

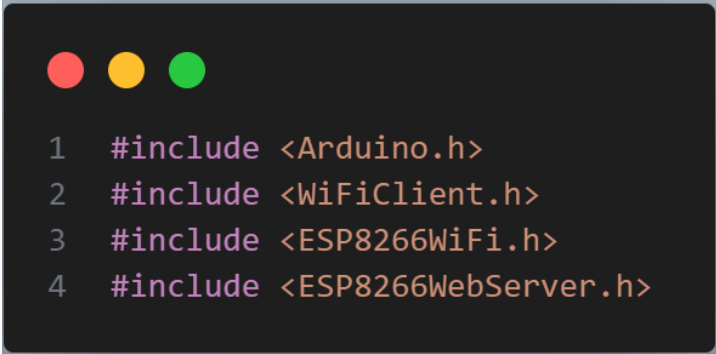
Nama : Destian Ardan Alfatanu

NIM : H1D022045

Shift : A-I-H

PENJELASAN SOURCE CODE LED IoT

1. Library yang Digunakan




```
1 #include <Arduino.h>
2 #include <WiFiClient.h>
3 #include <ESP8266WiFi.h>
4 #include <ESP8266WebServer.h>
```

Arduino.h: Library dasar Arduino.

WiFiClient.h & ESP8266WiFi.h: Untuk koneksi WiFi pada modul ESP8266.

ESP8266WebServer.h: Untuk membuat web server lokal pada ESP8266.

2. Inisialisasi WiFi dan Server



```
1 const char *ssid = "Fatan";
2 const char *password = "halohalo";
3
4 ESP8266WebServer server(80);
5 String webpage;
```

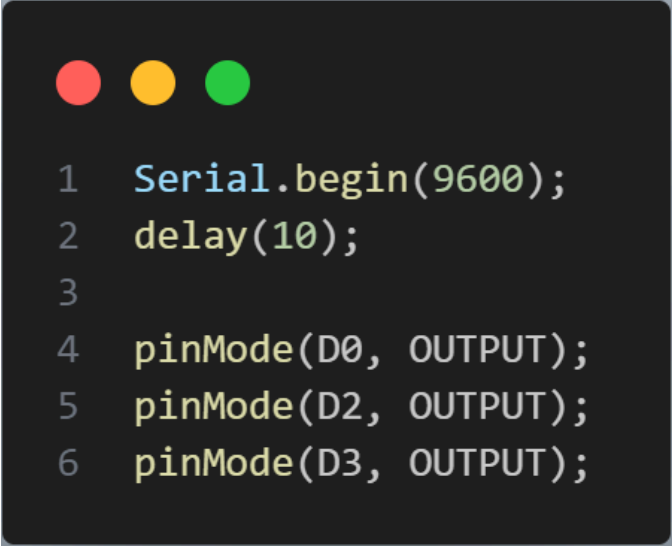
ssid dan password: Nama WiFi dan kata sandi untuk menghubungkan ESP ke jaringan.

server(80): Membuat server web yang berjalan di port 80 (port HTTP standar).

webpage: Variabel untuk menyimpan isi HTML halaman kontrol LED.

3. Fungsi Setup

a. Inisialisasi Serial dan Pin

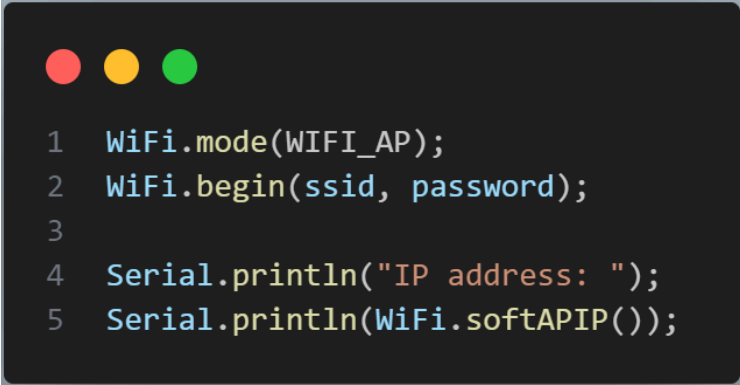


```
1 Serial.begin(9600);
2 delay(10);
3
4 pinMode(D0, OUTPUT);
5 pinMode(D2, OUTPUT);
6 pinMode(D3, OUTPUT);
```

Membuka komunikasi serial di kecepatan 9600 bps.

Mengatur pin D0, D2, dan D3 sebagai output (untuk mengontrol LED).

b. Membuat Access Point



```
1 WiFi.mode(WIFI_AP);
2 WiFi.begin(ssid, password);
3
4 Serial.println("IP address: ");
5 Serial.println(WiFi.softAPIP());
```

WiFi.mode(WIFI_AP); Digunakan untuk mengatur mode WiFi ESP8266 sebagai Access Point (AP), yaitu membuat ESP menjadi pemancar WiFi sendiri.

