

**LAPORAN AKHIR**  
**RANCANG BANGUN IOT CONTROLLING RELAY MENGGUNAKAN**  
**PROTOKOL HTTP DENGAN SERVER 000WEBHOST**

Dosen Pengampu : Sayyidul Aulia Alamsyah, S.T, M.T.



Disusun oleh : Kelompok 02

Nur Faiz Haqqoni      (20050874077)

Ikmal Mughni K.      (21050874002)

Rosi Fadila Mey S.      (21050874034)

Teknik Elektro 2021 B

(Konsentrasi Elektronika)

**S1 TEKNIK ELEKTRO**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS NEGERI SURABAYA**

**2023**

# **BAB I**

## **PENDAHULUAN**

### **A. Latar belakang**

Pembelajaran mengontrol relay menggunakan NodeMCU dengan 000webhost sebagai server IoT memiliki latar belakang yang sangat relevan dalam konteks perkembangan Internet of Things (IoT). Pertama-tama, NodeMCU menjadi pilihan yang sangat baik sebagai mikrokontroler untuk proyek-proyek IoT karena kehandalannya, ukurannya yang kecil, dan kemampuannya untuk terhubung ke jaringan WiFi. Kemampuan ini memungkinkan pengguna untuk mengontrol perangkat secara nirkabel, membuka potensi otomatisasi dan pengendalian perangkat yang lebih efisien dalam kehidupan sehari-hari.

Selanjutnya, penerapan konsep pengendalian relay melalui web server 000webhost memberikan pengalaman praktis dalam mendeploy aplikasi IoT ke cloud. Dengan menggunakan web server sebagai platform penyimpanan dan pengontrolan, seseorang dapat mengakses dan mengontrol perangkat secara online dari berbagai lokasi, mengatasi batasan geografis. Ini tidak hanya meningkatkan keterampilan praktis pengguna dalam implementasi IoT, tetapi juga menciptakan dasar untuk pengembangan proyek-proyek IoT yang lebih kompleks.

Pembelajaran ini juga mendukung penerapan prinsip-prinsip dasar IoT, seperti pengumpulan data, pengiriman data melalui jaringan, dan respons terhadap perintah dari pengguna melalui antarmuka web. Dengan demikian, proyek ini tidak hanya memenuhi kebutuhan pengendalian perangkat, tetapi juga memberikan pemahaman yang holistik tentang konsep-konsep dasar yang berkaitan dengan Internet of Things. Kesempatan ini dapat menjadi dasar bagi mahasiswa atau pengembang pemula untuk memajukan kemampuan mereka dalam memahami dan mengaplikasikan teknologi IoT secara lebih luas.

### **B. Tujuan**

1. Peningkatan Pemahaman Konsep Dasar IoT
2. Meningkatkan Keterampilan Kontrol Perangkat dengan NodeMCU
3. Memahami Integrasi dengan Web Server 000webhost

## **BAB II**

### **TINJAUAN PUSTAKA**

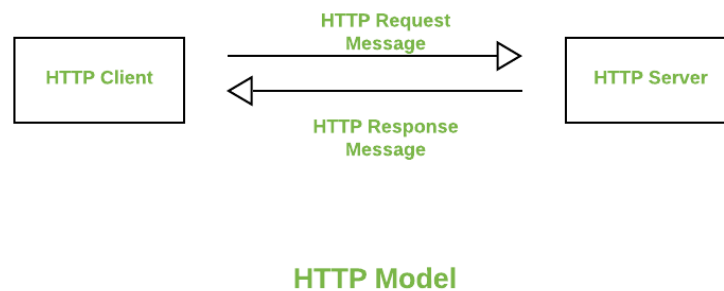
#### **A. Kajian pustaka**

##### **1. Internet of Things**

Internet of thing (IoT) merupakan suatu konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus [1]. Internet of thing (IoT) bisa dimanfaatkan pada gedung untuk mengendalikan peralatan elektronik seperti lampu ruangan yang dapat dioperasikan dari jarak jauh melalui jaringan komputer, tidak dapat dipungkiri kemajuan teknologi yang sedemikian cepat harus bisa dimanfaatkan, dipelajari serta diterapkan dalam kehidupan sehari-hari. Contohnya adalah perkembangan teknologi yang bisa dimanfaatkan dari adanya koneksi internet ini bisa mengakses peralatan elektronik seperti lampu ruangan yang dapat dioperasikan dengan cara online melalui mobile. Sehingga, dapat memudahkan pengguna memantau ataupun mengendalikan lampu kapanpun dan dimanapun dengan catatan di lokasi yang akan diterapkan teknologi kendali jarak jauh mempunyai jaringan internet yang memadai. Sistem kendali jarak jauh, memudahkan pengguna dalam mengontrol lampu gedung yang jaraknya cukup jauh lokasinya.

##### **2. Protokol HTTP**

Protokol Hypertext Transfer Protocol (HTTP) adalah protokol komunikasi yang digunakan untuk mentransfer data melalui World Wide Web. HTTP adalah bagian dari protokol aplikasi yang didefinisikan dalam model referensi OSI (Open Systems Interconnection) dan biasanya digunakan untuk mengakses dokumen-dokumen web. HTTP digunakan untuk mentransfer dokumen hypertext, seperti halaman web yang dapat berisi teks, gambar, audio, video, tautan, dan elemen-elemen multimedia lainnya.



Gambar 2.1 : Protokol HTTP

*(Sumber : [geeksforgeeks.org/difference-between-mqtt-and-http-protocols](https://www.geeksforgeeks.org/difference-between-mqtt-and-http-protocols) 2021)*

HTTP bekerja berdasarkan model klien-server. Klien adalah perangkat atau aplikasi yang meminta sumber daya, sedangkan server adalah tempat sumber daya tersebut disimpan atau diproses. HTTP adalah protokol stateless, yang berarti setiap permintaan dari klien ke server dianggap terpisah, dan server tidak menyimpan informasi tentang status sebelumnya. Ini membuat setiap permintaan dianggap sebagai entitas yang independen. HTTP mendefinisikan beberapa metode atau tindakan yang dapat dilakukan pada sumber daya. Metode umum termasuk GET (mengambil informasi), POST (mengirimkan data untuk diproses ke sumber daya yang ditentukan), PUT (mengganti atau membuat sumber daya), dan DELETE (menghapus sumber daya). HTTP menggunakan URL untuk merujuk ke sumber daya yang diminta. URL terdiri dari protokol (contohnya, "http://" atau "https://"), nama domain, jalur sumber daya, dan mungkin juga parameter tambahan. Untuk meningkatkan keamanan, HTTPS (HTTP Secure) menggunakan enkripsi SSL/TLS untuk melindungi data yang ditransfer antara klien dan server.

### 3. 000Webhost

000webhost adalah layanan hosting web gratis yang memberi pengguna kemampuan untuk meng-host situs web mereka tanpa biaya apa pun. Ini dikenal karena menawarkan solusi hosting gratis, menjadikannya pilihan populer bagi individu atau usaha kecil dengan anggaran terbatas. Berikut adalah beberapa poin penting tentang 000webhost: Hosting Gratis: Seperti disebutkan, 000webhost menyediakan layanan hosting web gratis, memungkinkan pengguna meng-host situs web mereka tanpa perlu pembayaran. Pembuat Situs Web: Seringkali

dilengkapi dengan pembuat situs web yang memungkinkan pengguna membuat dan mendesain situs web mereka dengan mudah tanpa pengetahuan pengkodean yang luas.



Gambar 2.2 : 000webhost

*(Sumber : [blogs.powercode.id/bagaimana-cara-untuk-menghosting-di-000webhost](https://blogs.powercode.id/bagaimana-cara-untuk-menghosting-di-000webhost) 2023)*

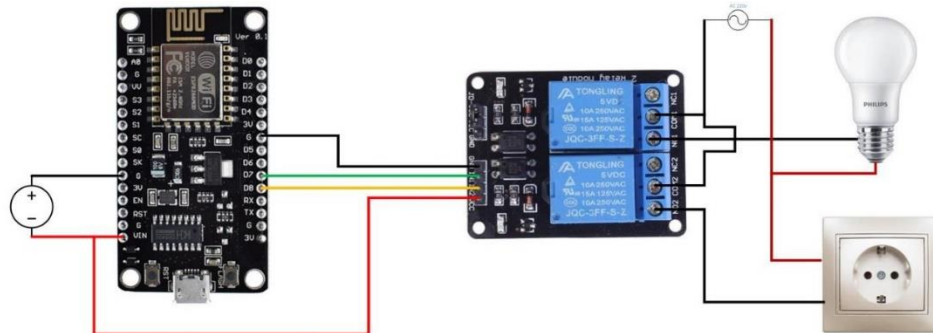
**Sumber Daya Terbatas:** Karena ini adalah layanan hosting gratis, terdapat batasan sumber daya seperti ruang disk, bandwidth, dan fitur. Hal ini dapat menjadi kelemahan bagi pengguna dengan situs web yang besar atau memiliki lalu lintas tinggi. **Subdomain:** Pengguna biasanya mendapatkan subdomain di bawah domain 000webhost untuk situs web mereka, kecuali mereka memilih untuk menggunakan nama domain mereka sendiri.

