

## **MODUL 4**

### **PRAKTIKUM MONITORING SUHU DENGAN THINGSPEAK**

#### **A. Tujuan Percobaan**

1. Dapat mengimplementasikan koneksi ESP32 dengan DHT11 dan OLED Display
2. Dapat mengintegrasikan ESP32 dengan platform thingspeak

#### **B. Alat dan Bahan**

1. ESP32
2. Laptop/komputer
3. Kabel USB *micro* B
4. Arduino IDE
5. DHT11

#### **C. Dasar Teori**

Thingspeak.com merupakan platform Internet of Things dibagian cloud dimana kita dapat mengirim atau menerima suatu data dengan protokol komunikasi HTTP dan juga dapat menampilkan nilai data melalui dashboard gratis yang diberikan.

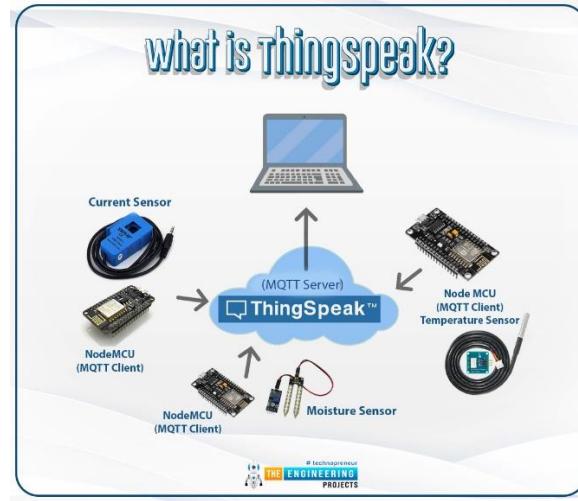
#### **Fungsi Thingspeak.com**

Thingspeak.com berfungsi sebagai pengumpul data yang berasal dari perangkat node berupa sensor-sensor yang sudah terhubung ke internet dan juga memungkinkan pengambilan data dari perangkat lunak untuk keperluan visualisasi ,notifikasi, kontrol dan analisis historis data.

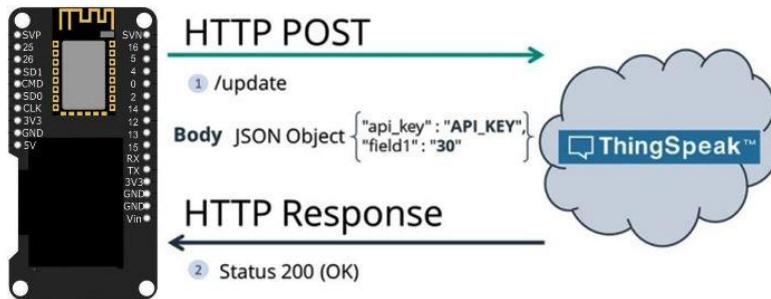
Unsur utama dari ThingSpeak adalah saluran, yang berisi bidang data, bidang lokasi, dan bidang status. Setelah membuat saluranThingSpeak, kita bisa menulis data ke saluran proses dan melihat hasil data lewat MATLAB. Dari situ dapat dilihat reaksi terhadap data dengan *tweet* dan alert lainnya.

#### **Fitur- Fitur Dari Thingspeak.com**

- Open Api
- Real-time data collection
- Geolocation data
- Data prosessing
- Data Visualizations
- Device status messages
- Plugins



Gambar 1. Platform thingspeak



Gambar 2. Komunikasi dengan HTTP

### HTTP POST Request Method

Hypertext Transfer Protocol (HTTP) adalah antara perkataan yang amat sinonim dengan zaman sekarang. Ianya berfungsi sebagai protokol komunikasi diantara client dan server. Di dalam kes kita kali ini, client adalah ESP32 manakala server adalah ThingSpeak. Berikut adalah contoh komunikasi diantara Client dan Server melalui komunikasi HTTP :

- ESP32 (yang bertindak sebagai Client) menghantar HTTP request ke ThingSpeak (yang bertindak sebagai Server).
- ThingSpeak (yang bertindak sebagai Server) membalas *respon* kepada ESP32 (yang bertindak sebagai Client) melalui komunikasi HTTP.
- Akhirnya, *respon* yang diterima ESP32 (yang bertindak sebagai Client) mengandungi info seperti status atau maklumat maklumat tertentu.

