

DIKTAT PETUNJUK PRAKTIKUM KATARINA 1.0



**Fakultas Teknik
Universitas Katolik Indonesia Atma Jaya**

JAKARTA 2021



BAB I

DHT Sensor with OLED Display

- I. Tujuan Percobaan
 - a. Mengetahui cara kerja Sensor DHT dengan tampilan OLED
 - b. Dapat merancang Sensor DHT dengan tampilan OLED
- II. Teori Singkat

OLED Display

OLED (Organic Light Emitting Diode) merupakan sebuah *LED (Light Emitting Diode)* yang mempunyai lapisan *emissive electroluminescent*, dimana lapisan ini merupakan lembaran senyawa organik yang akan memancarkan cahaya bila dilalui arus elektrik. Lapisan bahan semikonduktor organik ini terletak diantara dua elektroda dimana pada umumnya salah satu elektroda tersebut tembus pandang.

Devais *OLED* dibedakan menjadi dua macam, tergantung dari tipe lapisan organiknya, yaitu:

1. *Devais small molecule (SMOLED)*
2. *Devais organic polymer (PLED atau LEP).*

Devais *small molecule* difabrikasi menggunakan teknik evaporasi vakum, dengan meletakkan struktur polimer secara *spin-casting* atau memakai teknik cetak *ink-jet*. Sebutan *OLED* biasanya mengacu ke *small-molecular OLED* dimana struktur dasar *OLED* memiliki ketebalan sebesar 100 sampai 500 nanometer atau sekitar 1/200 ketebalan rambut manusia. Lapisan bahan organik yang digunakan untuk menyusun *OLED* bisa mencapai 2 atau 3 lapis.

Sensor DHT

Pada percobaan kali ini sensor DHT yang digunakan yaitu DHT11. Sensor DHT11 adalah modul sensor yang berfungsi untuk mendeteksi objek suhu dan kelembaban yang memiliki output tegangan analog yang dapat diolah lebih lanjut menggunakan mikrokontroler. Modul sensor ini tergolong kedalam elemen resistif seperti perangkat pengukur suhu seperti contohnya yaitu NTC. Kelebihan dari modul sensor ini dibanding modul sensor lainnya yaitu dari segi kualitas pembacaan data sensing yang lebih

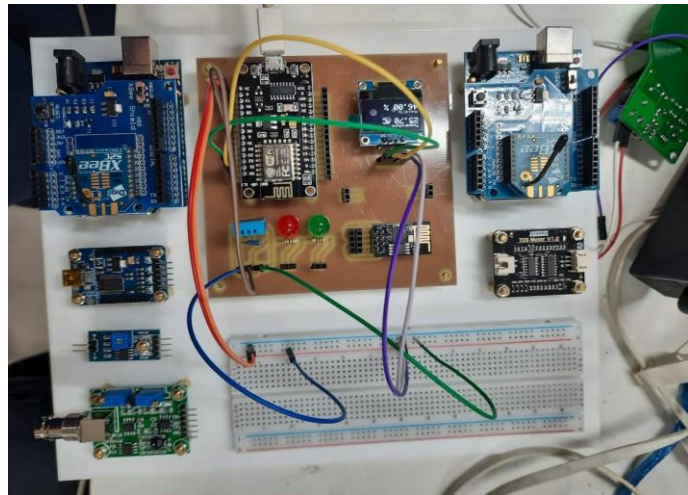
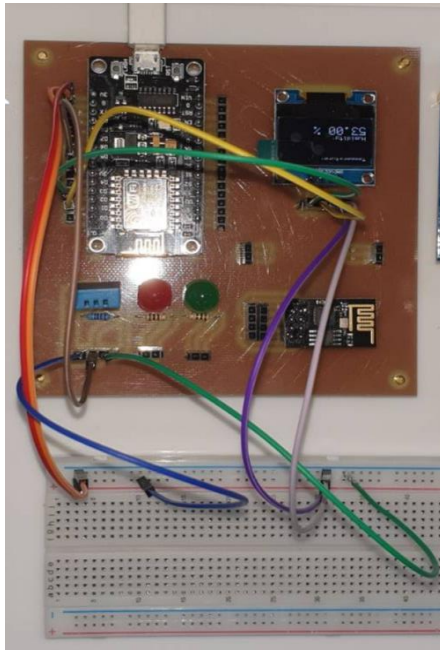
responsif yang memiliki kecepatan dalam hal sensing objek suhu dan kelembaban, dan data yang terbaca tidak mudah terinterferensi. Sensor DHT11 memiliki spesifikasi diantaranya tegangan masukan sebesar 5 Vdc, rentang temperatur sebesar 0-50° C kesalahan $\pm 2^\circ \text{C}$, kelembaban sebesar 20-90% RH $\pm 5\% \text{ RH error}$.

III. Alat dan Bahan

- Modul Praktikum IoT
- PC / Laptop
- Arduino IDE
- *OLED Display*
- *DHT Sensor*

IV. Percobaan

1. Buatlah rangkaian sesuai yang ada di gambar



P.s:

- Kabel Merah (3V NodeMCU ke V in ProjectBoard)
- Kabel Orange (Ground NodeMCU ke Ground ProjectBoard)
- Kabel Biru (Vcc DHT ke V in ProjectBoard)
- Kabel Coklat (D5 NodeMCU ke Kaki Data DHT)
- Kabel Kuning (D1 NodeMCU ke SCL OLED)



-
- Kabel Hijau Tua (D2 NodeMCU ke SDA OLED)
 - Kabel Hijau Muda (Ground DHT ke Ground ProjectBoard)
 - Kabel Putih (Ground OLED ke Ground ProjectBoard)
 - Kabel Ungu (Vcc OLED ke V in ProjectBoard)
2. Upload Coding yang telah dibuat ke NodeMCU
 3. Lihat Hasil pada OLED Display, Jika berhasil maka akan terbaca suhu pada display tersebut.



BAB II

DHT11 with XBee Connectivity

- I. Tujuan Percobaan
 - a. Mengetahui cara kerja DHT11 yang terhubung dengan Xbee
 - b. Dapat merancang DHT11 yang terhubung dengan Xbee
- II. Teori Singkat

ZigBee

ZigBee merupakan kependekan dari dua kata yaitu *zigzag* dan *bee*, yang berarti lebah terbang dengan perubahan arah. Namun secara umum, *ZigBee* merupakan sebuah spesifikasi untuk protokol komunikasi tingkat tinggi yang mengacu pada standar IEEE 802.15.4. Keunggulan dari *ZigBee* adalah rendahnya daya yang dibutuhkan karena biasa digunakan dalam jaringan berskala kecil (*personal*), sehingga dapat digunakan sebagai perangkat pengatur secara *wireless*, dengan penggunaan sumber dayanya yang rendah. Walaupun *coverage* area tidak begitu luas. Teknologi ini sesuai untuk diimplementasikan pada peralatan *wireless* yang berada didalam rumah, perkantoran, apartment, taman kota, dan lain-lain.

XBee

XBee adalah salah satu perangkat komunikasi data wireless yang bekerja pada frekuensi 2.4 GHz dengan menggunakan protokol standar IEEE 802.15.4. XBee sendiri berfungsi sebagai transceiver dan receiver pada komunikasi radio dengan berbagai jenis antena, di antaranya adalah whip antenna, U. Fl. RF connector dan in-build chip antenna. XBee juga mendukung komunikasi jaringan peer-to-peer serta point to multi-point wireless dengan kecepatan 250 kbits/s¹.

XBee yang digunakan ada dua varian.

1. Xbee S2C XBee ini digunakan untuk perangkat *end device*, Dan menggunakan empat buah Xbee S2C. Pada perangkat *end device*, PAN ID ditulis 1234, Destination High (DH) ditulis 13A200 mengikuti Serial High (SH). Dan Destination Low (DL) ditulis 414EED5D mengikuti Serial Low (SL) dari perangkat router.

¹ <http://prima.lecturer.pens.ac.id/Siskomnir/Modul%201.pdf> & www.arduino-tutorial.com



2. Xbee pro S2C Xbee ini digunakan untuk perangkat router dan server. Satu buah Xbee PRO S2C untuk server, dan satu buah Xbee PRO S2C untuk router. Pada perangkat router dan server, PAN ID ditulis 1234. Pada perangkat router, Join Notification (JN) diaktifkan dan Coordinator Enable (CE) dinonaktifkan. Destination High (DH) Dan Destination Low (DL) ditulis 0. Pada perangkat server Coordinator Enable (CE) diaktifkan dan Destination High (DH) ditulis 0, Destination Low (DL) ditulis. FFFF.

