Workshop Aplikasi Mikroprosesor & Antarmuka

PROGRAM STUDI
TEKNIK TELKOMUNIKASI

Akuwan Saleh, MT

PENILAIAN

- Laporan + Tugas + Presentasi = 60%
  - eval-1(Lap.1-5) = 20%
  - eval-2(Lap.6-10) = 20%
  - eval-3(Lap.11-15 & (PPT+ presentasi)) = 20%
- TPS = 40%
  - eval-4 (TPS = Tugas Proyek Semester)

REFERENSI

- Rui Santos & Sara Santos, “ESP32 Web Server With Arduino IDE.pdf: Step By Step Project Guide”, https://randomnerdtutorials.com/esp32-web-server-arduino-ide/, juni 2020.
- Julien Bayle, “C Programming for Arduino”, Packt Publishing Ltd, Birmingham, May 2013
- Fabian Winkler, “Arduino/Processing Communication Workshop”, Fall, 2013.
- Jack Purdum, “Beginning C for Arduino, Learn C Programming for the Arduino and Compatible Microcontrollers”, Apress, 2012.
- John-David Warren, Josh Adams, and Harald Molle, “Arduino Robotics”, Springer, New York, 2011.
- Casey Reas and Ben Fry, “Getting Started with Processing”, O’Reilly Media, Inc., June 2010.
- Joshua Noble, “Programming Interactivity: A Designer’s Guide to Processing, Arduino, and openFrameworks”, O’Reilly Media, Inc., July 2009.
- Peter Hoddie, Lizzie Prader, “IoT Development for ESP32 and ESP8266 with JavaScript; A Practical Guide to XS and Moddable SDK”, Apress, Menlo Park, CA,USA, 2020.

MATERI

PENDAHULUAN

1. KOMUNIKASI MIKROKONTROLER DENGAN SOFTWARE PROCESSING
2 . ANALOG INPUT DAN AUDIO PROCESSING
3. KONTROL MULTI LED MENGGUNAKAN ARDUINO DAN PROCESSING
4. PENCAMPUR WARNA VIRTUAL MENGGUNAKAN ARDUINO DAN PROCESSING
5. MONITORING SUHU DENGAN ARDUINO DAN PROCESSING
6. MONITORING INTENSITAS CAHAYA DENGAN ARDUINO DAN PROCESSING
7. KONTROL MOTOR DC MENGGUNAKAN ARDUINO DAN PROCESSING
8. APLIKASI SENSOR ULTRASONIC MENGGUNAKAN ARDUINO DAN PROCESSING

MATERI

9. KONTROL LAMPU AC 220 V BERBASIS ARDUINO DAN PROCESSING
10. MODUL WiFi ESP32 DENGAN ARDUINO IDE
11. KOMUNIKASI NIRKABEL MENGGUNAKAN MODUL RF 434 MHz DAN PROCESSING
12. ESP32 WEB SERVER UNTUK KONTROL LED DAN MENAMPILKAN GAMBAR
13. ANTARMUKA MODUL GPS DENGAN MIKROKONTROLER DAN PROCESSING
14. ESP32 WEB SERVER UNTUK PENGUKURAN SUHU DAN KELEMBABAN
15. KOMUNIKASI DATA BERBASIS BLUETOOTH DAN HP

