

# **GUIDE BOOK PROJECT**

**INTERNET OF THINGS**



**JAKARTA GLOBAL UNIVERSITY**

# MODUL 1

## PERANCANGAN INTERNET OF THINGS DENGAN NODE MCU ESP8266, DHT 11 DAN THINKSEPAK

### Alat dan Bahan

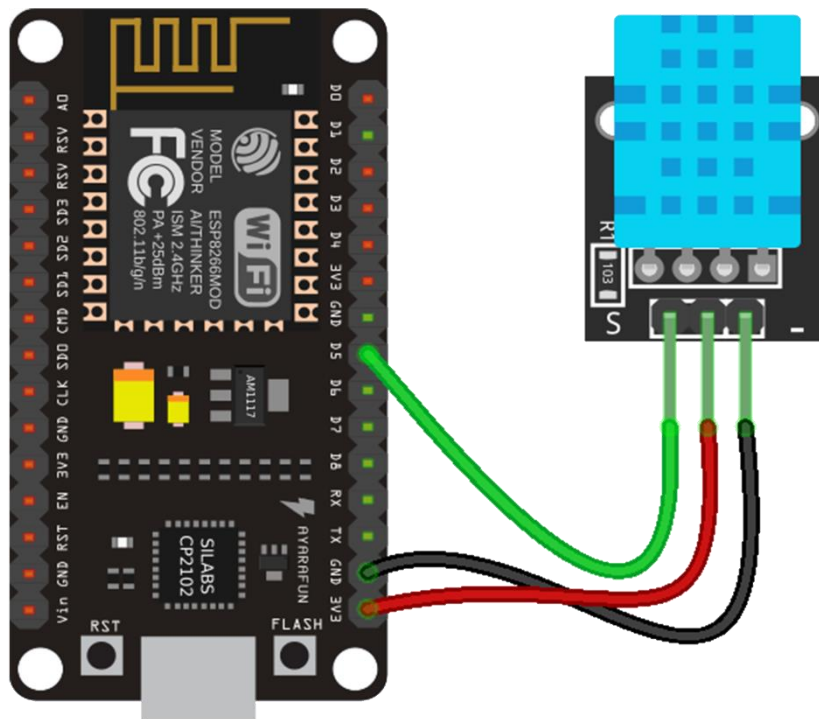
#### Hardware

- |                         |        |
|-------------------------|--------|
| 1. Node MCU ESP 8266    | 1 buah |
| 2. Sensor DHT 11        | 1 buah |
| 3. Kabel Jumper         |        |
| 4. Kabel USB power      | 1 buah |
| 5. Power bank / adaptor | 1 buah |

#### Software

1. Arduino IDE
2. CH341SER

### Konfigurasi Hardware



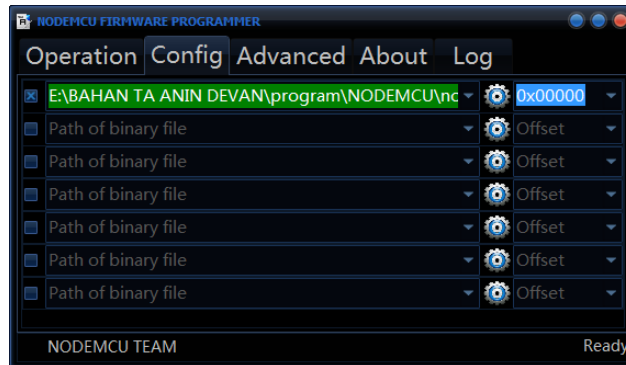
Koneksikan sensor dan Node MCU ESP8266 dengan sensor DHT 11. Pin Digital yang terkoneksi menyesuaikan program arduinonya. Gambar diatas terkoneksi ke pin D5.

### Konfigurasi Perancangan Antarmuka Sensor DHT 11 dengan Node MCU, dan Thingspeak.

Untuk menjalankan *node* sensor pada sistem diperlukan program. Program dibuat dengan menggunakan *Software* IDE Arduino untuk kemudian dapat dilakukan *verify* dan dimasukkan ke

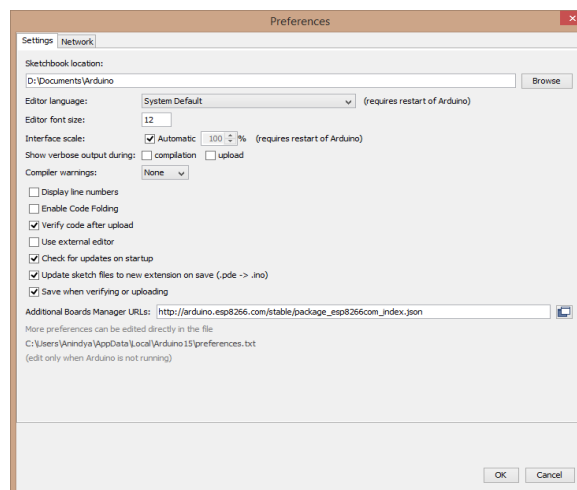
NodeMCU ESP8266. Berikut cara konfigurasi program pada arduino IDE.