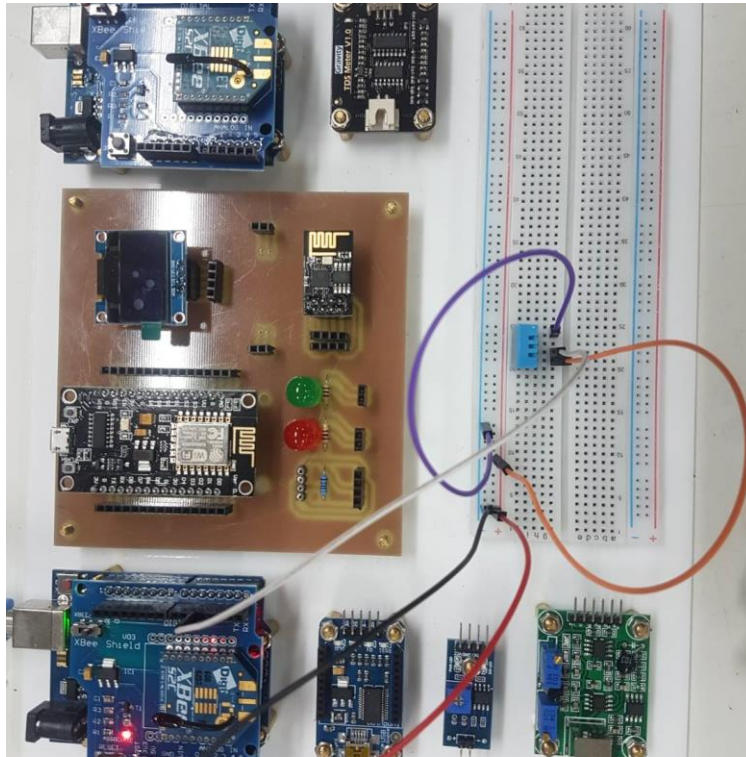
III. Alat dan Bahan

- Modul Praktikum IoT
- PC / Laptop
- Arduino IDE

IV. Percobaan

1. Lakukan konfigurasi pada Xbee dengan cara:
 - Pindahkan Xbee ke Zigbee USB adapter
 - Hubungkan Zigbee USB adapter dengan laptop/PC
 - Buka Software XCTU
 - Klik Search
 - Pilih COM Zigbee anda
 - Pilih perangkat yang ingin anda konfigurasi
 - Atur baud di angka 9600 dan data bits 8
 - Klik Finish
 - Atur PAN ID menjadi 1234
 - Atur destination address dan node identifier
 - Klik Write

2. Buatlah rangkaian sesuai yang ada di gambar



P.S :

- Kabel Ungu (Kaki Ground DHT ke Ground ProjectBoard)
- Kabel Putih (Input 5 Xbee ke Kaki Data DHT)
- Kabel Orange (V in ProjectBoard ke Kaki V in DHT)
- Kabel Hitam (Ground Xbee ke Ground ProjectBoard)
- Kabel Merah (V in Xbee ke ProjectBoard)

Percobaan sukses saat DHT11 berhasil terhubung ke Arduino (dapat diperiksa dengan pembacaan suhu melalui *serial console*), dan XBee terhubung ke Arduino (dapat *send/receive* paket dari XCTU atau Arduino *serial console*). Jika tidak dapat mengirimkan atau menerima paket, mohon periksa konfigurasi pada XCTU.



BAB III

LED with Local Host

I. Tujuan Percobaan

- Mengetahui cara kerja pengontrolan LED menggunakan web server

II. Teori Singkat

NodeMCU

Modul WiFi NodeMCU merupakan sebuah *firmware* interaktif berbasis LUA Espressif ESP8622 Wifi SoC. NodeMCU ESP8266 v0.9 memiliki 4MB flash, 11 pin GPIO dimana 10 diantaranya dapat digunakan untuk *Pulse Width Modulation* (PWM), 1 pin ADC, 2 pasang UART, WiFi 2,4GHz serta mendukung WPA/ WPA2 . NodeMCU selain dapat diprogram menggunakan bahasa LUA dapat juga diprogram menggunakan bahasa C menggunakan arduino IDE.

NodeMCU adalah sebuah *board* elektronik yang berbasis *chip* ESP8266 dengan kemampuan menjalankan fungsi mikrokontroler dan juga koneksi internet (WiFi). Terdapat beberapa pin *Input* dan *Output* sehingga dapat dikembangkan menjadi sebuah aplikasi *monitoring* maupun *controlling* pada proyek *Internet of Things* (IoT). NodeMCU ESP8266 dapat diprogram dengan *compiler*-nya Arduino, menggunakan Arduino IDE. Bentuk fisik dari NodeMCU ESP 8266, terdapat *port* USB (*mini* USB) sehingga akan memudahkan dalam pemrogramannya. NodeMCU ESP8266 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*”.

III. Alat dan Bahan

- PC/Laptop
- Modul Praktikum IoT



IV. Percobaan

1. Hubungkan rangkaian seperti pada gambar 3.1
2. Klik *tab tools* > *Board* > Pilih NodeMCU 1.0 (ESP-12E Module)
3. Klik *tab tools* > *Port* > Pilih *port* yang digunakan oleh NodeMCU
4. Masukkan *ssid* dan *password* pada Arduino IDE dari WiFi yang digunakan. (*input ssid* dan *password* dapat berbeda sesuai dengan keinginan *user*). Kode dapat diunduh melalui <https://github.com/annisasarah/KATARINA1.0>

```
web_server_led$
#include <ESP8266WiFi.h>

// Replace with your network details
const char* ssid = ".....";
const char* password = ".....";

// Web Server on port 8888
WiFiServer server(8888);

// variables
String header;
String gpio5_state = "Off";
String gpio4_state = "Off";
int gpio5_pin = 5;
int gpio4_pin = 4;

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

// only runs once
void setup() {
  // Initializing serial port for debugging purposes
  Serial.begin(9600);
  delay(10);
}
```



web_server_led\$

```
// preparing GPIOs
pinMode(gpio5_pin, OUTPUT);
digitalWrite(gpio5_pin, LOW);
pinMode(gpio4_pin, OUTPUT);
digitalWrite(gpio4_pin, LOW);

// Connecting to WiFi network
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

// Starting the web server
server.begin();
Serial.println("Web server running. Waiting for the ESP IP...");
delay(5000);

// Printing the ESP IP address
Serial.print("Go to: http://");
Serial.print(WiFi.localIP());
Serial.println(":8080");
```

web_server_led\$

```
// Printing the ESP IP address
Serial.print("Go to: http://");
Serial.print(WiFi.localIP());
Serial.println(":8080");
}

// runs over and over again
void loop() {
    // Listening for new clients
    WiFiClient client = server.available();

    if (client) {
        currentTime = millis();
        previousTime = currentTime;
        Serial.println("New client");
        // boolean to locate when the http request ends
        boolean blank_line = true;
        while (client.connected() && currentTime - previousTime <= timeoutTime) {
            currentTime = millis();
            if (client.available()) {
                char c = client.read();
                header += c;

                if (c == '\n' && blank_line) {
                    // checking if header is valid
                    // dXNlcjpwYXNz = 'user:pass' (user:pass) base64 encode
                    // username : user
                    // password : pass
                }
            }
        }
    }
}
```



web_server_led\$

```
if (c == '\n' && blank_line) {

    // checking if header is valid
    // dXNlcjpwYXNz = 'user:pass' (user:pass) base64 encode
    // username : user
    // password : pass

    Serial.print(header);

    // Finding the right credential string
    if(header.indexOf("dXNlcjpwYXNz") >= 0) {
        //successful login
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Connection: close");
        client.println();
        // turns the GPIOs on and off
        if(header.indexOf("GET / HTTP/1.1") >= 0) {
            Serial.println("Main Web Page");
        }
        else if(header.indexOf("GET /gpio5on HTTP/1.1") >= 0){
            Serial.println("GPIO 5 On");
            gpio5_state = "On";
            digitalWrite(gpio5_pin, HIGH);
        }
        else if(header.indexOf("GET /gpio5off HTTP/1.1") >= 0){
            Serial.println("GPIO 5 Off");
            gpio5_state = "Off";
            digitalWrite(gpio5_pin, LOW);
        }
    }
}
```

web_server_led\$

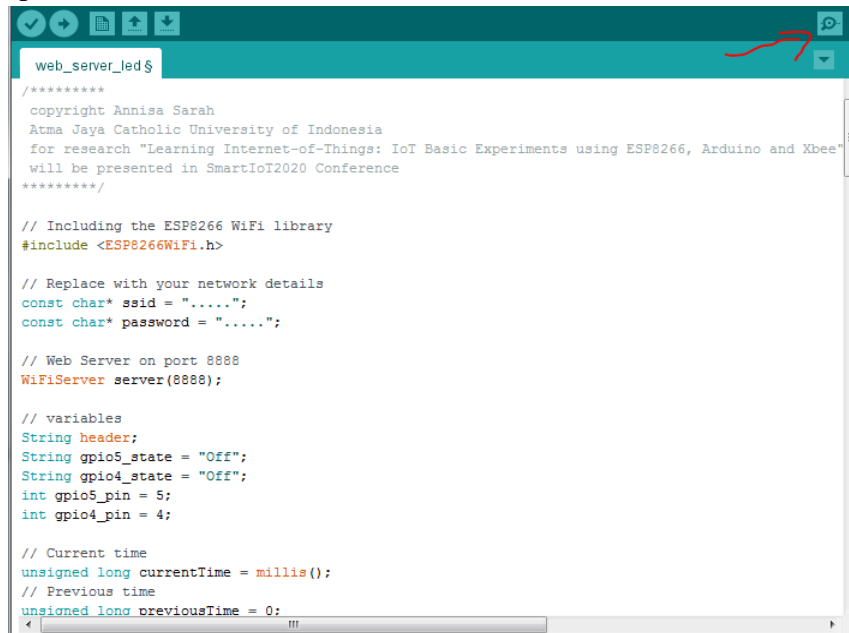
```
else if(header.indexOf("GET /gpio5off HTTP/1.1") >= 0){
    Serial.println("GPIO 5 Off");
    gpio5_state = "Off";
    digitalWrite(gpio5_pin, LOW);
}
else if(header.indexOf("GET /gpio4on HTTP/1.1") >= 0){
    Serial.println("GPIO 4 On");
    gpio4_state = "On";
    digitalWrite(gpio4_pin, HIGH);
}
else if(header.indexOf("GET /gpio4off HTTP/1.1") >= 0){
    Serial.println("GPIO 4 Off");
    gpio4_state = "Off";
    digitalWrite(gpio4_pin, LOW);
}

// your web page
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head>");
client.println("<meta name='viewport' content='width=device-width, initial-scale=1'>");
client.println("<link rel='stylesheet' href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css'>");
client.println("</head><div class='container'>");
client.println("<h1>Web Server</h1>");
client.println("<h2>STATUS LAMPU MERAH: " + gpio5_state);
client.println("<div class='col-md-2'><a href='"/gpio5on\"' class='btn btn-block btn-lg btn-success' role='button'>On</a></div>");
client.println("<div class='col-md-2'><a href='"/gpio5off\"' class='btn btn-block btn-lg btn-danger' role='button'>Off</a></div>");
client.println("</div>");
client.println("<h2>STATUS LAMPU KUNING: " + gpio4_state);
client.println("<div class='row'>");
client.println("<div class='col-md-2'><a href='"/gpio4on\"' class='btn btn-block btn-lg btn-success' role='button'>On</a></div>");
```



```
web_server_led$
// wrong user or passm, so http request fails...
else {
  client.println("HTTP/1.1 401 Unauthorized");
  client.println("WWW-Authenticate: Basic realm=\"Secure\"");
  client.println("Content-Type: text/html");
  client.println();
  client.println("<html>Authentication failed</html>");
}
header = "";
break;
}
if (c == '\n') {
  // when starts reading a new line
  blank_line = true;
}
else if (c != '\r') {
  // when finds a character on the current line
  blank_line = false;
}
}
// closing the client connection
delay(1);
client.stop();
Serial.println("Client disconnected.");
}
```