12. ESP32 WEB SERVER UNTUK KONTROL LED DAN MENAMPILKAN GAMBAR

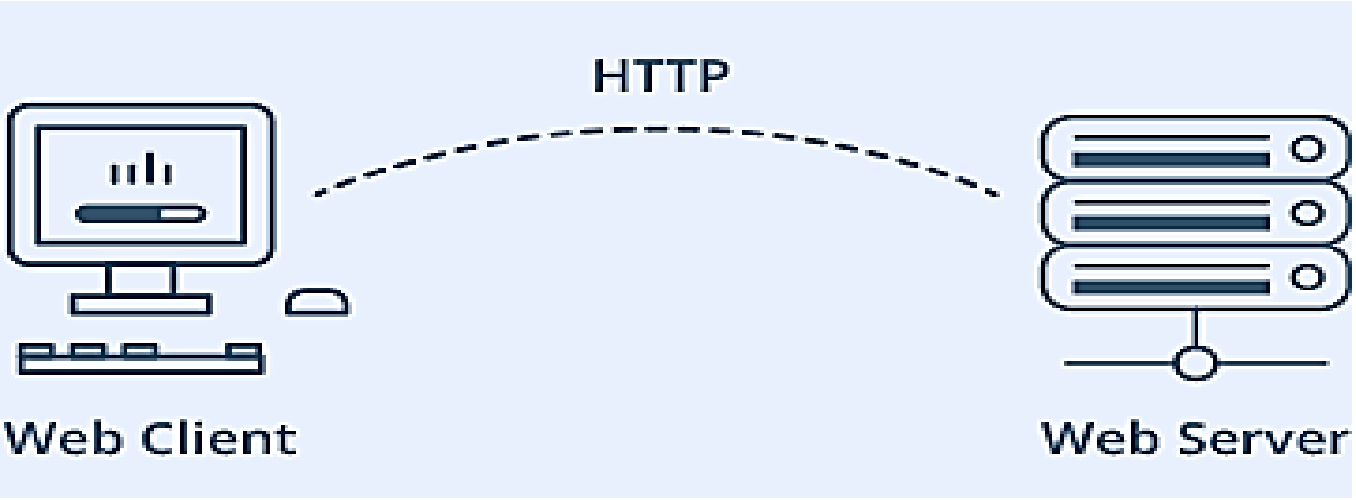
TUJUAN

- Membangun web server mandiri.
- Membuat program di IDE Arduino untuk membuat halaman web yang interaktif.
- Mengendalikan nyala LED berbasis web menggunakan modul WiFi ESP32

DASAR TEORI

- **Web server dan cara kerjanya**

- **Web Server** adalah tempat menyimpan, memproses dan mengirimkan halaman web ke klien Web.
- **Klien web** adalah web browser di laptop dan smartphone.
- **Komunikasi** antara klien dan server menggunakan protokol khusus yang disebut ***Hypertext Transfer Protocol*** (HTTP).



- Dalam protokol ini, klien memulai komunikasi dengan membuat permintaan untuk halaman web tertentu menggunakan HTTP
- Server merespons dengan konten halaman web tersebut atau pesan kesalahan jika tidak dapat melakukannya (seperti 404 Error).
- Halaman yang dikirimkan oleh server sebagian besar adalah dokumen HTML.

