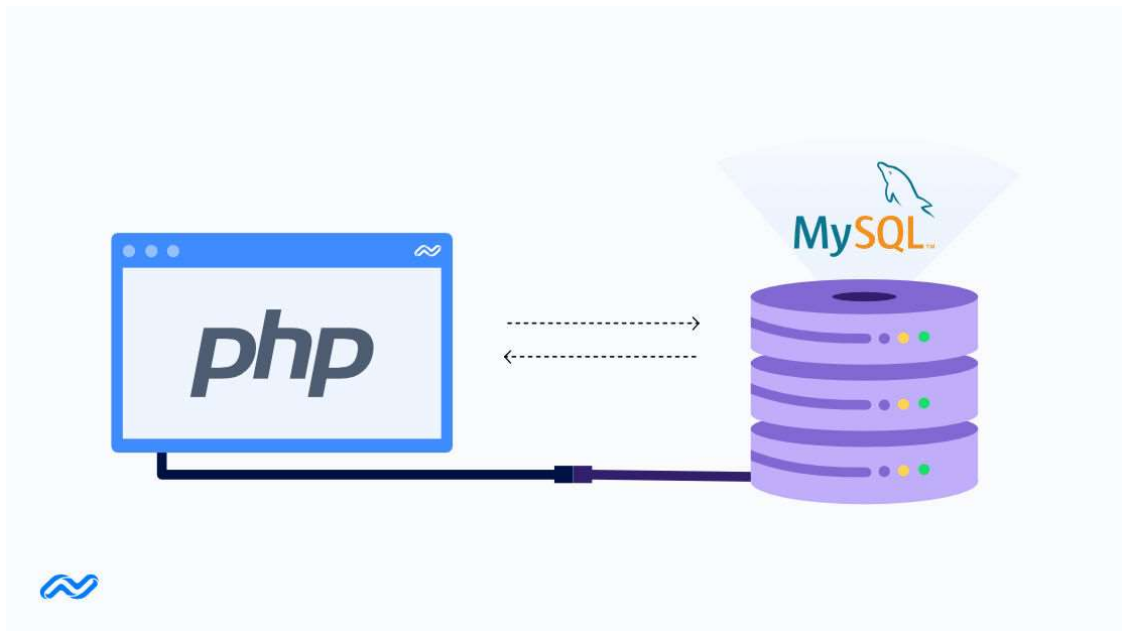
```

Gambar 8. Contoh DCL (almirvuk.blogspot.com)

### 1.3.3. PHP

**PHP** adalah singkatan dari **PHP:Hypertext Preprocessor**. PHP merupakan bahasa script yang dijalankan pada **sisi server (SSS: Server Side Scripting)**. Database yang didukung oleh PHP, antara lain meliputi: **MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL**, dan **Generic ODBC**. File PHP dapat berisi **teks, tag HTML**, dan **Script**. File PHP dapat berekstensi **.php, .php3**, atau **.phtml**. PHP mempunyai beberapa kelebihan, antara lain: dapat dijalankan di berbagai platform (**Windows, Linux**, dan lain-lain), kompatibel terhadap hampir semua server yang digunakan saat ini, bebas di unduh dari situs resmi (**www.php.net**), mudah dipelajari, dan berjalan dengan efisien pada sisi server.





Gambar 9. Konsep PHP ([www.niagahoster.co.id](http://www.niagahoster.co.id))

## 2. Alat/Instrumen/Aparatus/Bahan

1. Browser (Contoh : Google Chrome).
2. Wemos D1 R1/R2.
3. LED.
4. Sensor DHT11.
5. Resistor 470 ohm.
6. Project Board.
7. Kabel Jumper.

## 3. Keselamatan Kerja

### 3.1. Pemasangan Komponen

Perhatikan bagian pin yang digunakan. Terutama bagian komponen yang memiliki polaritas, jangan sampai terbalik antara kaki positif dan negatif.

