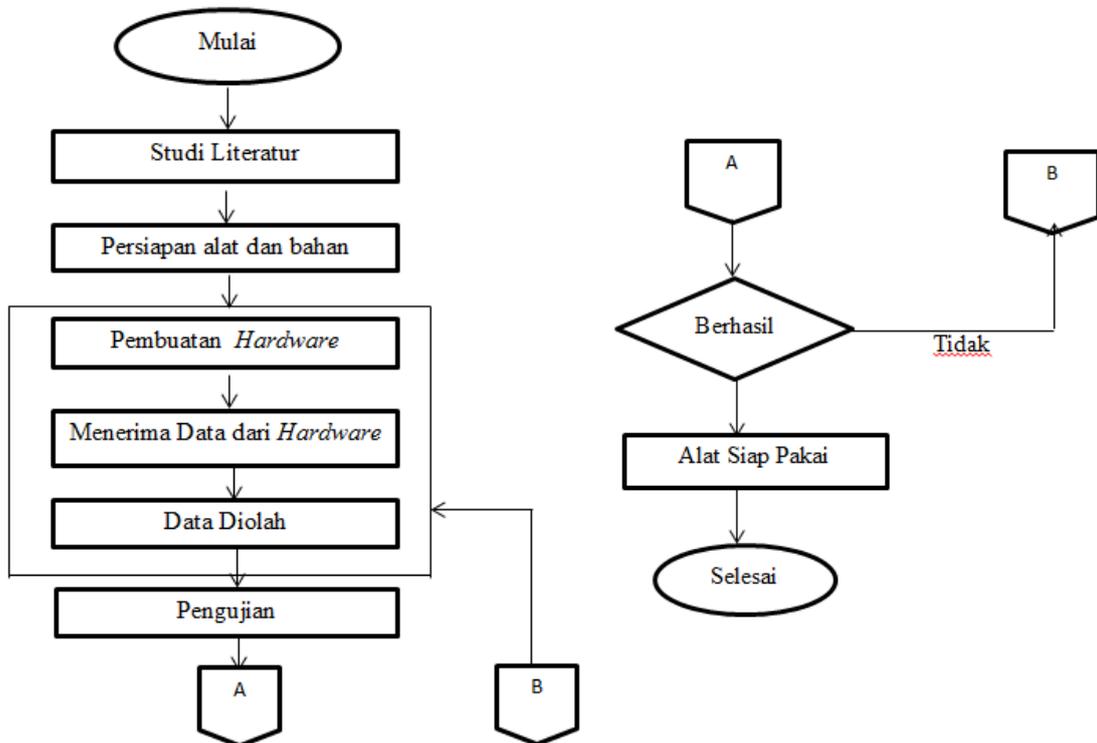


## BAB III

### METODELOGI RANCANG BANGUN ALAT

#### 3.1 Metode Perancangan

Pada perancangan alat ini terbagi menjadi dua metodologi, yang pertama pembuatan sistem *hardware* dan yang kedua pembuatan sistem yang akan mengirimkan data dari *hardware* ke laman web yang telah disediakan. Berikut gambar 3.1 diagram alir pada perancangan :



**Gambar 3.1** Diagram Alir Perancangan

### 3.2 Studi Literatur

Pada studi literatur ini yaitu mempelajari bahasan-bahasan mengenai Aplikasi ESP8266 Sebagai *Wireless Sensor Node* Temperatur, Sensor suhu DSB8210, dan *Access point* yang di dapat dari buku-buku terkait maupun melalui media *online*.

### 3.3 Persiapan Alat dan Bahan

Pada persiapan alat dan bahan ini, ada beberapa kebutuhan yang perlu dipersiapkan agar pembuatan *hardware* berhasil dengan sempurna. Kebutuhan-kebutuhan yang perlu dipersiapkan yaitu sebagai berikut

a. Alat

1. Sensor Suhu DS18B20
2. Modul ESP8266
3. Resistor
4. Laptop
5. Router TP-Link
6. Powerbank
7. Solder
8. Multimeter
9. Tang Potong