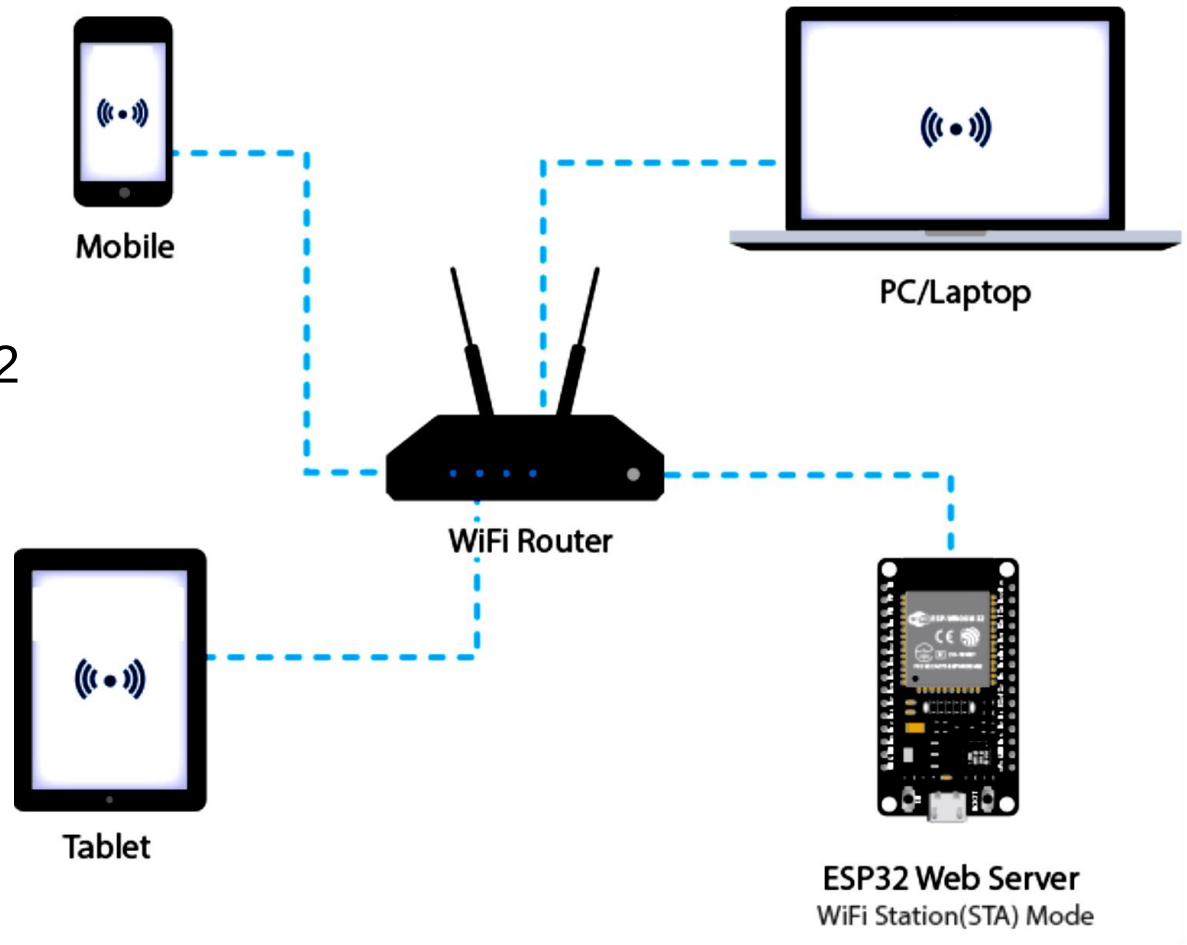
➢ Mode Pengoperasian ESP32

Salah satu fitur terbesar yang disediakan ESP32:

- tidak hanya dapat terhubung ke jaringan WiFi yang ada dan bertindak sebagai Web Server, tetapi juga dapat mengatur jaringannya sendiri,
- memungkinkan perangkat lain untuk terhubung langsung dan mengakses halaman web.
- ESP32 dapat beroperasi dalam tiga mode berbeda: **Mode Station**, **mode Soft Access Point**, dan **keduanya** pada saat bersamaan.

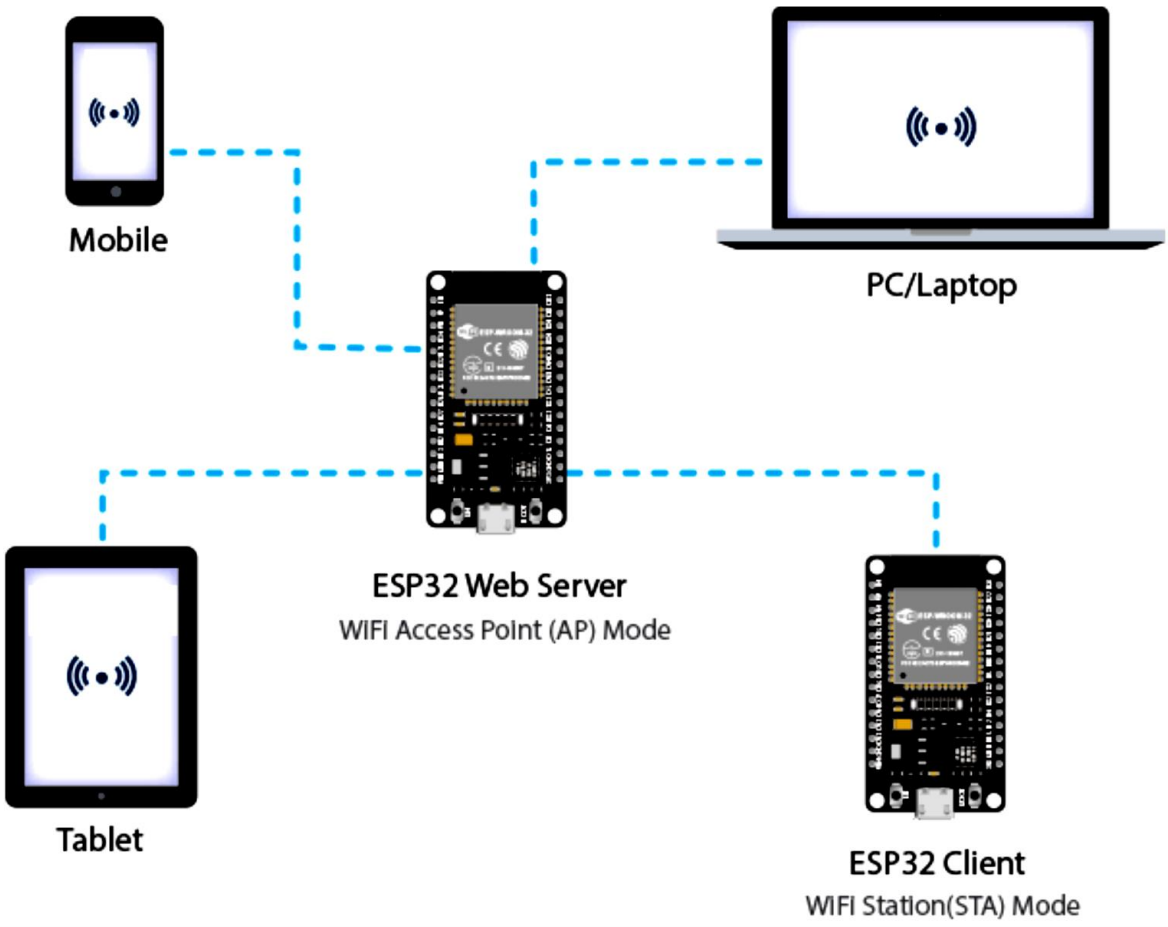
• Mode Station (STA)

