



UNIVERSITAS SEMARANG  
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI  
TEKNIK INFORMATIKA

---

**TIS18755P**  
**Internet of Thing**

---

Modul Praktikum Mahasiswa

*Oleh:*  
Alauddin Maulana Hirzan, S. Kom., M. Kom  
NIDN. 0607069401

# Daftar Isi

<b>Pendahuluan</b>	<b>2</b>
0.1 Mengenal <i>Internet of Things</i> . . . . .	2
0.2 Perangkat Board IoT . . . . .	2
<b>Persiapan Praktikum</b>	<b>6</b>
0.3 Perangkat Keras . . . . .	6
0.4 Perangkat Lunak . . . . .	6
<b>1 Praktikum 1</b>	<b>7</b>
1.1 Konfigurasi Arduino IDE dan ESP8266 . . . . .	7
1.2 Tutorial . . . . .	7
<b>2 Praktikum 2</b>	<b>14</b>
2.1 ESP8266, DHT11, dan AdafruitIO . . . . .	14
2.2 Tutorial . . . . .	14
<b>3 Praktikum 3</b>	<b>27</b>
3.1 ESP8266, DHT11, dan Thingspeak . . . . .	27
3.2 Tutorial . . . . .	27
<b>4 Praktikum 4</b>	<b>40</b>
4.1 ESP8266, DHT11, dan Firebase Realtime . . . . .	40
4.2 Tutorial . . . . .	40
<b>5 Praktikum 5</b>	<b>53</b>
5.1 ESP8266, DHT11, dan Telegram Bot . . . . .	53
5.2 Tutorial . . . . .	53
<b>6 Praktikum 6</b>	<b>59</b>
6.1 ESP8266, DHT11, dan Linear Regression . . . . .	59
6.2 Tutorial . . . . .	59
<b>7 Praktikum 7</b>	<b>70</b>
7.1 ESP8266, DHT11, dan Fuzzy Logic . . . . .	70
7.2 Tutorial . . . . .	70
<b>8 Praktikum 8</b>	<b>82</b>
8.1 Tugas Akhir Praktikum . . . . .	82

# Daftar Gambar

1	Internet of Things . . . . .	2
2	Board Arduino . . . . .	3
3	Board NodeMCU . . . . .	3
4	Board Pico . . . . .	3
5	Board Pi 4B . . . . .	4
6	Board NVidia Jetson . . . . .	4
7	Board Orange Pi . . . . .	4
8	Board Banana Pi . . . . .	5

# Pendahuluan

## 0.1 Mengenal *Internet of Things*

*Internet of Things* merupakan sebuah teknologi yang di mana mengizinkan setiap perangkat-perangkat yang memiliki kekuatan komputasi untuk berkomunikasi satu dengan yang lainnya tanpa campur tangan manusia untuk menyelesaikan suatu tugas atau fungsi.

Teknologi ini dapat diimplementasikan ke berbagai macam hal tergantung dari tugas atau fungsi yang ingin dicapai. Sebagai contoh untuk mendesain sebuah rumah pintar yang dapat mendeteksi lingkungan sekitar dan melakukan otomatisasi berdasarkan data tersebut.



Gambar 1: Internet of Things

## 0.2 Perangkat Board IoT

Untuk membangun sebuah perangkat berbasis IoT, komponen dasar seperti **Board** sangatlah vital untuk dipunyai. Terdapat berbagai macam board yang dapat dibeli secara luring maupun daring, dengan variasi harga yang juga berbeda mulai dari paling murah hingga mewah. Semakin kompleks masalah yang dapat diselesaikan oleh satu board, makin mahal harga board tersebut. Contoh : **NVidia Jetson** untuk *Image Processing* berbasis IoT.