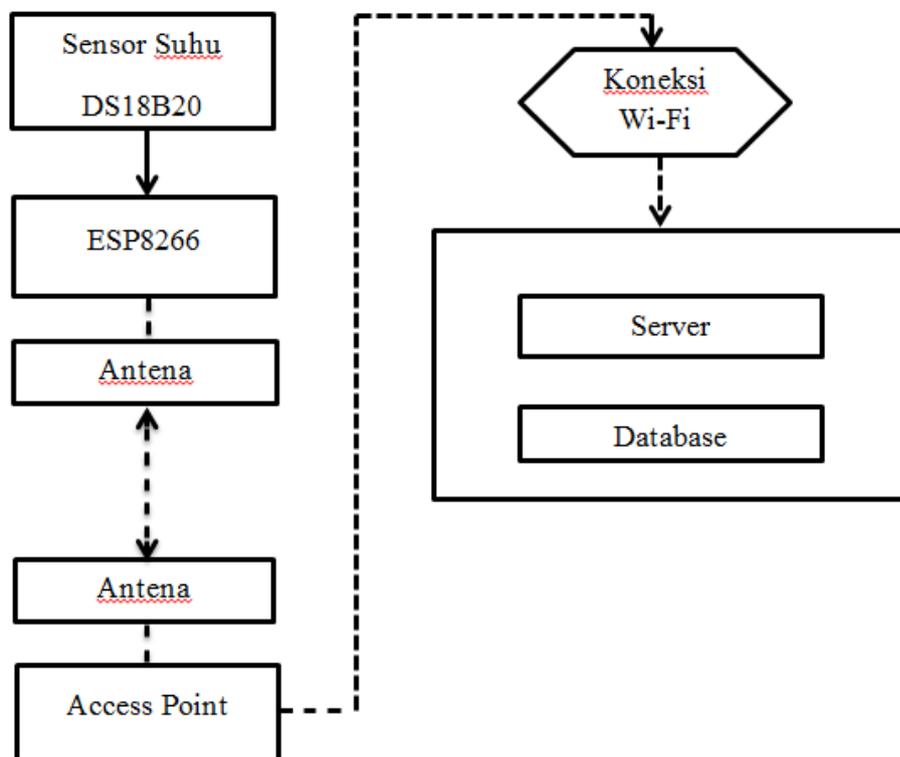
b. Bahan

1. Kabel
2. Timah

3. Selongsong bakar
4. Kabel Mikro-USB

### 3.4 Pembuatan *Hardware*

Pada tahapan kali ini, ketika semua bahan sudah disiapkan maka bahan akan dirancang sesuai desain yang telah dibuat. Perangkat keras atau *hardware* hasil akhirnya berupa rangkaian dan modul. *Hardware* tersebut baru bisa digunakan setelah diprogram dengan menggunakan *software arduino IDE*, agar sistem kontrol berjalan seperti yang diharapkan. Rancangan sistem digambar dalam bentuk diagram blok, berikut gambar 3.2 rancangan sistem *hardware* :



Gambar 3.2 Diagram Blok Sistem

Desain perangkat keras diatas meliputi ESP8266 sebagai modul *Wi-Fi* yang akan sekaligus menjadi mikrokontroler yang akan mengolah data suhu dari sensor suhu DS18B20. Lalu, modul *Wi-Fi* tersebut akan mengirimkan data suhu yang terukur ke *server*. Data *server* tersebut akan dirubah menjadi nilai suhu yang akan ditampilkan pada laman web dan dapat terukur secara *realtime*.

Setelah semua alat dan bahan telah tersedia, maka tahap selanjutnya adalah pembuatan alat. Tahap yang pertama yaitu mengetahui terlebih dahulu kaki-kaki dari modul ESP8266. Ditunjukkan pada gambar 3.3 yaitu Bentuk dari ESP8266 :