- ✓ ESP32 yang menghubungkan ke jaringan WiFi (yang dibuat oleh router nirkabel) disebut Station (STA)
- ✓ Dalam mode STA ESP32 mendapatkan IP dari router nirkabel yang terhubung.
- ✓ Dengan alamat IP ini, dapat mengatur web server dan mengirimkan halaman web ke semua perangkat yang terhubung di bawah jaringan WiFi yang ada.



• Mode Soft Access Point (AP)

- ✓ ESP32 menciptakan jaringan WiFi sendiri dan bertindak sebagai hub (seperti router WiFi) untuk satu atau lebih station disebut Access Point (AP).
- ✓ Tidak seperti router WiFi, tidak memiliki antarmuka ke jaringan kabel.
- ✓ Jadi, mode operasi semacam ini disebut Soft Access Point (soft-AP).
- ✓ Jumlah maksimum station yang dapat terhubung ke station ini dibatasi hingga lima.



- ✓ Dalam **mode AP** ESP32 membuat jaringan WiFi baru dan menetapkan **SSID** (Nama jaringan) dan alamat IP.
- ✓ Dengan alamat IP ini, dapat mengirimkan halaman web ke semua perangkat yang terhubung di bawah jaringannya sendiri.

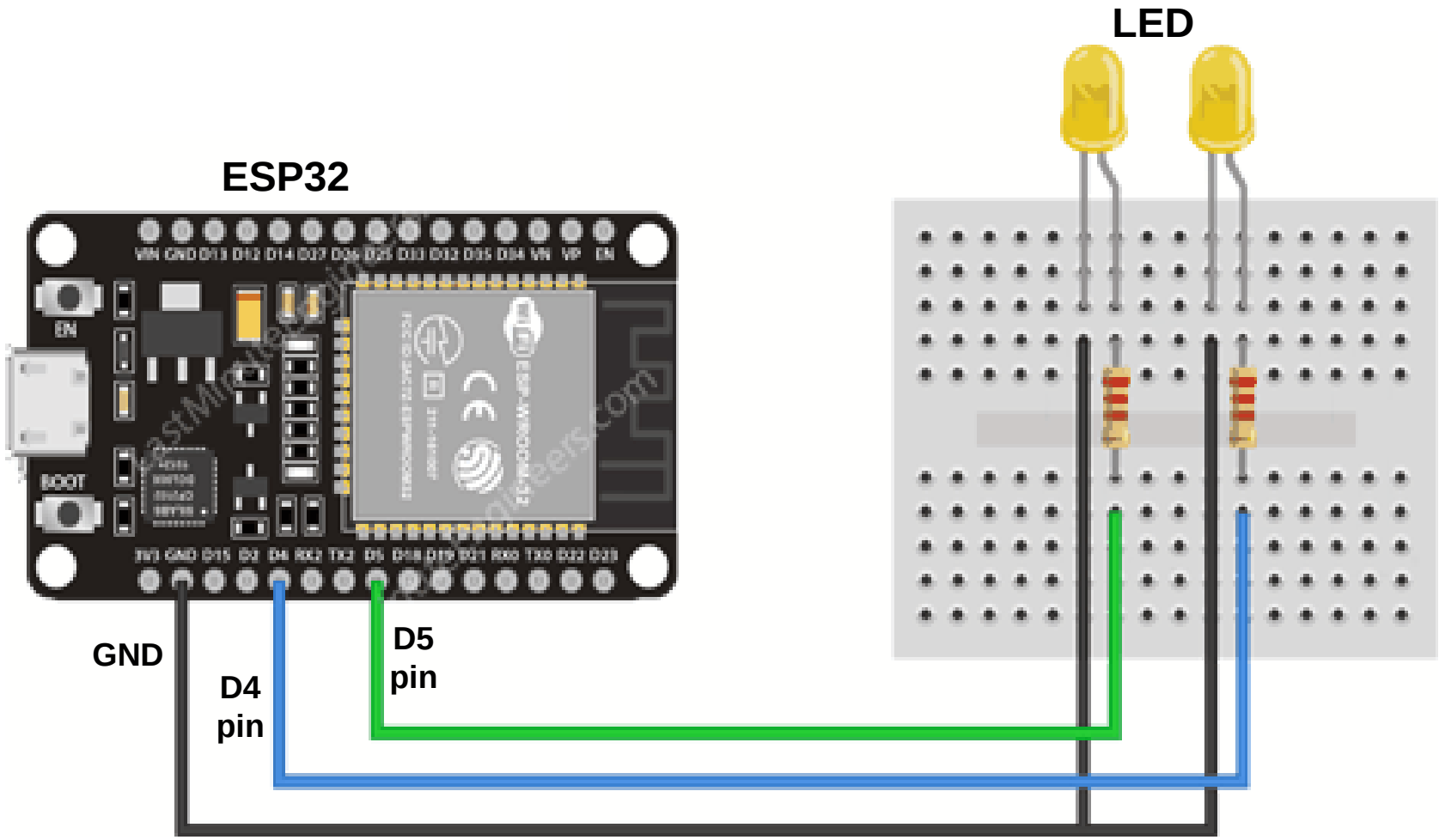
1. Mengontrol LED Dari ESP32 Web Server

- Mengendalikan sesuatu dengan mengakses URL tertentu. Misalnya, dengan cara memasukkan URL seperti http://192.168.1.1/ledon di browser.
- Browser kemudian mengirimkan permintaan HTTP ke ESP32 untuk menangani permintaan ini.
- Ketika ESP32 membaca permintaan ini, ia tahu bahwa pengguna ingin menyalakan LED. Jadi, ESP32 menyalakan LED dan mengirimkan halaman web dinamis ke browser yang menunjukkan status LED: ON.

Hardware :

- Breadboard(Optional)
- ESP32 Modul
- Arduino IDE
- 2 LED
- 2 resistor 220 Ω
- Kabel Micro USB
- Kabel Jumper

Rangkaian



PROGRAM:

Arduino IDE

```
#include <WiFi.h>

// Replace with your network credentials
const char* ssid     = "";
const char* password = "";
WiFiServer server(80); // Set web server port number to 80