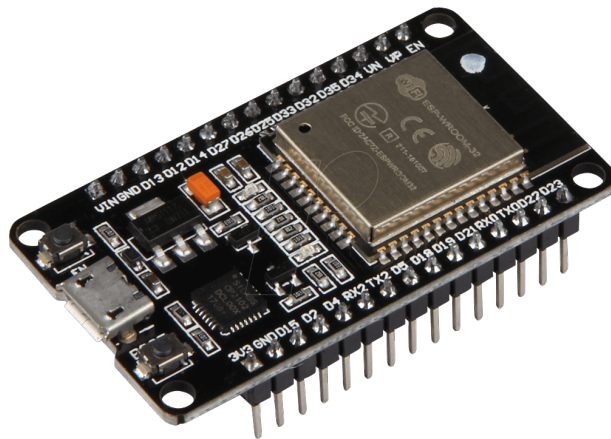
Berikut ini adalah daftar Board yang dapat dibeli dengan harga terjangkau:

1. Arduino



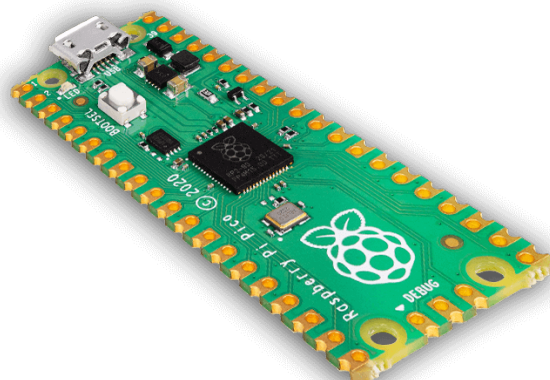
Gambar 2: Board Arduino

## 2. NodeMCU



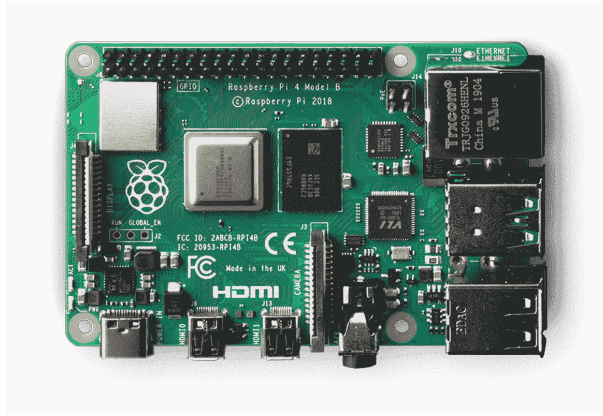
Gambar 3: Board NodeMCU

## 3. Raspberry Pi Pico



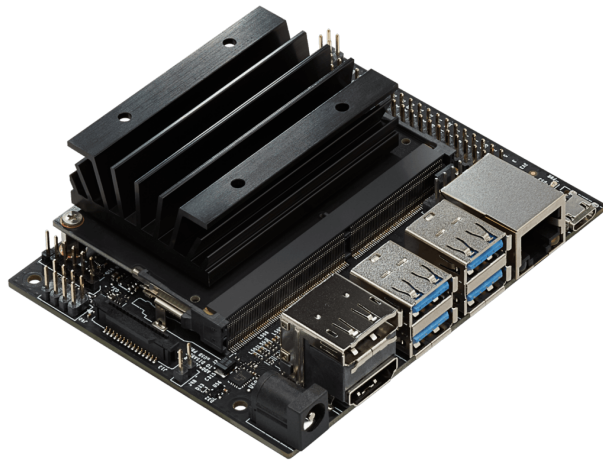
Gambar 4: Board Pico

## 4. Raspberry Pi B / 2B / 3B / 4B



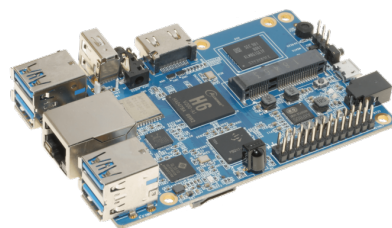
Gambar 5: Board Pi 4B

5. NVidia Jetson



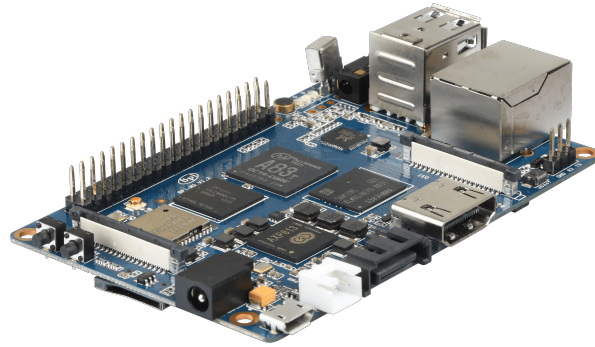
Gambar 6: Board NVidia Jetson

6. Orange Pi



Gambar 7: Board Orange Pi

7. Banana Pi



Gambar 8: Board Banana Pi

Perangkat IoT dapat berkomunikasi dengan berbagai cara seperti **Bluetooth**, **Wireless Network**, maupun jaringan kabel. Tergantung dari jenis *Board* yang digunakan, Board dengan SoC seperti Raspberry Pi biasanya dilengkapi dengan Port RJ45. Sedangkan Board mikrokontroler sederhana dilengkapi dengan nirkabel.

Selain perangkat komunikasi IoT, protokol komunikasi perangkat IoT juga mempengaruhi bagaimana proses pengiriman dan penerimaan data dari perangkat tersebut. Terdapat banyak sekali protokol maupun platform yang digunakan untuk berkomunikasi seperti: Platform dan Protokol Komunikasi IoT:

1. Blynk (Platform)
2. Cayenne (Platform)
3. Telegram Bot (Platform)
4. MQTT (Protocol)
5. Web Service

# Persiapan Praktikum

Agar praktikum dapat berjalan dengan lancar, mahasiswa diwajibkan memenuhi persyaratan berikut baik dalam bentuk perangkat keras maupun lunak:

## 0.3 Perangkat Keras

Mahasiswa sebaiknya memiliki perangkat yang sama dengan modul ini, berikut ini adalah perangkat keras yang digunakan untuk Praktikum:

- Komputer
  1. Keyboard
  2. Mouse
  3. Display
  4. Kabel Micro USB
- IoT Board
  1. NodeMCU ESP 8266
  2. Sensor DHT-11

## 0.4 Perangkat Lunak

Perangkat lunak berikut ini wajib diinstall oleh mahasiswa demi lancarnya praktikum:

- Arduino IDE (Terbaru)
  - Link : <https://www.arduino.cc/en/software>
- USB Serial Driver (Sesuaikan Model)
  - CH341 (Model ESP8266) [https://github.com/nodemcu/nodemcu-devkit/blob/master/Drivers/CH341SER\\_WINDOWS.zip](https://github.com/nodemcu/nodemcu-devkit/blob/master/Drivers/CH341SER_WINDOWS.zip)
  - CP210X (Model Amica ESP8266MOD) <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>