**Iklan:** Layanan hosting gratis mungkin menampilkan iklan di situs web pengguna sebagai cara untuk menghasilkan pendapatan bagi penyedia hosting. Ini adalah tradeoff umum untuk model hosting gratis. Penting untuk dicatat bahwa meskipun 000webhost adalah pilihan yang tepat bagi mereka yang mencari hosting gratis, 000webhost mungkin tidak cocok untuk situs web yang lebih besar atau kompleks yang memerlukan sumber daya dan fitur yang signifikan. Pengguna harus hati-hati mempertimbangkan kebutuhan spesifik mereka dan batasan layanan hosting gratis sebelum memilih penyedia hosting.

## BAB III

### HASIL DAN PEMBAHASAN

#### A. Implementasi



Gambar 3.1 : Skema rangkaian alat

(Sumber : Dokumentasi pribadi 2023)

Berikut merupakan konfigurasi *wiring* pada komponen aktuator relay dan output relay menggunakan board NodeMCU dan protokol HTTP.

Tabel 3.1 Konfigurasi pin NodeMCU

Konfigurasi pin NodeMCU	
Vin	5V
GND	GND
RX	In1 Relay
TX	In2 Relay

Tabel 3.2 Konfigurasi pin Relay with optocoupler

Konfigurasi pin Relay optocoupler	
VCC	Vin NodeMCU
GND	GND
In1 Relay	RX
In2 Relay	TX

#### B. Cara kerja

000webhost adalah sebuah penyedia jasa layanan server yang memiliki domain dan hosting tanpa membayar. Server Apache ini berbasis PHP dengan menggunakan database mySQL. Server bekerja menggunakan perintah GET pada address /espoutput-

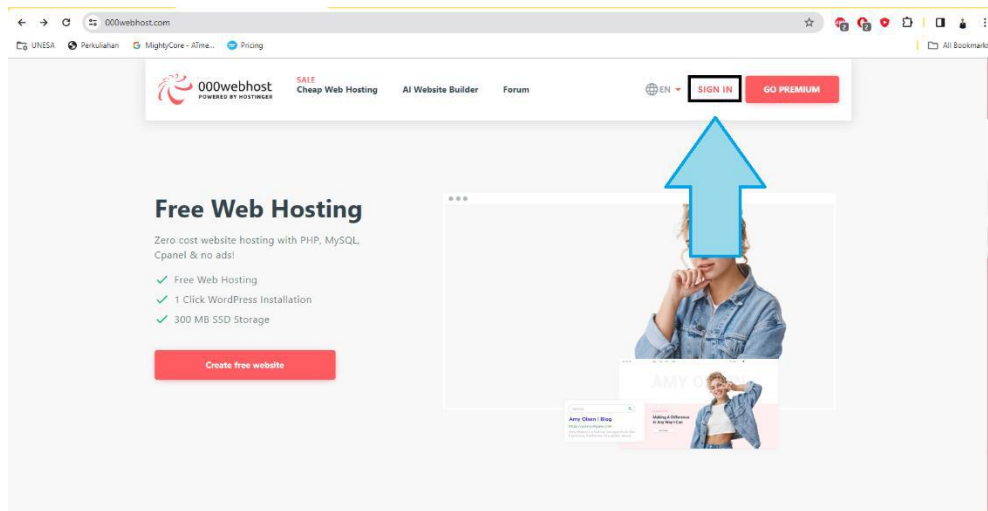
action.php lalu PHP bekerja dengan cara mengambil data pada database mySQL lalu menggabungkannya menjadi data JSON. Pada arduino menggunakan libraby ESP8266 http client, arduino Json dan ESP8266 Wifi multi yang menggunakan metode penerimaan data Json dari 000Webhost dengan algoritma Arduino mengirim perintah get ke 000webhost lalu menguraikan data Json yang diterima dan mengubah tipe data Json menjadi integer. Jadi data integer tersebut yang akan digunakan untuk controlling relay.

### C. Tutorial

1. Berikut adalah tutorial untuk upload program arduino menggunakan arduino IDE :
  - a. Menyalin program yang digunakan untuk controlling relay dengan 000webhost menggunakan protokol HTTP pada lampiran program HTTPS\_ESP32\_GET\_Request\_JSON.ino dibawah.
  - b. Menginstal library yang dibutuhkan diantaranya :
    - ESP8266HTTPClient.h
    - WiFiClient.h
    - ESP8266WiFi.h
    - Arduino\_JSON.h
    - <ESP8266WiFiMulti.h
  - c. Menyesuaikan SSID, Password, server name, interval pengiriman data, dan board ID pada program arduino Webhost.



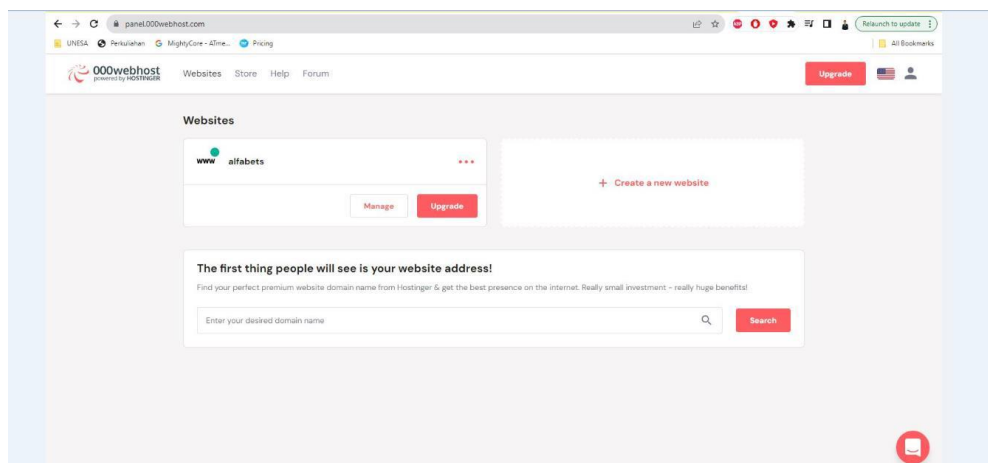
- d. mengCompile program dan mengupload program ke board NodeMCU menggunakan kabel MicroUSB.
2. Berikut adalah tutorial untuk controlling relay menggunakan protokol http dengan web server 000webhost :
  - a. Buka url 000webhost dengan menginputkan laman <https://000webhost.com> pada kolom pencarian browser.



Gambar 3.2 : URL 000webhost

(Sumber : Dokumentasi pribadi 2023)

- b. Daftar akun terlebih dahulu pada website 000webhost dan ikuti instruksi hingga masuk pada cPanel webhost.

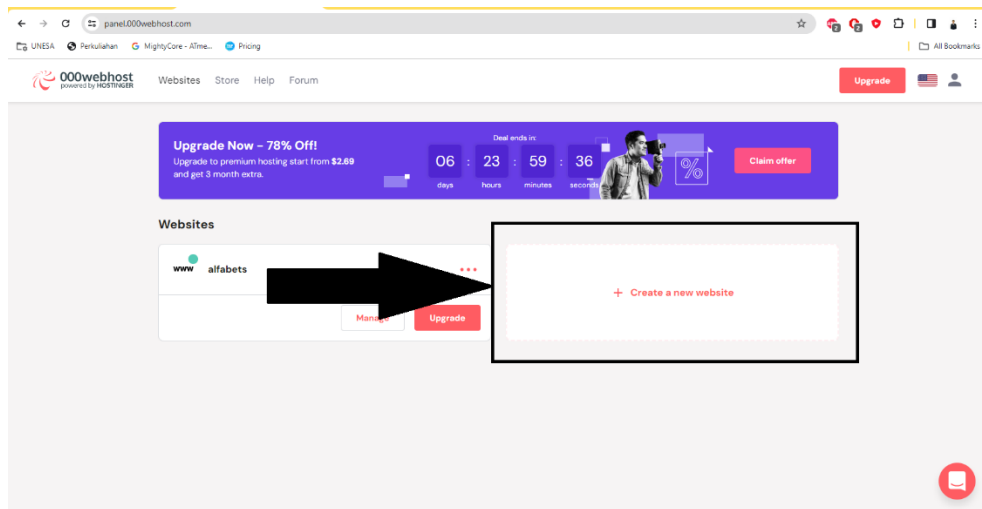


Gambar 3.3 : Cpanel 000webhost

(Sumber : Dokumentasi pribadi 2023)

- c. Masuk pada c panel, lalu klik create new website.

