**LAPORAN**

**DOJIN SMART WORKSHOP**

**KONTROL DAN MONITORING PERANGKAT LISTRIK JARAK JAUH**

**MENGGUNAKAN WEB BERBASIS NODEMCU ESP8266**



**Guru Pengampu :**

**Maria Bestarina Laili, S.T.**

**Disusun Oleh :**

|  |  |
| --- | --- |
| **Muhammad Ridho Darmawan** | **1920** |
| **Panji Anom** | **1920** |

**XII TEI 2**

**TEKNIK ELEKTRONIKA INDUSTRI**

**2022**

DAFTAR ISI

[BAB I 1](#_Toc100044004)

[PENDAHULUAN 1](#_Toc100044005)

[1.1. Latar Belakang Masalah 1](#_Toc100044006)

[1.2. Identifikasi Masalah 2](#_Toc100044007)

[1.3. Maksud 2](#_Toc100044008)

[1.4. Tujuan 3](#_Toc100044009)

[BAB II 4](#_Toc100044010)

[LANDASAN TEORI 4](#_Toc100044011)

[2.1. World Wide Web 4](#_Toc100044013)

[2.2. Web Application ( Aplikasi Web) 4](#_Toc100044014)

[2.3. HTML, CSS, dan JavaScript 4](#_Toc100044015)

[2.4. PHP dan MySQL 4](#_Toc100044016)

[2.5. Hosting dan Domain 5](#_Toc100044017)

[2.6. HTTP 5](#_Toc100044018)

[2.7. ESP 8266 5](#_Toc100044019)

[2.8. Komponen Pendukung 8](#_Toc100044020)

[BAB III 11](#_Toc100044021)

[KONSEP DAN RANCANGAN 11](#_Toc100044022)

[3.1. Identifikasi Kebutuhan 11](#_Toc100044024)

[3.2. Analisis Kebutuhan 12](#_Toc100044025)

[3.3. Blok Diagram Sistem 13](#_Toc100044026)

[3.4. Langkah Pembuatan Web dan Alat 14](#_Toc100044027)

[BAB IV 40](#_Toc100044028)

[HASIL PEMBUATAN 40](#_Toc100044029)

# BAB I

# PENDAHULUAN

## Latar Belakang Masalah

Dilansir dari *How Stuff Works* peralatan elektronik yang masih menyedot listrik walaupun kita telah mematikannya adalah peralatan elektronik dengan remote control seperti Televisi dan AC, selain itu masih banyak lagi peralatan elektronik yang sangat boros walau sudah dimatikan yaitu alat dengan adapter seperti Komputer dan Printer, Charger semua jenis gadget, dan alat yang terus-terusan menyala seperti kulkas. Namun studi dari Lawrence Barkeley National Laboratory menunjukkan bahwa peralatan elektronik yang masih menyedot listrik walaupun kita telah mematikannya bisa menghabiskan 5 hingga 25 persen dari total pengeluaran listriknya.

Permasalahan lainnya, ada hal lain yang juga dapat menimbulkan bahaya pada keadaan rumah, misalkan lupa mematikan peralatan listrik di rumah yang dapat terjadinya konsleting listrik dan dapat mengakibatkan kebakaran, maka dibutuhkannya sistem yang mampu mengendalikan peralatan listrik tersebut dari jarak jauh.

Mematikan dan menyalakan peralatan listrik yang dilakukan secara manual dengan menggunakan saklar atau stop kontak yang selalu terhubung dengan sumber listrik. Sehingga peralatan listrik penggunanya tidak dapat dimonitoring secara langsung dan dikendalikan dari jarak jauh karena terkadang ada peralatan listrik yang sudah tidak dipakai tidak dicabut dari stop kontak karena lupa atau peralatan listrik yang memang selalu terhubung pada sumber listrik tidak dimatikan saklarnya ketika tidak digunakan lagi. Semua hal ini berujung pada pemborosan listrik yang dapat terjadi secara tidak sadar. Oleh karena itu dibutuhkan suatu sistem kontrol yang dapat membantu dalam memonitoring peralatan listrik dan mengendalikannya agar pemborosan listrik dapat diminimalisir sesuai dengan kebutuhan dan kondisi yang dapat diatur.

Smart Home atau rumah pintar merupakan sistem yang telah diprogram dan dapat bekerja dengan bantuan komputer untuk mengintegrasikan dan mengendalikan sebuah perangkat atau peralatan listrik rumah secara otomatis dan efisien. Tujuan dari diciptakannya teknologi ini yaitu untuk mempermudah penghematan daya energi, meningkatkan keamanan, meningkatkan kenyamanan, dan lain sebagainya.

## Identifikasi Masalah

Berdasarkan latar belakang masalah yang di deskripsikan diatas maka di indetifikasi sebuah masalah, yaitu :

1. Terbatasnya jarak untuk mengendalikan pemakaian peralatan listrik dari jarak jauh.
2. Pemborosan energi listrik yang sering terjadi pada peralatan listrik yang dipakai sehari-hari.
3. Penggunaan listrik yang tidak dapat di monitoring.
4. Kurangnya keamanan peralatan listrik ketika ditinggal berpergian dalam jangka waktu yang tidak tertentu

## Maksud

Berdasarkan permasalahan yang ada, maka maksud dari tugas proyek ini adalah untuk membangun sistem smart home sederhana yang akan diterapkan pada bengkel teknik elektronika industri dengan menggunakan modul NodeMCU ESP8266 dan Web App atau aplikasi berbasis web serta memberikan kenyamanan dan kemudahan bagi penggunanya.

## Tujuan

Sedangkan tujuan yang akan dicapai dalam tugas proyek ini adalah :

1. Dapat memantau penggunaan peralatan listrik yang menyala untuk mencegah pemborosan listrik maupun konsleting listrik.
2. Bebasnya pengguna dalam mengendalikan pemakaian peralatan listrik dari jarak yang diinginkan.
3. Sebagai sarana implementasi pengetahuan yang didapatkan saat menempuh pendidikan di bangku sekolah.
4. Memberikan bekal pengalaman untuk mengaplikasikan ilmu pengetahuan yang didapat selama menempuh pendidikan di sekolah.

# BAB II

# LANDASAN TEORI



## World Wide Web

World Wide Web (WWW), umumnya dikenal sebagai Web, pada awalnya adalah sistem manajemen dokumen hypertext yang diakses melalui internet. Sejak itu berkembang menjadi platform perangkat lunak yang dominan di dunia.

## Web Application ( Aplikasi Web)

Aplikasi web adalah perangkat lunak yang berjalan di server web, tidak seperti program perangkat lunak yang berbasis komputer yang dijalankan secara lokal di sistem operasi perangkat. Aplikasi web diakses oleh pengguna melalui browser web dengan koneksi jaringan aktif.

## HTML, CSS, dan JavaScript

**HyperText Markup Language** atau **HTML** adalah bahasa markup standar untuk dokumen yang dirancang untuk ditampilkan di browser web. Sedangkan **Cascading Style Sheets** atau **CSS** adalah bahasa style sheet yang digunakan untuk menggambarkan presentasi dokumen yang ditulis dalam bahasa markup seperti HTML. **JavaScript** adalah bahasa pemrograman yang merupakan satu teknologi inti dari World Wide Web, di samping HTML dan CSS

## PHP dan MySQL

**PHP** singkatan dari **Hypertext Preprocessor** adalah bahasa scripting open source yang digunakan sebagai bahasa sisi server untuk membuat halaman web dinamis.

Sedangkan **MySQL** adalah sebuah database management system (manajemen basis data) menggunakan perintah dasar **SQL** (Structured Query Language) yang berfungsi untuk memanipulasi dan mengambil data dalam database. MySQL merupakan sistem manajemen data yang paling populer digunakan dengan PHP.

## Hosting dan Domain

**Hosting** merupakan tempat untuk menyimpan data yang ada di website. Sebuah hosting di simpan di dalam sebuah server. Layanan web hosting adalah jenis layanan hosting internet yang menghosting situs web untuk klien, yaitu menawarkan fasilitas yang diperlukan bagi mereka untuk membuat dan memelihara situs web dan membuatnya dapat di akses di World Wide Web. **Domain** adalah nama yang mudah diingat yang terkait dengan alamat IP fisik di internet. Nama unik ini muncul setelah tanda @ di alamat email, dan setelah www. Di alamat web.

## HTTP

Hypertext Transfer Protocol (HTTP) dirancang untuk memungkinkan komunikasi antara Klien dan Server. HTTP Berfungsi protokol permintaan tanggapan antara Klien dan Server. Browser sebagai Klien, dan aplikasi yang menghosting situs web adalah Server. Klien mengirimkan permintaan ke HTTP ke server, kemudia server mengembalikan respons ke Klien.

## ESP 8266

**Modul ESP8266** merupakan platform yang sangat murah tetapi benar-benar efektif untuk digunakan berkomunikasi atau kontrol melalui internet baik digunakan secara standalone (berdiri sendiri) maupun dengan menggunakan mikrokontroler tambahan, dalam hal ini Arduino sebagai pengendalinya.

**NodeMCU ESP826** merupakan modul turunan pengembangan dari modul platform IoT (Internet of Things) keluarga ESP8266 tipe ESP-12. Secara fungsi modul ini hampir menyerupai dengan platform modul arduino, tetapi yang membedakan yaitu dikhususkan untuk “Connected to Internet“.

**NodeMCU 1.0**

* Versi ini merupakan pengembangan dari versi 0.9. Dan pada versi 1.0 ini ESP8266 yang digunakan yaitu tipe ESP-12E yang dianggap lebih stabil dari ESP-12.
* Selain itu ukuran board modulnya diperkecil sehingga compatible digunakan membuat prototipe projek di breadboard.
* Serta terdapat pin yang dikhusukan untuk komunikasi SPI (Serial Peripheral Interface) dan PWM (Pulse Width Modulation) yang tidak tersedia di versi 0.9.

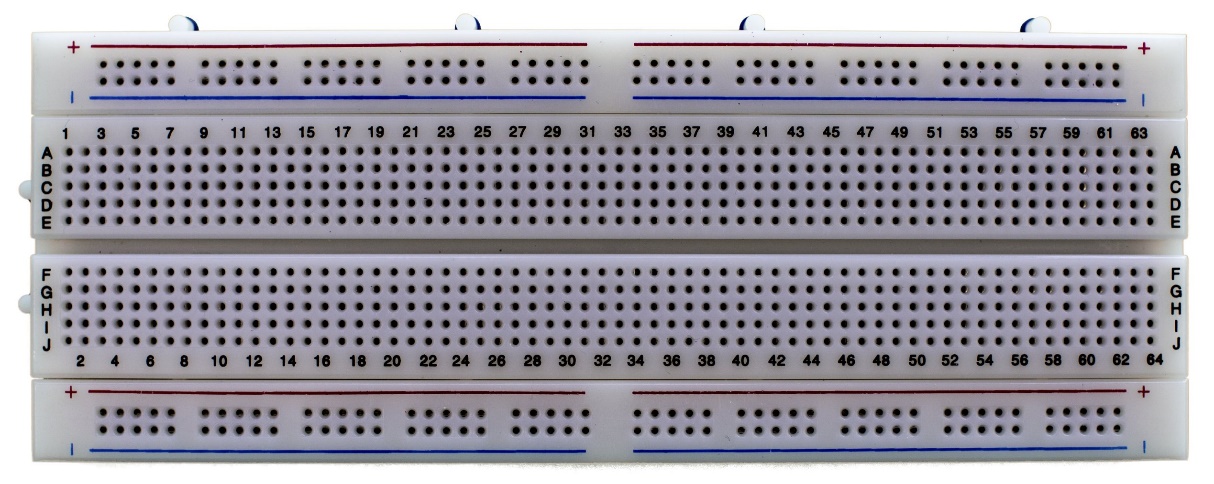


Spesifikasi dari NodeMCU diatas :