5. Upload kode tersebut ke NodeMCU, kemudian buka serial monitor



```
web_server_led$
/*****
copyright Annisa Sarah
Atma Jaya Catholic University of Indonesia
for research "Learning Internet-of-Things: IoT Basic Experiments using ESP8266, Arduino and Xbee"
will be presented in SmartIoT2020 Conference
*****/

// Including the ESP8266 WiFi library
#include <ESP8266WiFi.h>

// Replace with your network details
const char* ssid = ".....";
const char* password = ".....";

// Web Server on port 8080
WiFiServer server(8080);

// variables
String header;
String gpio5_state = "Off";
String gpio4_state = "Off";
int gpio5_pin = 5;
int gpio4_pin = 4;

// Current time
unsigned long currentTime = millis();
// Previous time
unsigned long previousTime = 0;
```

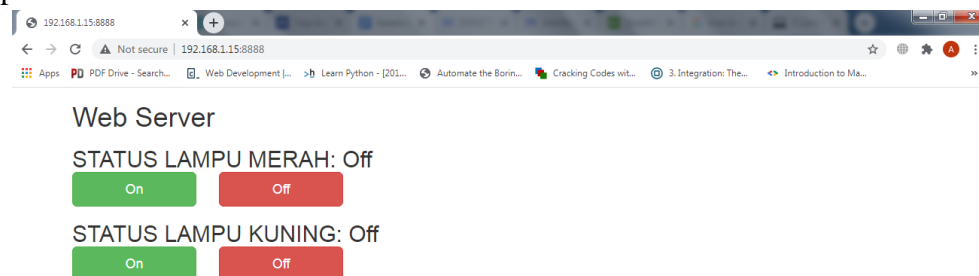
6. Ketikkan link yang digarisbawahi pada url browser

```
Connecting to ssid
.....
WiFi connected
Web server running. Waiting for the ESP IP...
Go to: http://192.168.1.15:8080
```

7. Masukkan "user" sebagai username dan "pass" sebagai password



8. Tekan tombol on atau off pada browser dan lihat perubahan LED pada modul praktikum



Percobaan berhasil jika kondisi LED dapat berubah saat kita menekan tombol on/off pada website



BAB IV

Tunneling with Ngrok (public)

I. Tujuan Percobaan

- Memahami konsep dari Internet of Things (IoT)
- Memahami konektivitas pada IoT dengan melakukan metode tunnelling pada eksperimen sebelumnya, via aplikasi ngrok.

II. Teori Singkat

Internet of Things (IoT)

Internet of Things (IoT) merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus ke dalam sebuah benda di sepegnungar sehingga dapat berkomunikasi antara satu sama lain melalui sebuah jaringan skala global seperti internet. IoT membentuk suatu koneksi antara mesin dengan mesin, sehingga mesin-mesin tersebut dapat berinteraksi dan bekerja secara independen sesuai dengan data yang diperoleh dan diolahnya secara mandiri. Tujuannya adalah untuk membuat manusia berinteraksi dengan benda menjadi lebih mudah, bahkan supaya benda juga bisa berkomunikasi dengan benda lainnya.

Suatu benda dikatakan ‘Internet of Things’ adalah apabila memiliki tiga komponen penyusun yaitu interface jaringan, sensor, dan mikroprosesor. Prinsip kerja dari Internet of Things adalah sebagai berikut. Setiap benda yang akan menjadi IoT harus memiliki sebuah IP address. IP address adalah sebuah identitas atau alamat dalam jaringan yang membuat benda tersebut dapat berkomunikasi dengan benda lainnya dalam jaringan yang sama. IP address dalam benda tersebut akan dikoneksikan ke jaringan yang dapat mencakup dalam skala global, yaitu internet. Kemudian benda tersebut juga harus dipasang sebuah sensor untuk memungkinkan benda tersebut memperoleh informasi yang dibutuhkan. Setelah informasi diperoleh, benda tersebut akan mampu mengolah informasi itu sendiri kedalam mikroprosesor. (Jurnal Ibu TG dan Ibu Sri)

Ngrok

Ngrok adalah sebuah aplikasi untuk menampilkan web server yang awalnya berjalan pada local host ke internet.

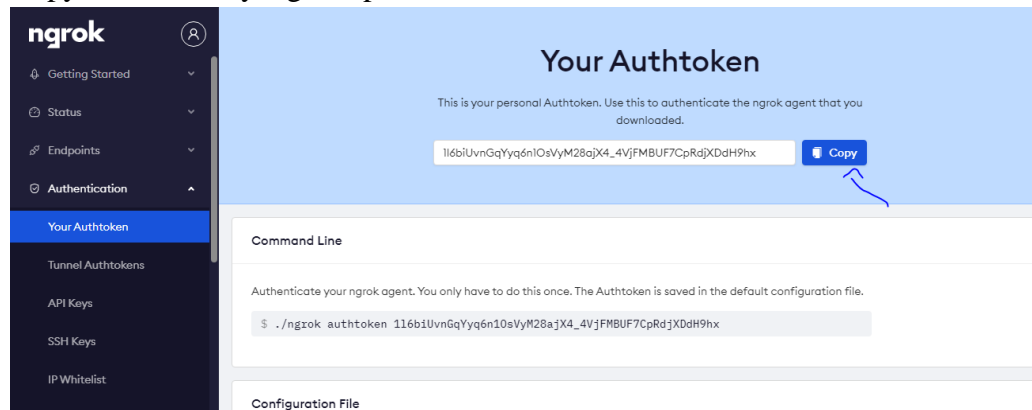


III. Alat dan Bahan

- PC/Laptop/Handphone/Tablet
- Modul Praktikum IoT
- Software Ngrok
- Arduino IDE

IV. Percobaan

1. Buat akun Ngrok
2. Copy AuthToken yang ada pada tab **Your Authtoken**



3. Hubungkan rangkaian seperti pada gambar
4. Klik tab tools > Board > Pilih NodeMCU 1.0 (ESP-12E Module)
5. Klik tab tools > Port > Pilih port yang digunakan oleh NodeMCU
6. Upload kode eksperimen “LED with Local Host” pada NodeMCU, kemudian klik serial monitor untuk melihat IP address ESP8266 (masukkan ssid dan password dari WiFi yang digunakan)



```
#include <ESP8266WiFi.h>

// Replace with your network details
const char* ssid = ".....";
const char* password = ".....";

// Web Server on port 8080
WiFiServer server(8080);

// variables
String header;
String gpio5_state = "Off";
String gpio4_state = "Off";
int gpio5_pin = 5;
int gpio4_pin = 4;

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

// only runs once
void setup() {
```

Connecting to ssid
.....
WiFi connected
Web server running. Waiting for the ESP IP...
Go to: http://192.168.1.15:8080 IP address:port