1. Buka repositori <https://github.com/aninanandah/TEMPERATURE-HUMIDITY-MONITORING-IOT-MQTT.git>
2. Install file CH341SER yang terdapat pada repository
3. Lakukan *flashing* pada perangkat NodeMCU ESP8266 dengan NodeMCU *firmware* programmer. Pilih COM → *Config* → Masukkan file *flashing node* MCU



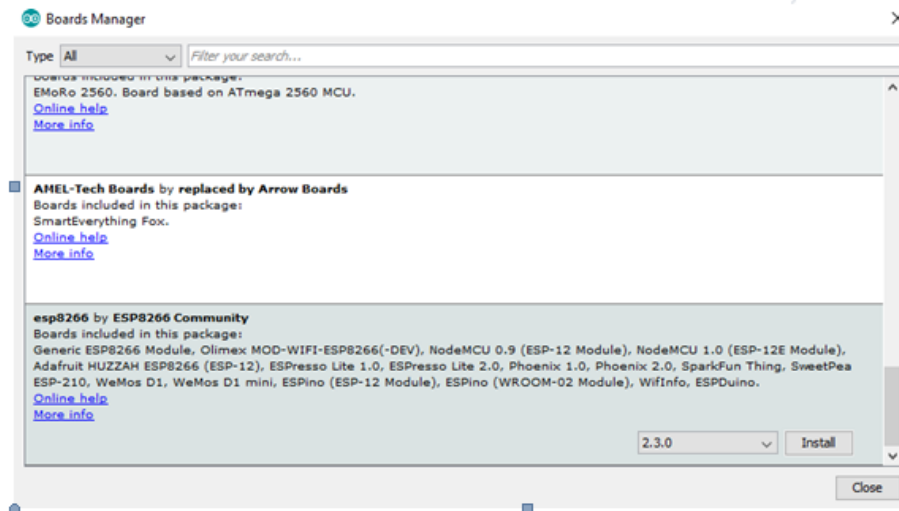
#### *Flashing NodeMCU ESP8266*

4. Install Arduino IDE yang dapat kalian download di website Arduino. Install dan jalankan arduino IDE. Install komunikasi serial untuk Node MCU pada folder CH341SER. Selanjutnya install board pada board manager di Arduino IDE.
5. Melakukan instalasi *board* NodeMCU ESP8266 pada Arduino IDE. Masuk ke Arduino IDE → *Files* → *Preferences* lalu masukkan URL [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json). Pada *Additional Boards Manager*, Lalu klik OK.



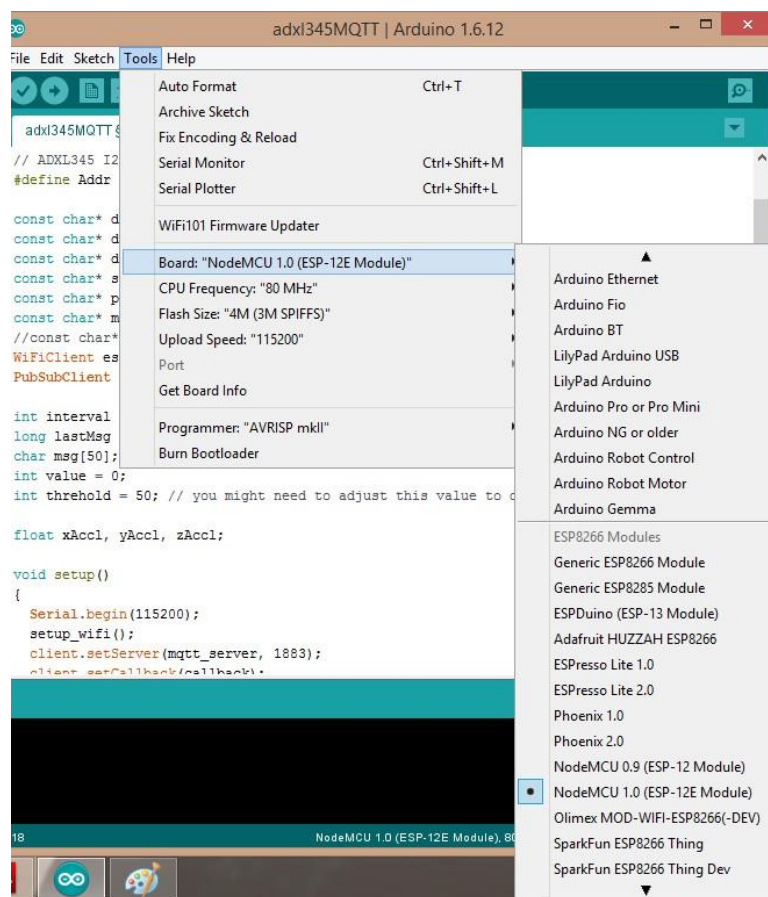
Gambar 1 *Install Node MCU* pada arduino IDE