String header; // Variable to store the HTTP request
String output4State = "off";
String output5State = "off";
const int output4 = 4;
const int output5 = 5;

void setup() {
  Serial.begin(115200);
  pinMode(output4, OUTPUT);
  pinMode(output5, OUTPUT);
  digitalWrite(output4, LOW);
  digitalWrite(output5, LOW);
```

PROGRAM: ***Lanjutan***

```
// Connect to Wi-Fi network with SSID and password
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Print local IP address and start web server
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.begin();
}
```

PROGRAM: *Lanjutan*

```
void loop(){
  WiFiClient client = server.available();   // Listen for incoming clients
  if (client) {                             // If a new client connects,
    Serial.println("New Client.");
    String currentLine = "";
   while (client.connected()) {             // loop while the client's connected
      if (client.available()) {             // if there's bytes to read from the client,
        char c = client.read();
        Serial.write(c);
        header += c;
        if (c == '\n') {
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Connection: close");
            client.println();
```

PROGRAM: *Lanjutan*

```
if (header.indexOf("GET /4/on") >= 0) {
  Serial.println("GPIO 4 on");
  output4State = "on";
  digitalWrite(output4, HIGH);
 } else if (header.indexOf("GET /4/off") >= 0) {
  Serial.println("GPIO 4 off");
  output4State = "off";
  digitalWrite(output4, LOW);
 } else if (header.indexOf("GET /5/on") >= 0) {
  Serial.println("GPIO 5 on");
  output5State = "on";
  digitalWrite(output5, HIGH);
 } else if (header.indexOf("GET /5/off") >= 0) {
  Serial.println("GPIO 5 off");
  output5State = "off";
  digitalWrite(output5, LOW);
 }
```