7. Kemudian buka aplikasi ngrok dan ketikkan kode berikut (ip address dilihat dari serial monitor dan authtoken dilihat dari tab Your Authtoken)

```
ngrok tcp 192.168.1.15:8080 --authtoken 1f6biUvnGqYyq6n1OsVyM28ajX4_4VjFMBUF7CpRdjXDdH9hx
```

```
ngrok by @inconshreveable (Ctrl+C to quit)

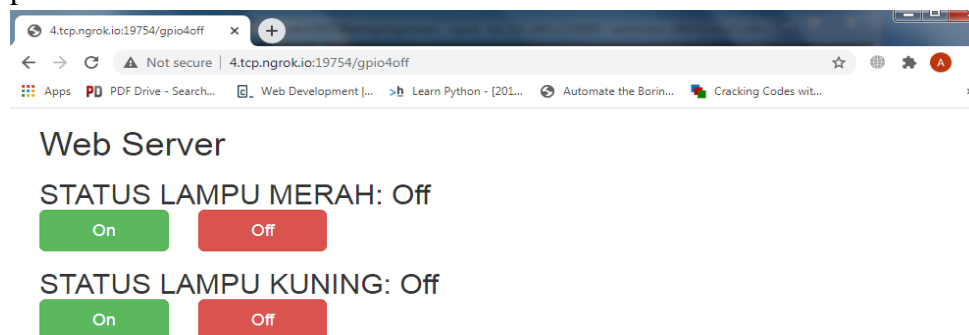
Session Status      online
Account             Anthony Surya (Plan: Free)
Version             2.3.35
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           tcp://4.tcp.ngrok.io:19754 -> 192.168.1.15:8080

Connections         ttl    opn    rt1    rt5    p50    p90
0                  0      0.00   0.00   0.00   0.00
```

8. Jika tunnelling sudah berhasil maka akan muncul tampilan seperti berikut, kemudian ketikkan link yang digarisbawahi pada url browser
9. Masukkan "user" sebagai username dan "pass" sebagai password



10. Tekan tombol on atau off pada browser dan lihat perubahan LED pada modul praktikum



Percobaan sukses jika kita bisa mengontrol status LED diluar jaringan Wi-Fi lokal.



BAB V

Xbee Network Connected to ESP8266

I. Tujuan Percobaan

- Menampilkan suhu dan kelembaban di website dengan cara menghubungkan Xbee dengan ESP8266

II. Teori Singkat

ESP 8266

Modul *ESP8266* adalah sebuah *modul wireless fidelity* (Wi-Fi) yang akhir-akhir ini semakin digemari para *hardware developer*. Selain karena harganya yang sangat terjangkau, *modul Wireless Fidelity* (Wi-Fi) serbaguna ini sudah bersifat *System on Chip* (SoC), sehingga dapat melakukan pemrograman langsung ke *ESP8266*.

Pada umumnya, *ESP8266* dapat diprogram dengan:

- a. Melalui *AT command* via *serial* komunikasi *Universal Asynchronous Receiver Transmitter* (UART)
- b. *ESP8266* dapat diprogram dengan menggunakan bahasa pemrograman yang sama dengan *Arduino*.

Kelebihan lain *ESP8266* adalah memiliki *deep sleep mode*, sehingga penggunaan daya akan relatif jauh lebih efisien dibandingkan dengan *modul WiFi* lain. Catatan penting yang harus digaris bawahi adalah *ESP8266* beroperasi pada tegangan 3.3V. Jika lebih dari 3.3V maka akan merusak *modul ESP 8266*.

III. Alat dan Bahan

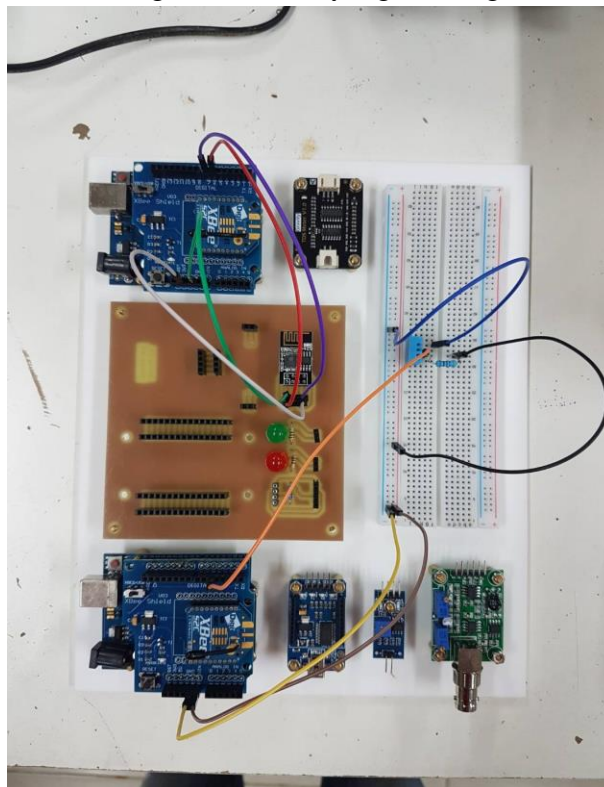
- Modul Praktikum IoT

IV. Percobaan

1. Lakukan konfigurasi kedua Xbee dengan software XCTU dengan cara:
 - Pindahkan Xbee ke Zigbee USB adapter
 - Hubungkan Zigbee USB adapter dengan laptop/PC

- Buka Software XCTU
- Klik Search
- Pilih COM Zigbee anda
- Pilih perangkat yang ingin anda konfigurasi
- Atur baud di angka 9600 dan data bits 8
- Klik Finish
- Atur PAN ID menjadi 1234
- Atur destination address dan node identifier
- Klik Write

2. Buatlah rangkaian sesuai yang ada di gambar



P.S:

- Kabel Kuning (Ground Xbee1 ke Ground ProjectBoard)
- Kabel Coklat (V in Xbee1 ke V in ProjectBoard)
- Kabel Orange (Input 5 Xbee1 ke Kaki Data DHT)
- Kabel Hitam (V in ProjectBoard ke Kaki Resistor)
- Kabel Biru (Ground DHT ke Ground ProjectBoard)



- Kabel Ungu (Input 7 Xbee2 ke Kaki RX ESP-01)
- Kabel Merah (Input 6 Xbee2 ke Kaki TX ESP-01)
- Kabel Putih (3V3 Xbee2 ke Vcc ESP-01)
- Kabel Hijau (Ground Xbee2 ke Ground ESP-01)

Percobaan akan sukses saat kita berhasil membaca status suhu pada website.

Catatan:

Pada eksperimen 5 terdapat beberapa kendala yang mengakibatkan eksperimen tersebut gagal. Yang pertama adalah komponen ESP8266 yang tidak dapat terhubung kepada jaringan WIFI yang digunakan. Walaupun sudah di test menggunakan example sketch tetap tidak dapat terhubung, ini kemungkinan dikarenakan komponen yang rusak, rangkaian yang kurang benar atau koding dan library yang digunakan salah. Karena ESP8266 tidak dapat terhubung kepada jaringan WIFI maka data temperatur dan kelembapan yang telah didapatkan tidak dapat di upload ke pada website yang telah dibuat. Kesimpulannya adalah, data yang sudah didapatkan tidak dapat di upload kepada website yang telah dibuat. Solusi yang dapat dilakukan agar eksperimen ini berjalan adalah menggunakan komponen ESP8266 yang baru, mengecek dan merevisi koding dan library yang digunakan, serta mengecek rangkaian dan menggunakan rangkaian baru jika salah.



BAB VI

Thingspeak

I. Tujuan Percobaan

- Mengetahui cara menggunakan platform cloud ThingSpeak
- Mengetahui cara mengirim data dari sensor secara live ke platform ThingSpeak dengan menggunakan WiFi

II. Teori Singkat

ThingSpeak

ThingSpeak merupakan platform Internet of Things (IoT) yang dapat diakses melalui www.thingspeak.com dan dapat digunakan secara gratis untuk mengumpulkan, memvisualisasikan dan menganalisis grafik suatu aliran data peralatan IoT di dalam cloud. Untuk menggunakan ThingSpeak, pengguna pertama-tama harus membuat akun terlebih dahulu dan membuat channel dengan jumlah field yang sesuai dengan kebutuhan pengguna.

Penjelasan singkat mengenai cara kerja percobaan ini adalah pertama sensor suhu dan kelembapan akan membaca data dari lingkungan, kemudian data tersebut akan dikirim ke server ThingSpeak melalui WiFi pengguna. Setelah data diterima di server thingspeak, data tersebut akan di visualisasikan ke dalam bentuk grafik yang bentuk grafiknya akan diperbaharui sesuai dengan data yang dikirim oleh sensor ke server ThingSpeak².