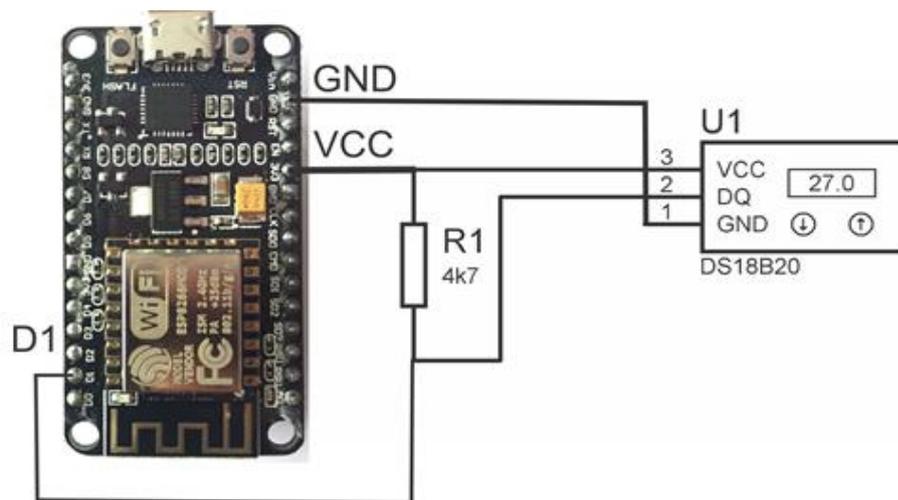


**Gambar 3.3** Bentuk dari ESP8266

### 3.5 Skematik Diagram ESP8266

Pada gambar dibawah merupakan skematik diagram node ESP8266 dengan sensor suhu DS18B20. Skematik ini digunakan untuk merancang pembuatan alat monitoring suhu melalui jaringan *Wi-Fi*. Pada penelitian ini digunakan tiga buah node dengan skematik yang sama, yang membedakan dari setiap node adalah pada pin input data yang masuk dari sensor suhu DS18B20 menuju ESP8266.

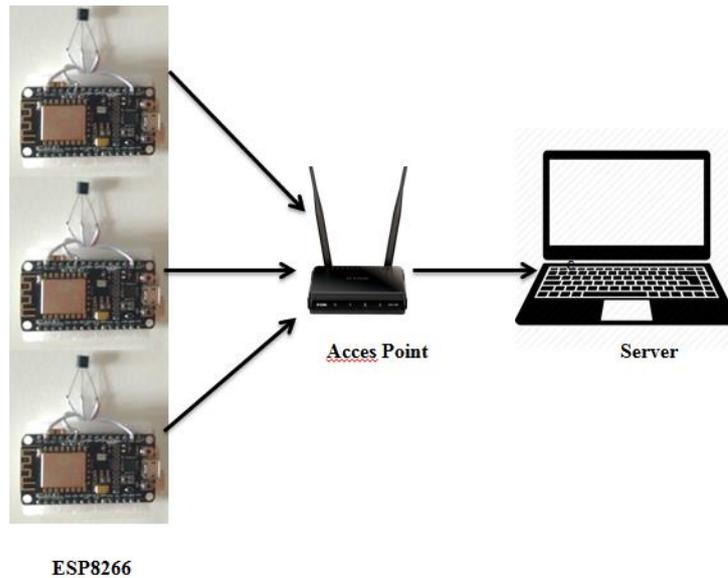
Pada skematik ini input data berada pada pin D1. Untuk tegangan kerja sensor menggunakan tegangan 3,3Volt DC sehingga dihubungkan dengan pin 3V3 dan *ground* sumber tegangan diambil dari port USB. Adapun gambar 3.4 hasil skematik diagram ESP8266



**Gambar 3.4** Skematik Diagram ESP8266

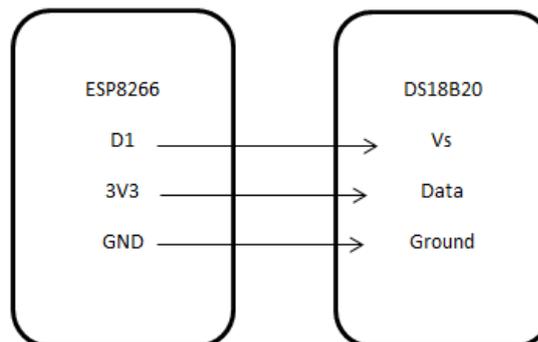
### 3.6 Perancangan alat

Pada perancangan alat ini terdapat tiga bagian utama yaitu pertama *server* yang berupa laptop ,kemudian acces point yang berguna untuk konektifitas *Wi-Fi* dan ESP8266 dengan sensor suhu yang berfungsi sebagai pengirim data menuju *server*. Penggunaan *access point* ditunjukkan untuk mendapatkan jarak jangkauan yang jauh. Sensor suhu terdapat pada setiap ESP8266 sebagai perangkat masukan atau sensor yang akan membaca besaran suhu menjadi nilai digital yang akan dikirimkan oleh ESP8266 menuju ke *database server*. Berikut gambar 3.5 Skematik blok diagram