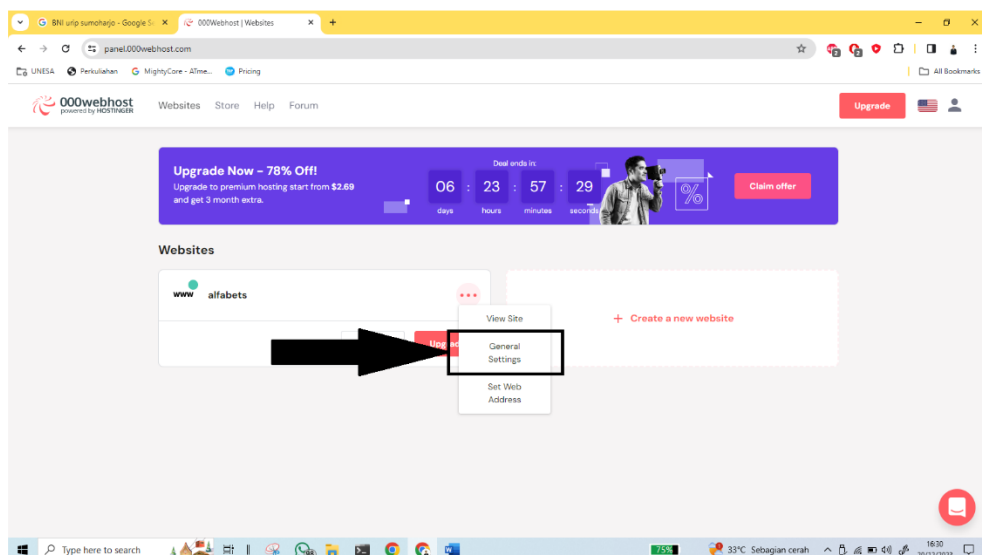




Gambar 3.4 : create new website cPanel

(Sumber : Dokumentasi pribadi 2023)

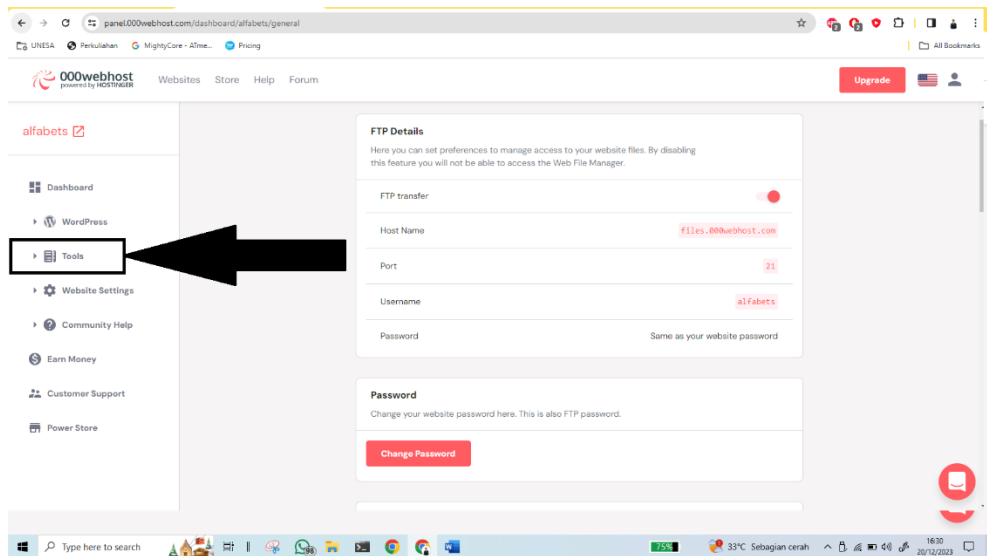
- d. Klik titik 3 pada website yang telah dibuat lalu pergi ke general settings.



Gambar 3.5 : general settings toggle 000webhost

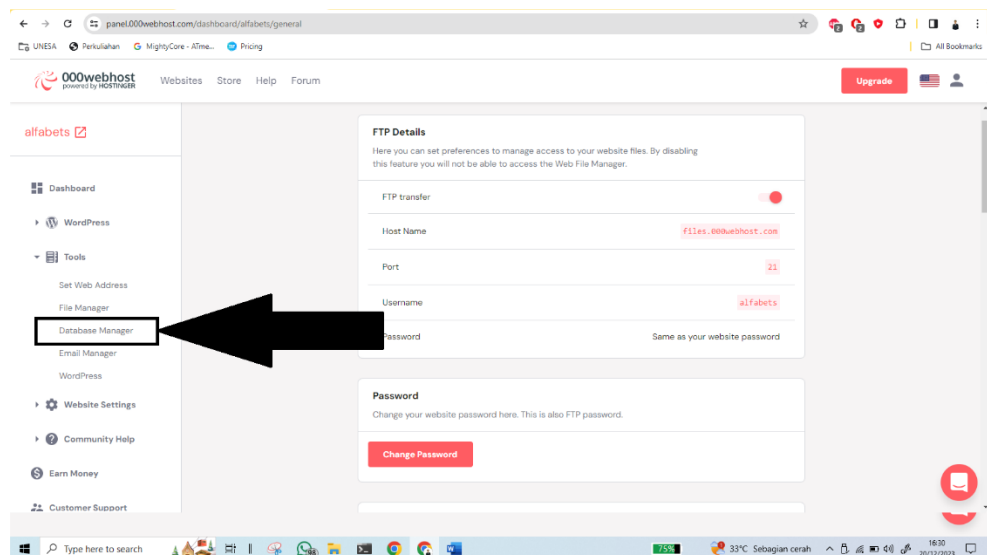
(Sumber : Dokumentasi pribadi 2023)

Setelah pada database settings, klik tools untuk menuju ke database manager.



Gambar 3.6 : general settings 000webhost

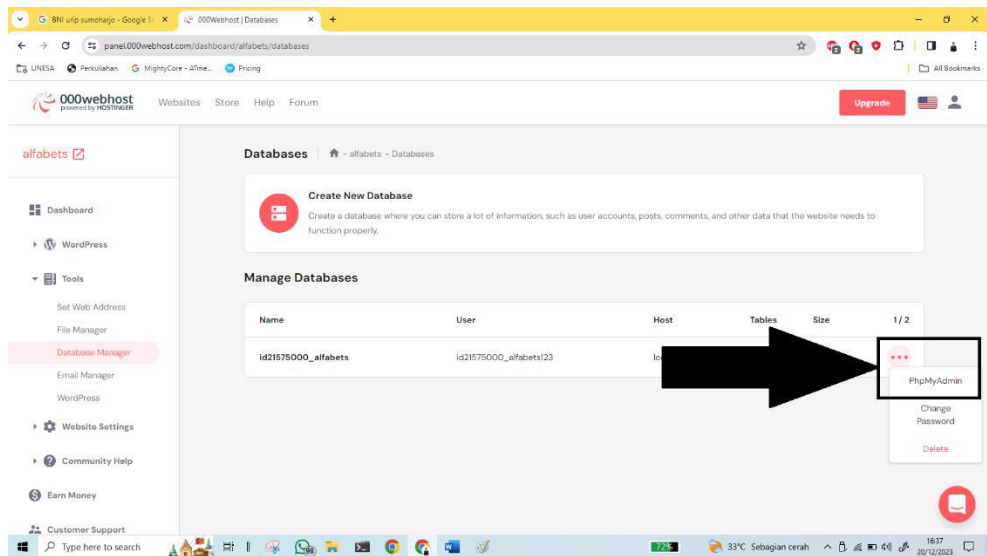
(Sumber : Dokumentasi pribadi 2023)



Gambar 3.7 : database manager toggle 000webhost

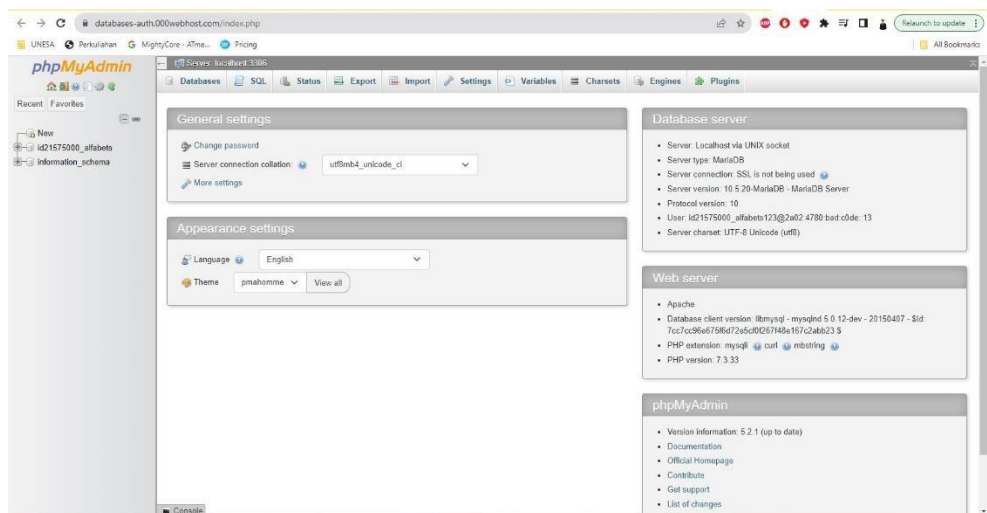
(Sumber : Dokumentasi pribadi 2023)