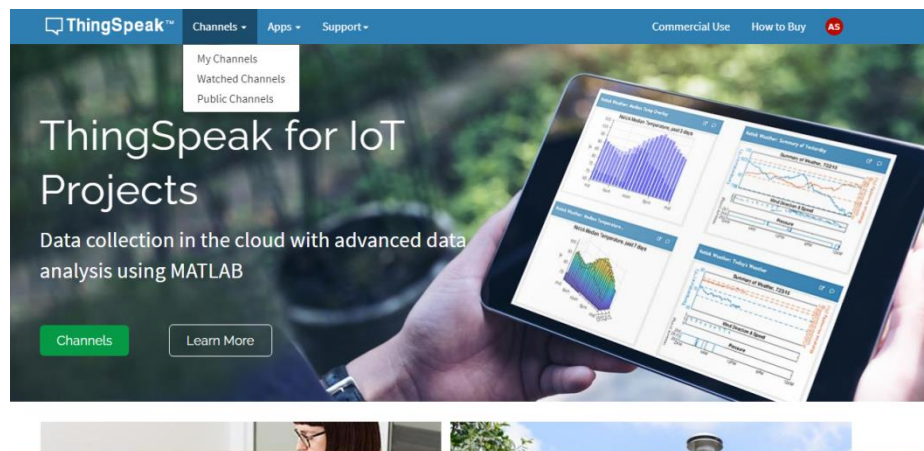
III. Alat dan Bahan

- PC
- Modul Praktikum IoT

IV. Percobaan

1. Buat akun ThingSpeak
2. Klik channel > mychannel

² <https://www.mathworks.com/help/thingspeak/product-description.html>



3. Klik New Channel

My Channels

New Channel

Search by tag



4. Masukkan nama channel dan field sesuai dengan gambar kemudian scroll kebawah dan klik save channel

New Channel

Name	Temperature dan Kelembapan
Description	
Field 1	Temperature <input checked="" type="checkbox"/>
Field 2	Kelembapan <input checked="" type="checkbox"/>
Field 3	<input type="text"/> <input type="checkbox"/>
Field 4	<input type="text"/> <input type="checkbox"/>
Field 5	<input type="text"/> <input type="checkbox"/>
Field 6	<input type="text"/> <input type="checkbox"/>
Field 7	<input type="text"/> <input type="checkbox"/>
Field 8	<input type="text"/> <input type="checkbox"/>

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
 - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5073

5. Hubungkan rangkaian seperti pada gambar

6. Ketikan code berikut pada Arduino IDE. (untuk variable ssid dan password masukkan ssid dan password dari WiFi yang digunakan, untuk variable myChannelNumber dan myWriteAPIKey lihat pada gambar)



```
Experiment_thingspeak$  
#include <ESP8266WiFi.h>  
#include <ThingSpeak.h>  
#include <DHT.h>  
  
#define DHTTYPE DHT11  
#define DHTPIN D5  
  
DHT dht(DHTPIN, DHTTYPE);  
WiFiClient client;  
  
const char* ssid = "....";  
const char* password = "....";  
  
unsigned long myChannelNumber = ....;  
const char* myWriteAPIKey = "....";  
  
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);  
  WiFi.begin(ssid, password);  
  ThingSpeak.begin(client);  
  dht.begin();  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  delay(2000);  
  
  float h = dht.readHumidity();  
  float t = dht.readTemperature();  
  
  ThingSpeak.setField(1, t);  
  ThingSpeak.setField(2, h);  
  ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);  
  
  delay(15000);  
}
```

Temperature dan Kelembapan

Channel ID: 1245184
Author: mwaa000019907376
Access: Private

mychannel number

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Write API Key

Key

[Generate New Write API Key](#)

Read API Keys

Key

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

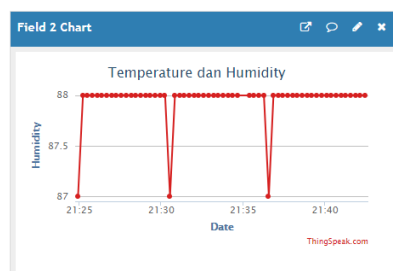
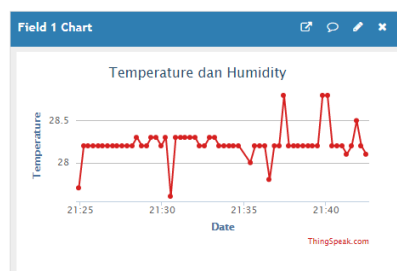
API Requests



7. Upload kode tersebut, kemudian perhatikan channel stat pada channel masing-masing

Channel Stats

Created: *about 2 hours ago*
Last entry: *less than a minute ago*
Entries: 262



Percobaan sukses saat kita berhasil melihat data dan grafik pada Thingspeak



BAB VII

Web Hosting

I. Tujuan Percobaan

- Mengetahui apa itu *Web Hosting*
- Mengetahui bagaimana cara membuat *Web Hosting*

II. Teori Singkat

Web Hosting

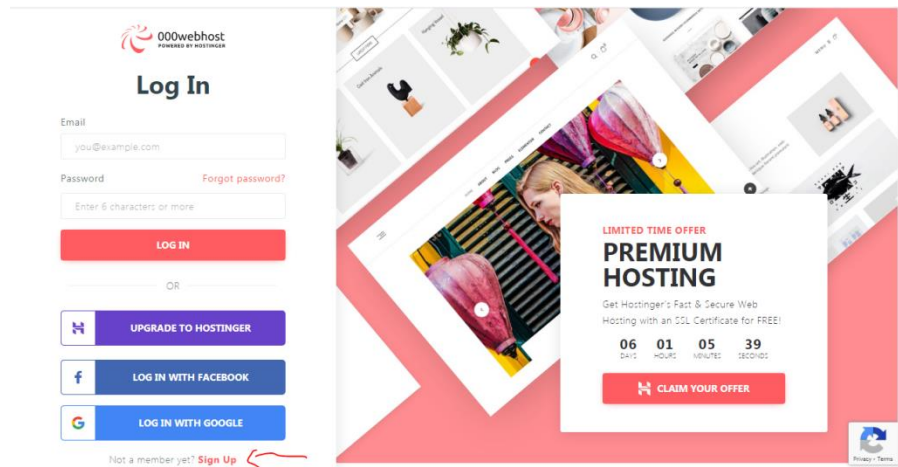
Pengertian hosting atau disebut juga *Web Hosting* atau sewa hosting adalah penyewaan tempat untuk menampung data data yang diperlukan oleh sebuah website sehingga dapat diakses lewat Internet. Data di sini dapat berupa file, gambar, email, aplikasi atau program atau script dan database. Hosting berasal dari kata host, komputer yang terhubung dalam jaringan layanan ini sebagai solusi untuk menempatkan halaman-halaman web agar selalu online. Dengan demikian, pengguna tidak perlu menyediakan satu komputer khusus yang selalu online di kantor atau di rumah, melainkan cukup dengan bekerja sama dengan penyedia jasa hosting ini. <https://docplayer.info/56812134-Hosting-untuk-memenuhi-salah-satu-tugas-mata-kuliah-praktikum-pemograman-internet.html>

III. Alat dan Bahan

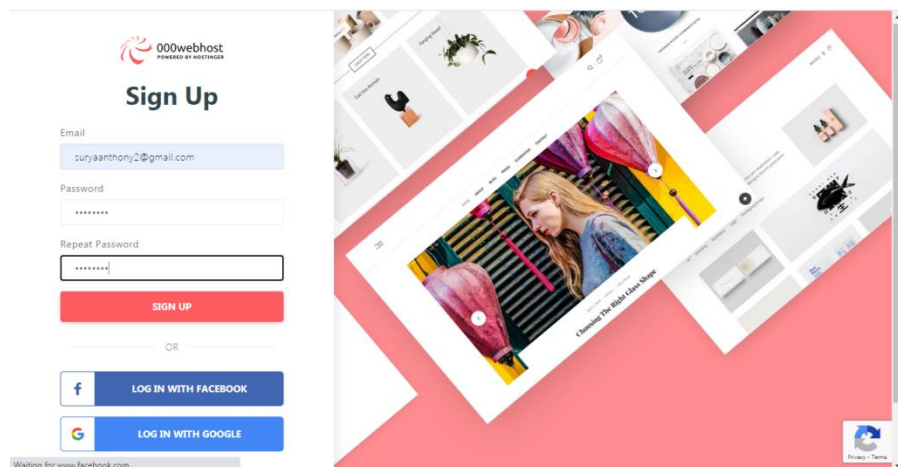
- PC / Laptop
- Arduino IDE

IV. Percobaan

1. Buat akun pada website www.000webhost.com



2. Masukkan email dan password kemudian klik sign up dan lakukan verifikasi email





3. Pilih opsi other, kemudian klik skip
4. Setelah itu masukkan nama website kemudian password

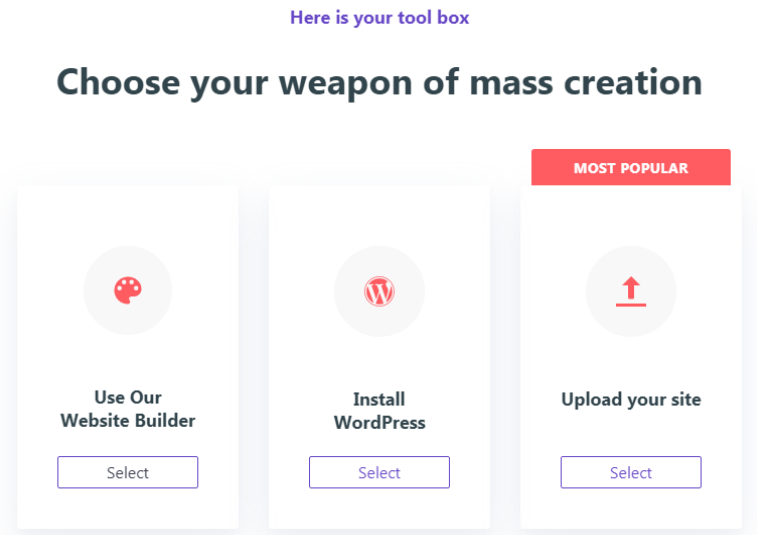
A great start is half the work

Name Your Project

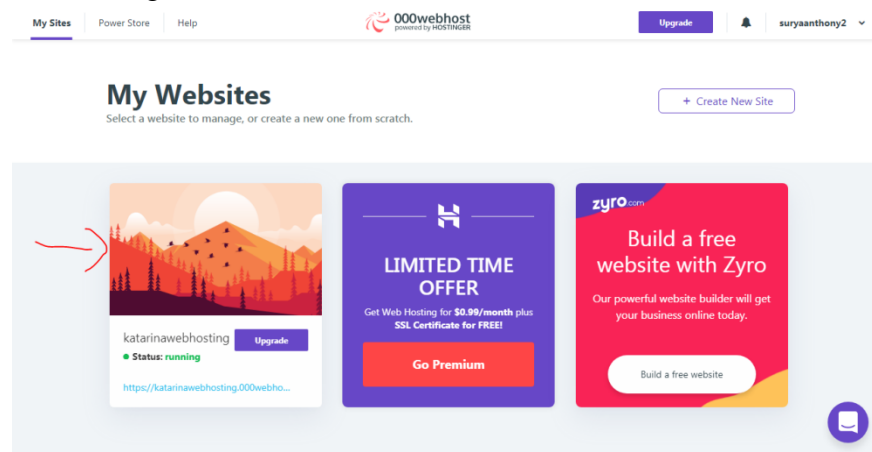
☐ Show password [GENERATE PASSWORD](#)