* Tegangan antarmuka komunikasi: 3.3V.
* Jenis antena: Tersedia antena PCB internal.
* Standar nirkabel 802.11 b / g / n
* WiFi di 2.4GHz, mendukung mode keamanan WPA / WPA2
* Mendukung tiga mode operasi STA / AP / STA + AP
* Tumpukan protokol TCP / IP bawaan untuk mendukung beberapa koneksi Klien TCP (5 MAX)
* D0 ~ D8, SD1 ~ SD3: digunakan sebagai GPIO, PWM, IIC, dll., Kemampuan driver port 15mA
* AD0: 1 saluran ADC
* Input daya: 4.5V ~ 9V (10VMAX), bertenaga USB
* Saat ini: transmisi kontinu: 70mA (200mA MAX), Siaga: 200uA
* Kecepatan transfer: 110-460800bps
* Mendukung antarmuka komunikasi data UART / GPIO
* Pembaruan firmware jarak jauh (OTA)
* Mendukung Smart Link Smart Networking
* Suhu kerja: -40 Deg ~ + 125 Deg
* Tipe Drive: Driver H-bridge ganda berdaya tinggi
* ESP8266 memiliki IO Pin
* Tidak perlu mengunduh pengaturan ulang
* Seperangkat alat yang bagus untuk mengembangkan ESP8266
* Ukuran flash: 4MByte

## Komponen Pendukung

1. Breadboard



Breadboard merupakan sebuah board atau papan yang berfungsi untuk merancang sebuah rangkaian elektronik sederhana. Breadboard tersebut nantinya akan dilakukan prototipe atau uji coba tanpa harus melakukan solder. Salah satu keuntungan menggunakan breadboard adalah komponen-komponen yang dirakit tersebut tidak akan mengalami kerusakan. Komponen tersebut juga masih bisa dirangkai kembali untuk membentuk rangkaian yang lainnya.

1. Kabel Jumper



Kabel jumper adalah kabel elektrik yang memiliki pin konektor di setiap ujungnya dan memungkinkanmu untuk menghubungkan dua komponen yang melibatkan Arduino tanpa memerlukan solder.

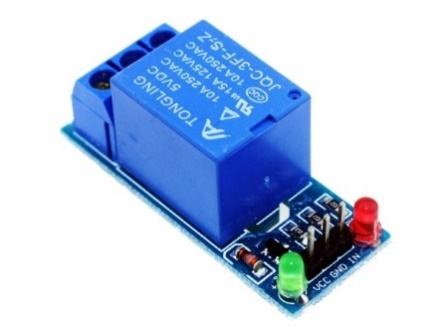
1. Kabel Serabut Merah dan Hitam



Warna standar kabel untuk listrik :

* Merah: Positif
* Hitam: Negatif

1. Relay



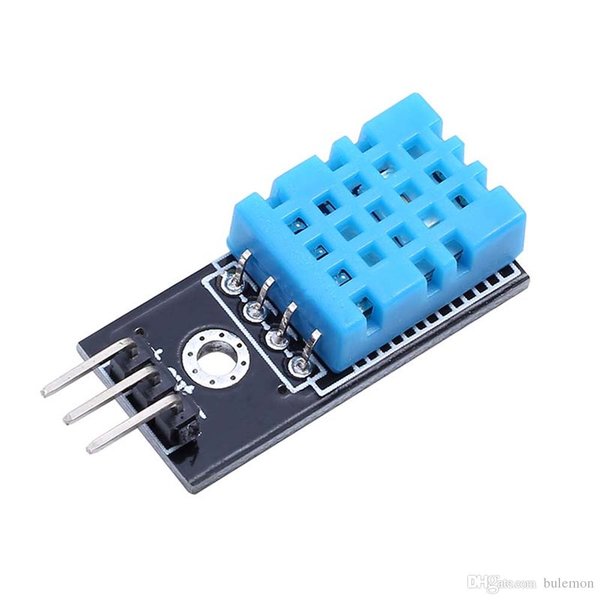
Relay adalah Saklar (Switch) yang dioperasikan secara listrik dan merupakan komponen Electromechanical (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat Kontak Saklar/Switch). Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (low power) dapat menghantarkan listrik yang bertegangan lebih tinggi.

1. Adapter 5v



Adaptor adalah sebuah perangkat berupa rangkaian elektronika untuk mengubah tegangan listrik yang besar menjadi tegangan listrik lebih kecil, atau rangkaian untuk mengubah arus bolak-balik (arus AC) menjadi arus searah (arus DC).

1. Sensor DHT 11



Sensor DHT11 adalah module sensor yang berfungsi untuk mendeteksi suhu dan kelembaban yang memiliki output tegangan analog yang dapat diolah lebih lanjut menggunakan mikrokontroler. Module sensor ini tergolong kedalam elemen resistif seperti perangkat pengukur suhu seperti contohnya yaitu NTC.

# BAB III

# KONSEP DAN RANCANGAN



## Identifikasi Kebutuhan

Beberapa kebutuhan terhadap prototype alat yang akan dibuat :

**Perangkat Lunak**

1. Arduino IDE
2. Code Editor apapun bisa menggunakan Notepad++, Sublime Text, atau Visual Studio Code

**Perangkat Keras**

|  |  |  |
| --- | --- | --- |
| Nama | Jumlah | Harga |
| NodeMCU ESP8266 Amica | 1 | Rp. 49.000 |
| Modul Relay Active LOW | 5 | Rp. 29.500 |
| Modul Sensor DHT 11 | 1 | Rp. 13.900 |
| Jumper Male to Female | 20 | Rp. 10.400 |
| Jumper Male to Male | 4 |
| Bread Board 400P | 1 | Rp. 10.000 |
| Kabel Sambungan 0,5m | 1 | Rp. 8.500 |
| Fitting Lampu | 2 | Rp. 5.600 |
| Lampu LED 3 watt | 2 | Rp. 4.700 |
| Steker Contra | 3 | Rp. 7.500 |
| Adapter AC to DC 5v | 1 | Rp. 10.00 |
| Kabel Merah | 5m | Rp.30.000 |
| Kabel Hitam | 5m |

## Analisis Kebutuhan

Berdasarkan identifikasi kebutuhan diatas, maka diperoleh beberapa analisis kebutuhan terhadap prototype alat yang akan dibuat :

**Perangkat Lunak**

1. Arduino IDE digunakan untuk membuat sketch pemrogaman. Arduino IDE ini berguna untuk mengedit, membuat, meng-upload ke NodeMCU ESP8266, dan meng-coding program.
2. Code editor atau text editor yang digunakan untuk menuliskan kode-kode dari perangkat lunak yang sedang dikembangkan.

**Perangkat Keras**

1. NodeMCU ESP8266 Amica sebagai Mikrokontroler digunakan untuk mengolah data input, output, dan mengendalikan seluruh sistem kerja pada protoype alat.
2. Modul Relay Active LOW digunakan sebagai saklar listrik yang dapat memutuskan dan menghubungkan listrik dengan mengirim sinyal inputan.
3. Modul Sensor DHT 11 berfungsi untuk mendeteksi suhu dan kelembaban yang ada disekitarnya. Sensor ini memiliki output tegangan analog yang dapat diolah lebih lanjut menggunakan mikrokontroler.
4. Jumper Male to Female digunakan untuk menghubungkan komponen.
5. Jumper Male to Male digunakan untuk menghubungkan sumber tegangan ke Mikrokontroler, Relay dan Sensor.
6. Bread Board 400P berfungsi untuk merancang sebuah rangkaian elektronik.
7. Kabel Sambungan 0,5m digunakan untuk menyambungkan sumber listrik AC.
8. Fitting Lampu berfungsi untuk menghubungkan lampu dengan listrik.
9. Lampu LED 3 watt
10. Steker Contra atau colokan berfungsi untuk mengalirkan listrik.
11. Adapter AC to DC 5v sebagai sumber tegangan untuk NodeMCU ESP8266, Relay, dan Sensor.
12. Kabel Merah dan Hitam 5 meter sebagai media penghantar listrik.

## Blok Diagram Sistem

Penjelasan :

Cara kerja saklar on off

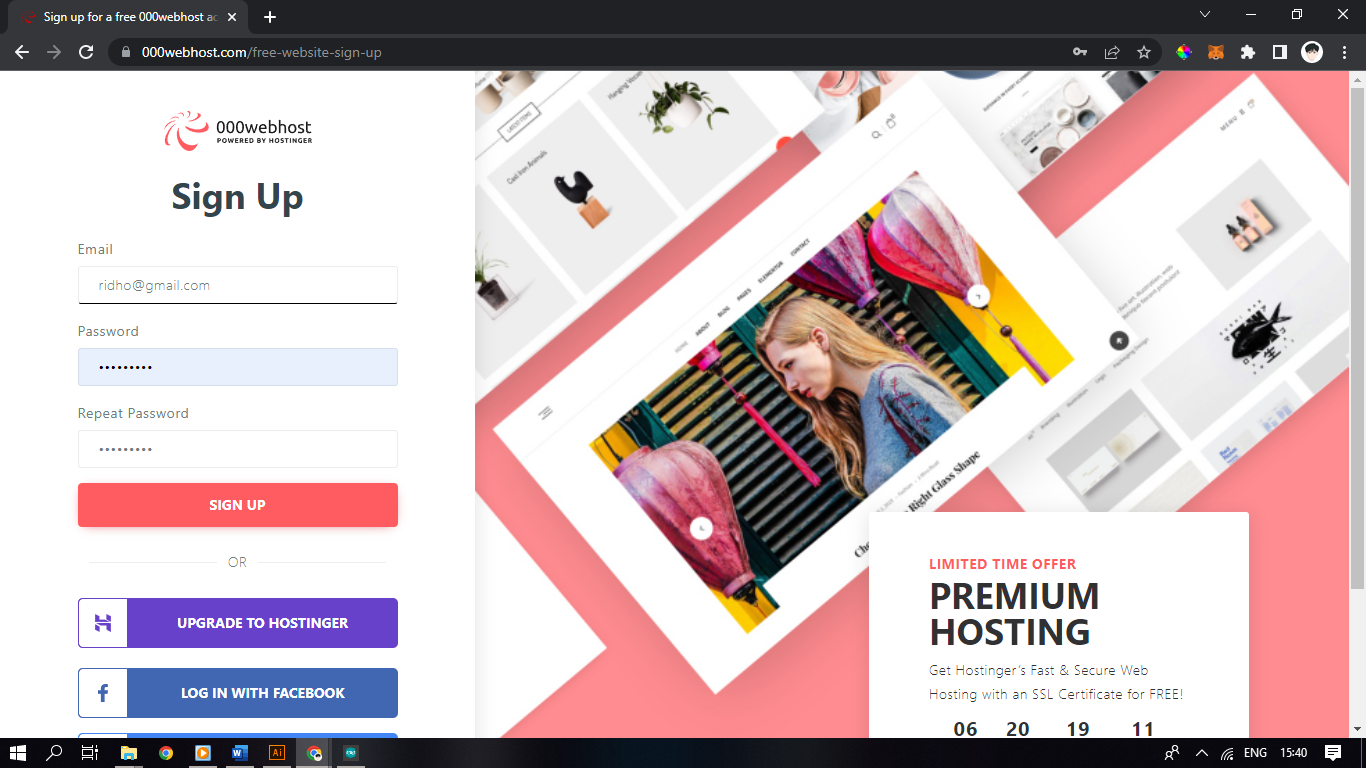
Pengguna Web akan mengirim perintah menyalakan atau mematikan Lampu dengan menekan saklar pada Web yang bernilai 1 jika on dan 0 jika off, lalu web akan menerima perintah tersebut lalu mengirimnya pada database MySQL yang tersimpan pada web server menggunakan sintaks PHP , lalu NodeMCU ESP8266 akan meminta data dari database setiap detiknya untuk mengecek data lampu 1 jika statusnya 1 maka lampu akan menyala dan jika statusnya 0 maka lampu akan mati.