WiFi.begin(ssid, password); Digunakan untuk memulai koneksi WiFi menggunakan SSID dan password yang telah ditentukan.

Serial.println(WiFi.softAPIP()); Menampilkan alamat IP dari Access Point yang telah dibuat ke Serial Monitor, sehingga kamu bisa tahu alamat IP untuk mengakses web server.

c. Membuat Halaman Web

```
1 webpage = "<html><head><title>ESP8266 Control</title></head><body>";
2 webpage += "<h1>Kontrol LED Destian Ardan Alfatanu</h1>";
3 webpage += "<p>LED1: ";
4 webpage += "<a href='\"LED1ON\"'><button>ON</button></a><a href='\"LED1OFF\"'><button>OFF</button></a></p><br>";
5 webpage += "<p>LED2: ";
6 webpage += "<a href='\"LED2ON\"'><button>ON</button></a><a href='\"LED2OFF\"'><button>OFF</button></a></p><br>";
7 webpage += "<p>LED3: ";
8 webpage += "<a href='\"LED3ON\"'><button>ON</button></a><a href='\"LED3OFF\"'><button>OFF</button></a></p><br>";
9
10 // Membuat tampilan webpage
11 server.on("/", []()
12 { server.send(200, "text/html", webpage); });
```

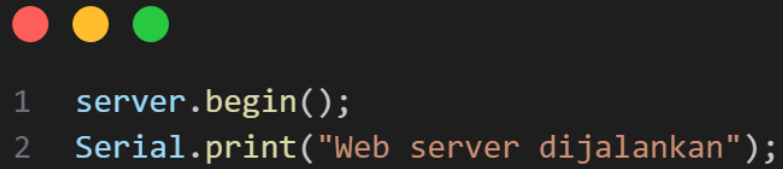
Membuat halaman HTML berisi tombol ON/OFF untuk 3 LED. Menggunakan hyperlink (****) dengan tombol HTML untuk mengakses URL yang akan mengaktifkan/mematikan LED. Ketika alamat utama (/) diakses, server akan menampilkan isi **webpage**.

d. Handler untuk Menyalakan LED

```
1 server.on("/LED1ON", []()
2 {
3   server.send(200, "text/html", webpage);
4   digitalWrite(D0, HIGH);
5   delay(100); });
6
7 server.on("/LED2ON", []()
8 {
9   server.send(200, "text/html", webpage);
10  digitalWrite(D2, HIGH);
11  delay(100); });
12
13 server.on("/LED3ON", []()
14 {
15   server.send(200, "text/html", webpage);
16   digitalWrite(D3, HIGH);
17   delay(100); });
18
19 server.on("/LED1OFF", []()
20 {
21   server.send(200, "text/html", webpage);
22   digitalWrite(D0, LOW);
23   delay(100); });
24
25 server.on("/LED2OFF", []()
26 {
27   server.send(200, "text/html", webpage);
28   digitalWrite(D2, LOW);
29   delay(100); });
30
31 server.on("/LED3OFF", []()
32 {
33   server.send(200, "text/html", webpage);
34   digitalWrite(D3, LOW);
35   delay(100); });
```

Jika URL **/LED1ON** dibuka, LED di pin D0 akan dinyalakan (**HIGH**). Halaman HTML dikembalikan sebagai respons. Begitu juga dengan **/LED2ON**, **/LED3ON**, **/LED1OFF**, **/LED2OFF**, **/LED3OFF** yang menyalakan atau mematikan masing-masing LED di D2 dan D3.

e. Menjalankan Server

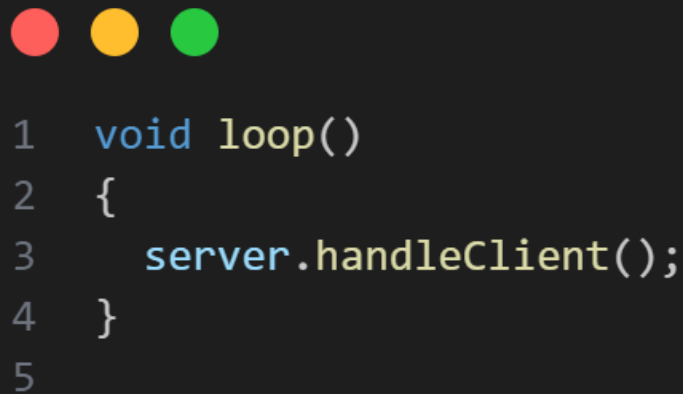


```
1  server.begin();
2  Serial.print("Web server dijalankan");
```

Memulai web server.

Menampilkan pesan bahwa server sudah aktif.

4. Fungsi Loop



```
1  void loop()
2  {
3      server.handleClient();
4  }
5
```

Fungsi ini terus dipanggil dalam loop utama.

server.handleClient() akan mengecek apakah ada request dari browser, dan jika ada akan dijalankan handler yang sesuai.

FOTO PRODUK