- e. Lalu masuk pada manage database klik titik tiga dan masuk ke php myadmin



Gambar 3.8 : php myAdmin toggle 000webhost

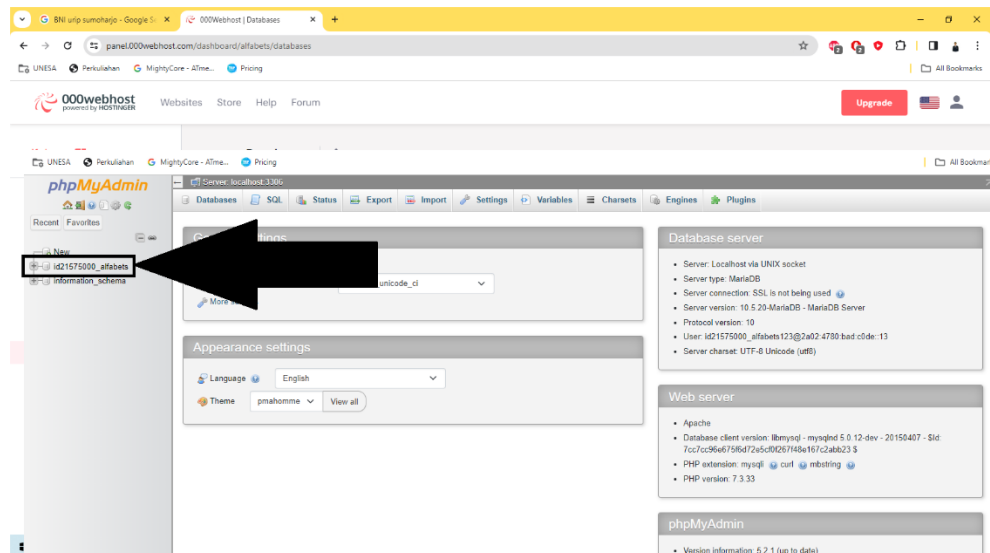
(Sumber : Dokumentasi pribadi 2023)



Gambar 3.9 : php myAdmin dashboard

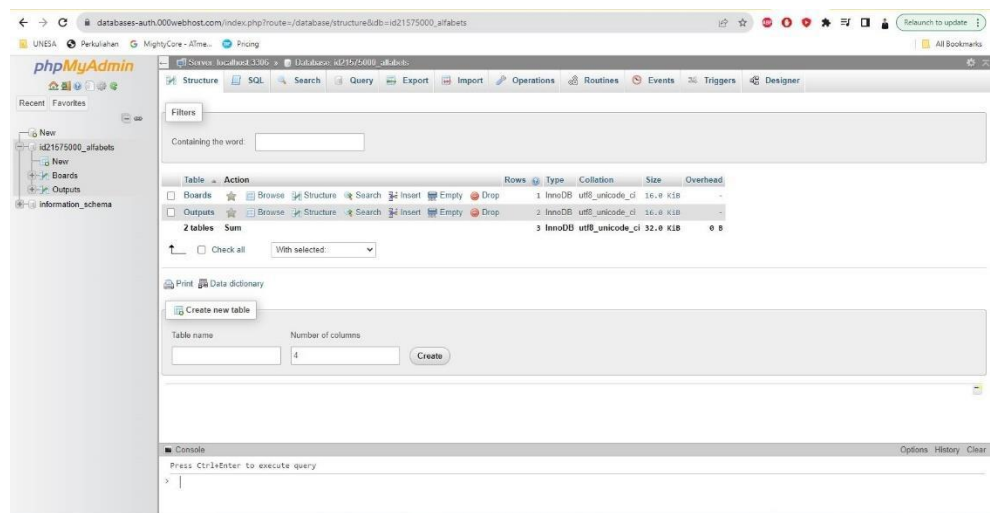
(Sumber : Dokumentasi pribadi 2023)

- f. Pada php myadmin klik id dan nama website pada bagian sebelah kiri lalu klik console pada bagian bawah.



Gambar 3.10 : id php myAdmin 000webhost

(Sumber : Dokumentasi pribadi 2023)



Gambar 3.11 : dashboard id dan console php myAdmin 000webhost

(Sumber : Dokumentasi pribadi 2023)

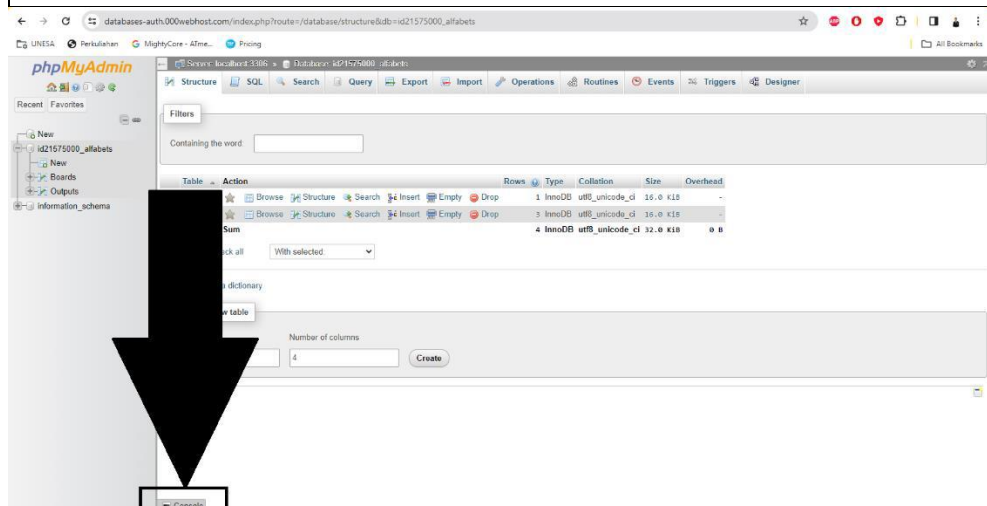
- g. Lalu copy paste program SQL berikut untuk menambahkan tabel pada SQL. Berikut adalah programnya :

```
CREATE TABLE Outputs (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(64),
    board INT(6),
    gpio INT(6),
```

```

state INT(6)
);
INSERT INTO Outputs(name, board, gpio, state) VALUES ("Built-in
LED", 1, 2, 0);
CREATE TABLE Boards (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    board INT(6),
    last_request TIMESTAMP DEFAULT CURRENT_TIMESTAMP
ON
UPDATE CURRENT_TIMESTAMP
);
INSERT INTO Boards(board) VALUES (1);

```



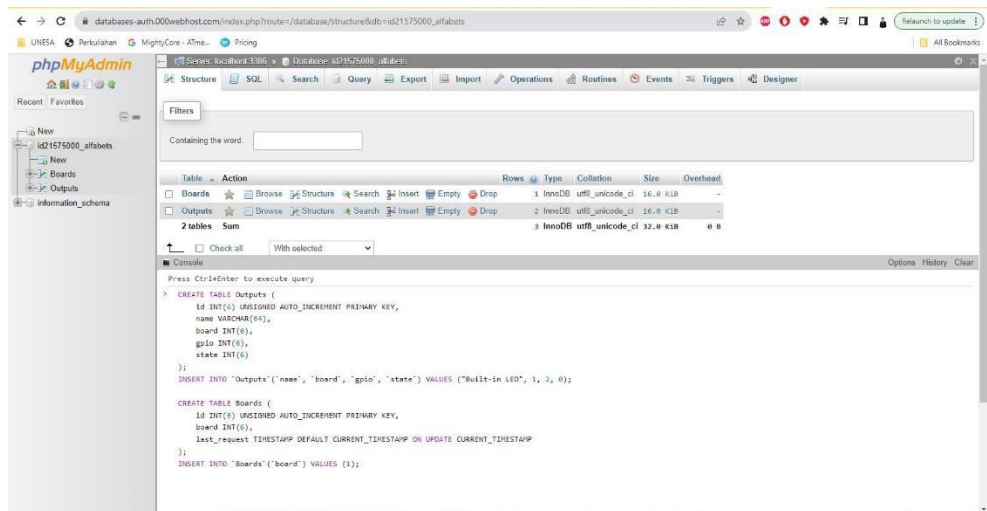
Gambar 3.12 : console toggle php myAdmin 000webhost

(Sumber : Dokumentasi pribadi 2023)