### HTTP POST

Kata kunci **POST** digunakan untuk menghantar data ke Server. Contohnya, ESP32 perlu menghantar kata kunci POST bersama sama data sensor bagi membolehkannya disimpan di dalam Server ThingSpeak. Maklumat maklumat yang dikirim bersama sama kata kunci POST melalui HTTP Request adalah seperti berikut :

```
POST /update HTTP/1.1
Host: api.thingspeak.com
api_key=api&field1=value
Content-Type: application/x-www-form-urlencoded
```

Walau bagaimanapun, HTTP Request yang diterangkan di atas telah dipermudahkan kepada kita melalui hanya satu fungsi berikut :

```
ThingSpeak.writeFields (myChannelNumber, myWriteAPIKey);
```

Untuk mengirim 1 data.

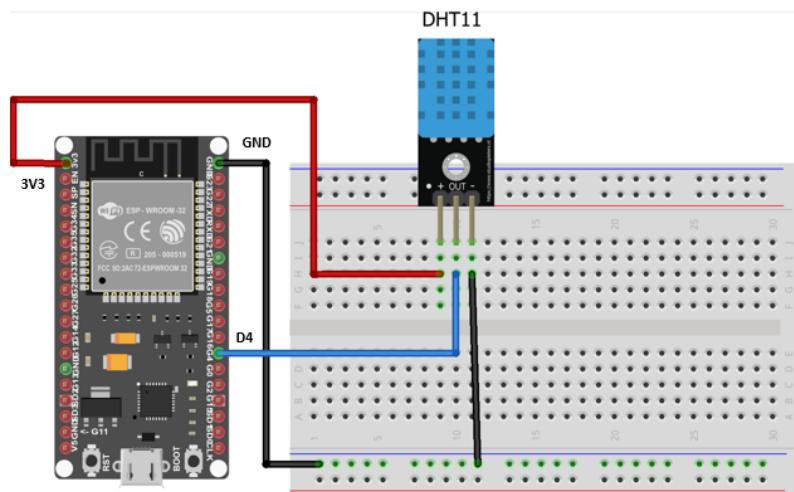
```
int writeField(channelNumber, field, value, writeAPIKey);
```

Untuk mengirim lebih dari 1 data.

```
int setField (field, value);
int writeFields (channelNumber, writeAPIKey);
```

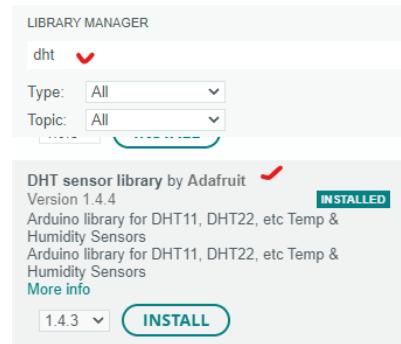
## D. Langkah Percobaan

### D.1. Monitoring suhu dengan DHT 11



Gambar 3. Rangkaian Percobaan

- Rangkailah seperti gambar 3.
- Jalankan Arduino IDE pada laptop/komputer.
- Install library dht



Gambar 5. Library DHT

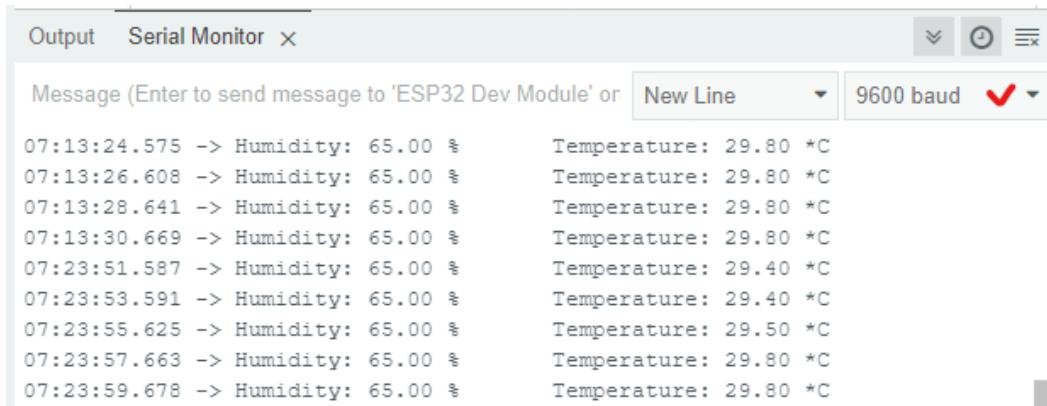
- Tambahkan kode berikut pada Arduino IDE.

```

#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  Serial.println("Deteksi Suhu dan kelembaban");
  dht.begin();
}
void loop() {
  delay(2000);
  float t = dht.readTemperature();
  float h = dht.readHumidity();
  if (isnan(h) || isnan(t)) {
    Serial.println("Sensor tidak terbaca!");
    return;
  }
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.println(" *C ");
}