5. Pilih upload your site

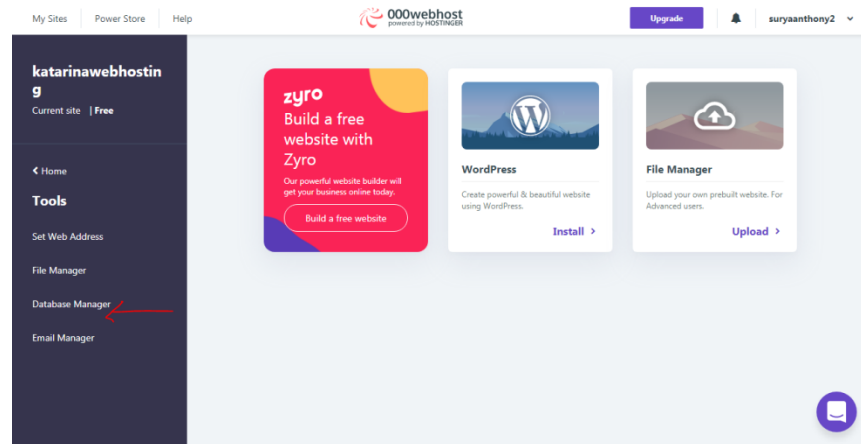


6. Klik manage website



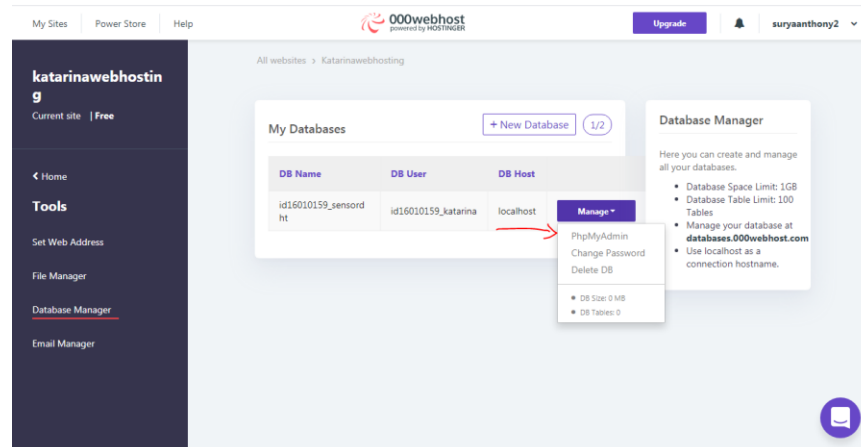


7. Pada bagian kiri klik Tools -> Database Manager

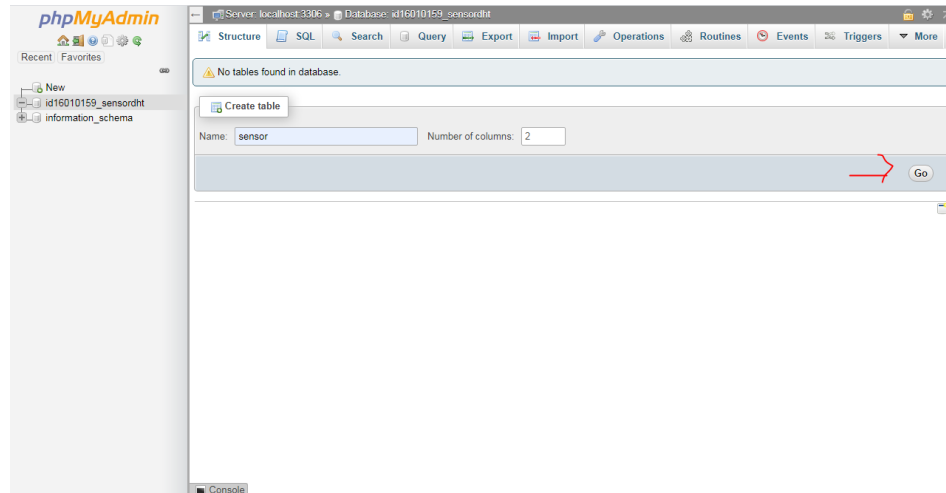


8. Klik new database, kemudian masukkan database name, database username, dan password lalu klik create dan tunggu sampai bar creating database penuh

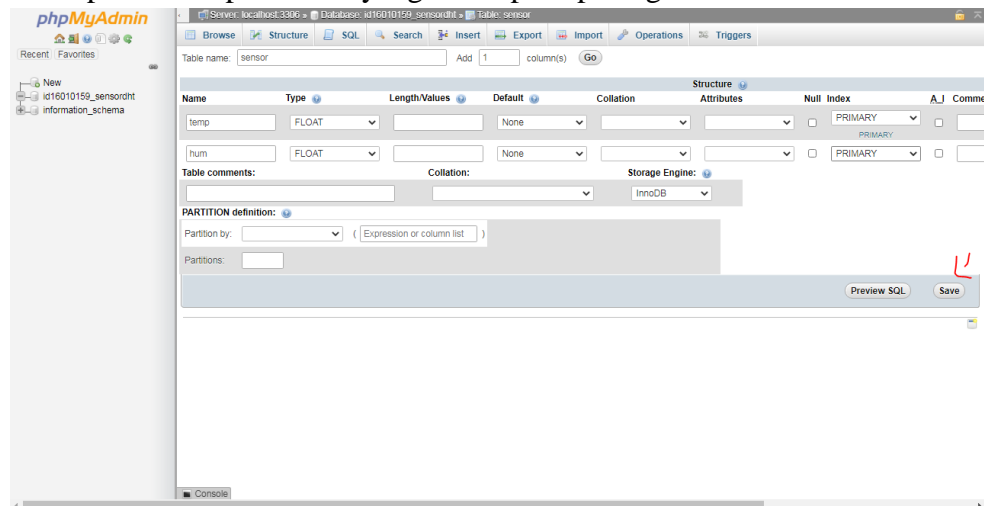
9. Klik manage dan masuk ke phpmyadmin



10. Klik database yang baru dibuat tadi kemudian masukkan “sensor” sebagai nama table, ubah nilai parameter column menjadi 2 dan tekan go



11. Ubah parameter-parameter yang ada seperti pada gambar kemudian tekan save





12. Klik insert, masukan nilai 0 pada temp dan hum lalu klik go

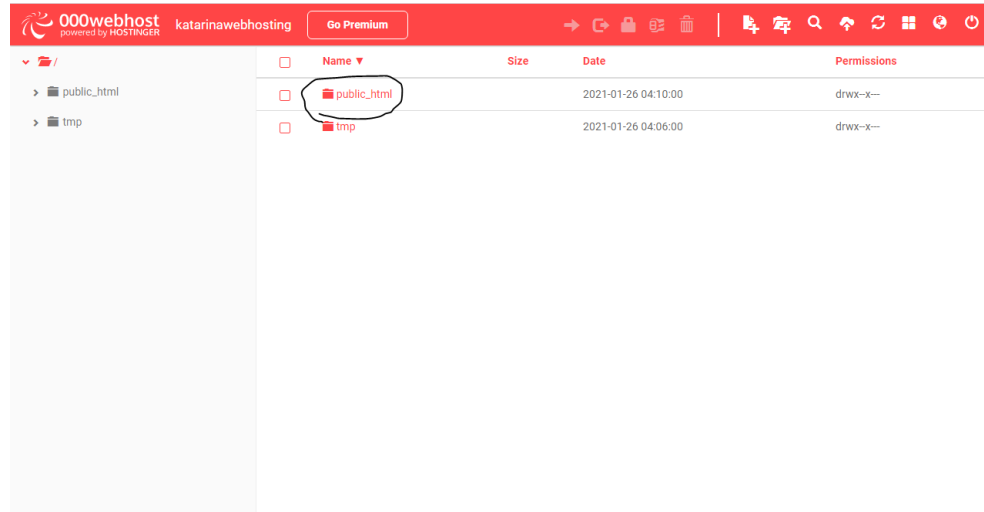
The first screenshot shows the phpMyAdmin interface for the 'sensor' table. The 'Table structure' tab is active, displaying the table's schema with two columns: 'temp' (float) and 'hum' (float). A red arrow points to the 'Insert' button in the top navigation bar.

The second screenshot shows the 'Insert' form in phpMyAdmin. The 'temp' and 'hum' columns are selected, and the value '0' is entered for both. A red arrow points to the 'Go' button at the bottom right of the form.

Below the 'Go' button, there is a section for 'Continue insertion with' rows, set to 2 rows.