**Gambar 3.5** Skematik Blok Diagram

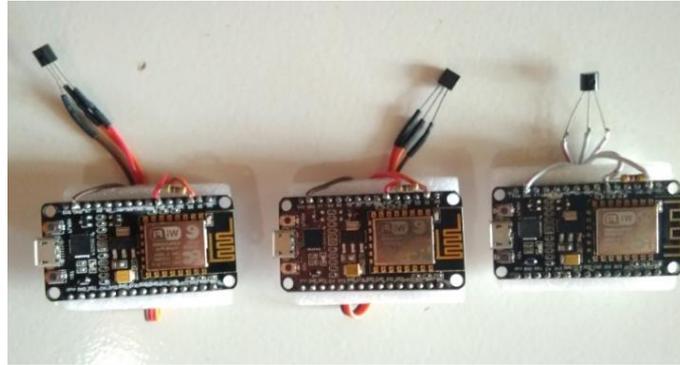
Penggunaan sensor suhu DS18B20 membutuhkan resistor sebagai penghambat arus yang disambungkan ke kaki sensor suhu DS18B20 menggunakan kabel pelangi. Pin D1 merupakan data masukan dari sensor suhu. Sensor suhu membutuhkan 3,3Volt sebagai tegangan masukan, maka sensor suhu perlu dihubungkan ke pin 3V3 dan *ground*. Berikut gambar 3.6 Skematik sensor suhu DS18B20



**Gambar 3.6** Skematik Sensor Suhu DS18B20

Pada penelitian ini sensor suhu yang digunakan sejumlah 3 buah dengan seri yang sama, dan modul *Wi-Fi* nya berjumlah 3 buah dengan seri ESP8266. Modul *Wi-Fi* ESP8266 berfungsi sebagai pengirim data melalui *Wi-Fi* ke *server* yang

telah tersedia. Data yang dikirim adalah berupa data sensor suhu yang telah terbaca oleh *arduino*. Berikut gambar 3.7 yaitu Hasil akhir hardware



**Gambar 3.7** Hasil akhir *Hardware*

### **3.7 Pengaktifan Modul ESP8266 Pada Komputer**

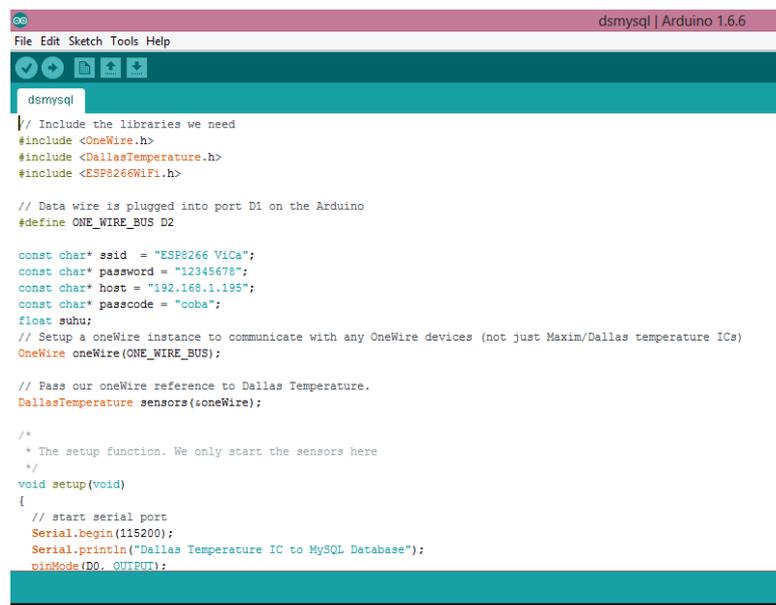
Sebelum ESP8266 ini aktif dan dapat *connect* ke komputer ada beberapa cara terlebih dahulu yang di lakukan terhadap ESP8266 ini supaya bisa aktif, berikut langkah-langkah pengaktifan modul ESP8266.