Cara kerja sensor

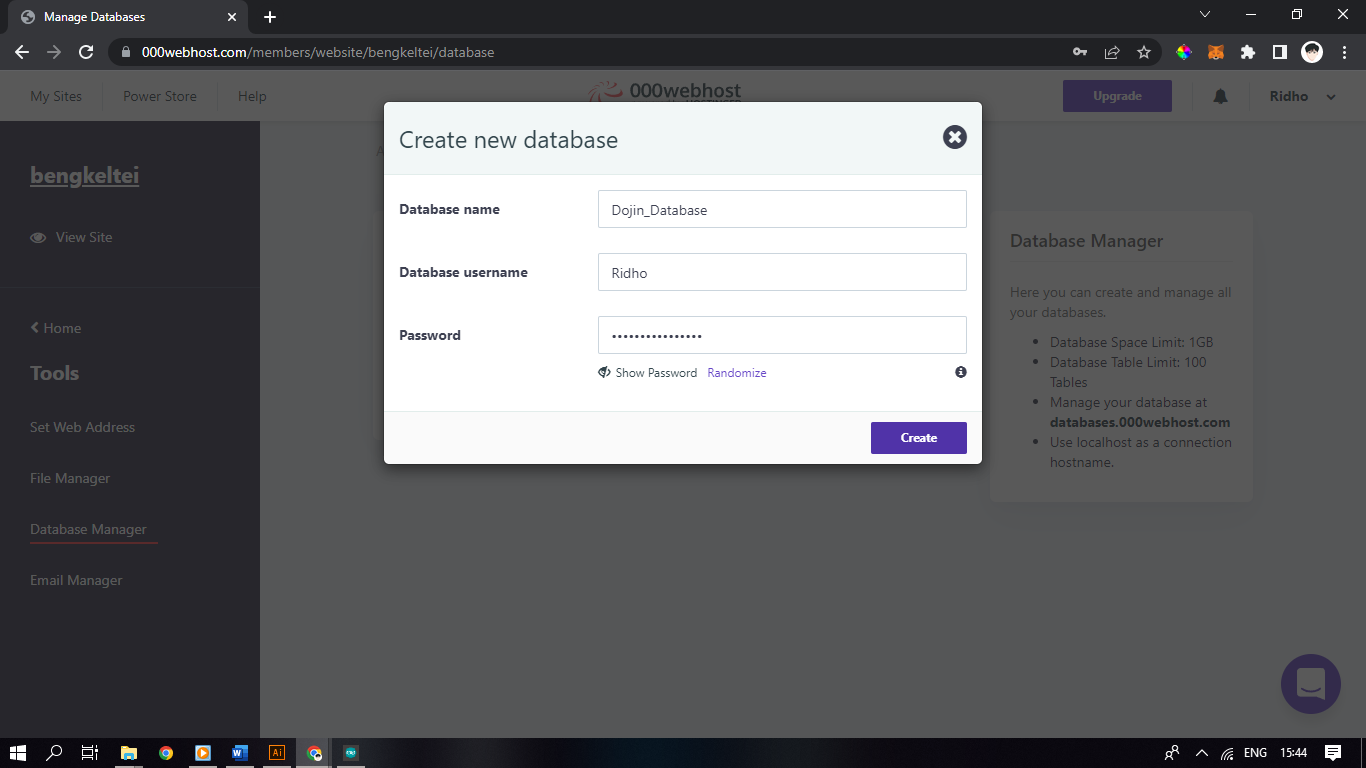
NodeMCU ESP8266 akan mengirimkan nilai suhu dan kelembapan yang dideteksi oleh sensor DHT11 ke database MySQL melalui url, setelah diterima nilai sensornya lalu di update di database dan ambil nilai sensor di dalam database menggunakan sintaks PHP dan akan ditampilkan dengan Html, Css, dan JavaScript pada tampilan Website.

## Langkah Pembuatan Web dan Alat

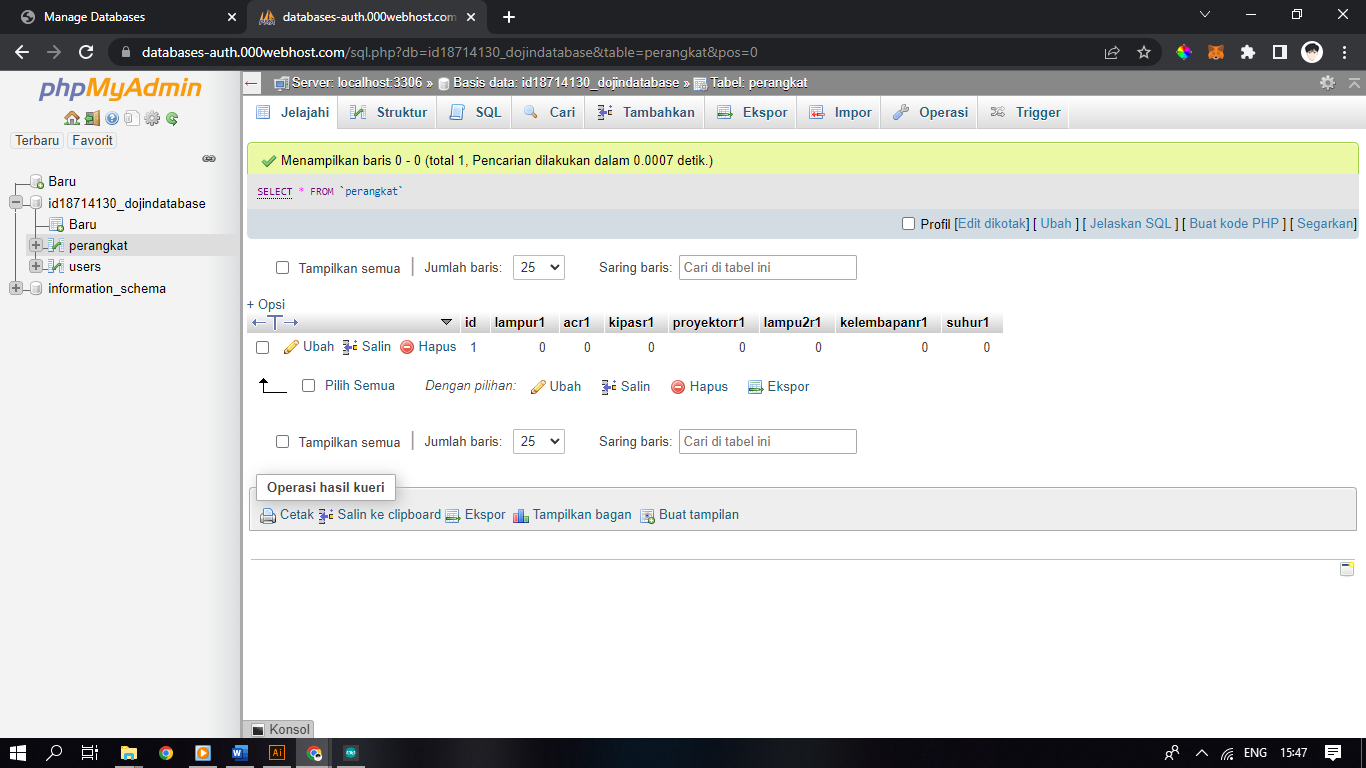
1. Perangkat Lunak
2. Buat akun layanan Hosting gratis untuk mengelola aplikasi web dan database.



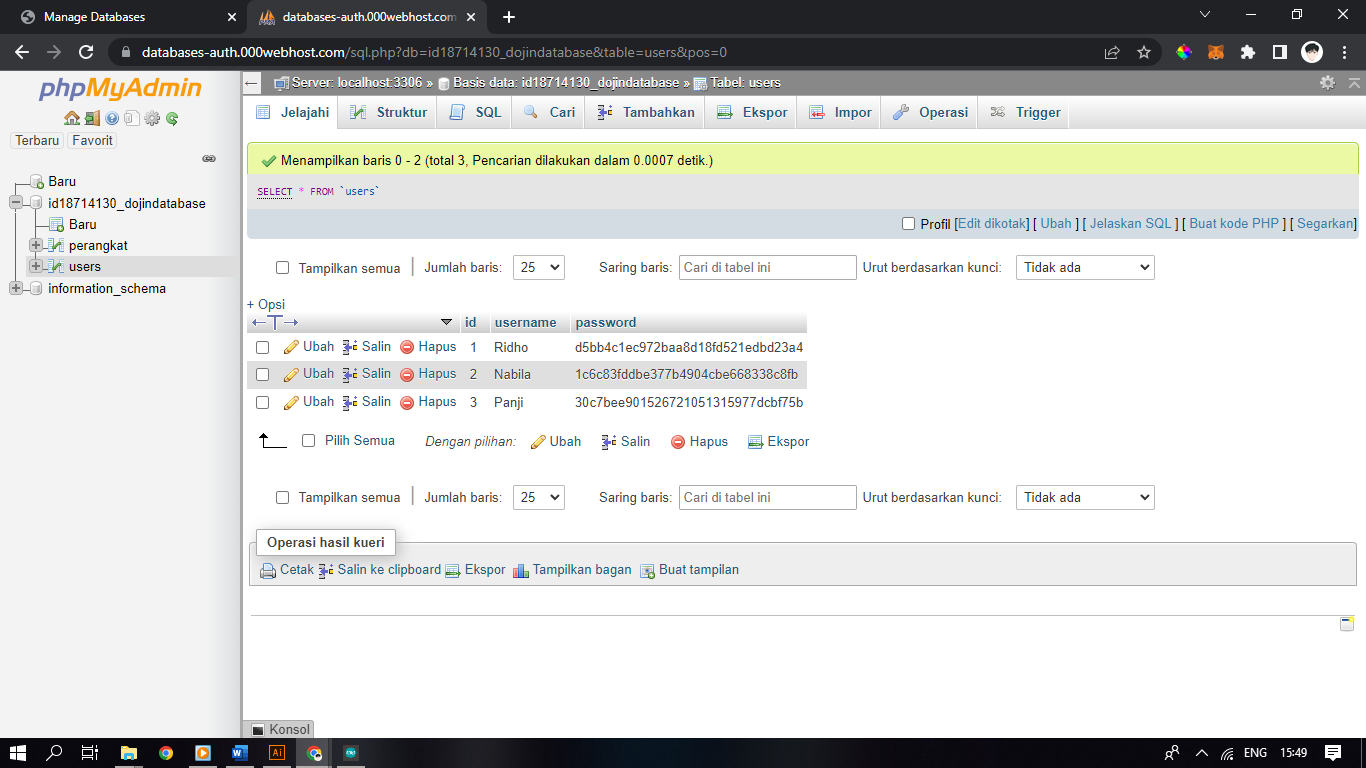
1. Buat akun database.



1. Buat tabel “perangkat” untuk menyimpan data peralatan listrik, suhu dan kelembapan.



1. Buat tabel “Users” untuk menyimpan data pengguna yang bisa mengakses web.



1. Buka aplikasi code editor lalu buat file *connect.php* untuk membuat koneksi atau menghubungkan web yang akan dibuat dengan database.