```

- e. Koneksikan ESP32 dengan Laptop/Komputer menggunakan kabel USB.
- f. Pastikan Board dan Port yang dipilih pada bagian Tools Arduino IDE sesuai.
- g. Klik ikon Verify (Ctrl+R) pada Arduino IDE untuk mengecek adanya error dan meng-compile kode.
- h. Jika tidak terdeteksi error, klik ikon Upload (Ctrl + U) untuk mengunggah kode ke dalam board ESP32. Setelah proses pengunggahan selesai, *serial monitor* akan menampilkan pesan seperti gambar di bawah ini.



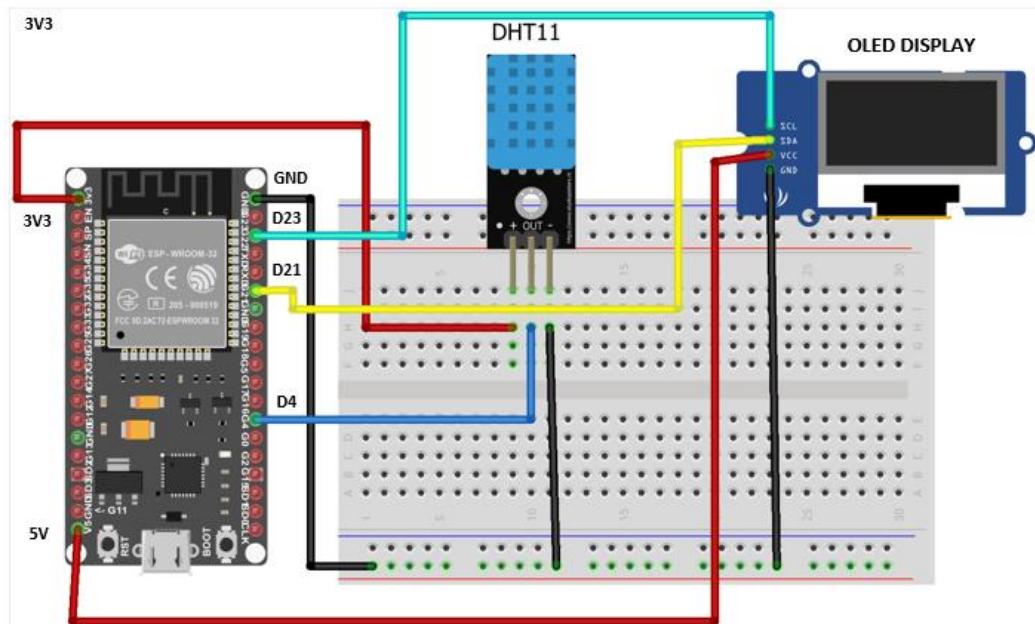
The screenshot shows the Arduino Serial Monitor window. The title bar says "Output" and "Serial Monitor". The main area displays a log of sensor readings. At the top, there is a message input field with placeholder text "Message (Enter to send message to 'ESP32 Dev Module' or", a "New Line" dropdown set to "New Line", and a "9600 baud" dropdown with a red checkmark. Below this, the log shows the following data:

```

07:13:24.575 -> Humidity: 65.00 %      Temperature: 29.80 *C
07:13:26.608 -> Humidity: 65.00 %      Temperature: 29.80 *C
07:13:28.641 -> Humidity: 65.00 %      Temperature: 29.80 *C
07:13:30.669 -> Humidity: 65.00 %      Temperature: 29.80 *C
07:23:51.587 -> Humidity: 65.00 %      Temperature: 29.40 *C
07:23:53.591 -> Humidity: 65.00 %      Temperature: 29.40 *C
07:23:55.625 -> Humidity: 65.00 %      Temperature: 29.50 *C
07:23:57.663 -> Humidity: 65.00 %      Temperature: 29.80 *C
07:23:59.678 -> Humidity: 65.00 %      Temperature: 29.80 *C

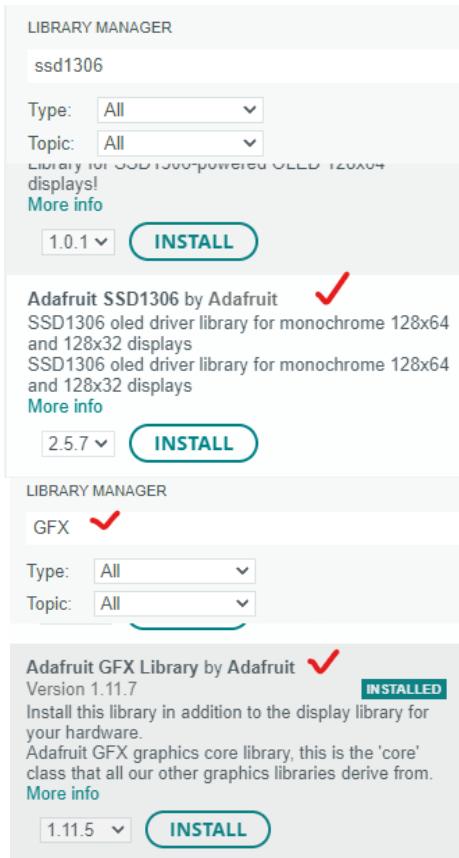
```