1. Pertama sambungkan ESP8266 ke komputer dengan menggunakan kabel data, kemudian buka file explorer, kemudian klik kanan komputer, kemudian pilih manage, lalu klik device manager, setelah itu lihat ports dan akan muncul com berapa.
2. Kemudian buka aplikasi ESP8266 flasher, lalu pilih operation, klik com port com ports ini yang sudah dilihat di manage, kemudian tekan tombol flash pada ESP8266, setelah itu tunggu sampai selesai prosesnya.
3. Setelah selesai proses flash buka folder ESPlorer, kemudian sesuaikan com dan *baud rate*, lalu kilik open dan tekan tombol RST pada ESP8266.

4. Kemudian jika lampu led pada ESP8266 sudah berkedip, masukan *script* dan setelah itu run atau jalankan.
5. Dengan langkah tersebut ESP8266 sudah terkoneksi dan dapat berkomunikasi dengan komputer.

### 3.8 Perancangan Perangkat Lunak

Perancangan perangkat lunak ini menggunakan 1 *input* dan 1 *output* yang diproses dalam mikrokontroller. Input tersebut berupa sensor DS18B20 sedangkan *output* berupa data dari sensor suhu yang dikirimkan ke *server* menggunakan modul ESP8266. Sebelum melakukan pemrograman pada *Arduino IDE* dipilih terlebih dahulu port dan jenis *arduino* yang digunakan pada menu *tools* kemudian pilih board NodeMCU 0.9 (ESP-12 Module) . Kemudian buka folder *dsmysql* yang telah di download dan buka aplikasi *arduino IDE*. Berikut gambar 3.8 tampilan program *dsmysql*



```

dsmysql | Arduino 1.6.6
File Edit Sketch Tools Help
dsmysql
// Include the libraries we need
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ESP8266WiFi.h>

// Data wire is plugged into port D1 on the Arduino
#define ONE_WIRE_BUS D2

const char* ssid = "ESP8266 ViCa";
const char* password = "12345678";
const char* host = "192.168.1.195";
const char* passcode = "coba";
float suhu;
// Setup a oneWire instance to communicate with any OneWire devices (not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

/*
 * The setup function. We only start the sensors here
 */
void setup(void)
{
  // start serial port
  Serial.begin(115200);
  Serial.println("Dallas Temperature IC to MySQL Database");
  pinMode(D0, OUTPUT);

```

**Gambar 3.8** Program *dsmysql*

Adapun *script* program tersebut, adalah sebagai berikut:

```
#include <OneWire.h>

#include <DallasTemperature.h>

#include <ESP8266Wi-Fi.h>

#define ONE_WIRE_BUS D2 (perintah untuk menyambungkan kaki ke
D2)

const char* ssid = "ESP8266 ViCa"; (menyambungkan ke access point)
const char* password = "12345678"; (password access point)
const char* host = "192.168.1.195"; ( IP address )
const char* passcode = "coba";

float suhu;

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

/*
 * The setup function. We only start the sensors here
 */

void setup(void)
{
  Serial.begin(115200);

  Serial.println("Dallas Temperature IC to MySQL Database");

  pinMode(D0, OUTPUT);

  delay(10);
```

```
sensors.begin();

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

Wi-Fi.begin(ssid, password);

while (Wi-Fi.status() != WL_CONNECTED) {

  delay(500);

  Serial.print(".");

  }

  Serial.println("");

  Serial.println("Wi-Fi connected");

  Serial.println("IP address: ");

  Serial.println(Wi-Fi.localIP());

}

/*

 * Main function, get and show the temperature

 */

void loop(void)

{

  Serial.print("Requesting temperatures...");

  sensors.requestTemperatures(); // Send the command to get temperatures
```

```

Serial.println("DONE");

Serial.print("Temperature : ");

suhu=sensors.getTempCByIndex(0);

Serial.println(suhu);

Serial.print("Connecting to ");

Serial.println(host);

Wi-FiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort))

{

    Serial.println("Connection failed!");

    return;

}

String url = "/ESP8266/temperatur3.php?t1="; (untuk mengakses file
temperatur sehingga bisa masuk ke database)

url += suhu;

url += "&pass=";

url += passcode;

client.print(String("GET ") + url + " HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

