

**LAPORAN PRAKTIKUM ELEKTRONIKA  
MESIN LISTRIK DAN TEKNIK KENDALI**  
*“Firmware Home Automatic”*



Disusun Oleh:

Akbar Hendrawan Pratama (19/441190/SV/16542)

Dika Puspito (19/441196/SV/16548)

Syarif Muhammad Nur Cahya (19/441211/SV/16563)

**TEKNOLOGI REKAYASA MESIN**  
**DEPARTEMEN TEKNIK MESIN SEKOLAH VOKASI**  
**UNIVERSITAS GADJAH MADA**  
**YOGYAKARTA**

**2022**

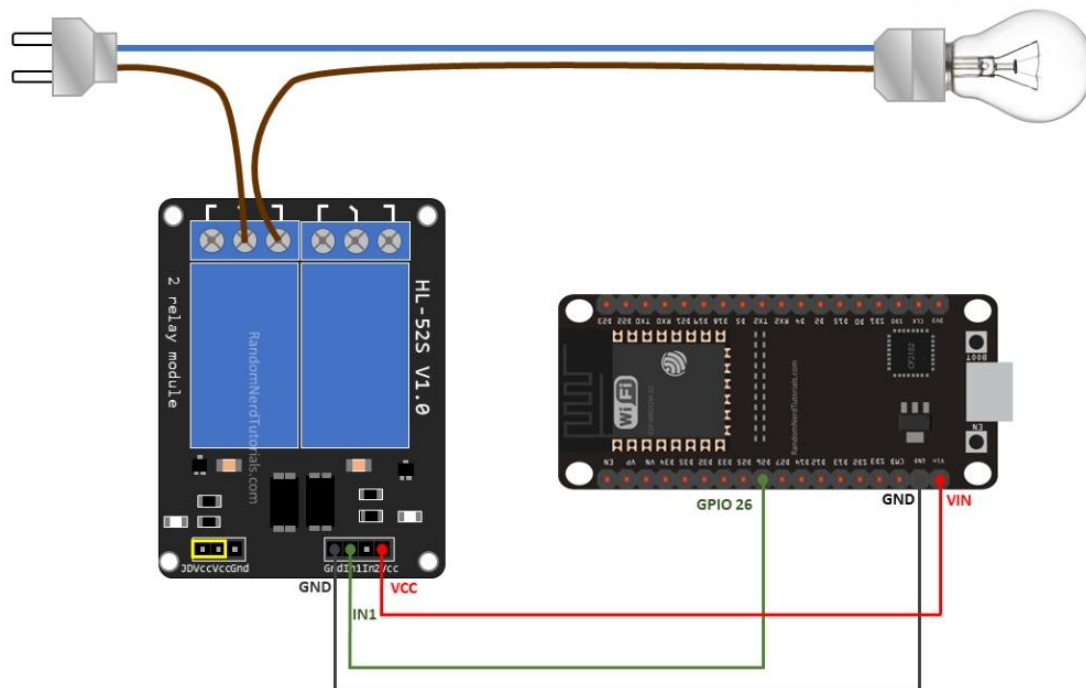
## BAB 1. DESKRIPSI KASUS

Perkembangan kemajuan teknologi dalam berbagai bidang sangat pesat terjadi di negeri nusantara sehingga membawa dampak positif bagi hampir seluruh lini kehidupan, terutama perkembangan teknologi yang setiap tahunnya semakin pesat sehingga menciptakan suatu inovasi baru dan berguna bagi khalayak umum terutama pada pengembangan inovasi teknologi sistem suatu barang elektronik.

Pada sistem kontrol jarak jauh turut memudahkan dalam hal pekerjaan rumah, seperti salah satu contohnya adalah pengontrolan lampu rumah pada saat masih menggunakan saklar dengan manual yang membuat jarak antara saklar dengan lampu berjauhan sehingga memerlukan waktu lebih lama dan tidak dapat menghemat tenaga, terlebih membuat suatu pekerjaan menjadi lebih lama.