Gambar 3.13 : console php myAdmin 000webhost

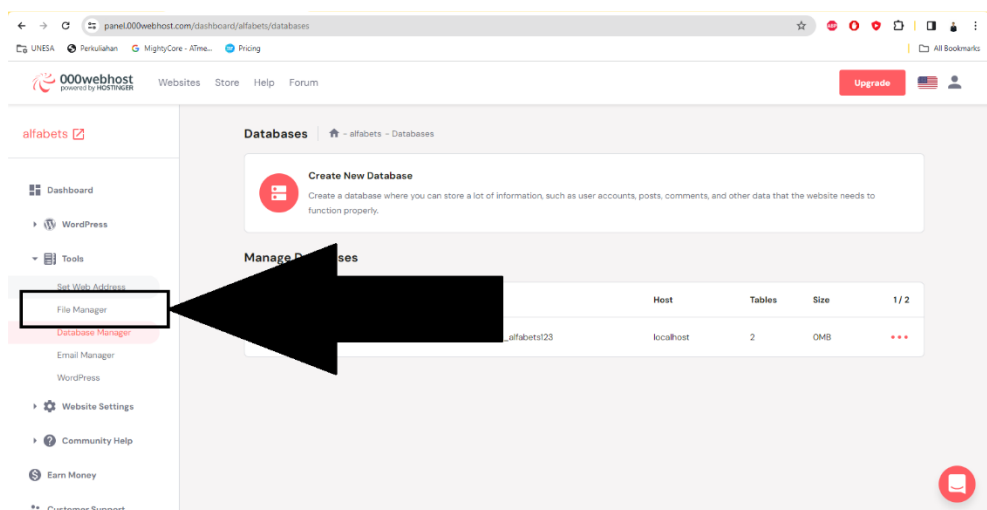
(Sumber : Dokumentasi pribadi 2023)



Gambar 3.14 : console code php myAdmin 000webhost

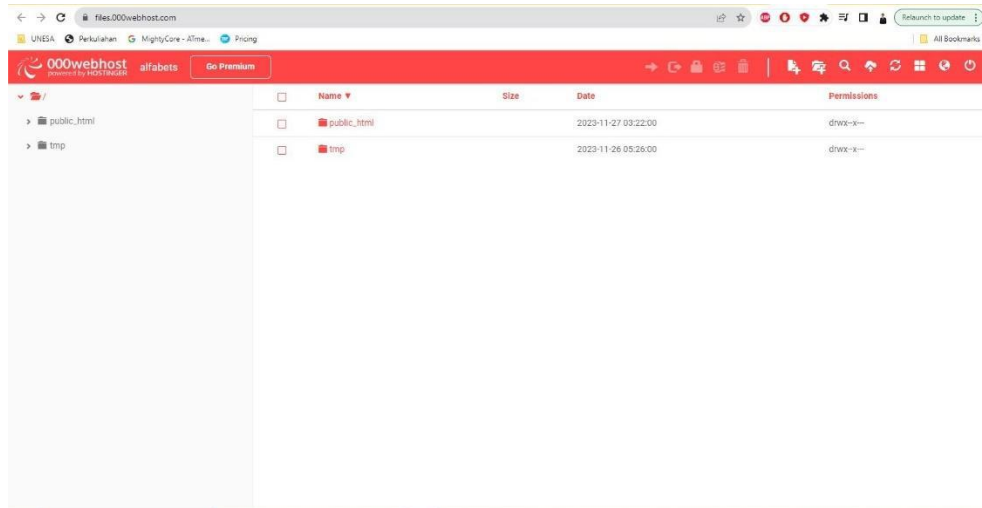
(Sumber : Dokumentasi pribadi 2023)

- h. Lalu kembali ke website 000webhost, masuk ke file manager.



Gambar 3.15 : file manager 000webhost

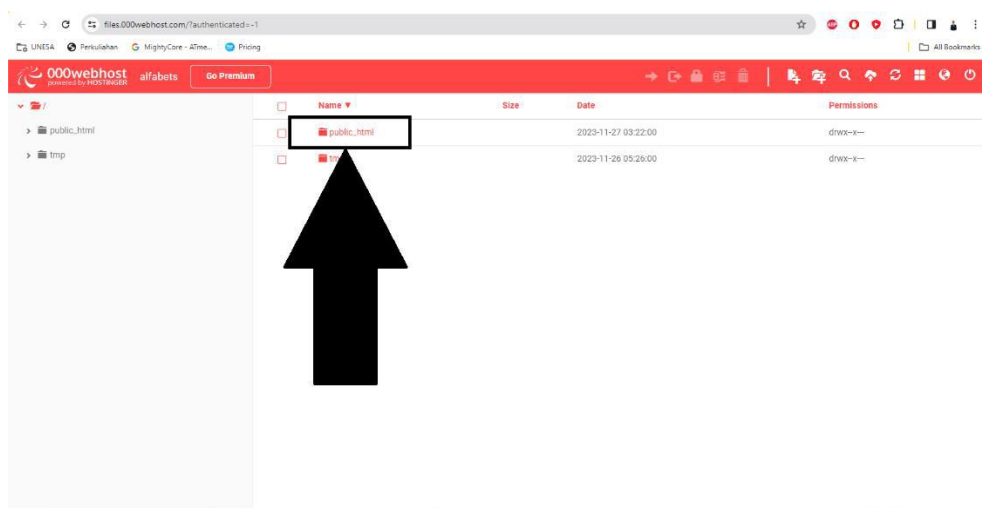
(Sumber : Dokumentasi pribadi 2023)



Gambar 3.16 : dashboard file manager 000webhost

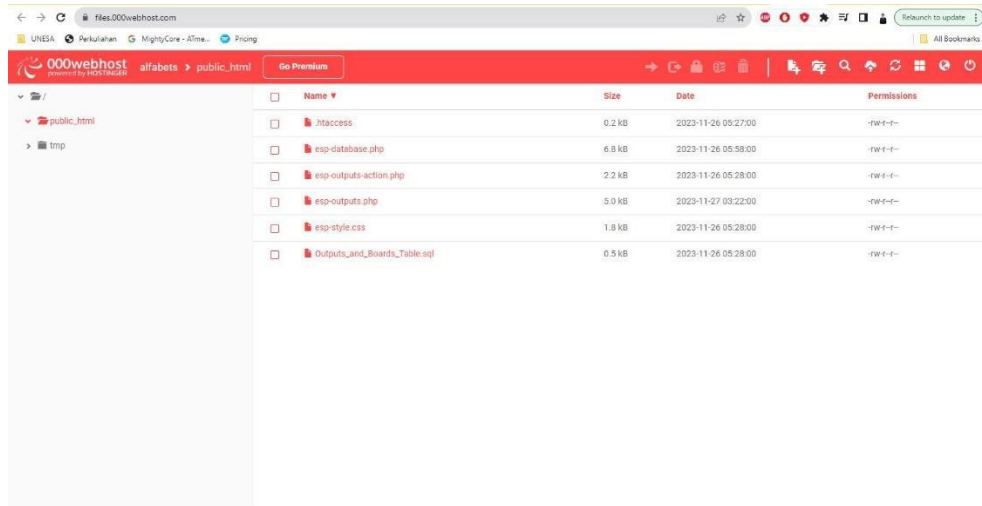
(Sumber : Dokumentasi pribadi 2023)

- i. Klik Public html pada file manager dan upload program berikut pada lampiran.
  - Esp-database.php
  - Esp-outputs-action.php
  - Esp-outputs.php
  - Esp-style.css
  - Outputs\_and\_Boards\_table.sql



Gambar 3.17 : folder public\_html 000webhost

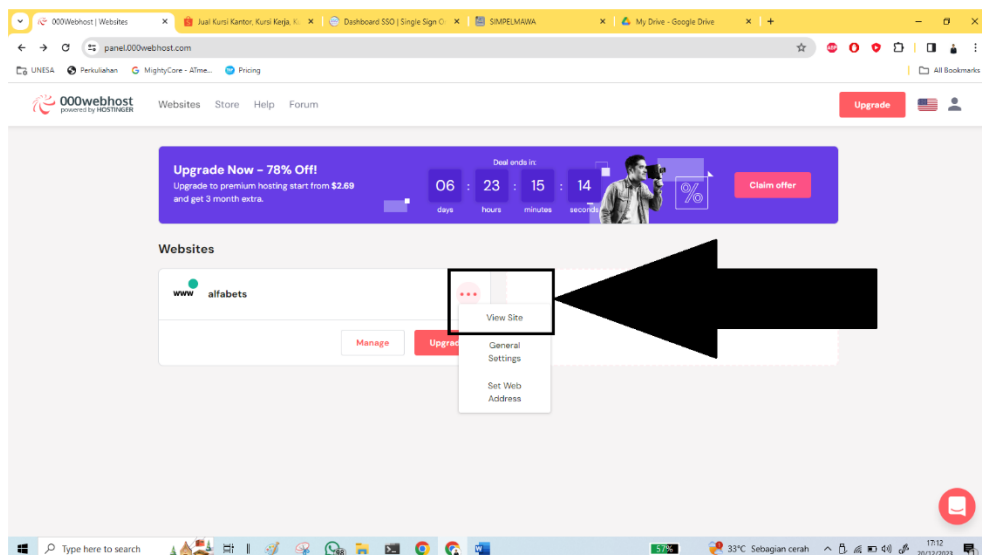
(Sumber : Dokumentasi pribadi 2023)