```

```

digitalWrite(D0, LOW);

delay(500);

digitalWrite(D0, HIGH);

delay(2500);

}

```

Cara kerja alat ini dimulai dari pembacaan suhu dengan menggunakan sensor DS18B20. Setelah didapat nilai suhu maka setiap beberapa detik sekali nilai suhu yang terbaca akan dikirimkan ke *server* menggunakan ESP8266. Hal yang perlu diingat adalah bahwa ESP8266 perlu dikoneksikan terlebih dahulu dengan *access point* supaya dapat mengirimkan data ke *server*. Data terkirim atau tidaknya ke *server* dapat dilihat pada serial monitor. Setelah pemrograman *Arduinoo IDE* selesai, maka selanjutnya yaitu mengolah data pada *server* dan pembuatan *database* menggunakan aplikasi WAMP.

### 3.9 Script Program Php

Sebelum ESP8266 ini mengirimkan data temperatur ke *database* ada beberapa cara terlebih dahulu yang dilakukan yaitu:

1. Buka folder WAMP kemudian klik folder www, di dalam folder www kemudian klik document temperature.php
2. Setelah itu buka document temperatur di notepad++
3. Dan tambahkan suhu yang akan dimasukkan temperatur tersebut di“INSERT INTO” setelah itu save.

Adapun script program temperatur.php tersebut, adalah sebagai berikut:

```
<?php
```

```
/*
```

Description: PHP code for NodeMCU project which inserts the date and two switch statuses into a MySQL database table.

Create a table in a database with columns: "date" (text), "s1" (boolean), "s2" (boolean).

/\* This PHP file is used to insert values into the database as well as display them on a webpage for you to view. The following PHP code at the top of this file is to get and insert values into the database, if the parameters are set in the URL. The PHP code later on is to display the values on the page.

```
*/
```

```
// Get values.
```

```
$t1 = $_GET['t1'];
```

```
$pass = $_GET['pass'];
```

```
date_default_timezone_set('Asia/Jakarta');
```

```
$date= date('m-d-Y H:i:s') ;
```

```
// Set password. This is just to prevent some random person from inserting values. Must be consistent with YOUR_PASSCODE in Arduino code.
```

```
$passcode = "coba";
```

```
// Check if password is right. (If there is no password, assume no data is trying to be entered and skip over this.)
```

```
if(isset($pass) && ($pass == $passcode)){
```

```
// If all values are present, insert it into the MySQL database.

if(isset($t1)){

    // Database credentials

    $servername = "localhost";

    $username = "root";

    $dbname = "cobaesp8266";

    $password = "";

    // Create connection.

    $conn = mysqli_connect($servername, $username,

$password, $dbname);

    if (!$conn) {

        die("Connection failed: " . mysqli_connect_error());

    }

    // Insert values into table. Replace YOUR_TABLE_NAME

with your database table name.

    $sql = "INSERT INTO suhu (date, suhu)

VALUES ('$date', '$t1')";

    if (mysqli_query($conn, $sql)) {

        echo "OK";

    } else

{

        echo "Fail: " . $sql . "<br/>" . mysqli_error($conn);

    }

}
```

```
// Close connection.

mysqli_close($conn);

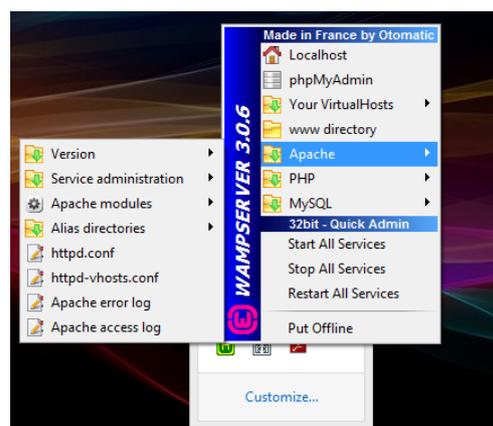
}

?>
```

### 3.10 Perancangan Perangkat Lunak pada *Server*

Dalam perancangan perangkat lunak pada sisi *server* digunakan perangkat lunak WAMP. WAMP memiliki kegunaan yang dapat membuat jaringan lokal sendiri dalam artian dapat membuat website secara *offline*.