## D.2. Tambahkan OLED Display



Gambar 6. Rangkaian percobaan dengan OLED

- Tambahkan OLED Display pada rangkaian.
- Install library pada Arduino IDE



Gambar 7. Library OLED

- c. Tambahkan library dan inisial pengaturan berikut ini:

```
//Tambahkan utk OLED
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <WiFi.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration SSD1306 display to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

- d. Tambahkan kode di bagian fungsi setup: `void setup ()`

```
//Tambahkan utk OLED
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
}
delay(500);
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
```

- e. Tambahkan kode di bagian fungsi loop: `void loop ()`

```
//Tambahkan utk OLED
display.clearDisplay();
display.setCursor(0, 0);
display.println("Monitoring Suhu\n");
display.print("Hum: ");
display.print(h);
display.println(" %");
display.print("Temp: ");
display.print(t);
display.println(" *C");
display.display();
```

- f. Compile dan upload program tersebut dan amati tampilan pada OLED Display.

### D.3. Interkoneksi dengan thingspeak

- a. Akses web thingspeak:  
<https://thingspeak.com/>



Gambar 8. Website thingspeak

- b. Lakukan registrasi bila belum mempunyai akun di thingspeak

Create MathWorks Account

Email Address  !  
Missing required information

To access your organization's MATLAB license, use your school or work email.

Location

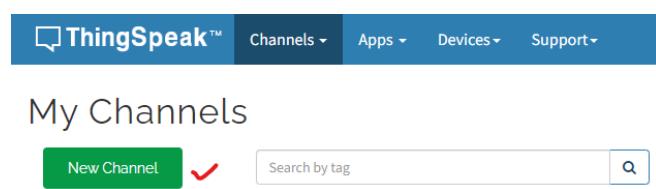
First Name

Last Name

**Continue**

Gambar 9. Pendaftaran akun

- c. Setelah registrasi, lakukan **Sign in**. Kemudian, buatlah **Channel** yang berisi **field** berupa hasil pengiriman dari sensor.



Gambar 10. Pembuatan Channel

- d. Buatlah field: **Suhu** dan **Kelembaban** untuk menerima data dari sensor DHT11.  
Klik "**Save Channel**" di bagian bawah.

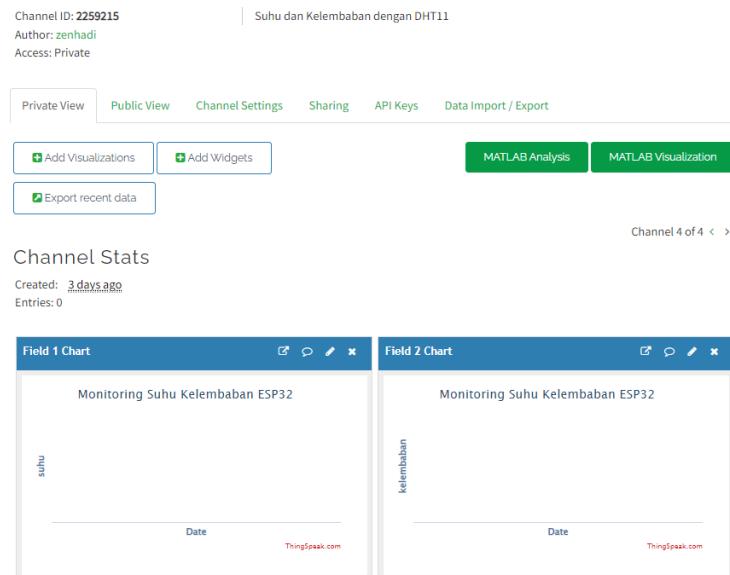
## New Channel

Name	Monitoring Suhu Kelembaban ESP32
Description	Suhu dan Kelembaban dengan DHT11
Field 1	suhu <input checked="" type="checkbox"/>
Field 2	kelembaban <input checked="" type="checkbox"/>

Gambar 11. Membuat Channel dan 2 field

- e. Catat **Channel ID** yang akan dimasukkan ke dalam program di Arduino IDE.

### Monitoring Suhu Kelembaban ESP32



Gambar 12. Field Chart