# Bab 1

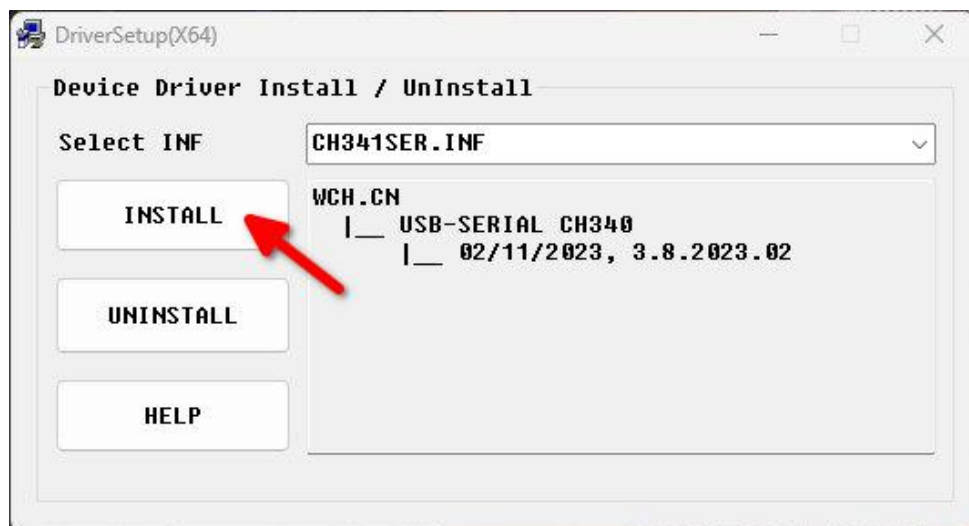
## Praktikum 1

### 1.1 Konfigurasi Arduino IDE dan ESP8266

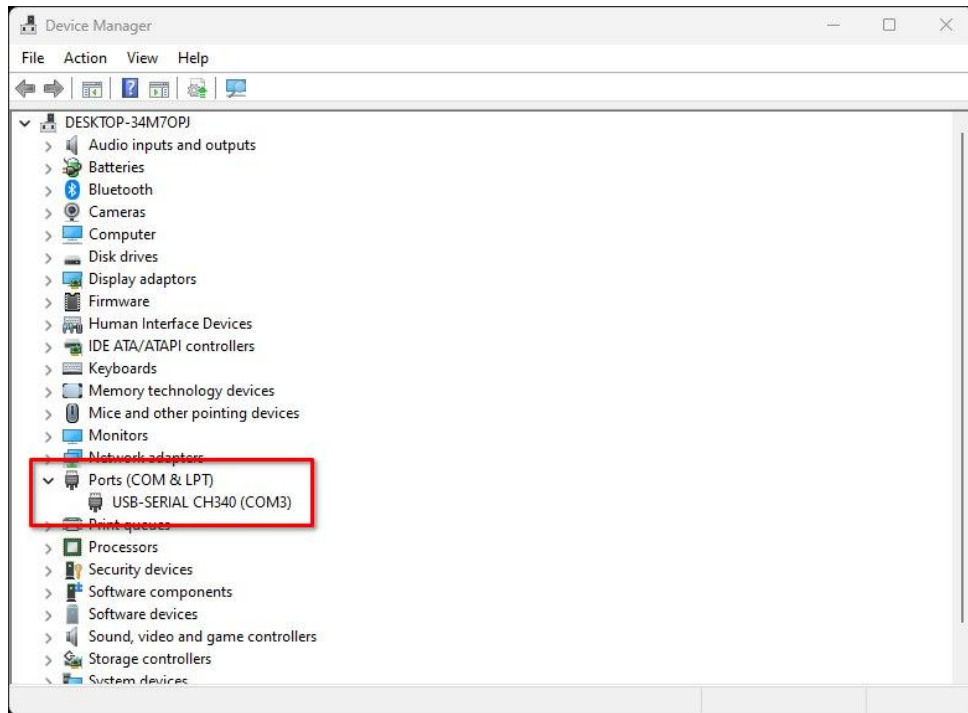
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke komputer beserta konfigurasinya hingga dapat dikenali oleh Arduino IDE. Mahasiswa diharapkan untuk membaca, dan memahami **Persiapan Praktikum** yang ada di halaman sebelumnya.

### 1.2 Tutorial

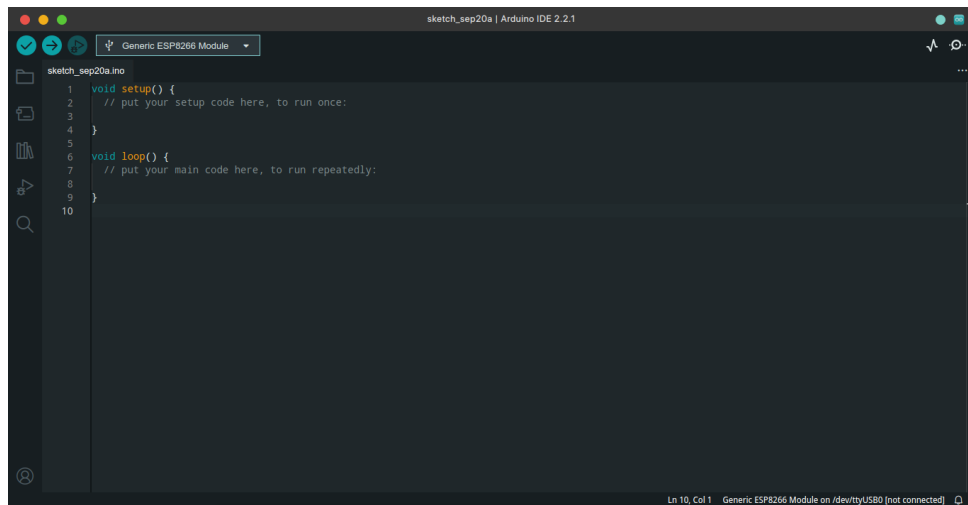
1. Setelah mahasiswa menyiapkan perangkat-perangkat yang diperlukan, maka langkah berikutnya adalah melakukan instalasi driver terlebih dahulu.
2. File driver **CH341SER** yang sudah diunduh, dibuka untuk diinstall. Cukup klik **Install** untuk memasang driver (Windows 10 ke bawah)