Sehingga firmware home automatic, seperti smart home, microcontroller yang terkoneksi terhadap website nantinya dapat dikendalikan atau mengoperasikan benda dari jarak jauh. Sehingga nantinya pengguna akan mengirimkan input berupa sinyal (pulsa) dilanjutkan dengan sinyal yang ditangkap oleh website serta disalurkan kepada mikrokontroler dan akan dilanjutkan berupa output.

## BAB 2. FLOW RANGKAIAN



2.2 Relay yang digunakan merupakan single relay berjumlah 2 buah:

- a. Relay 1 dibaca pada pin 26.
- b. Relay 2 dibaca pada pin 27.

2.3 Pin Vcc disambungkan pada pin Vin.

2.4 Pin Gnd disambungkan pada pin Gnd.

2.5 Rangkaian ini didasarkan pada pembuatan code berupa coding yang nantinya bertujuan untuk disambungkan melalui web server, sehingga diharapkan nantinya sistem kendali otomatis dapat diterapkan pada kasus firmware home automatic seperti yang ditugaskan. Sistem kendali otomatis jika sudah disambungkan melalui web server dapat dengan mudah diakses melalui IP Address yang sudah ditentukan pada jaringan yang telah terkoneksi dengan internet, dengan demikian pada web yang diakses akan muncul sebuah interface berupa tombol guna mengatur nyala dan matinya lampu.

Berikut merupakan hasil pemrograman guna mendukung flow rangkaian:

```
/******
```

```
Rui Santos
```

```
Complete project details at https://randomnerdtutorials.com
```

```
*****/
```

```
// Load Wi-Fi library
```

```
#include <WiFi.h>
```

```
// Replace with your network credentials
```

```
const char* ssid = "belalangsembah";
```

```
const char* password = "ahayhayam";
```

```
// Set web server port number to 80
```

```
WiFiServer server(80);
```

```
// Variable to store the HTTP request
```

```
String header;
```

```
// Auxiliar variables to store the current output state
```

```
String output26State = "off";
```

```
String output27State = "off";
```

```
// Assign output variables to GPIO pins
```

```
const int output26 = 26;
```

```
const int output27 = 27;
```

```

// Current time
unsigned long currentTime = millis();
// Previous time
unsigned long previousTime = 0;
// Define timeout time in milliseconds (example: 2000ms = 2s)
const long timeoutTime = 2000;

void setup() {
  Serial.begin(115200);
  // Initialize the output variables as outputs
  pinMode(output26, OUTPUT);
  pinMode(output27, OUTPUT);
  // Set outputs to LOW
  digitalWrite(output26, LOW);
  digitalWrite(output27, LOW);

  // 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();
}

void loop(){
  WiFiClient client = server.available(); // Listen for incoming clients

  if (client) { // If a new client connects,
    currentTime = millis();
    previousTime = currentTime;
    Serial.println("New Client."); // print a message out in the serial port
  }
}