- f. Catat **API Key** yang akan dimasukkan ke Arduino IDE.

### Monitoring Suhu Kelembaban ESP32

Channel ID: 2259215 | Suhu dan Kelembaban dengan DHT11  
Author: zenhadi  
Access: Private

Private View Public View Channel Settings Sharing API Keys Data I

Write API Key

Key JDCCCJ8Q97VMA930 ✓

Generate New Write API Key

### Read API Keys

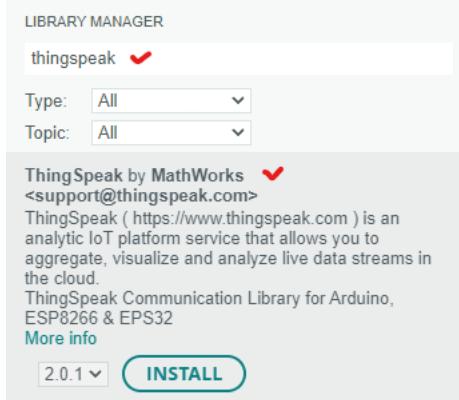
Key	N3EECCQQ4JH2HFZ4
Note	

Gambar 13. API Key

Write API Key: utk mengirim data ke channel.

Read API Key: utk mengijinkan orang lain melihat data di channel kita.

- g. Install library **thingspeak** pada Arduino IDE.



Gambar 14. Library thingspeak

- h. Masukkan kode program dibawah berikut yang sudah integrasi DHT11, modul OLED Display, koneksi ke thingspeak.

```
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT11

//Tambahkan utk OLED
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <WiFi.h>

#include "ThingSpeak.h" //header utk thingspeak

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

//Setting untuk WiFi
// ganti dengan API Keys anda di Thingspeak
#define CHANNEL_ID 2259215
#define CHANNEL_API_KEY "JDCCCJ8Q97VMA930"

const char* ssid = "AishaNet"; //masukkan dari tethering hp anda
const char* password = "████████";
const char* server = "api.thingspeak.com";
WiFiClient client;

DHT dht(DHTPIN, DHTTYPE);
void setup() {
  //Serial.begin(9600);
  Serial.begin(115200);
```

```

Serial.println("Deteksi Suhu dan kelembaban");
dht.begin();
//Tambahkan utk OLED
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
}
delay(500);
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);

//Initial WiFi
WiFi.begin(ssid, password);
Serial.println();
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");

ThingSpeak.begin(client); // Initialize ThingSpeak
}
void loop() {

float t = dht.readTemperature();
float h = dht.readHumidity();
if (isnan(h) || isnan(t)) {
    Serial.println("Sensor tidak terbaca!");
    return;
}
if (client.connect(server,80)) {
    ThingSpeak.setField(1,t);
    ThingSpeak.setField(2,h);
    int x = ThingSpeak.writeFields(CHANNEL_ID, CHANNEL_API_KEY);
    if(x == 200){
        Serial.println("Channel update successful.");
    }
    else{
        Serial.println("Problem updating channel. HTTP error code " +
String(x));
    }
}

```