3. Untuk mengecek apakah sudah sukses, gunakan **Device Manager** lalu tancapkan perangkat ke port USB

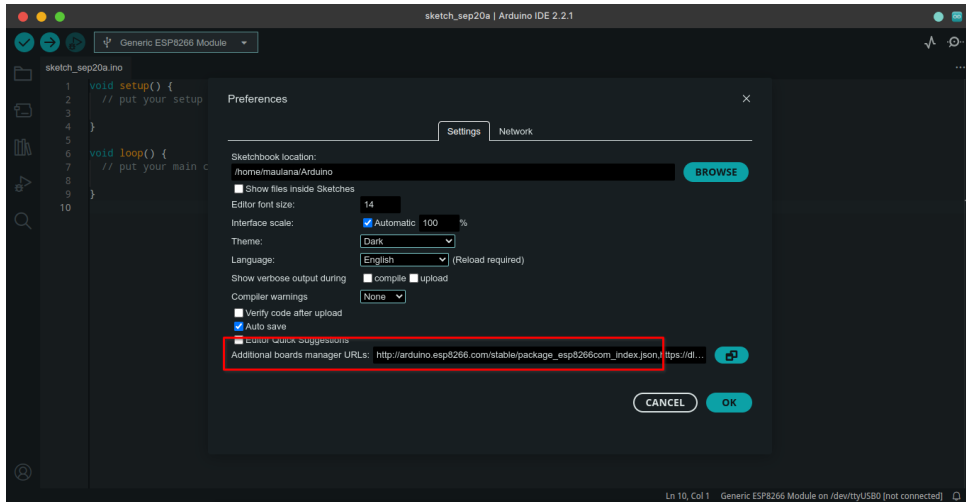


- Langkah berikutnya adalah mengunduh **Arduino IDE**, usahakan untuk mendapatkan versi terbaru. Setelah unduh, buka aplikasi tersebut

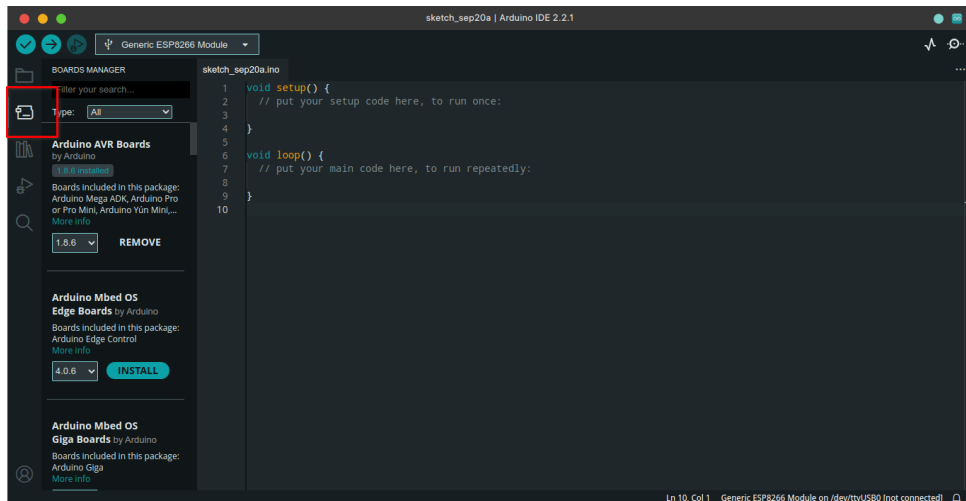


- Namun **Arduino IDE** ini belum mendukung perangkat yang kita gunakan. Langkah berikutnya buka **File** → **Preferences** →. Tambahkan baris **Alamat URL** berikut ke **Additional board manager URLs**. Klik **OK** untuk mengupdate otomatis.