*connect.php*

<?php

$conn = mysqli\_connect("Nama Host Database", "Username Database", "Password Database", "Nama Database");

?>

1. Buat folder baru bernama Login yang berisi file *index.php*, *auth.php*, dan *logout.php*

*Login/Index.php*

Sebagai halaman login yang tampil pada web untuk mengakses sebagai pengguna

<?php

require '../connect.php' //KONEKSI KE DATABASE

?>

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

</head>

<body>

<div class="container">

<!-- BUAT FORM LOGIN DENGAN METODE POST YANG MANA DATA YANG DIMASUKAN PENGGUNA AKAN DI AUTENTIKASI DI ARAHKAN KE FILE auth.php -->

<form action="auth.php" method="post">

<div class="form-wrap">

div class="login-title">

<h2>Login</h2>

</div>

<!-- PENGKONDISIAN HASIL DATA AUTENTIKASI YANG DIMASUKKAN PENGGUNA -->

<?php if (isset($\_GET['user'])) {

if ($\_GET['user'] == "fail") {echo "<p>Username dan password tidak sesuai</p>";}

elseif ($\_GET['user'] == "not-login") {echo "<p>Login terlebih dahulu</p>";} ?>

<!-- DATA NAMA -->

<div class="inputan">

<label for="username">Username</label>

<input type="text" name="username" id="username" placeholder="Username" value="<?php if (isset($\_COOKIE['name'])) {echo $\_COOKIE['name'];} ?>">

</div>

<!-- DATA PASSWORD -->

<div class="inputan">

<label for="password">Password</label>

<input type="password" name="password" id="password" placeholder="Password" value="<?php if (isset($\_COOKIE['pass'])) {echo $\_COOKIE['pass'];} ?>">

</div>

<!-- FITUR PENGINGAT JIKA DIAKTIFKAN MAKA PENGGUNA HANYA PERLU LOGIN SEKALI JIKA TIDAK LOGOUT AKAN TERUS DIINGAT OLEH WEB -->

<div class="remember">

<input type="checkbox" name="remember" id="remember" <?php if (isset($\_COOKIE[ 'name'])) {echo "checked";} ?>>

<div class="title-remember"><label for="remember">Ingat Aku</label></div>

</div>

<!-- TOMBOL SUBMIT UNTUK MENGIRIM DATA -->

<div class="btn">

<button type="submit" name="login">Login</button>

</div>

</div>

</form>

</div>

</body>

</html>

*Login/auth.php*

Berfungsi untuk menyamakan data pengguna dengan data yang ada di database

<?php

session\_start(); // MENGAKTIFKAN SESSION UNTUK FITUR LOGIN

require '../connect.php'; //KONEKSI KE DATABASE

$username = $\_POST['username']; //TANGKAP DATA NAMA YANG DIKIRIM PENGGUNA

$password = $\_POST['password']; //TANGKAP DATA PASSWORD YANG DIKIRIM PENGGUNA

$data = mysqli\_query($conn, "SELECT \* FROM users WHERE username='$username' "); //MENGAMBIL DATA NAMA YANG BERADA DI DATABASE

$result = mysqli\_fetch\_array($data); // MENGUBAH DATA MENJADI TERSTRUKTUR

// MENYAMAKAN DATA NAMA DAN PASSWORD YANG DIKIRIM PENGGUNA DENGAN DATA YANG ADA DI DATABASE

if ($result['username'] == $username && $result['password'] == md5($password)) {

$\_SESSION['login'] = "true"; // JIKA DATA SAMA MAKA SET SESSION LOGIN PENGGUNA MENJADI TRUE

// JIKA CHECKBOX INGAT AKU DICENTANG MAKA WEB AKAN MENGINGAT USERNAME DAN PASSWORD PENGGUNA

if (isset($\_POST['remember'])) {

setcookie('name', $username, time() + 60 \* 60 \* 24 \* 3);

setcookie('pass', $password, time() + 60 \* 60 \* 24 \* 3);

('cookie', 'true', time() + 60 \* 60 \* 24 \* 3);

}

else {

setcookie('name', '', time() - 60 \* 60 \* 24 \* 3);

setcookie('pass', '', time() - 60 \* 60 \* 24 \* 3);

setcookie('cookie', '', time() - 60 \* 60 \* 24 \* 3);

}

// MENGALIHKAN PENGGUNA KE HALAMAN UTAMA WEB

header("location:../");

}

// JIKA DATA TIDAK SAMA PENGGUNA AKAN DIALIHKAN KE HALAMAN LOGIN KEMBALI

else {

header("location:index.php?user=fail");

}

?>

*Login/logout.php*

Berfungsi untuk menghapus sesi login

<?php

session\_start(); // MENGAKTIFKAN SESSION UNTUK FITUR LOGIN

session\_destroy(); // MENGHAPUS DATA SESSION

header("location:index.php"); // MENGALIHKAN USER KEMBALI KE HALAMAN LOGIN

?>

1. Buat halaman utama web dengan membuat file *index.php.*

*Index.php*

<?php

session\_start(); // MENGAKTIFKAN SESSION UNTUK FITUR LOGIN

// MENGECEK SESSION LOGIN JIKA LOGIN TIDAK BER STATUS TRUE MAKA ALIHKAN PENGGUNA KEMBALI KE HALAMAN LOGIN

if ($\_SESSION['login'] != true) {

header("location:Login");

}

require 'connect.php'; // KONEKSI KE DATABASE

$sql = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$data = mysqli\_fetch\_array($sql); // MENGUBAH DATA MENJADI TERSTRUKTUR

// MENJUMLAHKAN DATA PERANGKAT YANG BERNILAI 1 ATAU AKTIF

$aktif = $data['lampur1'] + $data['lampu2r1'] + $data['acr1'] + $data['kipasr1'] + $data['proyektorr1'];

?>

<!DOCTYPE html>

<html>

<head>

<title>Home</title>

<body>

<header>

<div class="main-title">

<h2>Bengkel</h2>

</div>

</header>

<div class="container">

<!-- MEMBUAT KOTAK PILIHAN RUANGAN YANG AKAN MENGALIHKAN KE HALAMAN KONTROL RUANGAN -->

<!-- RUANG 1 -->

<div class="card">

<div class="title-room">

<!-- MEMBUAT TOMBOL LINK MENUJU HALAMAN RUANG 1 -->

<a href="Ruang-1" class="ruang1">

<h4>Ruang 1</h4>

<div class="teacher">

<!-- MENGECEK JUMLAH PERANGKAT YANG AKTIF DAN MENAMPILKANNYA PADA KOTAK RUANGAN -->

<p>

<?php if ($aktif > 0) { echo $aktif . " Perangkat menyala";}

else {echo "";} ?>

</p>

</div>

</a>

</div>

</div>

</div>

<div class="foot">

<div class="copyright">&copy; Copyright 2022, Dojin. <br>Dibuat Oleh <span><a href="#">Ridho</a></span></div>

<!-- TOMBOL LOGOUT YANG BERFUNGSI UNTUK MEMBUKA FILE logout.php -->

<div class="logout"><a href="Login/logout.php">Logout</a></div>

</div>

</body>

</html>

1. Buat folder baru bernama *Ruang-1* dan tambahakan file *index.php* halaman kontrol perangkat dan monitoring suhu kelembapan ruangan 1

*Ruang-1/index.php*

<?php

require '../connect.php'; // KONEKSI KE DATABASE

session\_start(); // MENGAKTIFKAN SESSION UNTUK FITUR LOGIN

// MENGECEK SESSION LOGIN JIKA LOGIN TIDAK BER STATUS TRUE MAKA ALIHKAN PENGGUNA KEMBALI KE HALAMAN LOGIN

if ($\_SESSION['login'] != true) {

header("location:../Login/index.php?user=not-login");

}

$sql = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$data = mysqli\_fetch\_array($sql); // MENGUBAH DATA MENJADI TERSTRUKTUR

$lampu1 = $data['lampur1']; // MENGAMBIL DATA LAMPU 1 DI DATABASE

$lampu2 = $data['lampu2r1']; // MENGAMBIL DATA LAMPU 2 DI DATABASE

$ac = $data['acr1']; // MENGAMBIL DATA AC DI DATABASE

$kipas = $data['kipasr1']; // MENGAMBIL DATA KIPAS DI DATABASE

$proyektor = $data['proyektorr1']; // MENGAMBIL DATA PROYEKTOR DI DATABASE

$suhu = $data['suhur1']; // MENGAMBIL DATA SUHU DI DATABASE

$kelembapan = $data['kelembapanr1']; // MENGAMBIL DATA KELEMBAPAN DI DATABASE

?>

<!DOCTYPE html>

<html>

<head>

<title>Ruang 1</title>

</head>

<!-- JALANKAN FUNGSI SENSOR SUHU DAN SENSOR LEMBAP-->

<body onload="sensorSuhu();

sensorLembap();">

<header>

<!-- MEMBUAT LINK TOMBOL KEMBALI KE HALAMAN UTAMA -->

<div class="back"><a href="../"><svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 34 57.61"><defs><style>.cls-1{fill:none;stroke:#fff;stroke-linecap:round;stroke-linejoin:round;stroke-width:8px;}</style></defs><title>back</title><g id="Layer\_2" data-name="Layer 2"><g id="Layer\_1-2" data-name="Layer 1"><polyline class="cls-1" points="28.61 52.61 5 29 29 5"/></g></g></svg></a></div>

<h2 class="ruang">Ruang 1</h2>

<div class="suhudankelembapan">

<div class="suhu">

<!-- TAMPILAN NILAI SUHU -->

<span id="suhu"></span><span class="celcius">&degC</span><br>Suhu

</div>

<div class="kelembapan">

<!-- TAMPILAN NILAI KELEMBAPAN -->

<span id="kelembapan"></span><span class="humidity">%<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24">

<path fill="#fff" d="M18 3a2 2 0 0 0-2 2 2 2 0 0 0 2 2 2 2 0 0 0 2-2 2 2 0 0 0-2-2zm0 1a1 1 0 0 1 1 1 1 1 0 0 1-1 1 1 1 0 0 1-1-1 1 1 0 0 1 1-1zM18 14.7c0 3.479394-2.686291 6.3-6 6.3-3.3137078 0-6-2.820606-6-6.3 2e-7-3.479394 6-11.7 6-11.7s6 8.220606 6 11.7z" paint-order="stroke fill markers"/></svg>

</span><br>Lembap

</div>

</div>

</header>

<div class="container">

<!-- MEMBUAT SAKLAR UNTUK SETIAP PERANGKAT SEBAGAI PENGONTROL PERANGKAT -->

<!-- LAMPU -->

<div class="card">

<div class="card-list">

<div class="card-icon"></div>

<div class="card-title">Lampu 1</div>

</div>

<label class="switch">

<!-- JIKA CHECKBOX DI CENTANG MAKA JALANKAN FUNGSI UBAH STATUS LAMPU DAN MENGECEK STATUS LAMPU -->

<input type="checkbox" onchange="ubahStatusLampu(this.checked)" <?php if($lampu1 == 1) echo "checked"; ?>>

<span class="slider"></span>

</label>

</div>

<!-- LAMPU 2 -->

<div class="card">

<div class="card-list">

<div class="card-icon"></div>

<div class="card-title">Lampu 2</div>

</div>

<label class="switch">

<!-- JIKA CHECKBOX DI CENTANG MAKA JALANKAN FUNGSI UBAH STATUS LAMPU 2 DAN MENGECEK STATUS LAMPU 2 -->

<input type="checkbox" onchange="ubahStatusLampu2(this.checked)" <?php if($lampu2 == 1) echo "checked"; ?>>