### 3.2. Penggunaan Ukuran Resistor

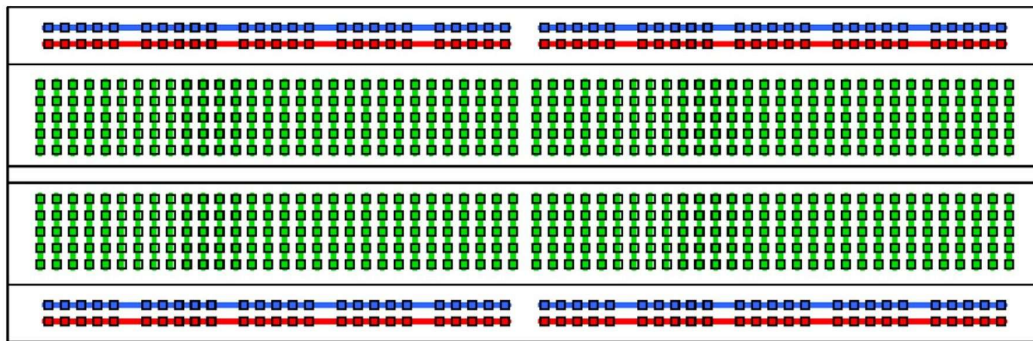
Hal yang perlu diperhatikan lainnya adalah mengenai ukuran resistor. Ukuran resistor dapat menyesuaikan dengan gambar rangkaiannya.

### 3.3. Perhatikan Pin

Selanjutnya kita juga perlu memperhatikan pin yang ada dalam wemos D1 maupun sensor.

### 3.4. Pemahaman Jalur Project Board

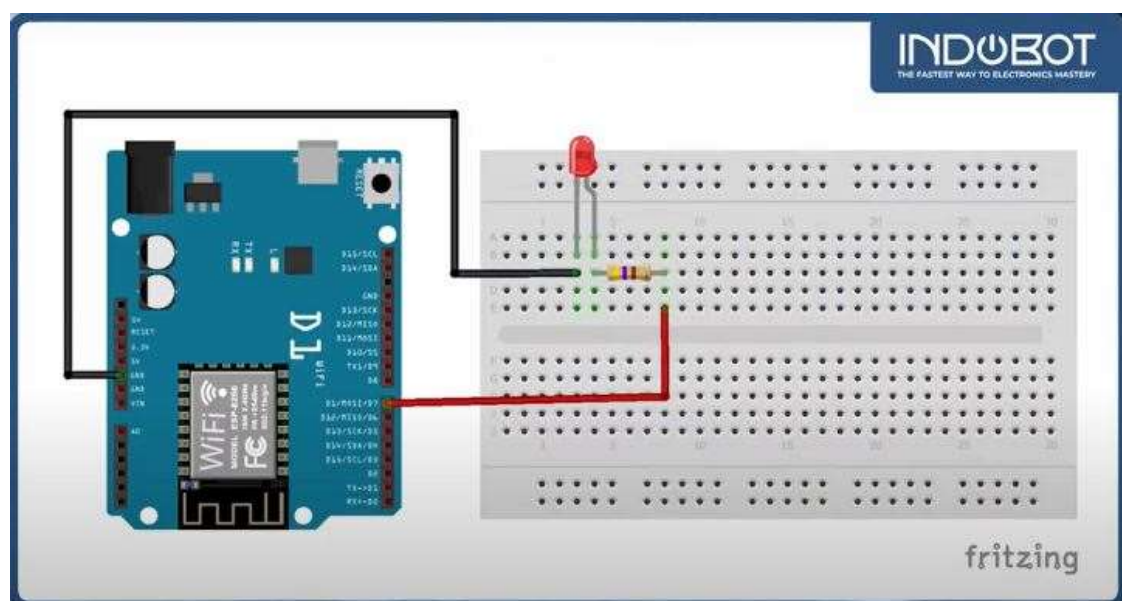
Agar kita mengetahui tentang jalur yang ada pada project board, kita bisa melihat gambar skema dalam project board seri MB-102 berikut ini.



- Bagian tengah project board akan saling terhubung secara vertikal setiap 5 pin. Kemudian akan ada celah, nah celah ini dapat anda gunakan untuk meletakkan push button atau komponen lainnya.
- Untuk bagian atas dan bawah ini terhubung secara horizontal, dengan celah juga di bagian tengah dari project board.

## 4. Langkah Kerja Praktikum 1 – Membuat Kontrol LED Pada Web Browser Melalui Arduino IDE

### 4.1. Buatlah rangkaian seperti gambar di bawah ini.



Gambar 10. Rangkaian LED



**Note : Harap periksa kembali Wemos yang anda miliki! Dikarenakan ada 2 versi yaitu Wemos D1 R1 dan Wemos D1 R2.**

**Keterangan :**

- Hubungkan satu Pin kaki LED pada Pin GND Wemos D1.
- Hubungkan Pin kaki LED lainnya pada Resistor.
- Hubungkan Pin kaki resistor lainnya pada Pin D7 Wemos D1.

## **4.2. Coding**