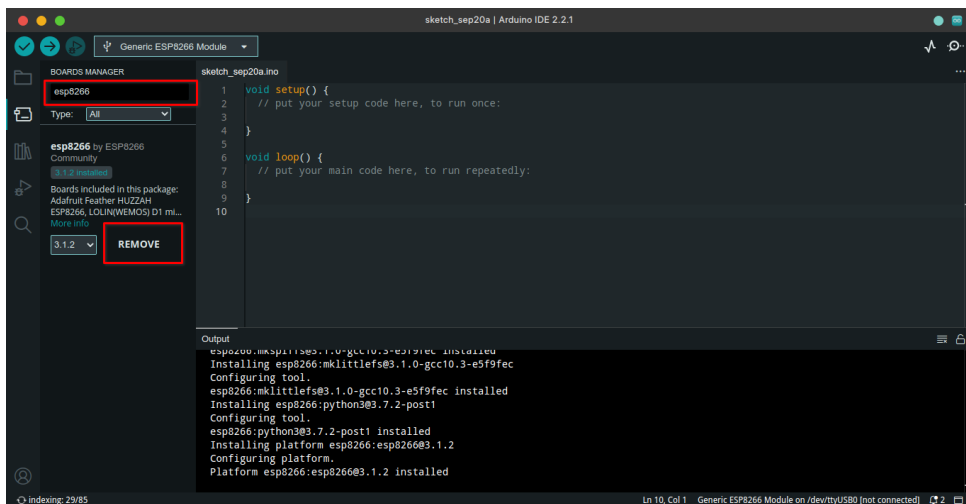
- [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



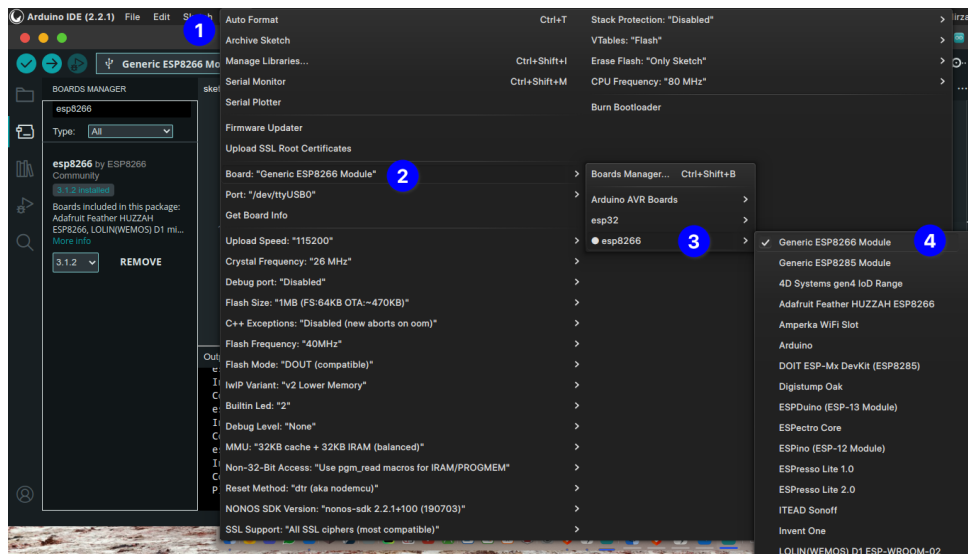
6. Jika sudah, install driver **ESP8266** dengan klik **Boards Manager** di **Sidebar Kanan** atau **Tools** → **Board:** → **Boards Manager**