6. Masuk ke *Tools* lalu *board*, dan pilih *board manager* . dan cari *esp8266 by ESP8266 community*.



Gambar 2 Download ESP8266 board manager

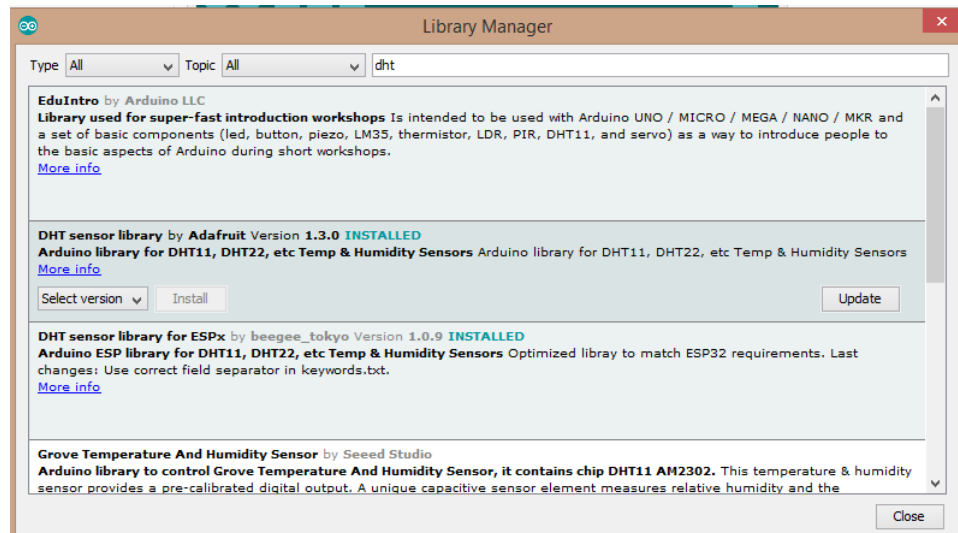
Setelah selesai proses *download* akan muncul *board* NodeMCU ESP8266 pada *board manager*.



Gambar 3 NodeMCU ESP8266 pada *board manager* Arduino IDE

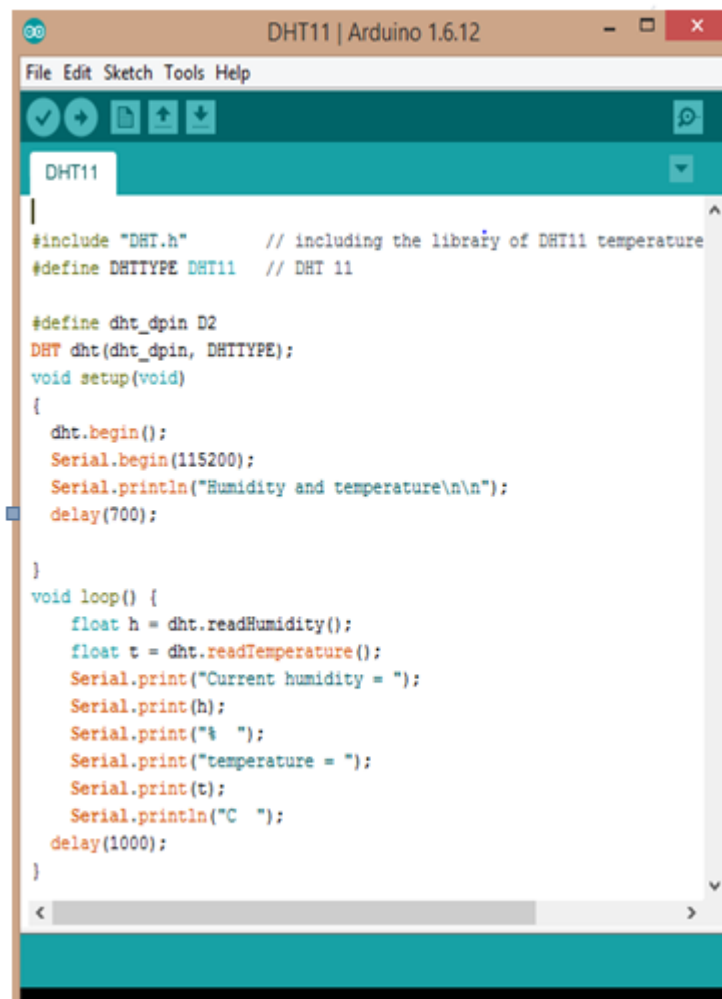
7. Mengecek koneksi *port* yang terhubung dengan NodeMCU ESP8266 pada *Tools*→*Port* lalu pilih *Port* NodeMCU ESP8266 , pilih *board* NodeMCU ESP8266 yang sebelumnya telah di *Install* pada Arduino IDE pilih *Node MCU 1.0*.

8. Menginstall library DHT 11 sensor yang akan digunakan pada Arduino IDE



Gambar 4 menginstall DHT 11 sensor library

9. Memasukkan program sesuai dengan kebutuhan disesuaikan terhadap *node* sensor. dengan menggunakan perintah *upload* seperti pada gambar



Gambar 5 Konfigurasi Program *Node* Sensor

10. Jalankan program dan coba buka serial monitor. Lihat hasilnya dan dokumentasikan! Apakah data sensor sudah terbaca dengan benar.

11. Selanjutnya adalah menghubungkan Node MCU , sensor DHT , atau node sensor ke antarmuka Thingspeak. Buat akun atau registrasi pada <https://thingspeak.com/>.