<span class="slider"></span>

</label>

</div>

<!-- AC -->

<div class="card">

<div class="card-list">

<div class="card-icon"></div>

<div class="card-title">Air Conditioner</div>

</div>

<label class="switch">

<!-- JIKA CHECKBOX DI CENTANG MAKA JALANKAN FUNGSI UBAH STATUS AC DAN MENGECEK STATUS AC -->

<input type="checkbox" onchange="ubahStatusAc(this.checked)" <?php if($ac == 1) echo "checked"; ?>>

<span class="slider"></span>

</label>

</div>

<!-- KIPAS -->

<div class="card">

<div class="card-list">

<div class="card-icon"></div>

<div class="card-title">Kipas</div>

</div>

<label class="switch">

<!-- JIKA CHECKBOX DI CENTANG MAKA JALANKAN FUNGSI UBAH STATUS KIPAS DAN MENGECEK STATUS KIPAS -->

<input type="checkbox" onchange="ubahStatusKipas(this.checked)" <?php if($kipas == 1) echo "checked"; ?>>

<span class="slider"></span>

</label>

</div>

<!-- PROYEKTOR -->

<div class="card">

<div class="card-list">

<div class="card-icon"></div>

<div class="card-title">Proyektor</div>

</div>

<label class="switch">

<!-- JIKA CHECKBOX DI CENTANG MAKA JALANKAN FUNGSI UBAH STATUS PROYEKTOR DAN MENGECEK STATUS PROYEKTOR -->

<input type="checkbox" onchange="ubahStatusProyektor(this.checked)" <?php if($proyektor == 1) echo "checked"; ?>>

<span class="slider"></span>

</label>

</div>

</div>

<script type="text/javascript">

// FUNGSI SENSOR SUHU

function sensorSuhu(){

// MEMBUAT OBJEK AJAX JAVASCRIPT YANG TERHUBUNG KE DATABASE SECARA REALTIME

var ajax = new XMLHttpRequest();

ajax.onload = function() {

// MENGISI TAMPILAN SUHU DENGAN NILAI SUHU YANG TERDAPAT DI DATABASE SETIAP

document.getElementById("suhu").innerHTML = ajax.responseText;

ajax.open( "GET", "status/datasuhu.php", true);

ajax.send();

}

// MENJALAN FUNGSI SENSOR SUHU SETIAP DETIK

setInterval(function(){sensorSuhu();}, 1000);

// FUNGSI SENSOR KELEMBAPAN

function sensorLembap(){

var ajax = new XMLHttpRequest();

ajax.onload = function() {

// MENGISI TAMPILAN KELEMBAPAN DENGAN NILAI KELEMBAPAN YANG TERDAPAT DI DATABASE

document.getElementById("kelembapan").innerHTML = ajax.responseText

}

ajax.open( "GET", "status/datakelembapan.php", true);

ajax.send();

}

// MENJALAN FUNGSI SENSOR KELEMBAPAN SETIAP DETIK

setInterval(function(){sensorLembap();}, 1000);

// FUNGSI LAMPU 1

function ubahStatusLampu(status){

// MEMBUAT OBJEK AJAX JAVASCRIPT YANG TERHUBUNG KE DATABASE SECARA REALTIME

var ajax = new XMLHttpRequest();

ajax.onreadystatechange = function() {

if (ajax.readyState == 4 && ajax.status == 200) {

}

}

// MENGIRIM STATUS LAMPU 1 KE DATABASE

ajax.open( "GET", "device/lampu1.php?stat=" + status, true);

ajax.send();

}

// FUNGSI LAMPU 2

function ubahStatusLampu2(status){

// MEMBUAT OBJEK AJAX JAVASCRIPT YANG TERHUBUNG KE DATABASE SECARA REALTIME

var ajax = new XMLHttpRequest();

ajax.onreadystatechange = function() {

if (ajax.readyState == 4 && ajax.status == 200) {

}

}

// MENGIRIM STATUS LAMPU 2 KE DATABASE

ajax.open( "GET", "device/lampu2.php?stat=" + status, true);

ajax.send();

}

// FUNGSI AC

function ubahStatusAc(status){

// MEMBUAT OBJEK AJAX JAVASCRIPT YANG TERHUBUNG KE DATABASE SECARA REALTIME

var ajax = new XMLHttpRequest();

ajax.onreadystatechange = function() {

if (ajax.readyState == 4 && ajax.status == 200) {

}

}

// MENGIRIM STATUS AC KE DATABASE

ajax.open( "GET", "device/ac.php?stat=" + status, true);

ajax.send();

}

// FUNGSI KIPAS

function ubahStatusKipas(status){

// MEMBUAT OBJEK AJAX JAVASCRIPT YANG TERHUBUNG KE DATABASE SECARA REALTIME

var ajax = new XMLHttpRequest();

ajax.onreadystatechange = function() {

if (ajax.readyState == 4 && ajax.status == 200) {

}

}

// MENGIRIM STATUS KIPAS KE DATABASE

ajax.open( "GET", "device/kipas.php?stat=" + status, true);

ajax.send();

}

// FUNGSI PROYEKTOR

function ubahStatusProyektor(status){

// MEMBUAT OBJEK AJAX JAVASCRIPT YANG TERHUBUNG KE DATABASE SECARA REALTIME

var ajax = new XMLHttpRequest();

ajax.onreadystatechange = function() {

if (ajax.readyState == 4 && ajax.status == 200) {

}

}

// MENGIRIM STATUS PROYEKTOR KE DATABASE

ajax.open( "GET", "device/proyektor.php?stat=" + status, true);

ajax.send();

}

</script>

</body>

</html>

1. Buat folder bernama *device* didalam folder *Ruang-1* untuk menyimpan file yang mengubah data di database dengan yang dikirim ajax

*Ruang-1/device/lampu1.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$result = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$stat = $\_GET['stat'];

// JIKA STATUS SAMA DENGAN TRUE MAKA UBAH DATA LAMPU 1 YANG YANG BERADA DALAM DATABASE MENJADI 1 JIKA TIDAK UBAH MENJADI 0

if ($stat == "true") {

mysqli\_query($conn, "UPDATE perangkat SET lampur1=1");

} else {

mysqli\_query($conn, "UPDATE perangkat SET lampur1=0");

}

?>

*Ruang-1/device/lampu2.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$result = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$stat = $\_GET['stat'];

// JIKA STATUS SAMA DENGAN TRUE MAKA UBAH DATA LAMPU 2 YANG YANG BERADA DALAM DATABASE MENJADI 1 JIKA TIDAK UBAH MENJADI 0

if ($stat == "true") {

mysqli\_query($conn, "UPDATE perangkat SET lampu2r1=1");

} else {

mysqli\_query($conn, "UPDATE perangkat SET lampu2r1=0");

}

?>

*Ruang-1/device/ac.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$result = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$stat = $\_GET['stat']; // MENGAMBIL STATUS AC3 YANG DIKIRIM AJAX

// JIKA STATUS SAMA DENGAN TRUE MAKA UBAH DATA AC YANG YANG BERADA DALAM DATABASE MENJADI 1 JIKA TIDAK UBAH MENJADI 0

if ($stat == "true") {

mysqli\_query($conn, "UPDATE perangkat SET acr1=1");

} else {

mysqli\_query($conn, "UPDATE perangkat SET acr1=0");

}

?>

*Ruang-1/device/kipas.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$result = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$stat = $\_GET['stat'];

// JIKA STATUS SAMA DENGAN TRUE MAKA UBAH DATA KIPAS YANG YANG BERADA DALAM DATABASE MENJADI 1 JIKA TIDAK UBAH MENJADI 0

if ($stat == "true") {

mysqli\_query($conn, "UPDATE perangkat SET kipasr1=1");

} else {

mysqli\_query($conn, "UPDATE perangkat SET kipasr1=0");

}

?>

*Ruang-1/device/proyektor.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$result = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$stat = $\_GET['stat'];

// JIKA STATUS SAMA DENGAN TRUE MAKA UBAH DATA PROYEKTOR YANG YANG BERADA DALAM DATABASE MENJADI 1 JIKA TIDAK UBAH MENJADI 0

if ($stat == "true") {

mysqli\_query($conn, "UPDATE perangkat SET proyektorr1=1");

} else {

mysqli\_query($conn, "UPDATE perangkat SET proyektorr1=0");

}

?>

1. Buat folder baru bernama *status* didalam folder *Ruang-1* yang berfungsi mencetak status setiap perangkat yang berada dalam database

*Ruang-1/status/datalampu1.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$sql = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$data = mysqli\_fetch\_array($sql); // MENGUBAH DATA MENJADI TERSTRUKTUR

$lampu1 = $data['lampur1']; // MENGAMBIL DATA LAMPU 1 DI DALAM DATABASE

echo $lampu1; // MENCETAK STATUS LAMPU 1 KE HALAMAN WEB

?>

*Ruang-1/status/datalampu2.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$sql = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$data = mysqli\_fetch\_array($sql); // MENGUBAH DATA MENJADI TERSTRUKTUR

$lampu2 = $data['lampu2r1']; // MENGAMBIL DATA LAMPU 2 DI DALAM DATABASE

echo $lampu2; // MENCETAK STATUS LAMPU 2 KE HALAMAN WEB

?>

*Ruang-1/status/dataac.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$sql = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$data = mysqli\_fetch\_array($sql); // MENGUBAH DATA MENJADI TERSTRUKTUR

$ac = $data['acr1']; // MENGAMBIL DATA AC DI DALAM DATABASE

echo $ac; // MENCETAK STATUS AC KE HALAMAN WEB

?>

*Ruang-1/status/datakipas.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$sql = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$data = mysqli\_fetch\_array($sql); // MENGUBAH DATA MENJADI TERSTRUKTUR

$kipas = $data['kipasr1']; // MENGAMBIL DATA KIPAS DI DALAM DATABASE

echo $kipas; // MENCETAK STATUS KIPAS KE HALAMAN WEB

?>

*Ruang-1/status/dataproyektor.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$sql = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$data = mysqli\_fetch\_array($sql); // MENGUBAH DATA MENJADI TERSTRUKTUR

$proyektor = $data['proyektorr1']; // MENGAMBIL DATA PROYEKTOR DI DALAM DATABASE

echo $proyektor; // MENCETAK STATUS PROYEKTOR KE HALAMAN WEB

?>

*Ruang-1/status/datasuhu.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$sql = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$data = mysqli\_fetch\_array($sql); // MENGUBAH DATA MENJADI TERSTRUKTUR

$suhu = $data['suhur1']; // MENGAMBIL DATA SUHU DI DALAM DATABASE

echo $suhu; // MENCETAK NILAI SUHU KE HALAMAN WEB

?>

*Ruang-1/status/datakelembapan.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$sql = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$data = mysqli\_fetch\_array($sql); // MENGUBAH DATA MENJADI TERSTRUKTUR

$kelembapan = $data['kelembapanr1']; // MENGAMBIL DATA KELEMBAPAN DI DALAM DATABASE

echo $kelembapan; // MENCETAK NILAI KELEMBAPAN KE HALAMAN WEB

?>

*Ruang-1/status/sendsensor.php*

<?php

require '../../connect.php'; // MEMBUAT KONEKSI KE DATABASE

$sql = mysqli\_query($conn, "SELECT \* FROM perangkat"); // MENGAMBIL DATA PERANGKAT DI DALAM DATABASE

$data = mysqli\_fetch\_array($sql); // MENGUBAH DATA MENJADI TERSTRUKTUR

$suhu = $\_GET ['suhu']; // MENGAMBIL DATA SUHU YANG DI KIRIM ESP8266

$kelembapan = $\_GET ['kelembapan']; // MENGAMBIL DATA KELEMBAPAN YANG DIKIRIM ESP8266