Berikut gambar 3.9 Tampilan dari WAMP Server



**Gambar 3.9** Tampilan WAMP Server

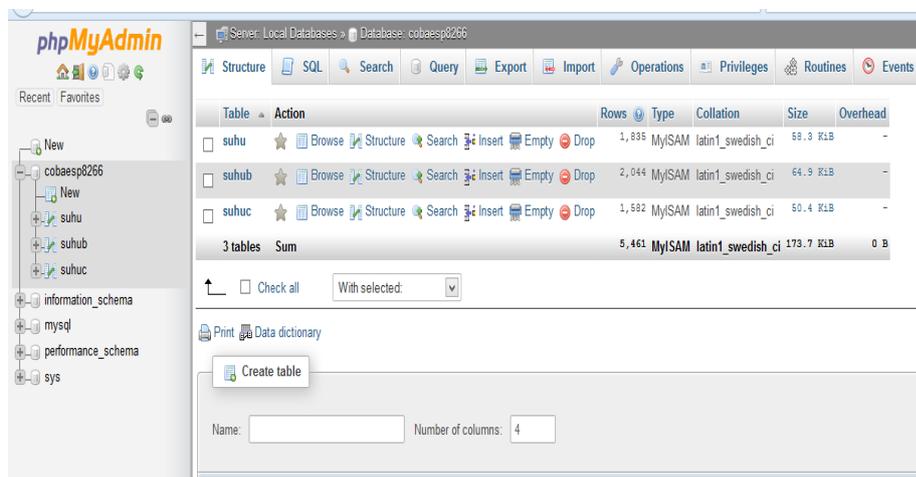
### 3.11 Pembuatan Database

Tahap pertama pada perancangan ini adalah bagaimana membuat tempat data yang dikirim dari sisi *hardware*. *Database* dapat diakses pada alamat <http://localhost/phpmyadmin/>. Langkah- langkah cara membuat *database* baru yaitu :

1. Klik database pada home phpMyAdmin, kemudian klik new
2. Kemudian klik create database dan beri nama yang diinginkan
3. Kemudian muncul Table name isi dengan suhu dengan nama yang sudah dibuat
4. Kemudian isi tabel tanggal dan nama suhu nya

### 3.12 Penyimpanan Database

Tahap pertama pada perancangan ini adalah bagaimana membuat tempat penyimpanan data yang dikirim dari sisi *hardware*. *Database* dapat diakses pada alamat <http://localhost/phpmyadmin/>. Cara membuat *database* baru yaitu klik pada tab “*Database*” dan beri nama tabel penyimpanan yang akan jadi tempat penampung. Berikut 3.10 tampilan tabel data yang dibuat pada *database*:



**Gambar 3.10** Pembuatan Tabel Data *Database*

### 3.13 Pengaktifan ESP8266 Pada *Database*

Tahap pertama yaitu dengan cara mengaktifkan aplikasi software WAMP terlebih dahulu untuk mengkoneksikan penyimpanan *database* ke laman web.

Berikut langkah-langkah pengaktifannya :

1. Hidupkan aplikasi WAMP terlebih dahulu samapai WAMP online atau lambang WAMP berwarna hijau dan sudah terkoneksi dengan *access point*.
2. Kemudian ketahui terlebih dahulu IP address ESP8266 tersebut.
3. Kemudian setelah selesai mengatur IP address, buka Mozilla Firefox, kemudian masukkan alamat <http://localhost/phpMyAdmin/>.
4. Kemudian muncul phpMyAdmin klik Go, setelah itu muncul *database* yang berupa phpMyAdmin.
5. Setelah itu, ESP8266 sudah bisa mengirimkan data suhu ke *database* tersebut.