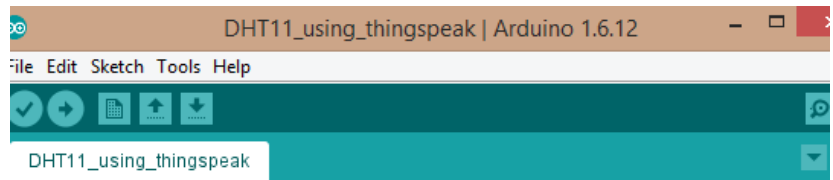


Gambar 6 Konfigurasi Thingspeak.com

12. Setelah registrasi (Sign UP) lanjutkan dengan Sign In dan kemudian buat channel pada thingspeak. Dan field untuk memvisualisasikan data pembacaan sensor.

Gambar 7. Pembuatan Channel pada Thingspeak.com

13. Buat program pada Arduino IDE untuk menghubungkan sensor dengan Thingspeak dan data dapat terupload melalui internet. Lalu data dapat divisualisasikan dan tersimpan melalui cloud Thingspeak. Masukkan API Thingspeak yang didapatkan pada account kalian dan koneksikan dengan program.



```
// Simple code upload the tempeature and humidity data using thingspeak.com
// Hardware: NodeMCU,DHT11

#include <DHT.h> // Including library for dht
#include <ESP8266WiFi.h>

String apiKey = "HX3J13EHYWT4C1D"; // Enter your Write API key from Thi

const char *ssid = "TOP"; // replace with your wifi ssid and wpa2 key
const char *pass = "jungyonghwablue";
const char* server = "api.thingspeak.com";

#define DHTPIN 2 //pin where the dht11 is connected

DHT dht(DHTPIN, DHT11);

WiFiClient client;

void setup()
{
    dht.begin();
    Serial.begin(115200);
    delay(1000);

    Serial.println("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, pass);

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(1000);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
}

void loop()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    if (isnan(h) || isnan(t))
    {
```

```

return;

if (client.connect(server,80)) // "184.106.153.149" or api.thingsp
{
    String postStr = apiKey;
    postStr += "sfield1=";
    postStr += String(t);
    postStr += "sfield2=";
    postStr += String(h);
    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);

    Serial.print("Temperature: ");

    Serial.print(t);
    Serial.print(" degrees Celcius, Humidity: ");
    Serial.print(h);
    Serial.println("%. Send to Thingspeak.");

}

client.stop();

Serial.println("Waiting...");

// thingspeak needs minimum 15 sec delay between updates,
delay(10000);
}

```



Gambar 8. Program Arduino IDE node sensor dengan thingspeak.com

```

// Simple code upload the tempeature and humidity data using thingspeak.com

// Hardware: NodeMCU,DHT11

#include <DHT.h> // Including library for
dht #include <ESP8266WiFi.h>

String apiKey = "HX3J13EHYWB4C1D"; // Enter your Write API key from
ThingSpeak const char *ssid = "namawifi"; // replace with your wifi ssid and
wpa2 key

const char *pass = "passwordwifi";

const char* server = "api.thingspeak.com";

#define DHTPIN 5 //pin where the dht11 is

```



```

connected DHT dht(DHTPIN, DHT11);

WiFiClient
client; void
setup()
{
    dht.begin();
    Serial.begin(115200);
    delay(1000);
    Serial.println("Connecting
to "); Serial.println(ssid);
    WiFi.begin(ssid, pass);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(1000)
        ;
        Serial.print(
        ".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
}

void loop()
{
    float h =
    dht.readHumidity(); float t =
    dht.readTemperature();
    if (isnan(h) || isnan(t))
    {
        // Serial.println("Failed to read from DHT sensor!");
        return;
    }

    if (client.connect(server,80)) // "184.106.153.149" or api.thingspeak.com
    {
        String postStr =

```

```

apiKey; postStr
+ "&field1=";

postStr +=

String(t); postStr
+ "&field2=";

postStr +=

String(h); postStr
+ "\r\n\r\n";

client.print("POST /update
HTTP/1.1\n"); client.print("Host:
api.thingspeak.com\n");
client.print("Connection: close\n");
client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
client.print("Content-Type: application/x-www-form-
urlencoded\n"); client.print("Content-Length: ");
client.print(postStr.length
h()); client.print("\n\n");
client.print(postStr);

Serial.print("Temperatur
e: "); Serial.print(t);

Serial.print(" degrees Celcius, Humidity: ");
Serial.print(h);

Serial.println("%. Send to Thingspeak.");
}

client.stop();

Serial.println("Waiting...");

// thingspeak needs minimum 15 sec delay between updates,
delay(10000);
}

```

14. Lihat channel yang telah dibuat pada thingspeak.

# My Channels

[New Channel](#)

Search by tag

Name	Created	Updated
MONITORING SUHU	2019-07-01	2019-07-01 15:30
<a href="#">Private</a> <a href="#">Public</a> <a href="#">Settings</a> <a href="#">Sharing</a> <a href="#">API Keys</a> <a href="#">Data Import / Export</a>		