$send = mysqli\_query($conn, "UPDATE perangkat SET suhur1='$suhu', kelembapanr1='$kelembapan'"); // MENGUBAH NILAI SUHU DAN KELEMBAPAN DI DATABASE MENJADI NILAI YANG DI KIRIM ESP8266

// JIKA DATA DI DATABASE BERHASIL DIUBAH MAKA CETAK TERKIRIM KE HALAMAN WEB JIKA TIDAK CETAK GAGAL

if ($send) {

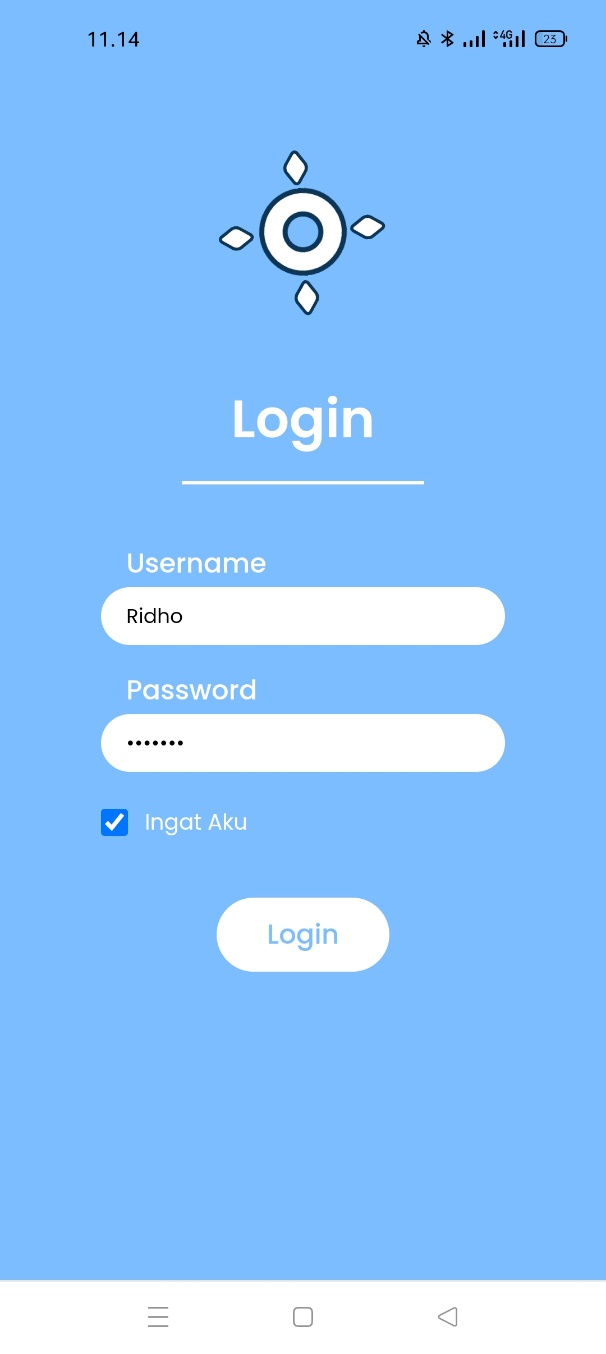
echo "Terkirim";

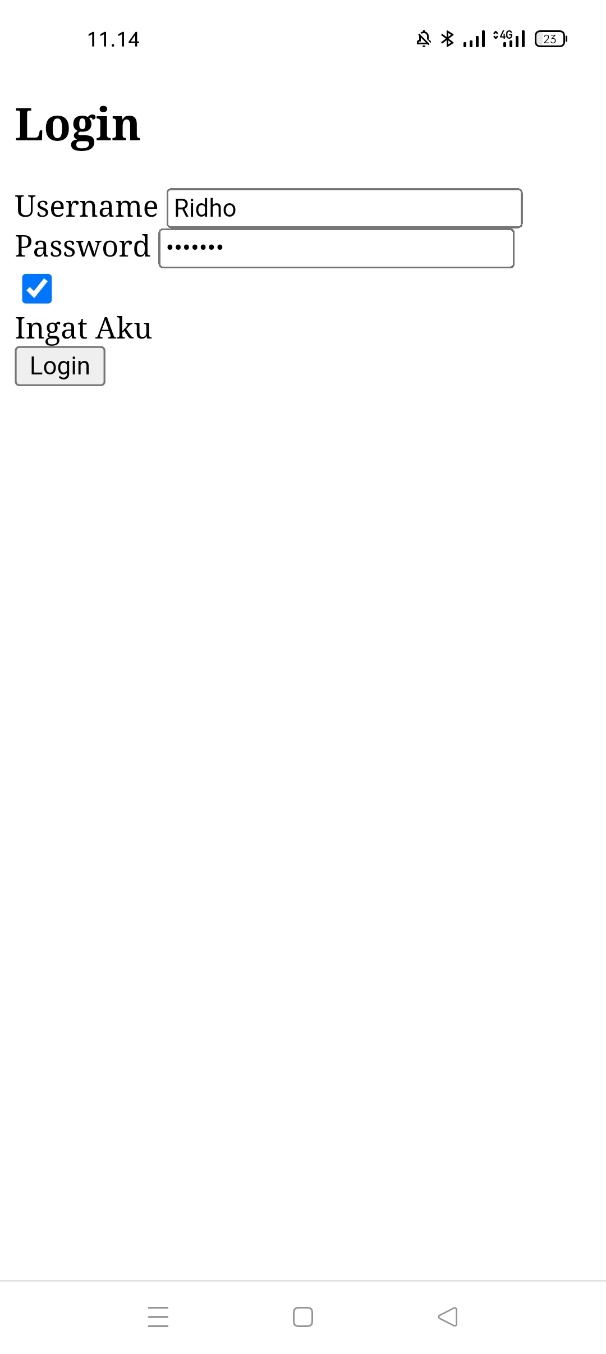
} else {

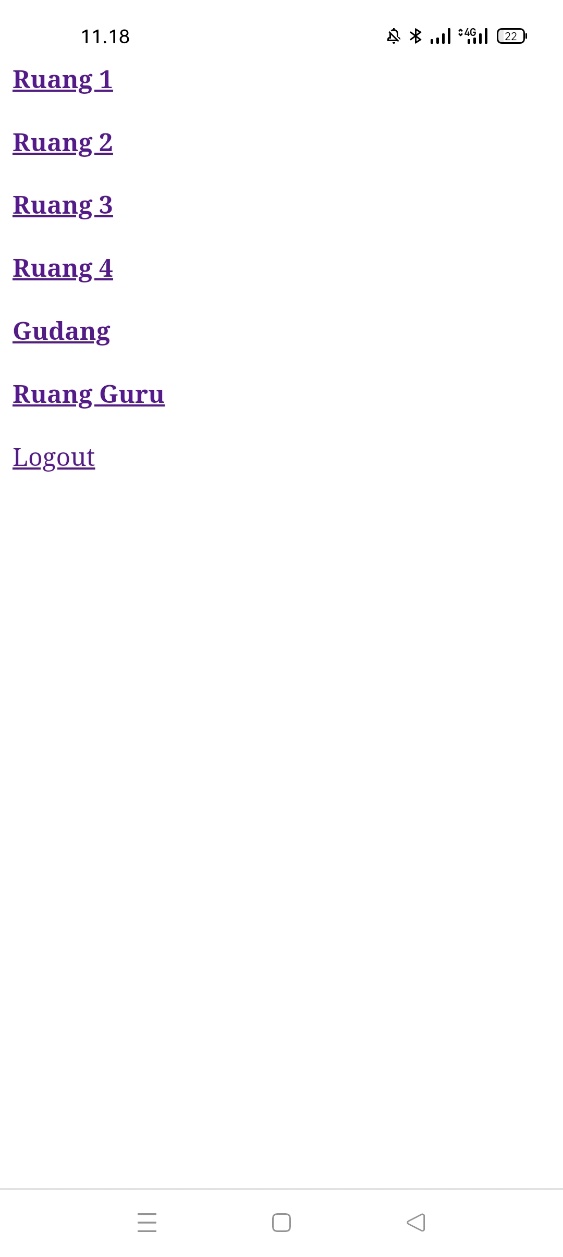
echo "Gagal";

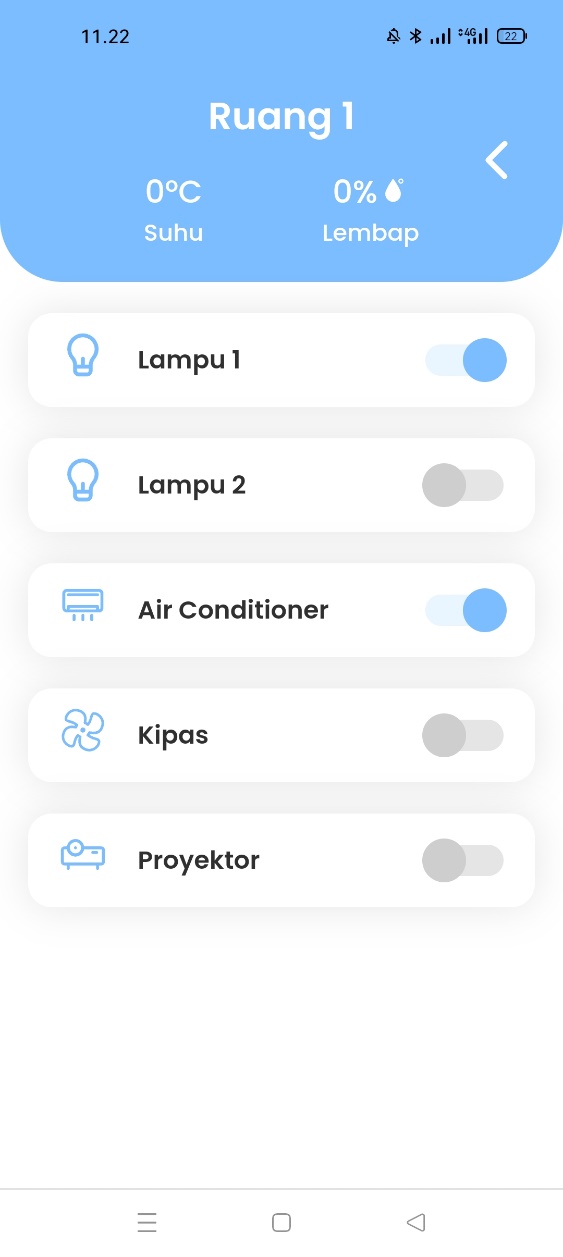
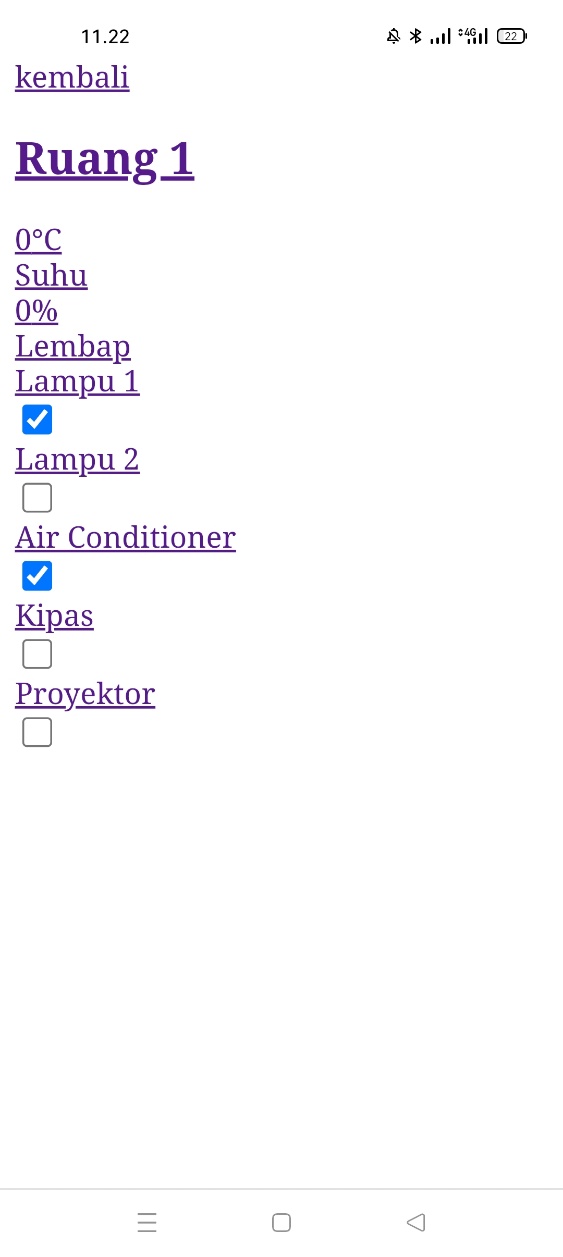
}