```
// Kumpulan Library
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
// Inisiasi pin LED
#define LED D7

// Nama Hotspot atau Access Point yang dipakai oleh
Wemos anda
const char* ssid = "INDOBOTACADEMY";
// Password Hotspot atau Access Point yang dipakai
oleh Wemos anda
const char* password = "belajariot";

// Buat objek ESP8266WebServer pada port 80
ESP8266WebServer Server(80);
// String untuk menampung sintaks webpage
String webpage;

// Method yang dijalankan sekali
void setup() {
    // Inisiasi baudrate dan status hardware
    Serial.begin(115200);
    delay(10);
    pinMode(LED, OUTPUT);

    // Mengkoneksikan ke jaringan WiFi
    Serial.println();
    Serial.print("Configuring access point...");
```

```

// Mengatur WiFi
WiFi.mode(WIFI_AP);
WiFi.softAP(ssid, password);

// Menunggu Koneksi
Serial.println("");
Serial.print("Nama Hotspot/Access Point Anda: ");
Serial.println(ssid);
Serial.println("IP Address : ");
Serial.println(WiFi.softAPIP());

// Tampilan yang akan muncul pada Web Browser
webpage+= "<h1><center>Kendali LED INSONERS</center>
</h1>";
webpage+= "<br>";
webpage+= "<center>LED :</center>";
webpage+= "<center><a href=\"LEDON\">
<button>ON</button><a href=\"LEDOFF\">
<button>OFF</button></center>";

// Pengiriman HTML
Server.on("/", [] () {
    Server.send(200, "text/html", webpage);
});

// Bagian ini untuk merespon perintah /LEDON
Server.on("/LEDON", [] () {
    Server.send(200, "text/html", webpage);
    digitalWrite(LED, HIGH);
    delay(1000);
});

// Bagian ini untuk merespon perintah /LEDOFF
Server.on("/LEDOFF", [] () {
    Server.send(200, "text/html", webpage);
    digitalWrite(LED, LOW);
    delay(1000);
});

```

```

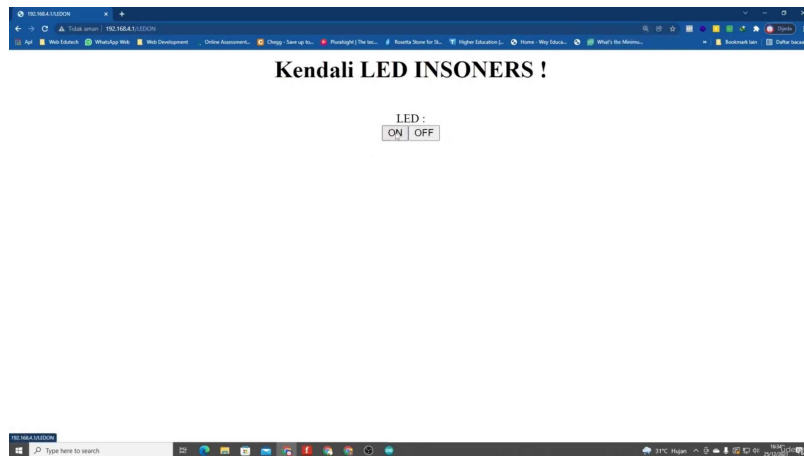
}

// Method yang dijalankan berkali-kali
void loop() {
    Server.handleClient();
}