Gambar 3.18 : isi folerr public\_html 000webhost

(Sumber : Dokumentasi pribadi 2023)

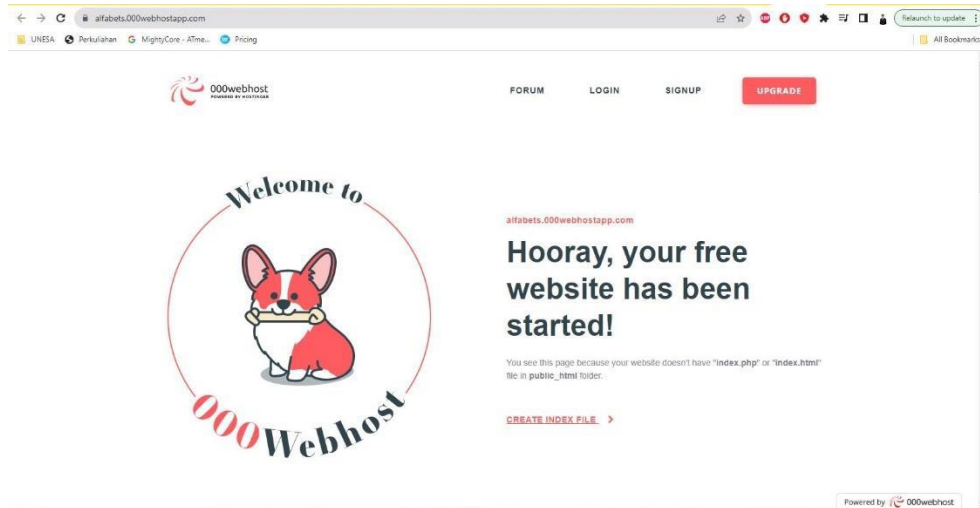
- j. Lalu kembali ke c panel, klik titik tiga lalu pergi ke view site.



Gambar 3.18 : view site cPanel 000webhost

(Sumber : Dokumentasi pribadi 2023)

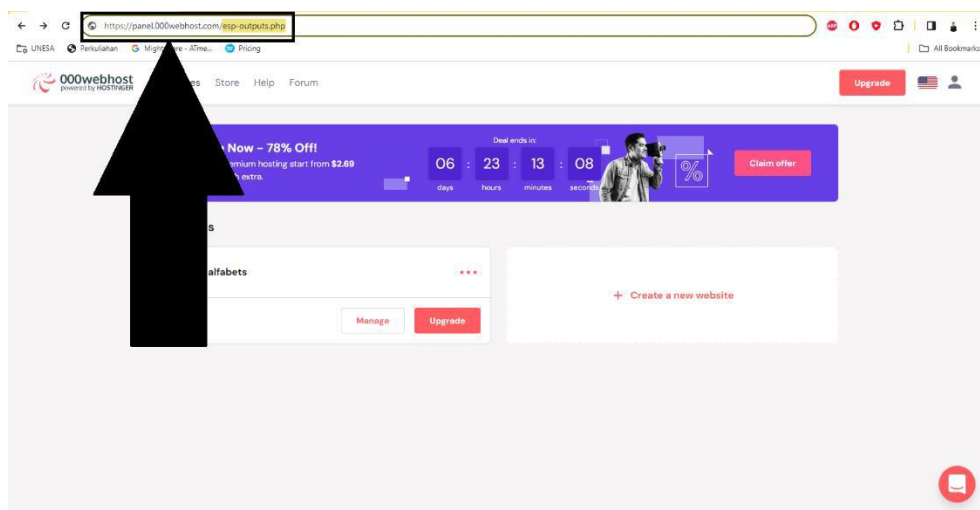




Gambar 3.19 : dashboard website 000webhost

(Sumber : Dokumentasi pribadi 2023)

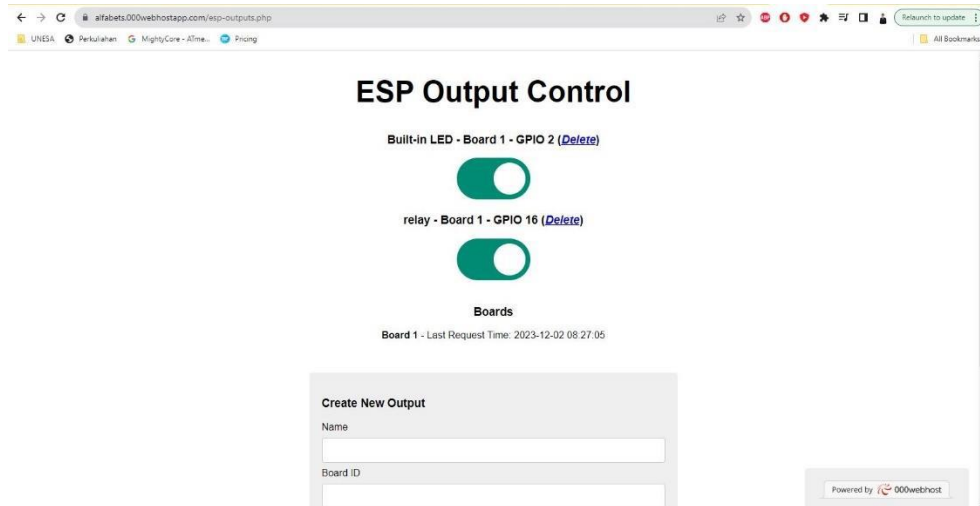
- k. Lalu tambahkan alamat URL dengan mengetikkan esp-outputs.php



Gambar 3.20 : penambahan *esp-outputs.php* pada url dashboard 000webhost

(Sumber : Dokumentasi pribadi 2023)

1. Masuk pada dashboard Iot yang telah dibuat meliputi ; judul, toggle, create new Output.



Gambar 3.21 : dashboard final IoT hosting 000webhost

(Sumber : Dokumentasi pribadi 2023)

- m. Untuk menambah board dan GPIO pada create new output, tambahkan nama board ID, GPIO dan initial GPIO State lalu klik new output.

### Create New Output

Name

Board ID

GPIO Number

Initial GPIO State

0 = OFF

Create Output

**Note:** in some devices, you might need to refresh the page to see your newly created buttons or to remove deleted buttons.

Gambar 3.22 : output added toggle dashboard 000webhost

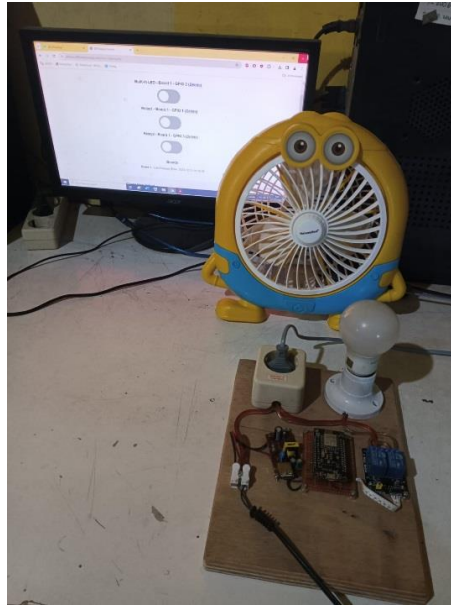
(Sumber : Dokumentasi pribadi 2023)