13. Download folder AdminLTE pada google drive experiment_7, kemudian ekstrak dan copy semua file yang ada di dalam folder tersebut ke dalam folder public_html dengan cara drag and drop



14. Buka thingspeak dan masuk ke channel temperature dan humidity yang sudah dibuat pada experiment 6

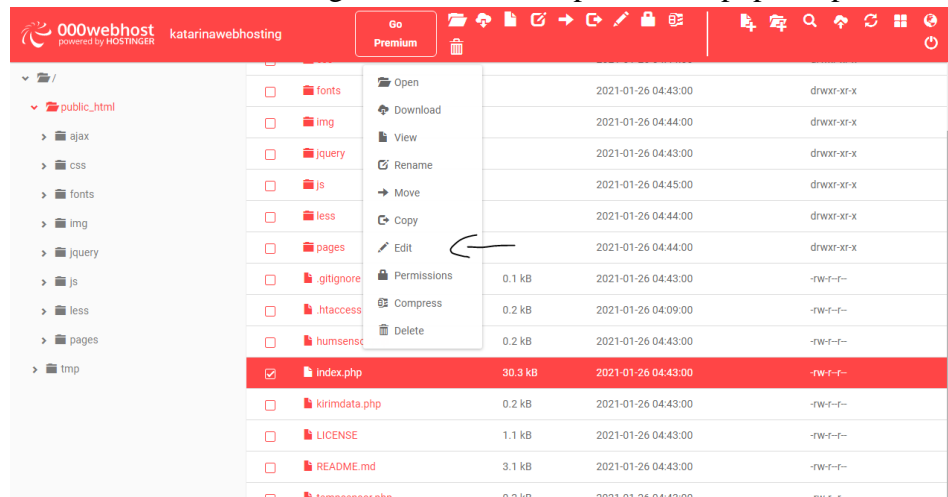


15. Pada field 1 chart klik icon yang ditunjuk oleh panah, kemudian copy teks yang muncul pada halaman

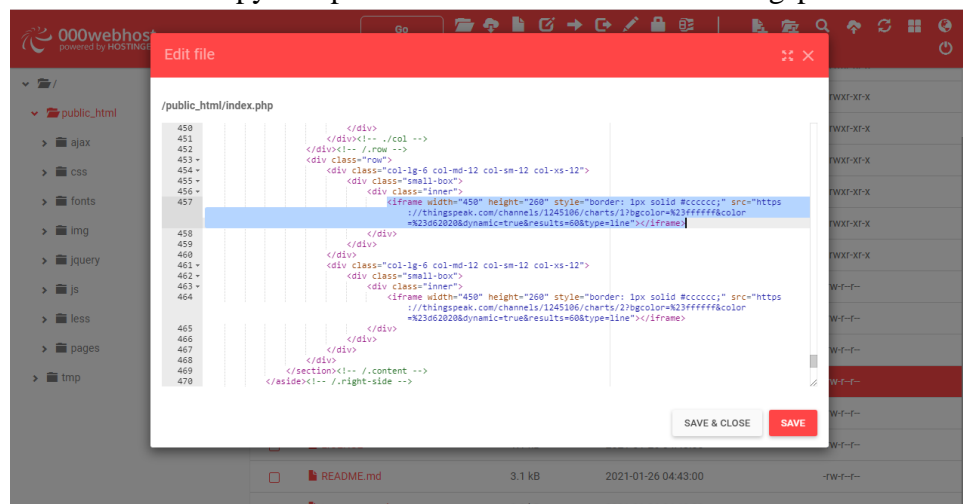
The screenshot shows the ThingSpeak web interface. At the top, there's a navigation bar with 'Channels', 'Apps', and 'Support' menus. Below this, the 'Channel Stats' section displays 'Created: about a month ago', 'Last entry: less than a minute ago', and 'Entries: 1213'. There are two charts: 'Field 1 Chart' and 'Field 2 Chart'. Both charts show 'Temperature dan Humidity' data over time. A black arrow points to the share icon (a square with a plus sign) in the top right corner of the 'Field 1 Chart'. A pop-up window titled 'Field 1 Chart IFrame' is open, displaying the following HTML code:

```
<iframe width="450" height="260" style="border: 1px solid #cccccc;" src="http://..."/>
```

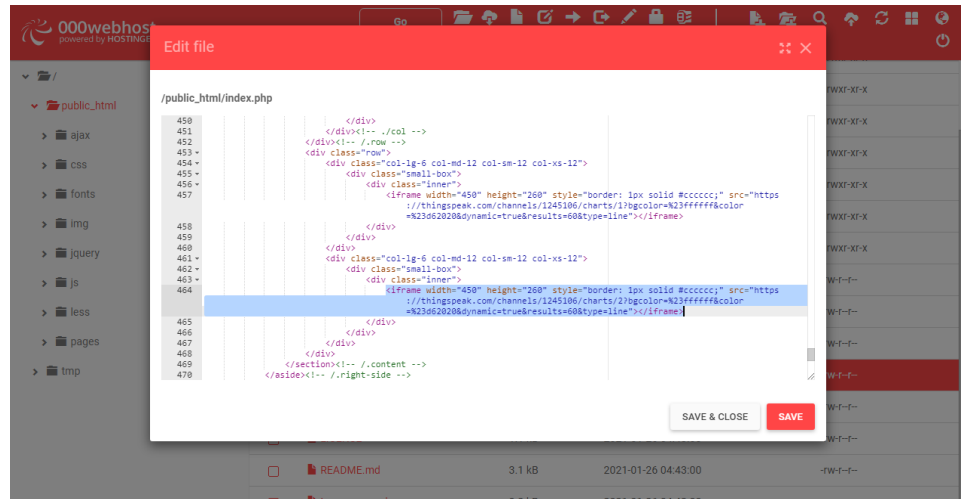
16. Buka kembali files manager dan klik kanan pada index.php dan pilih edit



17. Pada baris 457 copy dan paste teks iframe dari website thingspeak

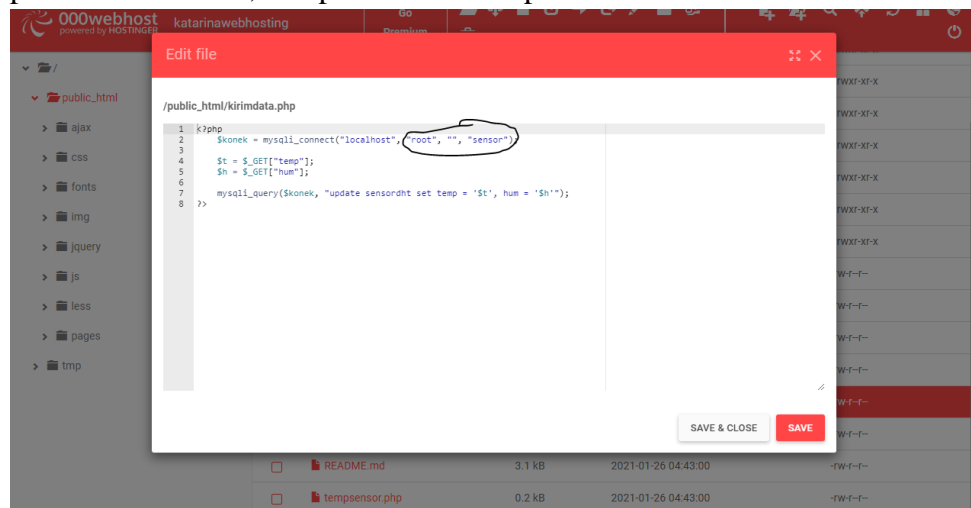


18. Lakukan hal yang sama pada field 2 chart dan copy paste teks tersebut pada baris 464



```
450 </div>
451 </div>
452 </div>
453 <div class="row">
454 <div class="col-lg-6 col-md-12 col-sm-12 col-xs-12">
455 <div class="small-box">
456 <div class="inner">
457 <iframe width="450" height="260" style="border: 1px solid #cccccc;" src="https://thingspeak.com/channels/1245186/charts/2?bgcolor=#238b45&color=#238b45&dynamic=true&results=60&type=line"></iframe>
458 </div>
459 </div>
460 <div class="col-lg-6 col-md-12 col-sm-12 col-xs-12">
461 <div class="small-box">
462 <div class="inner">
463 <iframe width="450" height="260" style="border: 1px solid #cccccc;" src="https://thingspeak.com/channels/1245186/charts/2?bgcolor=#238b45&color=#238b45&dynamic=true&results=60&type=line"></iframe>
464 </div>
465 </div>
466 </div>
467 </div>
468 </div>
469 </div>
470 </div>
```

19. Selanjutnya, edit file kirimdata.php dengan cara mengklik kanan -> edit. Setelah itu, ubah parameter yang dilingkari dengan informasi database yang sudah dibuat tadi. Parameter kedua diisi dengan DB User, parameter ketiga diisi dengan password database, dan parameter keempat diisi oleh DB Name.



```
1 <?php
2 $koneksi = mysqli_connect("localhost", "root", "", "sensor");
3
4 $t = $_GET["temp"];
5 $h = $_GET["hum"];
6
7 mysqli_query($koneksi, "update sensor set temp = '$t', hum = '$h'");
8 >>
```



My Sites | Power Store | Help

000webhost powered by HOSTINGER

Upgrade

suryaanthony2

katarinawebhosting

Current site | Free

Home

Tools

Set Web Address

File Manager

Database Manager

Email Manager

All websites > Katarinawebhosting

My Databases

+ New Database 1/2

DB Name	DB User	DB Host	
id16010159_sensordht	id16010159_katarina	localhost	Manage

Database Manager

Here you can create and manage all your databases.