```

Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" *C ");
Serial.println("% send to Thingspeak");

//Tambahan utk OLED
display.clearDisplay();
display.setCursor(0, 0);
display.println("Monitoring Suhu\n");
display.print("Hum: ");
display.print(h);
display.println(" %");
display.print("Temp: ");
display.print(t);
display.println(" *C");
display.display();
}

client.stop();
Serial.println("Waiting...");
delay(20000);
}

```

i. Amati hasil di serial monitor seperti dibawah ini.

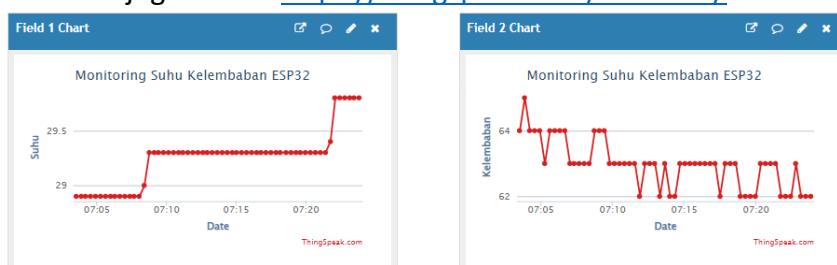
```

Output Serial Monitor X
Message (Enter to send message to 'ESP32 Dev Module' on 'COM6')
New Line 115200 baud
08:01:34.907 ->
08:01:34.907 -> Connecting to AishaNet
08:01:35.388 -> ..
08:01:35.911 -> WiFi connected
08:01:39.757 -> Channel update successful.
08:01:39.757 -> Humidity: 63.00 %      Temperature: 29.30 *C % send to Thingspeak
08:01:39.789 -> Waiting...
08:02:00.761 -> Channel update successful.
08:02:00.761 -> Humidity: 63.00 %      Temperature: 29.30 *C % send to Thingspeak
08:02:00.784 -> Waiting...
08:02:21.761 -> Channel update successful.
08:02:21.761 -> Humidity: 63.00 %      Temperature: 29.40 *C % send to Thingspeak
08:02:21.761 -> Waiting...

```

Gambar 15. Hasil di serial monitor

j. Amati juga di web <https://thingspeak.com/channels/>

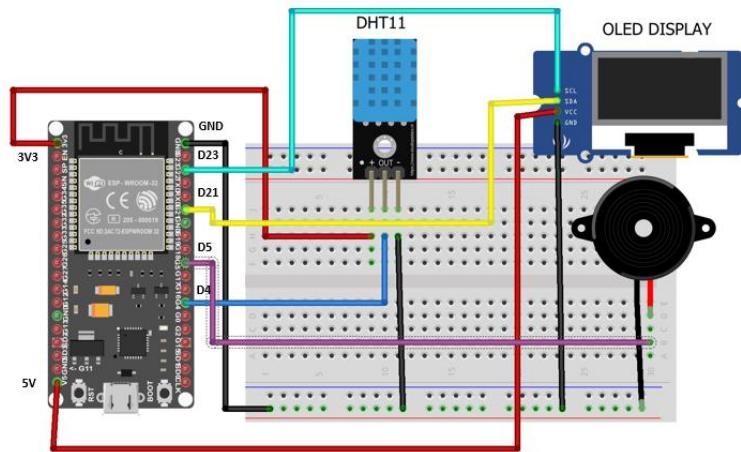


Gambar 16. Hasil di thingspeak

- k. Lakukan analisis data di thingspeak. Pilih **Apps | Matlab Analysis**.

## E. Tugas Praktikum

1. Tambahan buzzer ke dalam rangkaian tsb.



Buat threshold pada nilai suhu/kelembaban, bila mencapai nilai tertentu maka akan membunyikan buzzer, dan bila dibawah nilai tersebut maka buzzer akan mati.

2. Buatlah aplikasi web untuk melakukan monitoring dan control action.

### MONITORING SUHU DAN KELEMBABAN BERBASIS WEBSITE

Klik Tombol berikut untuk monitoring suhu dan kelembaban  
melalui web thingspeak.com

**Monitoring**

Klik Tombol berikut untuk on/off buzzer  
Buzzer ON

**ON**