?>

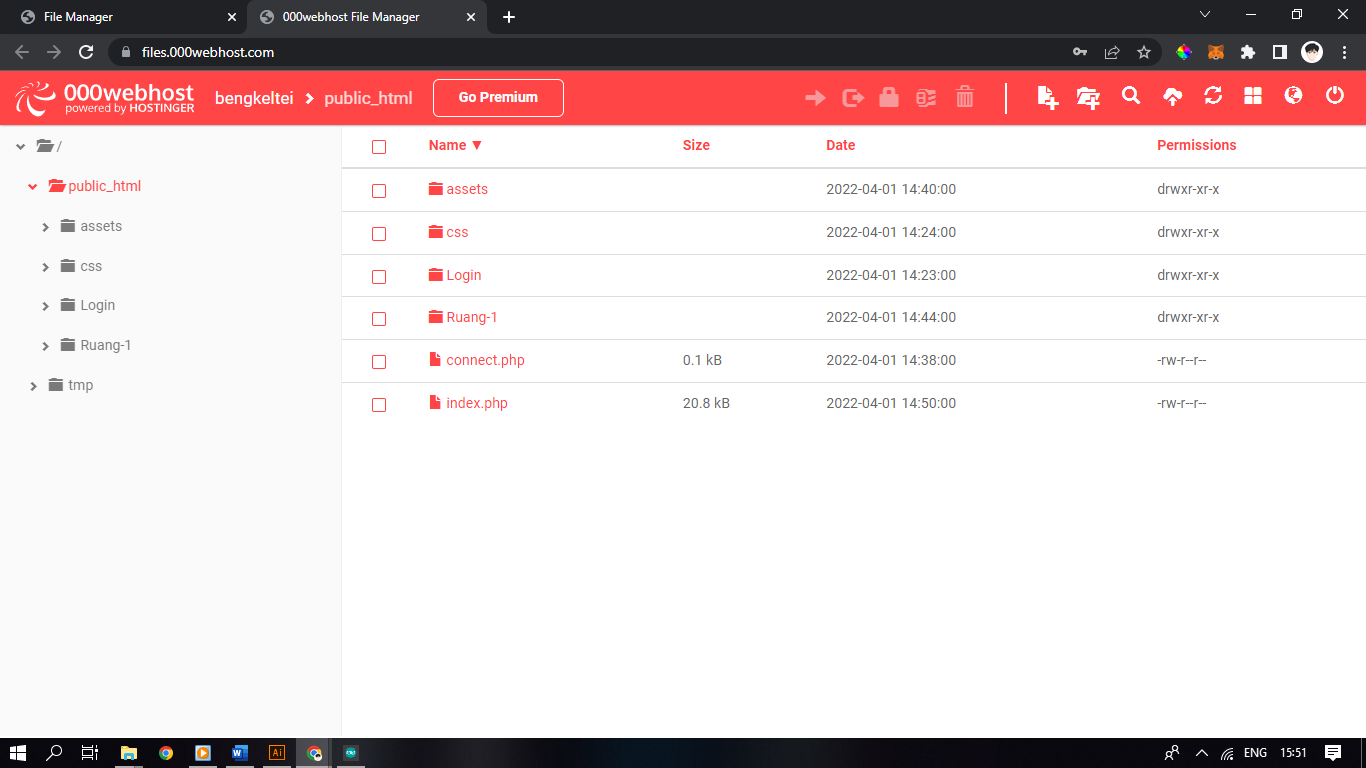
1. Agar tampilan webnya bagus dan menarik sehingga nyaman digunakan oleh pengguna gunakan beberapa file CSS ,font, gambar dan icon lalu hubungkan pada setiap halaman web yang akan diubah tampilannya.





1. Upload semua file ke layanan hosting



1. Buka aplikasi Arduino IDE instal library ESP, dan Sensor DHT11

Ubah board Arduino menjadi Generic ESP8266 Module.

// IMPORT LIBRARY SENSOR DHT

#**include** <DHT.h>

// IMPORT LIBRARY ESP8266 YANG DIBUTUHKAN

#**include** <ESP8266WiFi.h>

#**include** <ESP8266HTTPClient.h>

**const** **char**\* ssid = "Idho"; // NAMA WIFI

**const** **char**\* password = "munyenye"; // PASSWORD WIFI

**const** **char**\* host = "bengkeltei.000webhostapp.com"; // NAMA WEB YANG AKAN DIAKSES

#**define** dhtPin 5 // PIN 5 GPIO SEBAGAI PIN DATA SENSOR

#**define** dhtType DHT11 // TIPE DHT SENSOR

DHT dht(dhtPin,dhtType); // INSIALISASI SENSOR DHT

#**define** pinLampu1 4 // PIN 4 GPIO SEBAGAI PIN LAMPU 1

#**define** pinLampu2 14 // PIN 14 GPIO SEBAGAI PIN LAMPU 2

#**define** pinAc 12 // PIN 12 GPIO SEBAGAI PIN AC

#**define** pinKipas 13 // PIN 13 GPIO SEBAGAI PIN KIPAS

#**define** pinProyektor 3 // PIN 3 GPIO SEBAGAI PIN PROYEKTOR

**void** setup() {

Serial.begin(115200); // MEMULAI SERIAL MONITOR

Serial.println("Menghubungkan ke WiFi");

WiFi.begin(ssid, password); // MEMULAI WIFI

while(WiFi.status() != WL\_CONNECTED) {

Serial.println(".");

delay(500);

}

Serial.println("WiFi Terhubung");

dht.begin(); // MEMULAI SENSOR DHT

// SET PIN SEBAGAI OUTPUT UNTUK MENGIRIM SINYAL KE RELAY

pinMode(pinLampu1,OUTPUT);

pinMode(pinLampu2,OUTPUT);

pinMode(pinAc,OUTPUT);

pinMode(pinKipas,OUTPUT);

pinMode(pinProyektor,OUTPUT);

}

**void** loop() {

// MENGHUBUNGKAN KE WEB

WiFiClient client;

**const** **int** port = 80;

if(!client.connect(host, port)){

Serial.println("Gagal Menghubungkan ke Web");

digitalWrite(pinLampu1,LOW);

digitalWrite(pinLampu2,LOW);

digitalWrite(pinAc,LOW);

digitalWrite(pinKipas,LOW);

digitalWrite(pinProyektor,LOW);

return;

}

Serial.println("Terhubung dengan Web");

**float** suhu = dht.readTemperature(); // MEMBACA NILAI SUHU DARI SENSOR DHT

**int** kelembapan = dht.readHumidity();// MEMBACA NILAI KELEMBAPAN DARI SENSOR DHT

Serial.println("Suhu : " + **String**(suhu)); // MENCETAK NILAI SUHU DI SERIAL MONITOR

Serial.println("Kelembapan : " + **String**(kelembapan)); // MENCETAK NILAI KELEMBAPAN DI SERIAL MONITOR

HTTPClient httpSensor; // SET SEBAGAI CLIENT

// MENGIRIM DATA NILAI SENSOR KE WEB MELALUI METODE GET LINK ATAU URL

**String** sendSensor = "http://" +**String**(host)+ "/Ruang-1/status/sendsensor.php?suhu=" + **String**(suhu) + "&kelembapan=" + **String**(kelembapan);

httpSensor.begin(client,sendSensor); // MENGAKSES LINK WEB

httpSensor.GET(); // MENGAMBIL DATA SENSOR TERKIRIM ATAU TIDAK KE DATABASE WEB

**String** nilaiSensor = httpSensor.getString(); // MERUBAH DATA SENSOR MENJADI HURUF

Serial.println(nilaiSensor); // CETAK HASIL PENGIRIMAN DI SERIAL MONITOR

httpSensor.end(); // MENUTUP LINK WEB

HTTPClient httpLampu1; // SET SEBAGAI CLIENT

// LINK WEB YANG BERISI DATA PERANGKAT

**String** LinkLampu1 = "http://" +**String**(host)+ "/Ruang-1/status/datalampu1.php";

httpLampu1.begin(client,LinkLampu1); // MENGAKSES LINK WEB

httpLampu1.GET(); // MENGAMBIL DATA PERANGKAT DI WEB

**String** nilaiLampu1 = httpLampu1.getString(); // MERUBAH DATA PERANGKAT WEB MENJADI HURUF

Serial.println(nilaiLampu1); // MENCETAK DATA PERANGKAT WEB DI SERIAL MONITOR

httpLampu1.end(); // MENUTUP LINK WEB

digitalWrite(pinLampu1, nilaiLampu1.toInt()); // MANGAMBIL DATA PERANGKAT DI WEB LALU SET STATUS PIN PERANGKAT

HTTPClient httpLampu2;

**String** LinkLampu2 = "http://" +**String**(host)+ "/Ruang-1/status/datalampu2.php";

httpLampu2.begin(client,LinkLampu2);

httpLampu2.GET();

**String** nilaiLampu2 = httpLampu2.getString();

Serial.println(nilaiLampu2);

httpLampu2.end();

digitalWrite(pinLampu2, nilaiLampu2.toInt());

HTTPClient httpAc;

**String** LinkAc = "http://" +**String**(host)+ "/Ruang-1/status/dataac.php";

httpAc.begin(client,LinkAc);

httpAc.GET();

**String** nilaiAc = httpAc.getString();

Serial.println(nilaiAc);

httpAc.end();

digitalWrite(pinAc, nilaiAc.toInt());

HTTPClient httpKipas;

**String** LinkKipas = "http://" +**String**(host)+ "/Ruang-1/status/datakipas.php";

httpKipas.begin(client,LinkKipas);

httpKipas.GET();

**String** nilaiKipas = httpKipas.getString();

Serial.println(nilaiKipas);

httpKipas.end();

digitalWrite(pinKipas, nilaiKipas.toInt());

HTTPClient httpProyektor;

**String** LinkProyektor = "http://" +**String**(host)+ "/Ruang-1/status/dataproyektor.php";

httpProyektor.begin(client,LinkProyektor);

httpProyektor.GET();

**String** nilaiProyektor = httpProyektor.getString();

Serial.println(nilaiProyektor);