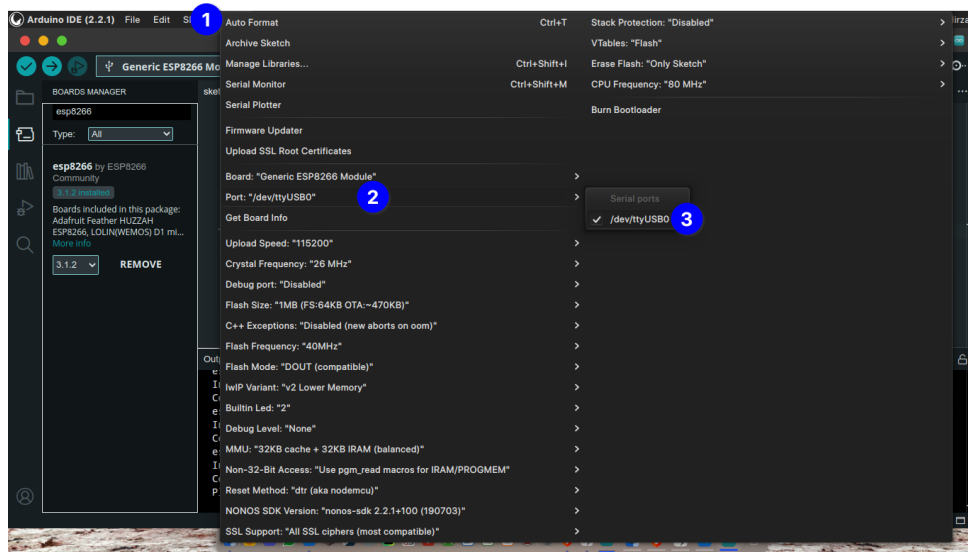
7. Di kolom **Pencarian**, ketik **ESP8266** dan klik **Install**



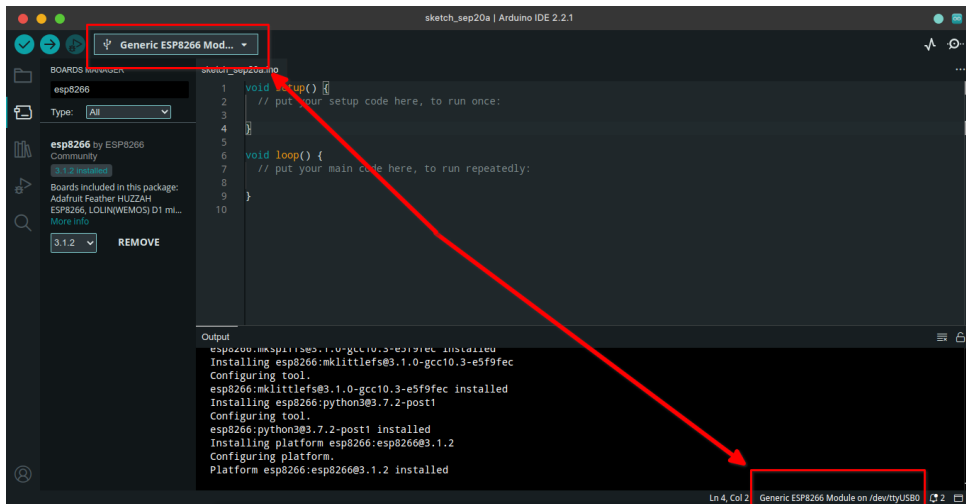
8. **Arduino IDE** sudah siap, namun belum terhubung ke perangkat. Untuk menghubungkan antara **IDE** dengan **ESP8266**, pilih **Tools** → **Board:** → **esp8266** → **Generic ESP8266 Module**



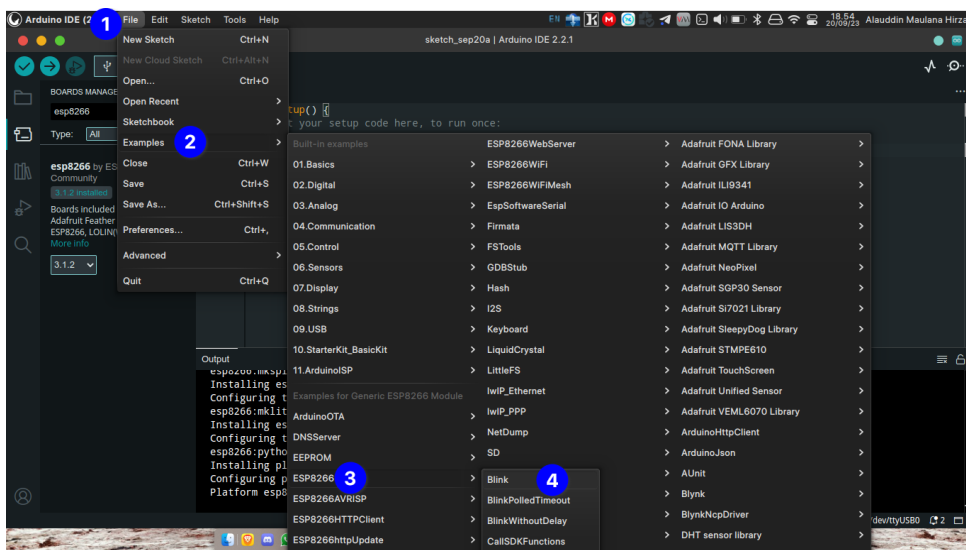
9. Kemudian pastikan **Port Serial** yang digunakan, sama dengan yang ada di **Device Manager**. Cek dengan menu **Tools** → **Port:** → **Pilih COM Sesuai Device Manager**