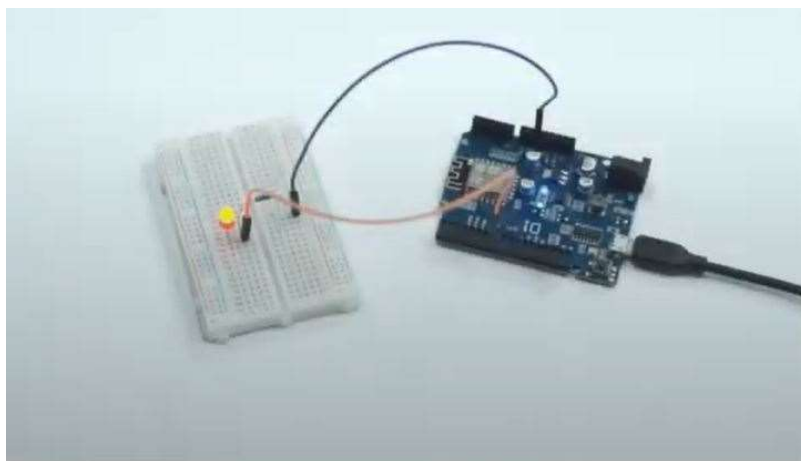
```

### 4.3. Hasil

Setelah program berhasil diupload, maka silahkan cek IP Address di bagian Serial Monitor Arduino IDE. Lalu, bukalah browser terlebih dahulu untuk dapat menuliskan IP Addressnya. Adapun tampilan yang ada pada web browser dapat anda ketahui pada gambar di bawah ini.

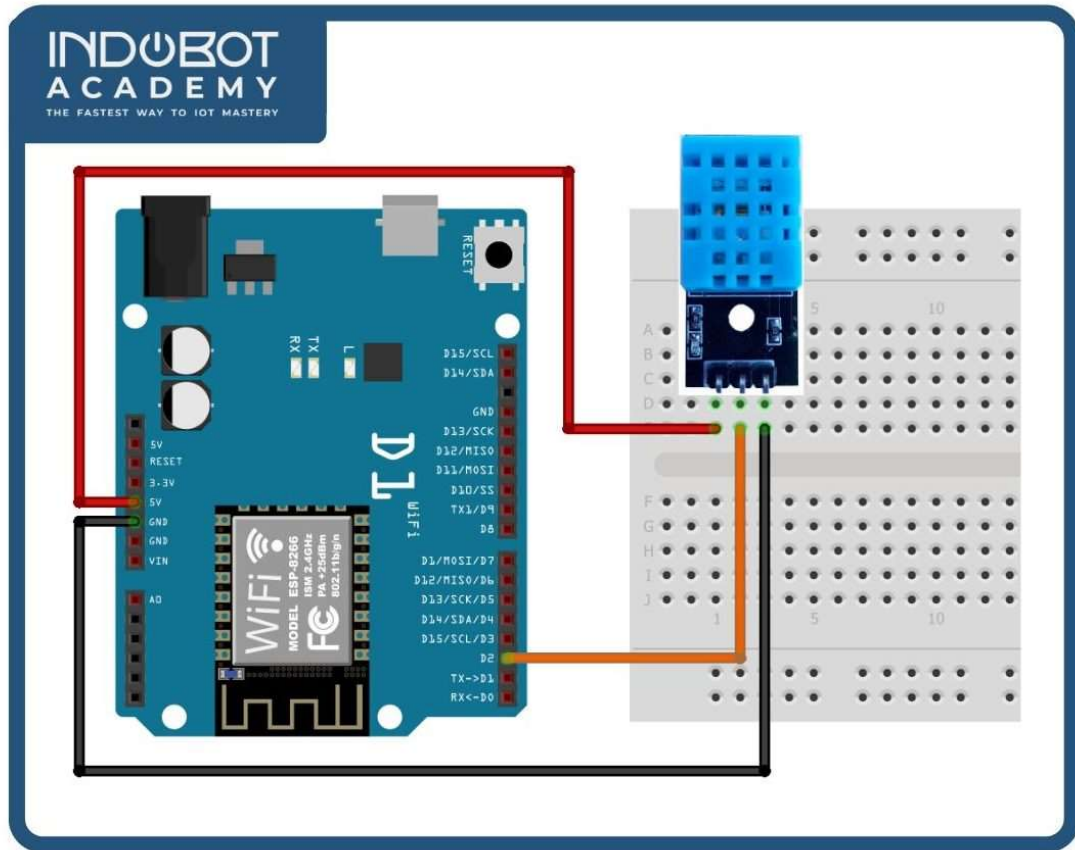


Tombol ON-OFF yang ada pada web browser merupakan saklar yang dapat mengatur hidup atau matinya suatu LED. Ketika anda menekan tombol ON maka LED akan menyala, sedangkan jika anda menekan tombol OFF maka LED akan mati. Reaksi dari LED dapat anda lihat pada gambar berikut ini.



## 5. Langkah Kerja Praktikum 2 – Membuat Monitoring Suhu dan Kelembapan Udara Pada Web Browser Melalui Arduino IDE

5.1. Buatlah rangkaian seperti gambar di bawah ini.



Gambar 11. Rangkaian Sensor DHT11

**Note :** Harap periksa kembali Wemos yang anda miliki! Dikarenakan ada 2 versi yaitu Wemos D1 R1 dan Wemos D1 R2.

**Keterangan :**

- Hubungkan PIN VCC sensor pada PIN 5V Wemos.
- Hubungkan PIN GND Sensor DHT pada PIN GND Wemos.
- Hubungkan PIN Data Sensor pada PIN D2 Wemos.

### 5.2. Penginstalan Library

Library yang dipakai dalam proyek ini ada 4, yaitu :

- ESP8266.h (sudah bawaan)
- ESPAsyncTCP.h (belum terinstall)

- ESPAsyncWebServer.h (belum terinstall)
- DHT.h (belum terinstall)

### 5.2.1. Library ESPAsyncTCP

Library ESPAsyncTCP merupakan library TCP yang mampu membuat lingkungan jaringan multi-koneksi bebas dari masalah dan asinkron sepenuhnya. Library ini adalah basis untuk ESPAsyncWebServer. Pemasangan library ini dapat dilakukan secara manual pada software Arduino IDE atau juga dapat melalui library manager. Jika anda ingin memasang library secara manual, anda dapat mengikuti langkah-langkah berikut ini :

- Download library ESPAsyncTCP :  
<https://github.com/me-no-dev/ESPAsyncTCP>.