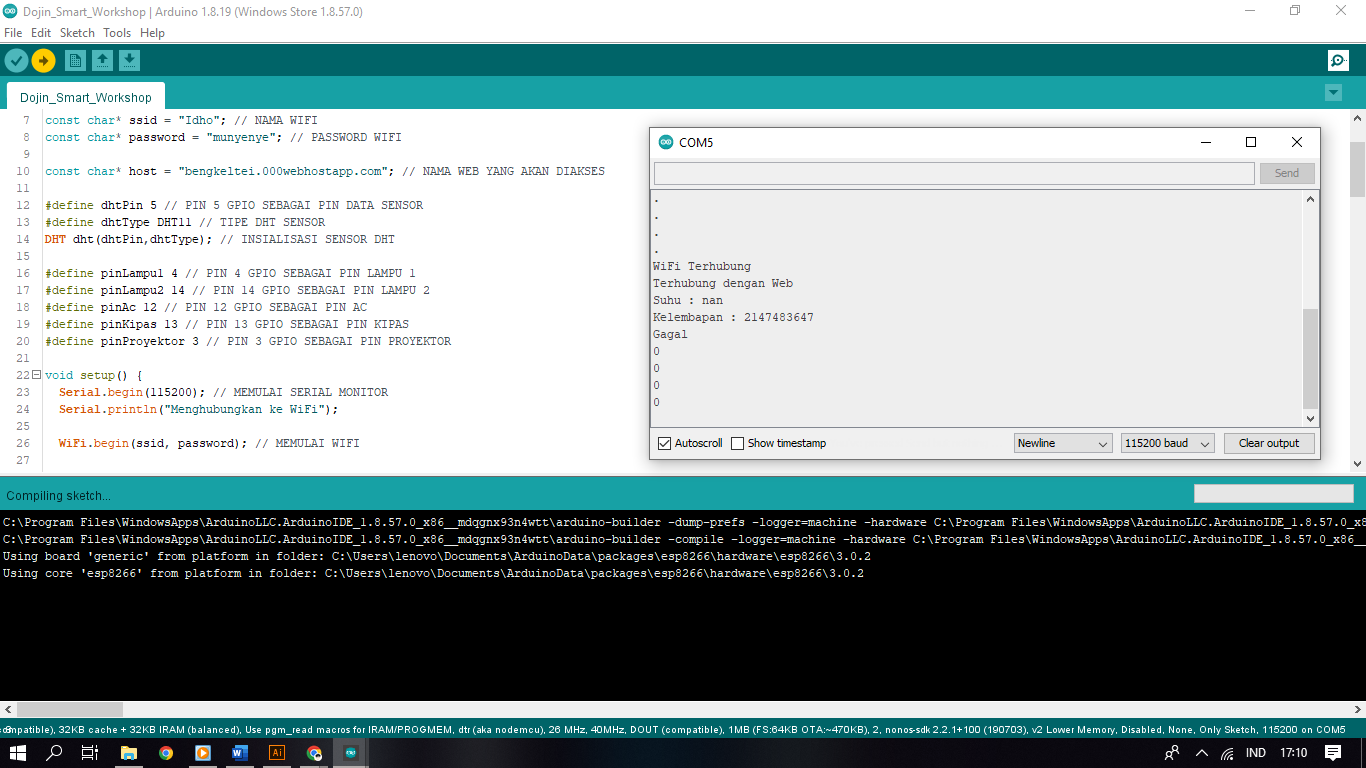
httpProyektor.end();

digitalWrite(pinProyektor, nilaiProyektor.toInt());

delay(1000); // JEDA 1 DETIK UNTUK MENJALANKAN PROGRAM

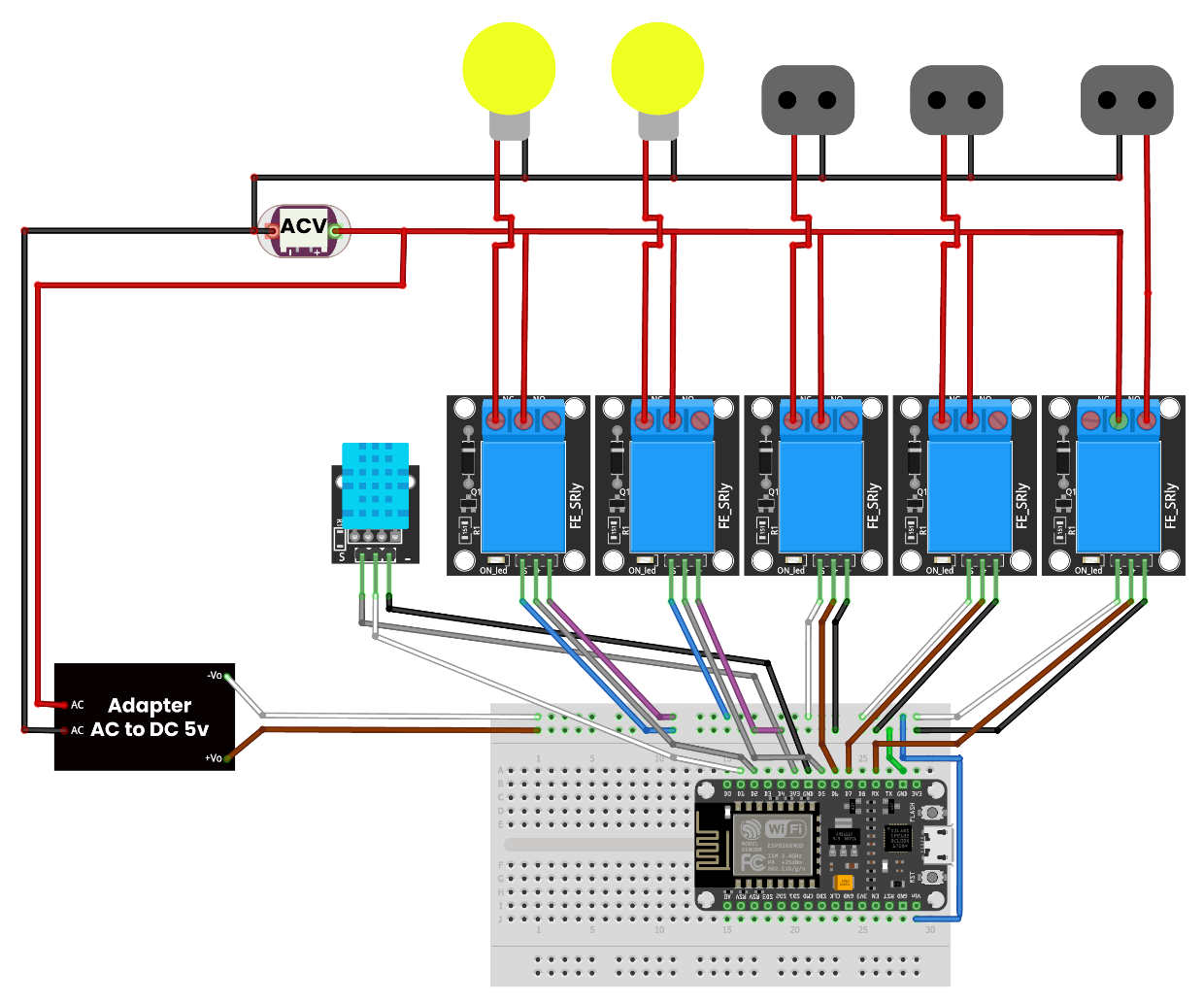
}

1. Atur port lalu Upload Program ke NodeMCU ESP8266 menggunakan kabel USB.



1. Perangkat Keras

Rangkaian Alat



Pasangkan NodeMCU ESP8266 ke Bread board

Sambungkan sumber tegangan AC ke Adapter 5v lalu sambungkan pin + - ke Bread board

Sensor DHT11

Sambungkan pin + DHT ke pin 3.3V ESP8266

Sambungkan pin – DHT ke pin GND ESP8266

Sambungkan pin OUT DHT ke pin D1(GPIO 5) ESP 8266

Relay

Sambungkan COM relay ke sumber listrik AC

Sambungkan semua pin VCC relay ke + Adapter

Sambungkan semua pin GND relay ke – Adapter

Sambungkan pin relay lampu 1 ke pin D2(GPIO 4) ESP 8266

Sambungkan NC relay lampu 1 ke lampu 1

Sambungkan pin relay lampu 2 ke pin D5(GPIO 14) ESP 8266

Sambungkan NC relay lampu 2 ke lampu 2

Sambungkan pin relay Ac ke pin D6(GPIO 12) ESP 8266

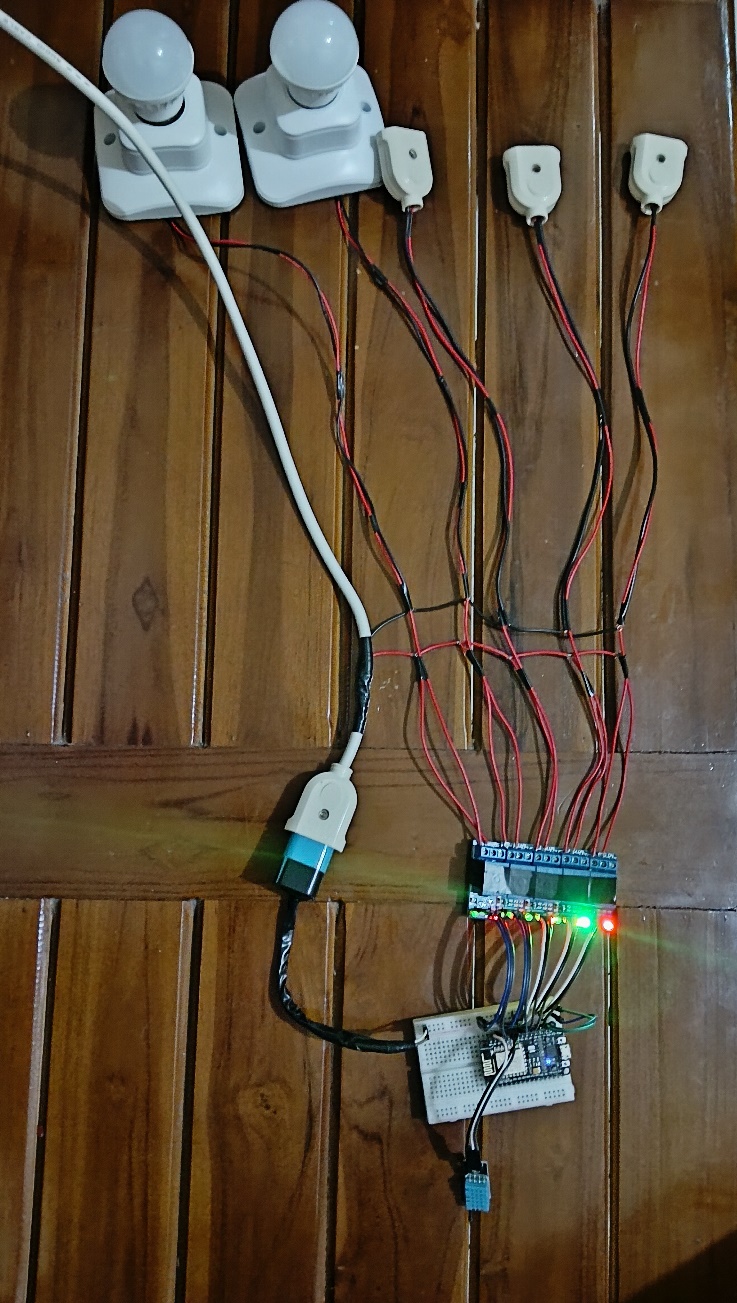
Sambungkan NC relay Ac ke colokan Ac

Sambungkan pin relay Kipas ke pin D7(GPIO 13) ESP 8266

Sambungkan NC relay Kipas ke colokan Kipas

Sambungkan pin relay Proyektor ke pin RX(GPIO 3) ESP 8266

Sambungkan NC relay Proyektor ke colokan Proyektor



# BAB IV

# HASIL PEMBUATAN