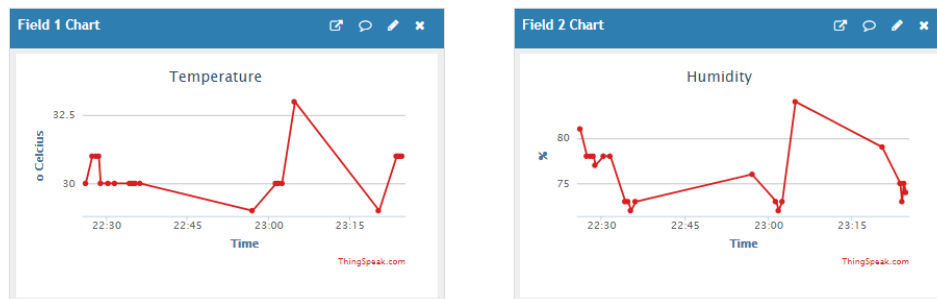
**ThingSpeak™**

[Channels](#) [Apps](#) [Community](#) [Support](#) [Commercial Use](#) [How to Buy](#) [Account](#) [Sign Out](#)

[+ Add Visualizations](#) [+ Add Widgets](#) [Export recent data](#) [MATLAB Analysis](#) [MATLAB Visualization](#)

## Channel Stats

Created: [about 14 hours ago](#)  
Last entry: [about 13 hours ago](#)  
Entries: 21



Gambar 9. Antarmuka visualisasi data sensor dalam bentuk grafik di Thinspeak channel yang telah dibuat.

15. Tampilkan grafik suhu dan kelembaban yang dibaca oleh sensor ke interface Channel Thingspeak kalian ! Buat visualisasi dalam bentuk lainnya seperti gauge, numeric display, dan lamp indikator pada thingspeak.
16. Kirim link visualisasi thinkspeak yang telah kalian buat saat waktu pengumpulan project. Nyalakan modul node sensor kalian sehingga dapat menampilkan hasil pembacaan suhu dan kelembaban dengan metode Internet of Things.

## MODUL 2

### PERANCANGAN INTERNET OF THINGS DENGAN NODE MCU ESP8266, DHT 11 DAN MQTT

#### Alat dan Bahan

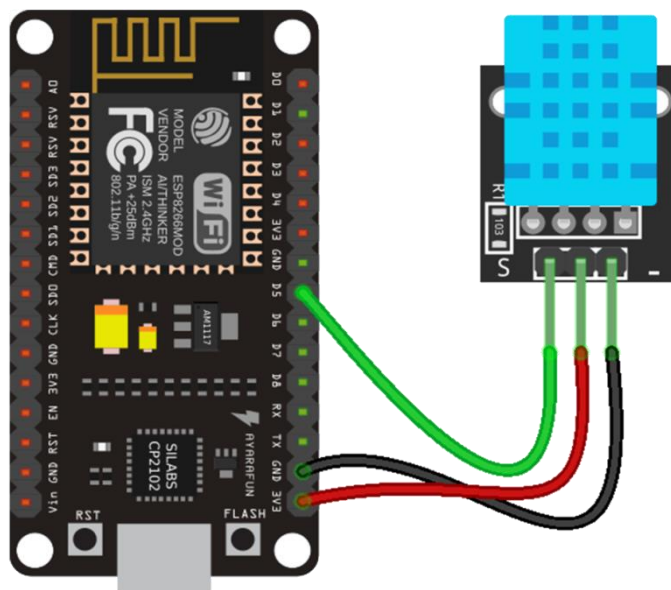
##### Hardware

- |                         |        |
|-------------------------|--------|
| 1. Node MCU ESP 8266    | 1 buah |
| 2. Sensor DHT 11        | 1 buah |
| 3. Kabel Jumper         |        |
| 4. Kabel USB power      | 1 buah |
| 5. Power bank / adaptor | 1 buah |

##### Software

1. Arduino IDE
2. CH341SER
3. Pubsubclient

#### Konfigurasi Hardware

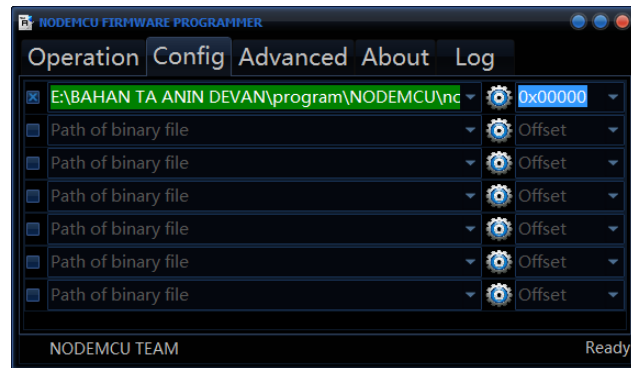


Koneksikan sensor dan Node MCU ESP8266 dengan sensor DHT 11. Pin Digital yang terkoneksi menyesuaikan program arduinonya. Gambar diatas terkoneksi ke pin D5. (Sesuaikan nantinya dengan program kalian).

Untuk menjalankan *node* sensor dan *node* penerima pada sistem diperlukan program. Program dibuat dengan menggunakan *Software* IDE Arduino untuk kemudian dapat dilakukan *verify* dan dimasukkan ke NodeMCU ESP8266. Berikut cara konfigurasi program pada arduino IDE.