PROGRAM: *Lanjutan*

```
            // Display the HTML web page
            client.println("<!DOCTYPE html><html>");
            client.println("<head><meta
name=\"viewport\“content=\"width=device-width, initial-scale=1\">");
            client.println("<link rel=\"icon\" href=\"data:,\">");
            client.println("<style>html { font-family: Helvetica; display:
inline-block; margin: 0px auto; text-align: center;}");
            client.println(".button { background-color: #4CAF50; border:
none; color: white; padding: 16px 40px;");
            client.println("text-decoration: none; font-size: 30px; margin:
2px; cursor: pointer;}");
            client.println(".button2 {background-color:
#555555;}</style></head>");
            // Web Page Heading
            client.println("<body><h1>ESP32 Web Server</h1>");
            client.println("<p>GPIO 4 - State " + output4State + "</p>");
```

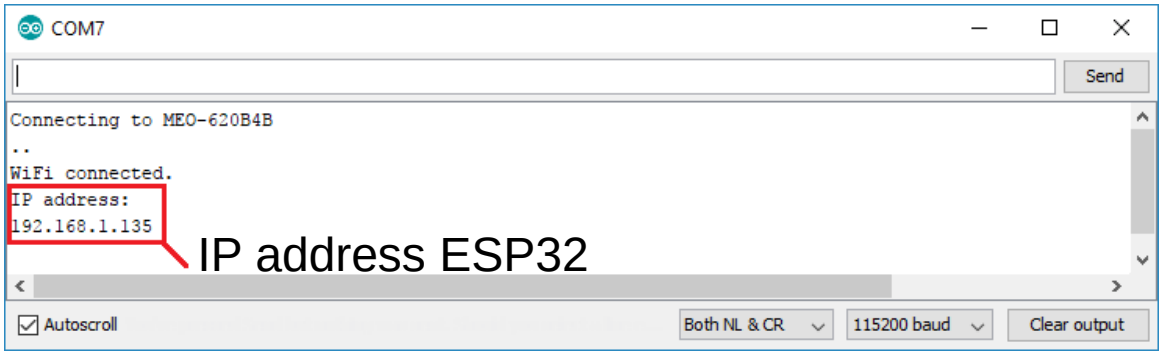
PROGRAM: *Lanjutan*

```
            if (output4State=="off") {
              client.println("<p><a href=\"/4/on\"><button
class=\"button\">ON</button></a></p>");
             } else {
              client.println("<p><a href=\"/4/off\"><button class=\"button
button2\">OFF</button></a></p>");
             }
            client.println("<p>GPIO 5 - State " + output5State + "</p>");
            if (output5State=="off") {
              client.println("<p><a href=\"/5/on\"><button
class=\"button\">ON</button></a></p>");
             } else {
              client.println("<p><a href=\"/5/off\"><button class=\"button
button2\">OFF</button></a></p>");
             }
             client.println("</body></html>");
             client.println();
```