```

```

String currentLine = "";          // make a String to hold incoming data from the client
while (client.connected() && currentTime - previousTime <= timeoutTime) { // loop while
the client's connected
    currentTime = millis();
    if (client.available()) {      // if there's bytes to read from the client,
        char c = client.read();    // read a byte, then
        Serial.write(c);           // print it out the serial monitor
        header += c;
        if (c == '\n') {          // if the byte is a newline character
            // if the current line is blank, you got two newline characters in a row.
            // that's the end of the client HTTP request, so send a response:
            if (currentLine.length() == 0) {
                // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
                // and a content-type so the client knows what's coming, then a blank line:
                client.println("HTTP/1.1 200 OK");
                client.println("Content-type:text/html");
                client.println("Connection: close");
                client.println();

                // turns the GPIOs on and off
                if (header.indexOf("GET /26/on") >= 0) {
                    Serial.println("GPIO 26 on");
                    output26State = "on";
                    digitalWrite(output26, HIGH);
                } else if (header.indexOf("GET /26/off") >= 0) {
                    Serial.println("GPIO 26 off");
                    output26State = "off";
                    digitalWrite(output26, LOW);
                } else if (header.indexOf("GET /27/on") >= 0) {
                    Serial.println("GPIO 27 on");
                    output27State = "on";
                    digitalWrite(output27, HIGH);
                } else if (header.indexOf("GET /27/off") >= 0) {
                    Serial.println("GPIO 27 off");
                    output27State = "off";
                    digitalWrite(output27, LOW);
                }
            }

            // 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:;\">");
        // CSS to style the on/off buttons
        // Feel free to change the background-color and font-size attributes to fit your preferences
        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>Hayoo mau nyalain lampu yang mana?</h1>");

// Display current state, and ON/OFF buttons for GPIO 26
client.println("<p>Lampu Kamar - State " + output26State + "</p>");
// If the output26State is off, it displays the ON button
if (output26State=="off") {
                                client.println("<p><a href=\"/26/on\"><button
class=\"button\">Duarr</button></a></p>");
        } else {
                                client.println("<p><a href=\"/26/off\"><button class=\"button
button2\">Mati</button></a></p>");
        }

// Display current state, and ON/OFF buttons for GPIO 27
client.println("<p>Lampu WC - State " + output27State + "</p>");
// If the output27State is off, it displays the ON button
if (output27State=="off") {
                                client.println("<p><a href=\"/27/on\"><button
class=\"button\">Duarr</button></a></p>");
        } else {
                                client.println("<p><a href=\"/27/off\"><button class=\"button
button2\">Mati</button></a></p>");
        }
        client.println("</body></html>");

// The HTTP response ends with another blank line
client.println();

```

```

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

```

### **BAB 3. SISTEM KERJA PROGRAM**

1. Membuat kode koneksi pada jaringan Wi-Fi yang ingin dipakai.

```

// Load Wi-Fi library
#include <WiFi.h>

// Replace with your network credentials
const char* ssid = "belalangsembah";
const char* password = "ahayhayam";

// Set web server port number to 80
WiFiServer server(80);

// Variable to store the HTTP request
String header;

```

2. Menentukan Pin Out yang akan digunakan untuk membaca input output dari mikrokontroler pada relay.

```
// Auxiliar variables to store the current output state
String output26State = "off";
String output27State = "off";

// Assign output variables to GPIO pins
const int output26 = 26;
const int output27 = 27;
```

3. Mengatur timer.

```
// Current time
unsigned long currentTime = millis();
// Previous time
unsigned long previousTime = 0;
// Define timeout time in milliseconds (example: 2000ms = 2s)
const long timeoutTime = 2000;
```

4. Set Output keadaan awal dalam keadaan OFF atau mati.

```
void setup() {
  Serial.begin(115200);
  // Initialize the output variables as outputs
  pinMode(output26, OUTPUT);
  pinMode(output27, OUTPUT);
  // Set outputs to LOW
  digitalWrite(output26, LOW);
  digitalWrite(output27, LOW);
}
```

5. Connecting Wi-Fi dan IP Address pada code.

```
// 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();
}

```

6. Setup Wi-Fi Client dari code ke server agar terkoneksi satu sama lain, serta setup looping.

```

void loop(){
  WiFiClient client = server.available(); // Listen for incoming clients

  if (client) { // If a new client connects,
    currentTime = millis();
    previousTime = currentTime;
    Serial.println("New Client."); // print a message out in the serial port
    String currentLine = ""; // make a String to hold incoming data from the
    client
    while (client.connected() && currentTime - previousTime <= timeoutTime) { // loop
    while the client's connected
      currentTime = millis();
      if (client.available()) { // if there's bytes to read from the client,
        char c = client.read(); // read a byte, then
        Serial.write(c); // print it out the serial monitor
        header += c;
        if (c == '\n') { // if the byte is a newline character
          // if the current line is blank, you got two newline characters in a row.
          // that's the end of the client HTTP request, so send a response:
          if (currentLine.length() == 0) {
            // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
            // and a content-type so the client knows what's coming, then a blank line:
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Connection: close");
            client.println();