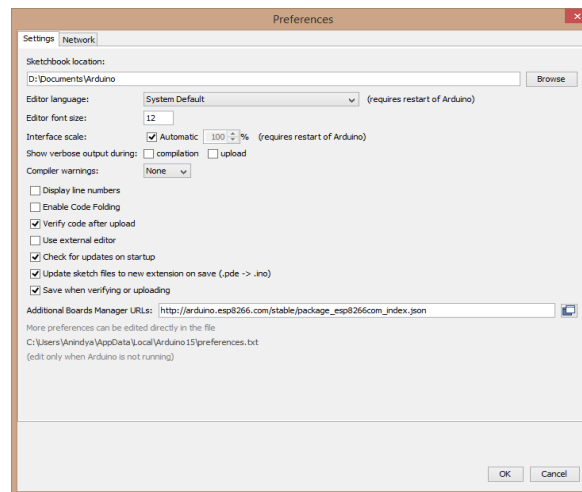
1. Buka repositori <https://github.com/aninanandah/TEMPERATURE-HUMIDITY-MONITORING-IOT-MQTT.git>

2. Install file CH341SER yang terdapat pada repository
3. Lakukan *flashing* pada perangkat NodeMCU ESP8266 dengan NodeMCU *firmware* programmer. Pilih COM → *Config* → Masukkan file *flashing node MCU*



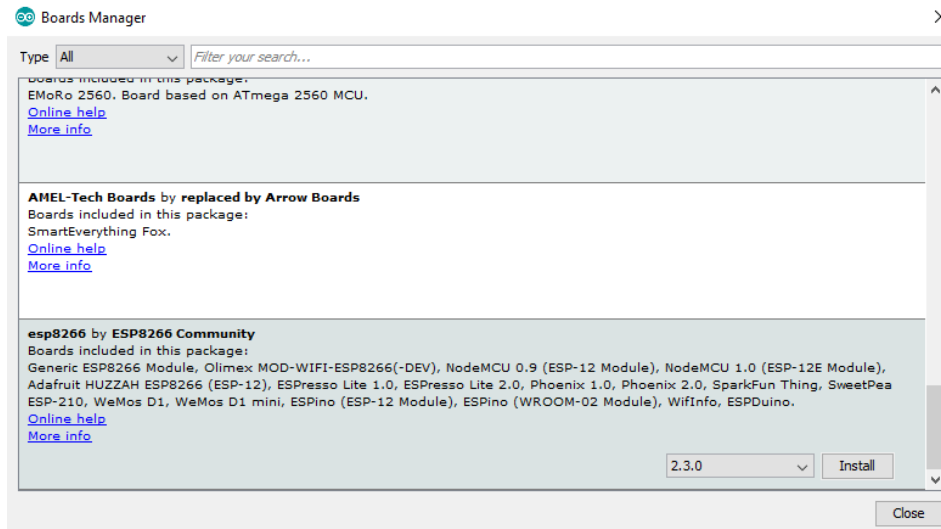
Gambar 1 *Flashing* NodeMCU ESP8266

4. Melakukan instalasi *board* NodeMCU ESP8266 pada Arduino IDE. Masuk ke Arduino IDE → *Files* → *Preferences* lalu masukkan URL [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json). Pada *Additional Boards Manager*, Lalu klik OK.



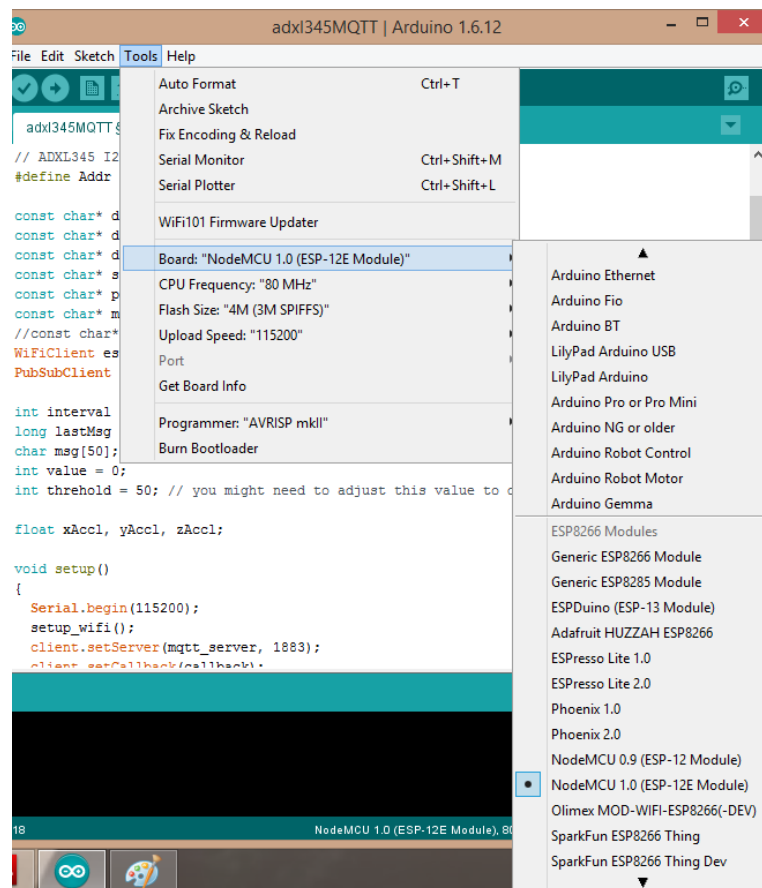
Gambar 2 *Install Node MCU* pada arduino IDE