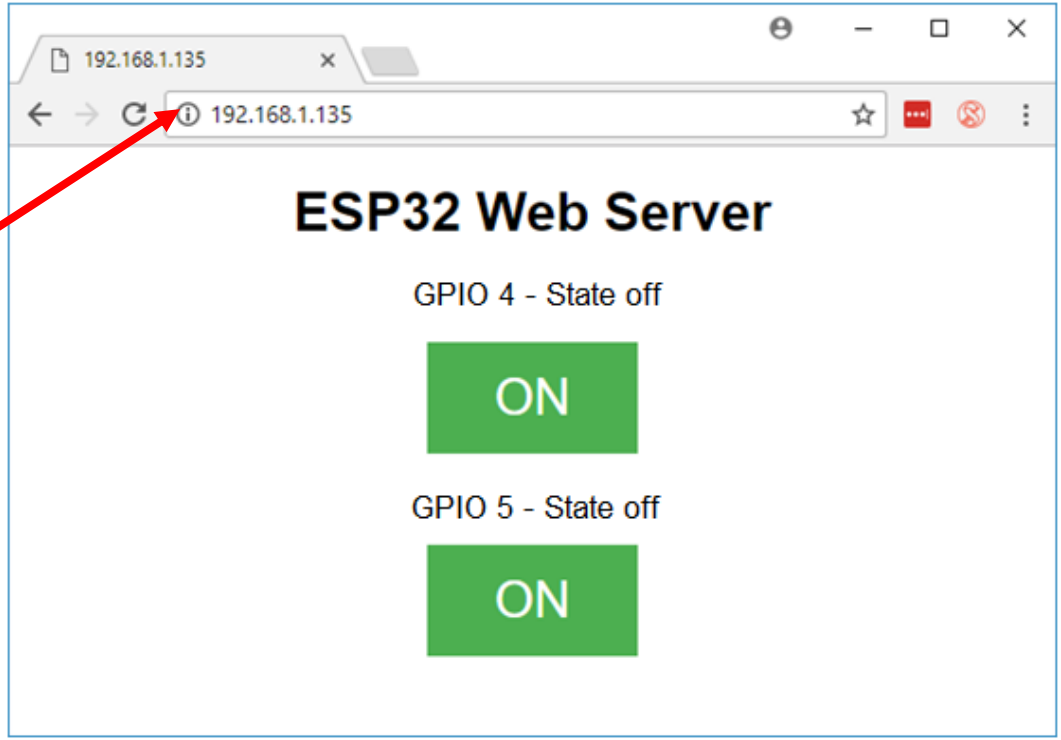
PROGRAM: *Lanjutan*

```
          break;
        } else {                    // if you got a newline, then clear currentLine
          currentLine = "";
        }
      } else if (c != '\r') {
        currentLine += c;           // add it to the end of the currentLine
      }
    }
  }
  header = "";                      // Clear the header variable
  client.stop();                    // Close the connection
  Serial.println("Client disconnected.");
  Serial.println("");
 }
}
```

- Setelah program diupload, buka Serial Monitor dengan baud rate = 115200.



- Tekan tobol button “EN”.
- Buka browser, paste IP address ESP32, dan lihat halaman web.



Hasil :

- Amati/foto dan catat hasil dari program ESP32 Web Server

Latihan :

1. Buatlah eksperimen meggunakan modul ESP32 untuk web server dengan SSID (nama perangkat internet) dan Password (kata sandi perangkat internet) tertentu.
2. Buatlah eksperimen meggunakan modul WiFi ESP32 untuk mengakses Web Server dalam mode AP dan mode STA

2. Gambar pada ESP32 Web Server

- Menyiapkan library ESPAsyncWebServer https://github.com/me-no-dev/ESPAsyncWebServer
- Library yang harus di-install adalah : ESPAsyncWebServer dan AsyncTCP
- Ada 3 cara yang harus dilakukan untuk menampilkan gambar pada ESP32 web server :
  1. Convert gambar menjadi base64 enkoder pada web https://www.base64-image.de/
  2. Copy kode encoder gambar yang telah didapatkan dan diletakkan di dalam **<img src =""\>** tag
  3. Masukkan **<img src = ""\>** di dalam kode program ESP32

Cara Menampilkan Gambar:

- Menampilkan gambar pada web server seperti cara-cara di bawah ini :
  1. Menggunakan Hyperlynk lokasi tempat menyimpan gambar
  2. Mengubah gambar tersebut kedalam format base64
  3. Memasukkan gambar kedalam SPI Flash System

PROGRAM: *ESP Async Web Server*

```
#ifdef ESP32
  #include <WiFi.h>
  #include <ESPAsyncWebServer.h>
#else
  #include <Arduino.h>
  #include <ESP8266WiFi.h>
  #include <Hash.h>
  #include <ESPAsyncTCP.h>
  #include <ESPAsyncWebServer.h>
#endif

// Replace with your network credentials
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

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

PROGRAM: *Lanjutan*

```
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
  <h2>ESP Image Web Server</h2>
//Gambar Logo PENS
  <img
src="
YAAADIGPXJAAAAGXRFWHRTb2Z……(copy file kode gambar.docs)">
</body>
</html>)rawliteral";

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

PROGRAM: *Lanjutan*

```
// Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
// Print ESP32 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);
  });
  // Start server
  server.begin();
}
 void loop(){
}
```

Hasil :

- Amati/foto dan catat hasil dari program menampilkan gambar pada ESP32 web server

Latihan :

1. Buatlah eksperimen menampilkan beberapa gambar pada ESP32 web server.