- Kemudian jika anda telah berhasil mendownload, ubah nama zip dari ~~ESPAsyncTCP-master~~ menjadi *ESPAsyncTCP*.
- Lalu Klik "sketch" pada menu Arduino IDE.
- Klik "include library".
- Klik ikon "add .Zip Libraries.
- Setelah itu cari file .Zip yang sudah anda download tadi.

### 5.2.2. Library ESPAsyncWebServer

Library ESPAsyncWebServer merupakan library pada ESP yang mampu mengkomunikasikan antara client dengan server dalam waktu bersamaan. Pemasangan library ini dapat dilakukan secara manual pada software Arduino IDE atau juga dapat melalui library manager. Jika anda ingin memasang library secara manual, anda dapat mengikuti langkah-langkah berikut ini :

- Download library ESPAsyncWebServer :  
<https://github.com/me-no-dev/ESPAsyncWebServer>.
- Kemudian jika anda telah berhasil mendownload, ubah nama zip dari ~~ESPAsyncWebServer-master~~ menjadi *ESPAsyncWebServer*.
- Lalu Klik "sketch" pada menu Arduino IDE.
- Klik "include library".
- Klik ikon "add .Zip Libraries.
- Setelah itu cari file .Zip yang sudah anda download tadi.

### 5.2.3. Library Sensor DHT

Library DHT merupakan library untuk mengakses sensor suhu dan kelembapan udara, baik untuk tipe: DHT11, DHT22, dan lain sebagainya. Pemasangan library ini dapat dilakukan secara manual pada software Arduino IDE atau juga dapat melalui library manager. Jika anda ingin memasang library secara manual, anda dapat mengikuti langkah-langkah berikut ini :

- Download library sensor DHT11 :  
<https://www.dropbox.com/s/l6a6g5czeekbzce/DHT-sensor-library-master.zip?dl=0>.
- Kemudian jika anda telah berhasil mendownload, ubah nama zip dari ~~DHT-sensor-library-master~~ menjadi *DHT-sensor-library*.
- Lalu Klik "sketch" pada menu Arduino IDE.
- Klik "include library".
- Klik ikon "add .Zip Libraries.
- Setelah itu cari file .Zip yang sudah anda download tadi.

### 5.3. Coding