5. Masuk ke *Tools* lalu *board*, dan pilih *board manager* . dan cari *esp8266 by ESP8266 community*.



Gambar 3 Download ESP8266 board manager

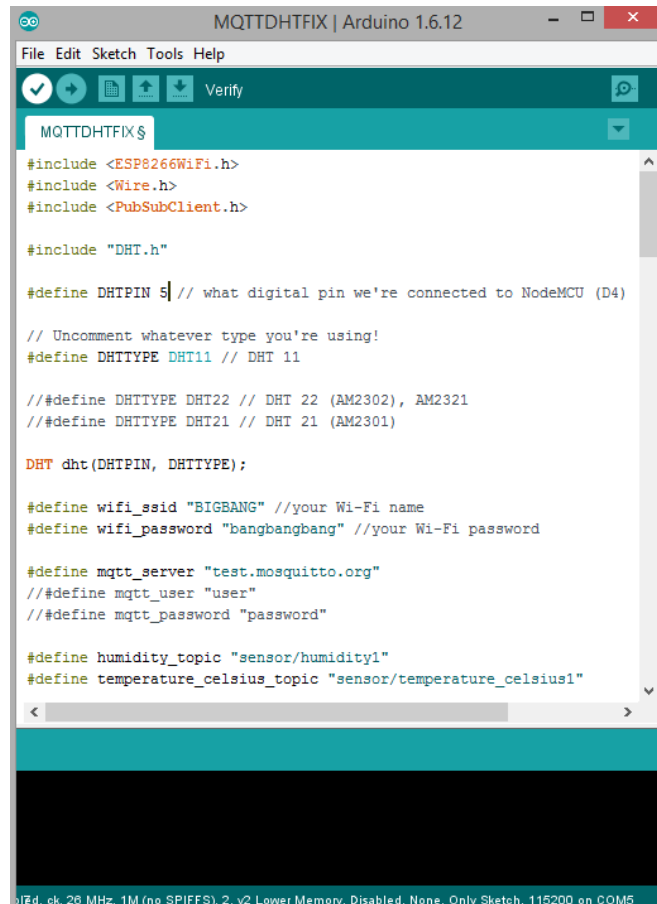
6. Setelah selesai proses *download* akan muncul *board* NodeMCU ESP8266 pada *board manager*.



Gambar 4 NodeMCU ESP8266 pada *board manager* Arduino IDE

7. Mengecek koneksi *port* yang terhubung dengan NodeMCU ESP8266 pada *Tools*→*Port* lalu pilih *Port* NodeMCU ESP8266 , pilih *board* NodeMCU ESP8266 yang sebelumnya telah di *Install* pada Arduino IDE pilih *Node MCU 1.0*.

8. Membuat koding untuk koneksi Wi-Fi nde MCU ke router, program untuk membaca data sensor *dht 11*, *print* data atau baca data sensor, dan koneksi ke MQTT . Simpan program yang telah dibuat.
9. Memasukkan program sesuai dengan kebutuhan disesuaikan terhadap *node* sensor. dengan menggunakan perintah *upload* seperti pada gambar



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

```
#include <Wire.h>
```

```
#include <PubSubClient.h>
```

```
#include "DHT.h"
```

```
#define DHTPIN 5 // what digital pin we're connected to NodeMCU (D4)
```

```
// Uncomment whatever type you're using!
```

```
#define DHTTYPE DHT11 // DHT 11
```

```
// #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
```

```
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
#define wifi_ssid "BIGBANG" //your Wi-Fi name
```

```
#define wifi_password "bangbangbang" //your Wi-Fi password
```

```
#define mqtt_server "test.mosquitto.org" //server yang di gunakan dapat diubah
```

```
//#define mqtt_user "user"
```

```
//#define mqtt_password "password"
```

```
#define humidity_topic "sensor/humidity1_anin" //ganti dengan namamu
```

```
#define temperature_celsius_topic "sensor/temperature_celsius_anin" //ganti dengan  
namamu
```

```
WiFiClient espClient;
```

```
PubSubClient client(espClient);
```

```
void setup() {
```

```
  Serial.begin(115200);
```

```
  dht.begin();
```

```
  setup_wifi();
```

```
  client.setServer(mqtt_server, 1883);
```

```
}
```

```
String macToStr(const uint8_t* mac)
```

```
{
```

```
  String result;
```

```
  for (int i = 0; i < 6; ++i) {
```

```
    result += String(mac[i], 16);
```

```
    if (i < 5)
```

```
      result += ':';
```

```
  }
```

```
  return result;
```

```
}
```



```

void setup_wifi() {
    delay(10);
    // We start by connecting to a WiFi network
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(wifi_ssid);

    WiFi.begin(wifi_ssid, wifi_password);

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

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");

        // Generate client name based on MAC address and last 8 bits of microsecond counter
        String clientName;
        clientName += "esp8266-";
        uint8_t mac[6];
        WiFi.macAddress(mac);
        clientName += macToStr(mac);
        clientName += "-";
        clientName += String(micros() & 0xff, 16);
    }
}