## BAB IV

### LAMPIRAN

#### A. Dokumentasi

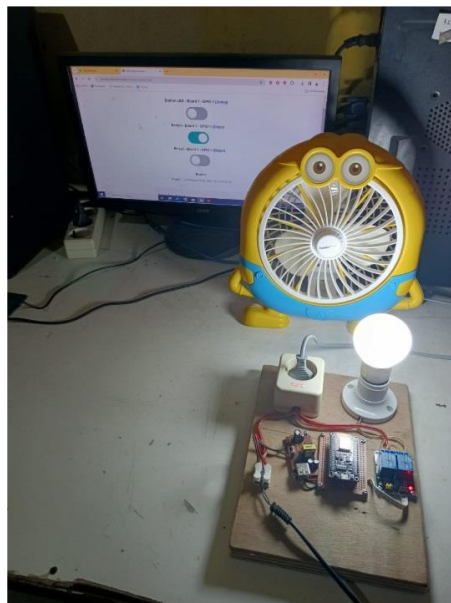
1. Toggle relay 1, 2 kondisi off



Gambar 4.1 : Toggle relay 1, 2 kondisi off

*(Sumber : Dokumentasi pribadi 2023)*

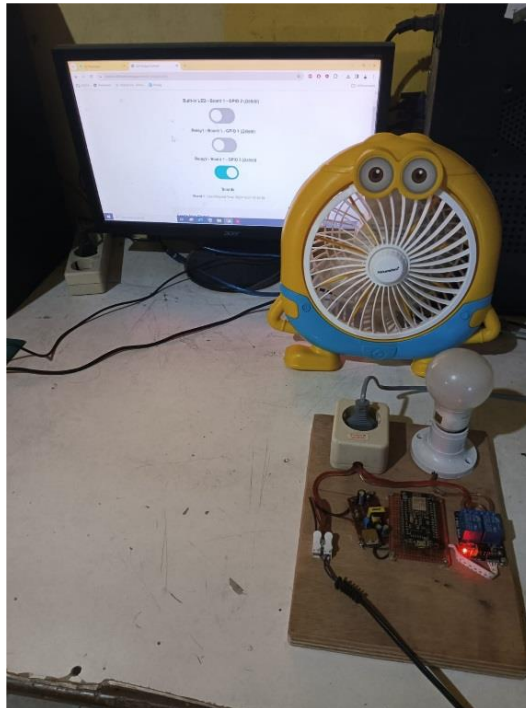
2. Toggle relay 1 off dan 2 on



Gambar 4.2 : Toggle relay 1 off dan 2 on

*(Sumber : Dokumentasi pribadi 2023)*

3. Toggle relay 1 on dan 2 off



Gambar 4.3 : Toggle relay 1 on dan 2 off

*(Sumber : Dokumentasi pribadi 2023)*

4. Toggle relay 1, 2 on



Gambar 4.4 : Toggle relay 1, 2 on

*(Sumber : Dokumentasi pribadi 2023)*

## 5. Source code

### 1. Esp-database.php

```
<?php
    $servername = "localhost";
    // Your Database name
    $dbname = "id21575000_alfabets";
    // Your Database user
    $username = "id21575000_alfabets123";
    // Your Database user password
    $password = "Alfabets+123";

    function createOutput($name, $board, $gpio, $state) {
        global $servername, $username, $password, $dbname;

        // Create connection
        $conn = new mysqli($servername, $username, $password,
            $dbname);
        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        $sql = "INSERT INTO Outputs (name, board, gpio, state)
        VALUES ('" . $name . "', '" . $board . "', '" . $gpio . "', '" . $state .
        "')";

        if ($conn->query($sql) === TRUE) {
            return "New output created successfully";
        }
        else {
            return "Error: " . $sql . "<br>" . $conn->error;
        }
    }
}
```

```

$conn->close();
}

function deleteOutput($id) {
    global $servername, $username, $password, $dbname;

    // Create connection
    $conn = new mysqli($servername, $username, $password,
$dbname);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "DELETE FROM Outputs WHERE id='". $id . "'";

    if ($conn->query($sql) === TRUE) {
        return "Output deleted successfully";
    }
    else {
        return "Error: " . $sql . "<br>" . $conn->error;
    }
    $conn->close();
}

function updateOutput($id, $state) {
    global $servername, $username, $password, $dbname;

    // Create connection
    $conn = new mysqli($servername, $username, $password,
$dbname);

    // Check connection
    if ($conn->connect_error) {

```

```

        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "UPDATE Outputs SET state='" . $state . "' WHERE id='".
    $id . "'";

    if ($conn->query($sql) === TRUE) {
        return "Output state updated successfully";
    }
    else {
        return "Error: " . $sql . "<br>" . $conn->error;
    }
    $conn->close();
}

function getAllOutputs() {
    global $servername, $username, $password, $dbname;

    // Create connection
    $conn = new mysqli($servername, $username, $password,
    $dbname);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT id, name, board, gpio, state FROM Outputs
    ORDER BY board";
    if ($result = $conn->query($sql)) {
        return $result;
    }
    else {
        return false;
    }
}

```

```

    }
    $conn->close();
}

function getAllOutputStates($board) {
    global $servername, $username, $password, $dbname;

    // Create connection
    $conn = new mysqli($servername, $username, $password,
$dbname);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT gpio, state FROM Outputs WHERE board=" .
$board . " ";
    if ($result = $conn->query($sql)) {
        return $result;
    }
    else {
        return false;
    }
    $conn->close();
}

function getOutputBoardById($id) {
    global $servername, $username, $password, $dbname;

    // Create connection
    $conn = new mysqli($servername, $username, $password,
$dbname);

    // Check connection

```



```

        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        $sql = "SELECT board FROM Outputs WHERE id='" . $id . "'";
        if ($result = $conn->query($sql)) {
            return $result;
        }
        else {
            return false;
        }
        $conn->close();
    }

function updateLastBoardTime($board) {
    global $servername, $username, $password, $dbname;

    // Create connection
    $conn = new mysqli($servername, $username, $password,
    $dbname);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "UPDATE Boards SET last_request=now() WHERE
board='" . $board . "'";

    if ($conn->query($sql) === TRUE) {
        return "Output state updated successfully";
    }
    else {
        return "Error: " . $sql . "<br>" . $conn->error;
    }
}

```

```

    }
    $conn->close();
}

function getAllBoards() {
    global $servername, $username, $password, $dbname;

    // Create connection
    $conn = new mysqli($servername, $username, $password,
$dbname);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "SELECT board, last_request FROM Boards ORDER BY
board";
    if ($result = $conn->query($sql)) {
        return $result;
    }
    else {
        return false;
    }
    $conn->close();
}

function getBoard($board) {
    global $servername, $username, $password, $dbname;

    // Create connection
    $conn = new mysqli($servername, $username, $password,
$dbname);

    // Check connection

```

```

        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        $sql = "SELECT board, last_request FROM Boards WHERE
board='" . $board . "'";
        if ($result = $conn->query($sql)) {
            return $result;
        }
        else {
            return false;
        }
        $conn->close();
    }

function createBoard($board) {
    global $servername, $username, $password, $dbname;

    // Create connection
    $conn = new mysqli($servername, $username, $password,
$dbname);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "INSERT INTO Boards (board) VALUES ('" . $board . "')";

    if ($conn->query($sql) === TRUE) {
        return "New board created successfully";
    }
    else {
        return "Error: " . $sql . "<br>" . $conn->error;
    }
}

```

```

    }
    $conn->close();
}

function deleteBoard($board) {
    global $servername, $username, $password, $dbname;

    // Create connection
    $conn = new mysqli($servername, $username, $password,
$dbname);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "DELETE FROM Boards WHERE board='". $board . "'";

    if ($conn->query($sql) === TRUE) {
        return "Board deleted successfully";
    }
    else {
        return "Error: " . $sql . "<br>" . $conn->error;
    }
    $conn->close();
}

?>

```

## 2. Esp-outputs-action.php

```

<?php
    include_once('esp-database.php');

    $action = $id = $name = $gpio = $state = "";

```

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $action = test_input($_POST["action"]);
    if ($action == "output_create") {
        $name = test_input($_POST["name"]);
        $board = test_input($_POST["board"]);
        $gpio = test_input($_POST["gpio"]);
        $state = test_input($_POST["state"]);
        $result = createOutput($name, $board, $gpio, $state);

        $result2 = getBoard($board);
        if (!$result2->fetch_assoc()) {
            createBoard($board);
        }
        echo $result;
    }
    else {
        echo "No data posted with HTTP POST.";
    }
}

if ($_SERVER["REQUEST_METHOD"] == "GET") {
    $action = test_input($_GET["action"]);
    if ($action == "outputs_state") {
        $board = test_input($_GET["board"]);
        $result = getAllOutputStates($board);
        if ($result) {
            while ($row = $result->fetch_assoc()) {
                $rows[$row["gpio"]] = $row["state"];
            }
        }
        echo json_encode($rows);
        $result = getBoard($board);
    }
}

```

```

        if($result->fetch_assoc()) {
            updateLastBoardTime($board);
        }
    }
    else if ($action == "output_update") {
        $id = test_input($_GET["id"]);
        $state = test_input($_GET["state"]);
        $result = updateOutput($id, $state);
        echo $result;
    }
    else if ($action == "output_delete") {
        $id = test_input($_GET["id"]);
        $board = getOutputBoardById($id);
        if ($row = $board->fetch_assoc()) {
            $board_id = $row["board"];
        }
        $result = deleteOutput($id);
        $result2 = getAllOutputStates($board_id);
        if(!$result2->fetch_assoc()) {
            deleteBoard($board_id);
        }
        echo $result;
    }
    else {
        echo "Invalid HTTP request.";
    }
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

```

```
}  
?>
```

### 3. Esp-outputs.php

```
<?php  
    include_once('esp-database.php');  
  