```
// Kumpulan Library
#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <DHT.h>

// Nama Hotspot atau Access Point yang dipakai oleh
Wemos anda
const char* ssid = "INDOBOTACADEMY";
// Password Hotspot atau Access Point yang dipakai
oleh Wemos anda
const char* password = "belajariot";

// Buat objek AsyncWebServer pada port 80
AsyncWebServer server(80);
```



```

// Inisiasi pin DHT
#define DHTPIN D2
// Tentukan tipe DHT menjadi DHT11
#define DHTTYPE DHT11
// Buat objek dht
DHT dht(DHTPIN, DHTTYPE);

// Nilai default untuk suhu dan kelembapan udara
float t = 0.0;
float h = 0.0;

// Variabel untuk default waktu, yang akan menyimpan
data DHT terakhir kali diperbarui
unsigned long previousMillis = 0;

// Memperbarui pembacaan DHT setiap 10 detik
const long interval = 10000;

// Bagian ini yang akan ditampilkan pada halaman
website
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
integrity="sha384-
fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpx
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">°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>
<script>
// Update data suhu udara secara otomatis yang
ditampilkan pada web browser
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 ) ;

```

```
// Update data kelembapan udara secara otomatis yang
ditampilkan pada web browser
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";
```

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

```
// Method yang dijalankan sekali
void setup(){
    // Inisiasi baudrate dan status hardware
    Serial.begin(115200);
    dht.begin();

    // Menyambungkan ke WiFi
    WiFi.begin(ssid, password);
    Serial.println("Connecting to WiFi");
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println(".");
    }
}
```

```

}

// Cetak Local IP Address pada Serial Monitor
Serial.println(WiFi.localIP());

// Rute yang mengatur request pada web browser
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());
});

// Memulai server
server.begin();
}

// Method yang dijalankan berkali-kali
void loop() {
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
        // Menyimpan terakhir kali ketika ada pembaruan
        nilai dari sensor DHT
        previousMillis = currentMillis;

        // Pengecekan sensor
        if (isnan(newT)) {
            Serial.println("Failed to read from DHT
sensor!");
        }
        else {

```