```

7. Command untuk memberi sinyal/pulsa pada output.

```
// turns the GPIOs on and off
if (header.indexOf("GET /26/on") >= 0) {
  Serial.println("GPIO 26 on");
  output26State = "on";
  digitalWrite(output26, HIGH);
} else if (header.indexOf("GET /26/off") >= 0) {
  Serial.println("GPIO 26 off");
  output26State = "off";
  digitalWrite(output26, LOW);
} else if (header.indexOf("GET /27/on") >= 0) {
  Serial.println("GPIO 27 on");
  output27State = "on";
  digitalWrite(output27, HIGH);
} else if (header.indexOf("GET /27/off") >= 0) {
  Serial.println("GPIO 27 off");
  output27State = "off";
  digitalWrite(output27, LOW);
}
```

8. Mengatur interface website beserta button command pada output.

```
// 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:;\">");
// CSS to style the on/off buttons
// Feel free to change the background-color and font-size attributes to fit your
preferences
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>Hayoo mau nyalain lampu yang mana?</h1>");

// Display current state, and ON/OFF buttons for GPIO 26
client.println("<p>Lampu Kamar - State " + output26State + "</p>");
// If the output26State is off, it displays the ON button
if (output26State=="off") {
    client.println("<p><a href=\"/26/on\"><button
class=\"button\">Duarr</button></a></p>");
} else {
    client.println("<p><a href=\"/26/off\"><button class=\"button
button2\">Mati</button></a></p>");
}

// Display current state, and ON/OFF buttons for GPIO 27
client.println("<p>Lampu WC - State " + output27State + "</p>");
// If the output27State is off, it displays the ON button
if (output27State=="off") {
    client.println("<p><a href=\"/27/on\"><button
class=\"button\">Duarr</button></a></p>");
} else {
    client.println("<p><a href=\"/27/off\"><button class=\"button
button2\">Mati</button></a></p>");
}
client.println("</body></html>");

```

9. Set respon HTTP pada baris kosong beserta command memutus koneksi jaringan pada website.

```

// The HTTP response ends with another blank line
client.println();
// Break out of the while loop
break;
} else { // if you got a newline, then clear currentLine
    currentLine = "";
}
} else if (c != '\r') { // if you got anything else but a carriage return character,
    currentLine += c; // add it to the end of the currentLine
}
}
}

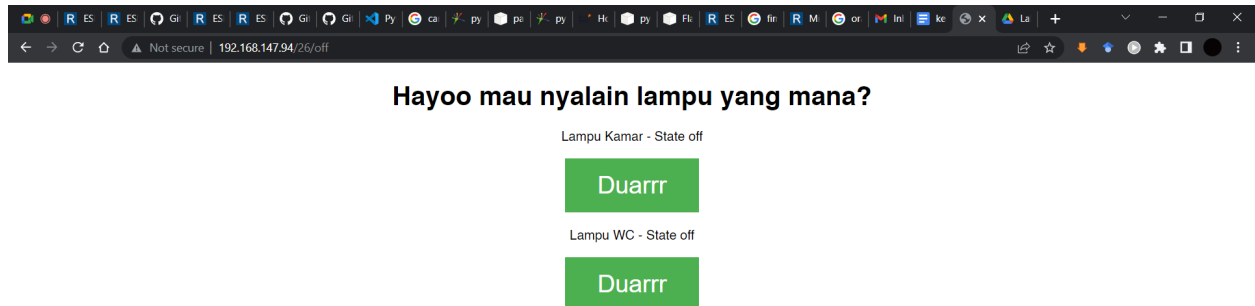
```

```

}
// Clear the header variable
header = "";
// Close the connection
client.stop();
Serial.println("Client disconnected.");
Serial.println("");
}
}

```

## BAB 4. KESIMPULAN



Setelah dilakukannya pembuatan program firmware home automatic dapat ditarik kesimpulan bahwa:

1. Fungsi kode yang terdapat pada ESP merupakan proses pengambilan data dari web server yang nantinya akan diteruskan terhadap pin 26 dan 27 yang terdapat pada ESP32.
2. Proses yang dijalankan berawal dari web server memberikan perintah kepada ESP untuk dapat melakukan perubahan pada aliran listrik yang akan disalurkan terhadap relay sehingga sistem pada kendali jarak jauh dapat dilakukan.