- Database Space Limit: 1GB
- Database Table Limit: 100 Tables
- Manage your database at databases.000webhost.com
- Use localhost as a connection hostname.

Edit file

```
/public_html/kirimdata.php
1 <?php
2 $koneksi = mysqli_connect("localhost", "id16010159_katarina", "Password123!", "id16010159_sensordht");
3
4 $t = $_GET["temp"];
5 $h = $_GET["hum"];
6
7 mysqli_query($koneksi, "update sensor set temp = '$t', hum = '$h'");
8 >>
```

SAVE & CLOSE SAVE

public_html

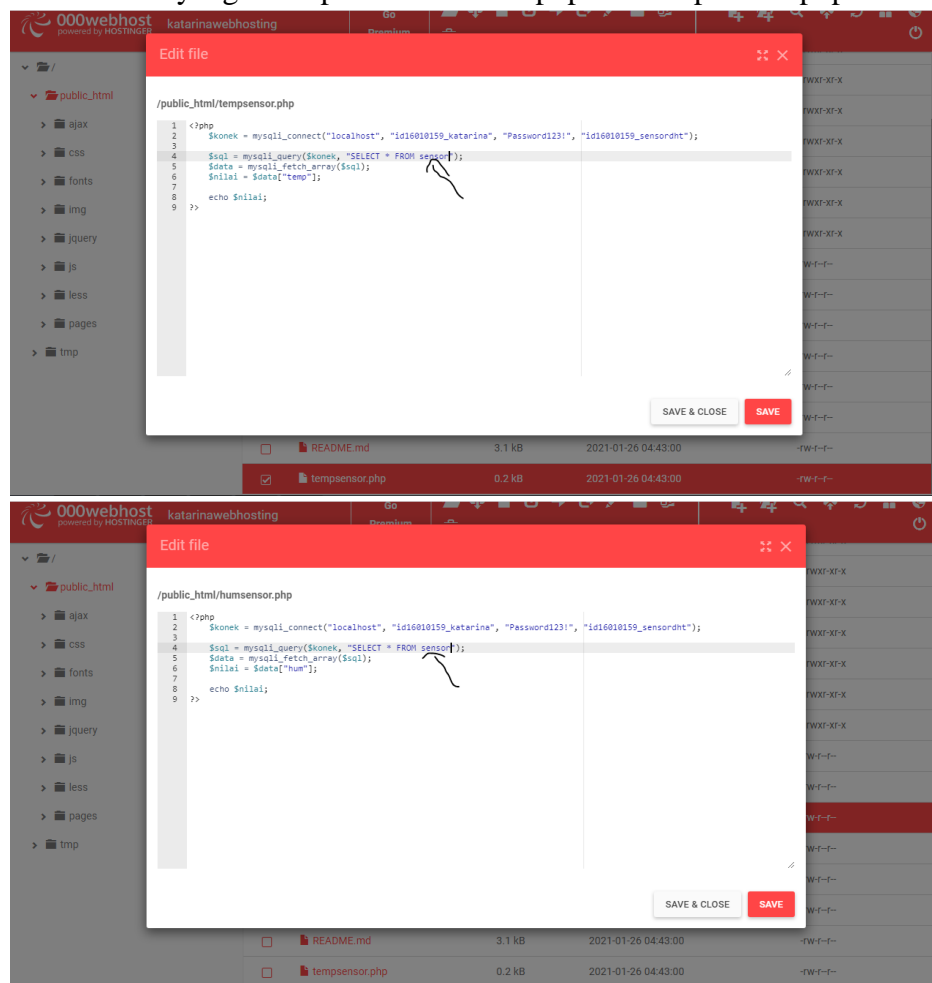
- ajax
- css
- fonts
- img
- jquery
- js
- less
- pages
- tmp

README.md 3.1 kB 2021-01-26 04:43:00

tempsensor.php 0.2 kB 2021-01-26 04:43:00



20. Lakukan hal yang sama pada humsensor.php dan tempsensor.php



21. Buka google drive experiment_7 dan download kode arduino experiment_7.ino



22. Isi parameter yang kosong dengan nilai yang sesuai dengan proyek masing-masing.

```
Experiment_7
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ThingSpeak.h>
#include <DHT.h>

#define DHTTYPE DHT11
#define DHTPIN 14

DHT dht(DHTPIN, DHTTYPE);
WiFiClient client;

//masukkan ssid dan password wifi
const char* ssid = ".....";
const char* password = ".....";

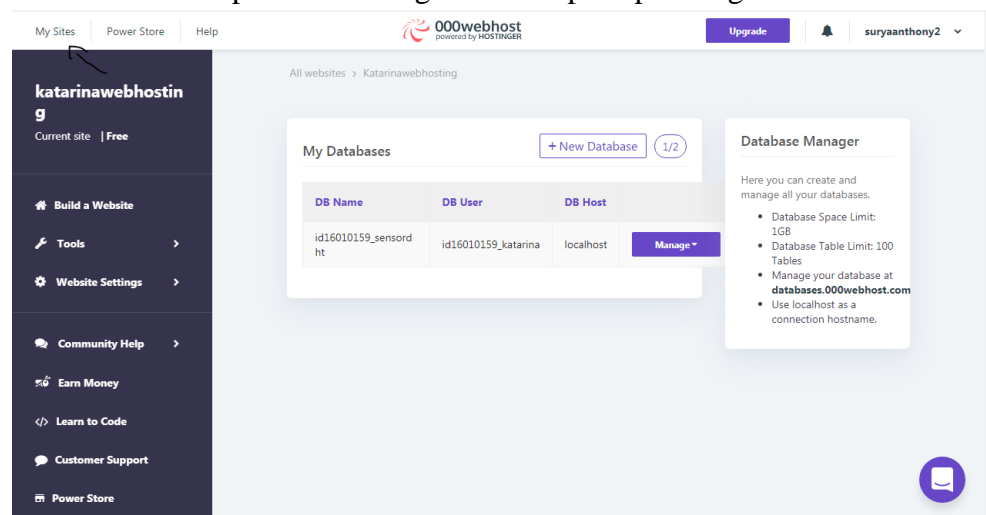
//masukkan nama website
const char* host = ".....";

//masukkan myChannelNumber dan myWriteAPIKEY channel thingspeak
unsigned long myChannelNumber = ....;
const char* myWriteAPIKey = ".....";

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  WiFi.hostname("NodeMCU");
  WiFi.begin(ssid, password);
  ThingSpeak.begin(client);
  dht.begin();

  while(WiFi.status() != WL_CONNECTED) {
```

23. Nama website dapat dilihat dengan cara seperti pada digambar





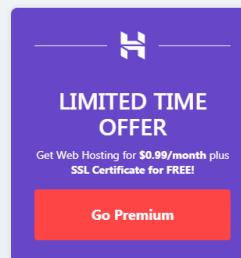
My Websites

Select a website to manage, or create a new one from scratch.

+ Create New Site



katarinawebhosting
Status: running
[https://katarinawebhosting.000webho...](https://katarinawebhosting.000webhostapp.com)



Browser address bar: <https://katarinawebhosting.000webhostapp.com>

AdminLTE Dashboard

Hello, Jane
Online

Search...

Dashboard Widgets Charts UI Elements Forms Tables Calendar Mailbox Examples

Dashboard Control panel

0 Suhu saat ini

0 Kelembapan saat ini

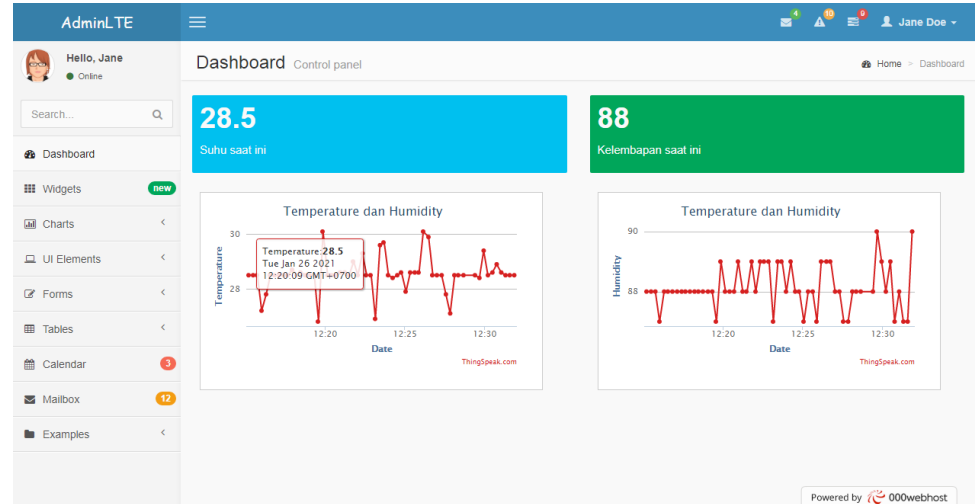
Temperature dan Humidity

Temperature dan Humidity

Powered by 000webhost



24. Jika percobaan berhasil maka nilai pada suhu dan kelembapan yang tadinya 0 akan berubah



Percobaan sukses saat kita sudah dapat melihat grafik dan tampilan web sesuai yang kita inginkan.