    $result = getAllOutputs();  
    $html_buttons = null;  
    if ($result) {  
        while ($row = $result->fetch_assoc()) {  
            if ($row["state"] == "1"){  
                $button_checked = "checked";  
            }  
            else {  
                $button_checked = "";  
            }  
            $html_buttons .= '<h3>' . $row["name"] . ' - Board '  
$row["board"] . ' - GPIO ' . $row["gpio"] . ' (<i><a  
onclick="deleteOutput(this)" href="javascript:void(0);" id="' .  
$row["id"] . '">Delete</a></i></h3><label class="switch"><input  
type="checkbox" onchange="updateOutput(this)" id="' . $row["id"] . '" '  
. $button_checked . '><span class="slider"></span></label>';  
        }  
    }  
  
    $result2 = getAllBoards();  
    $html_boards = null;  
    if ($result2) {  
        $html_boards .= '<h3>Boards</h3>';  
        while ($row = $result2->fetch_assoc()) {  
            $row_reading_time = $row["last_request"];
```

```

        // Uncomment to set timezone to - 1 hour (you can change 1 to
any number)

        //$row_reading_time = date("Y-m-d H:i:s",
strtotime("$row_reading_time - 1 hours"));

        // Uncomment to set timezone to + 4 hours (you can change 4 to
any number)

        //$row_reading_time = date("Y-m-d H:i:s",
strtotime("$row_reading_time + 7 hours"));

        $html_boards .= '<p><strong>Board ' . $row["board"] .
'</strong> - Last Request Time: ' . $row_reading_time . '</p>';
    }
}
?>

<!DOCTYPE HTML>
<html>
    <head><meta http-equiv="Content-Type" content="text/html;
charset=utf-8">

        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <link rel="stylesheet" type="text/css" href="esp-style.css">
        <title>ESP Output Control</title>
    </head>
    <body>
        <h2>ESP Output Control</h2>
        <?php echo $html_buttons; ?>
        <br><br>
        <?php echo $html_boards; ?>
        <br><br>
        <div><form onsubmit="return createOutput();">
            <h3>Create New Output</h3>

```



```

<label for="outputName">Name</label>
<input type="text" name="name" id="outputName"><br>
<label for="outputBoard">Board ID</label>
<input type="number" name="board" min="0" id="outputBoard">
<label for="outputGpio">GPIO Number</label>
<input type="number" name="gpio" min="0" id="outputGpio">
<label for="outputState">Initial GPIO State</label>
<select id="outputState" name="state">
  <option value="0">0 = OFF</option>
  <option value="1">1 = ON</option>
</select>
<input type="submit" value="Create Output">

<p><strong>Note:</strong> in some devices, you might need to
refresh the page to see your newly created buttons or to remove deleted
buttons.</p>
</form></div>

```

```

<script>
function updateOutput(element) {
  var xhr = new XMLHttpRequest();
  if(element.checked){
    xhr.open("GET", "esp-outputs-
action.php?action=output_update&id="+element.id+"&state=1", true);
  }
  else {
    xhr.open("GET", "esp-outputs-
action.php?action=output_update&id="+element.id+"&state=0", true);
  }
  xhr.send();
}

function deleteOutput(element) {
  var result = confirm("Want to delete this output?");

```

```

        if (result) {
            var xhr = new XMLHttpRequest();
            xhr.open("GET", "esp-outputs-
action.php?action=output_delete&id="+element.id, true);
            xhr.send();
            alert("Output deleted");
            setTimeout(function(){ window.location.reload(); });
        }
    }

    function createOutput(element) {
        var xhr = new XMLHttpRequest();
        xhr.open("POST", "esp-outputs-action.php", true);

        xhr.setRequestHeader("Content-Type", "application/x-www-
form-urlencoded");

        xhr.onreadystatechange = function() {
            if (this.readyState === XMLHttpRequest.DONE &&
this.status === 200) {
                alert("Output created");
                setTimeout(function(){ window.location.reload(); });
            }
        }

        var outputName =
document.getElementById("outputName").value;
        var outputBoard =
document.getElementById("outputBoard").value;
        var outputGpio =
document.getElementById("outputGpio").value;
        var outputState =
document.getElementById("outputState").value;

```

```
        var httpRequestData =  
        "action=output_create&name="+outputName+"&board="+outputBoard  
       +"&gpio="+outputGpio+"&state="+outputState;  
        xhr.send(httpRequestData);  
    }  
</script>  
</body>  
</html>
```

#### 4. Esp-style.css

```
html {  
    font-family: Arial;  
    display: inline-block;  
    text-align: center;  
}  
  
h2 {  
    font-size: 3.0rem;  
}  
  
body {  
    max-width: 600px;  
    margin: 0px auto;  
    padding-bottom: 25px;  
}  
  
.switch {  
    position: relative;  
    display: inline-block;  
    width: 120px;  
    height: 68px;  
}
```

```
.switch input {  
    display: none  
}
```

```
.slider {  
    position: absolute;  
    top: 0;  
    left: 0;  
    right: 0;  
    bottom: 0;  
    background-color: #949494;  
    border-radius: 34px;  
}
```

```
.slider:before {  
    position: absolute;  
    content: "";  
    height: 52px;  
    width: 52px;  
    left: 8px; bottom: 8px;  
    background-color: #fff;  
    -webkit-transition: .4s;  
    transition: .4s;  
    border-radius: 68px;  
}
```

```
input:checked+.slider {  
    background-color: #008B74;  
}
```

```
input:checked+.slider:before {  
    -webkit-transform: translateX(52px);  
}
```

```
-ms-transform: translateX(52px);
transform: translateX(52px);
}

input[type=text], input[type=number], select {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    display: inline-block;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}

input[type=submit] {
    width: 100%;
    background-color: #008B74;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

input[type=submit]:hover {
    background-color: #005a4c;
}

div {
    text-align: left;
    border-radius: 4px;
    background-color: #efefef;
```

```
padding: 20px;
}
```

#### 5. HTTPS\_ESP32\_GET\_Request\_JSON.ino

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <Arduino_JSON.h>
#include <ESP8266WiFiMulti.h>
ESP8266WiFiMulti WiFiMulti;
const char* ssid = "TP-Link_AFBC";
const char* password = "Penelitian2023";
const char* serverName = "http://alfabets.000webhostapp.com/esp-outputs-action.php?action=outputs_state&board=1";
const long interval = 5000;
unsigned long previousMillis = 0;

String outputsState;

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");
}

void loop() {
  unsigned long currentMillis = millis();
  if(currentMillis - previousMillis >= interval) {
```

```

if ((WiFiMulti.run() == WL_CONNECTED)) {
    outputsState = httpGETRequest(serverName);
    Serial.println(outputsState);
    JSONVar myObject = JSON.parse(outputsState);
    if (JSON.typeof(myObject) == "undefined") {
        Serial.println("Parsing input failed!");
        return;}
    Serial.print("JSON object = ");
    Serial.println(myObject);
    JSONVar keys = myObject.keys();
    for (int i = 0; i < keys.length(); i++) {
        JSONVar value = myObject[keys[i]];
        Serial.print("GPIO: ");
        Serial.print(keys[i]);
        Serial.print(" - SET to: ");
        Serial.println(value);
        pinMode(atoi(keys[i]), OUTPUT);
        int pinVal = atoi(value);
        int pinValv2;
        if(pinVal == 0){
            pinValv2 = 1;
        }else{
            pinValv2 = 0;}
        digitalWrite(atoi(keys[i]), pinValv2);}
    previousMillis = currentMillis;
} else {
    Serial.println("WiFi Disconnected");}}}

String httpGETRequest(const char* serverName) {

    WiFiClient client;
    HTTPClient http;
    http.begin(client, serverName);
    int httpResponseCode = http.GET();

```

```
String payload = "{}";  
if (httpResponseCode>0) {  
    Serial.print("HTTP Response code: ");  
    Serial.println(httpResponseCode);  
    payload = http.getString();  
} else {  
    Serial.print("Error code: ");  
    Serial.println(httpResponseCode);}  
http.end();  
return payload;  
}
```



## DAFTAR PUSTAKA

geeksforgeeks.org. (2021, 11 November). "Difference between MQTT and HTTP protocols". Dalam <https://www.geeksforgeeks.org/difference-between-mqtt-and-http-protocols/>. Diakses tanggal 25 Desember 2023.

Priyanto. Sendi Agus. (2023, 30 Maret). "Bagaimana cara untuk menghosting di 000webhost?". Dalam <https://blogs.powercode.id/bagaimana-cara-untuk-menghosting-di-000webhost/>. Diakses tanggal 25 Desember 2023.

Santos. Rui. (2020, 24 Januari). "Control ESP32 and ESP8266 GPIOs from Anywhere in the World". Dalam <https://randomnerdtutorials.com/control-esp32-esp8266-gpios-from-anywhere/>. Diakses tanggal 26 Desember 2023