```

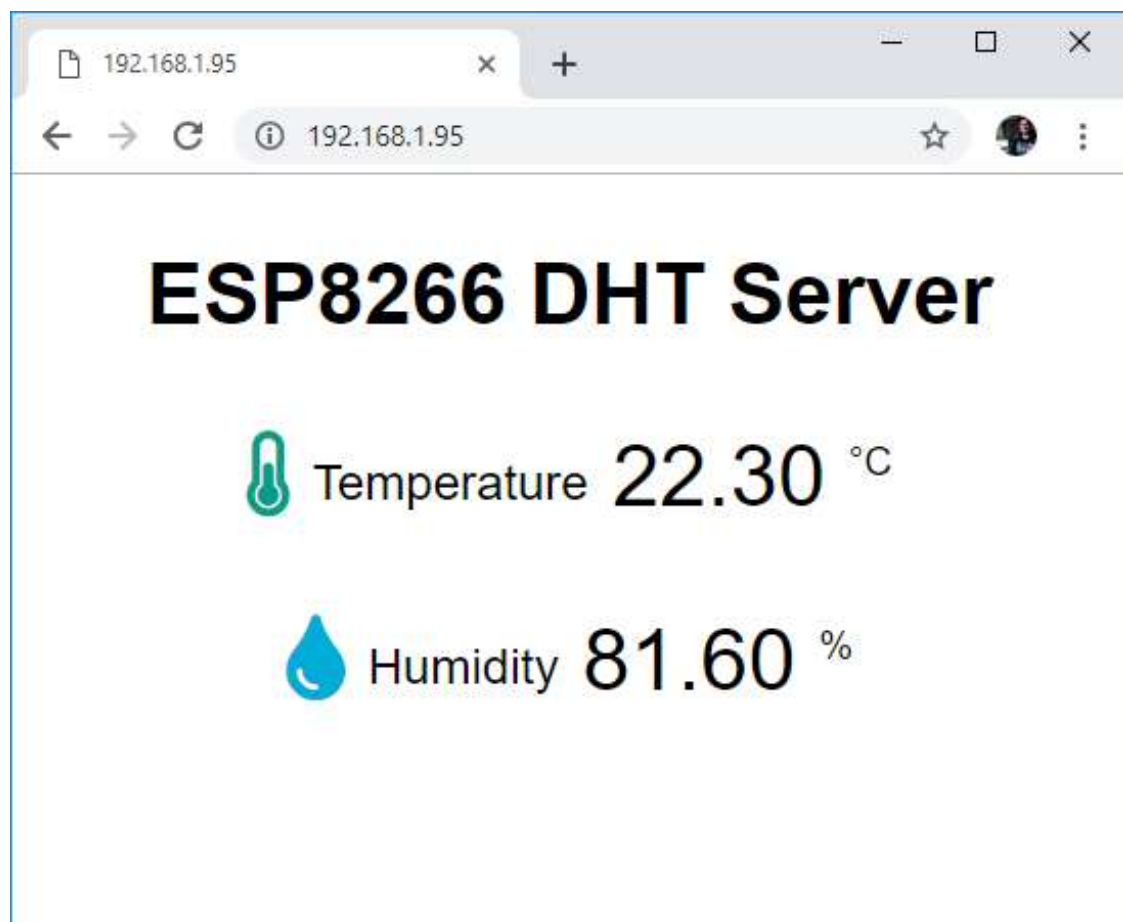
        t = newT;
        Serial.println(t);
    }
    if (isnan(newH)) {
        Serial.println("Failed to read from DHT
sensor!");
    }
    else {
        h = newH;
        Serial.println(h);
    }

    // Baca data suhu udara dalam satuan celsius
    (default)
    float newT = dht.readTemperature();
    // Baca data suhu udara dalam satuan fahrenheit
    //float newT = dht.readTemperature(true);
    // Baca data kelembapan udara
    float newH = dht.readHumidity();
}
}

```

## 5.4. Hasil

Setelah program berhasil diupload, maka silahkan cek IP Address di bagian Serial Monitor Arduino IDE. Lalu, bukalah browser terlebih dahulu untuk dapat menuliskan IP Addressnya. Pada web browser, akan ditampilkan sebuah halaman monitoring yang dapat anda lihat pada gambar di bawah ini.



## 6. Tugas dan Tantangan

Buatlah kontrol dan monitoring pada web browser melalui Arduino IDE.