```

```

Serial.print("Connecting to ");
Serial.print(mqtt_server);
Serial.print(" as ");
Serial.println(clientName);

// Attempt to connect
// If you do not want to use a username and password, change next line to
if (client.connect((char*) clientName.c_str())) {
    //if (client.connect((char*) clientName.c_str(), mqtt_user, mqtt_password)) {
        Serial.println("connected");
    } else {
        Serial.print("failed, rc=");
        Serial.print(client.state());
        Serial.println(" try again in 5 seconds");
        // Wait 5 seconds before retrying
        delay(5000);
    }
}

}

void loop() {

    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    // Wait a few seconds between measurements.
    delay(2000);

    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    // Read temperature as Celsius (the default)

```

```

float t = dht.readTemperature();
float h = dht.readHumidity();

// Check if any reads failed and exit early (to try again).
if (isnan(h) || isnan(t) ) {
  Serial.println("Failed to read from DHT sensor!");
  return;
}

// Compute heat index in Celsius (isFahreheit = false)
float hic = dht.computeHeatIndex(t, h, false);
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" % ");
Serial.print("Heat index: ");
Serial.print(hic);
Serial.print(" *C ");

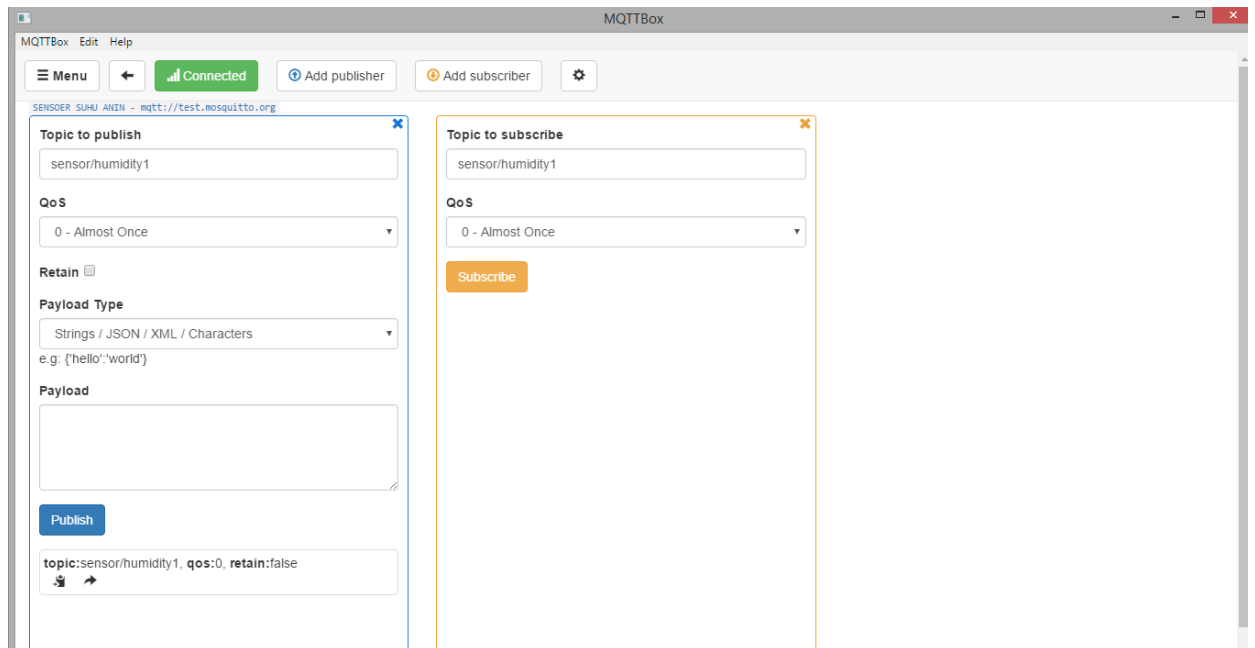
Serial.print("\t");
Serial.print("Temperature mqtt:");
Serial.print(String(t).c_str());
client.publish(temperature_celsius_topic, String(t).c_str(), true);
Serial.print("Humidity mqtt:");
Serial.println(String(h).c_str());
client.publish(humidity_topic, String(h).c_str(), true);
}

```

10. Cobalah download dashboard yang sudah tersedia online, contoh MQTT box (dapat di download di PC atau MQTT dashboard dapat di download di mobile phone.

11. Masukkan alamat MQTT data kalian pada dashboard dan sesuaikan dengan program yang telah kalian buat. Contoh dashboard yang di gunakan adalah MQTT Box.

<https://chrome.google.com/webstore/detail/mqttbox/kaajoficamnjjhkeomgfljpificfbkaf/related?hl=id>



12. Untuk penilaian kalian dapat mengirim alamat topic dan alamat server yang kalian gunakan untuk dapat dilakukan pengecekan data pada dashboard.