10. Jika sudah terhubung, akan ada tanda tulisan **Generic ESP8266 Module** on **COMXXX** di bawah kanan maupun simbol USB di atas kiri



11. NodeMCU ESP8266 siap diujikan. Untuk menguji alat, **Arduino IDE** sudah menyiapkan template dasar seperti **LED Blinking**. Untuk mengakses kode ini buka menu **File** → **Examples** → **ESP8266** → **Blink**



12. **Arduino IDE** akan membuka **Window Baru**. Tutup **Window** sebelumnya agar tidak terganggu.

```

1  /*
2  ESP8266 Blink by Simon Peter
3  Blink the blue LED on the ESP-01 module
4  This example code is in the public domain
5
6  The blue LED on the ESP-01 module is connected to GPIO1
7  (which is also the TXD pin, so we cannot use Serial.print() at the same time)
8
9  Note that this sketch uses LED_BUILTIN to find the pin with the internal LED
10 */
11
12 void setup() {
13   pinMode(LED_BUILTIN, OUTPUT); // Initialize the LED_BUILTIN pin as an output
14 }
15
16 // the loop function runs over and over again forever
17 void loop() {
18   digitalWrite(LED_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage level
19   // but actually the LED is on; this is because
20   // it is active low on the ESP-01)
21   delay(1000); // Wait for a second
22   digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH
23   delay(2000); // Wait for two seconds (to demonstrate the active low LED)
24 }
25

```

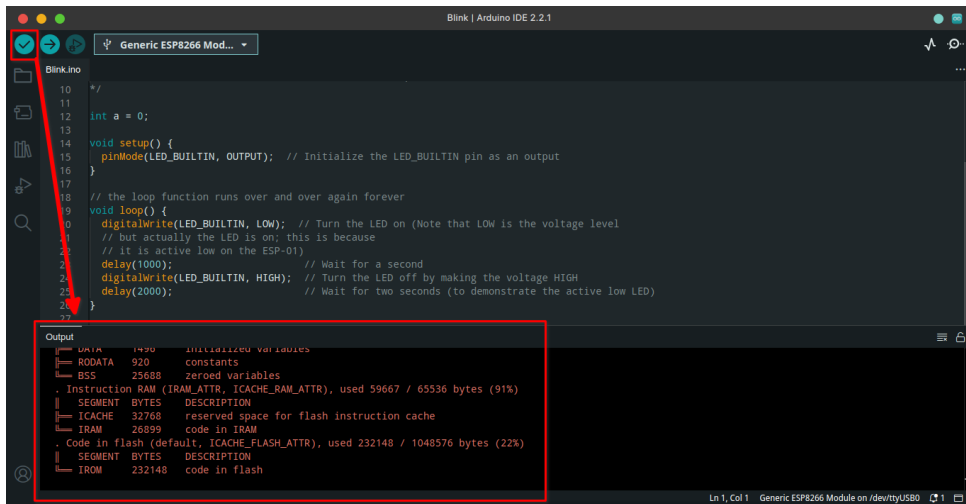
13. Mahasiswa **WAJIB MEMAHAMI ALUR KODE**. Kode dieksekusi dari atas ke bawah. **Fungsi SETUP** digunakan untuk mengatur inisialisasi yang dilakukan **SATU KALI**. Sedangkan **Fungsi LOOP** digunakan untuk proses yang diulang-ulang oleh alat. Kode-kode di atas kedua fungsi tersebut dianggap sebagai **PARAMETER GLOBAL**

```

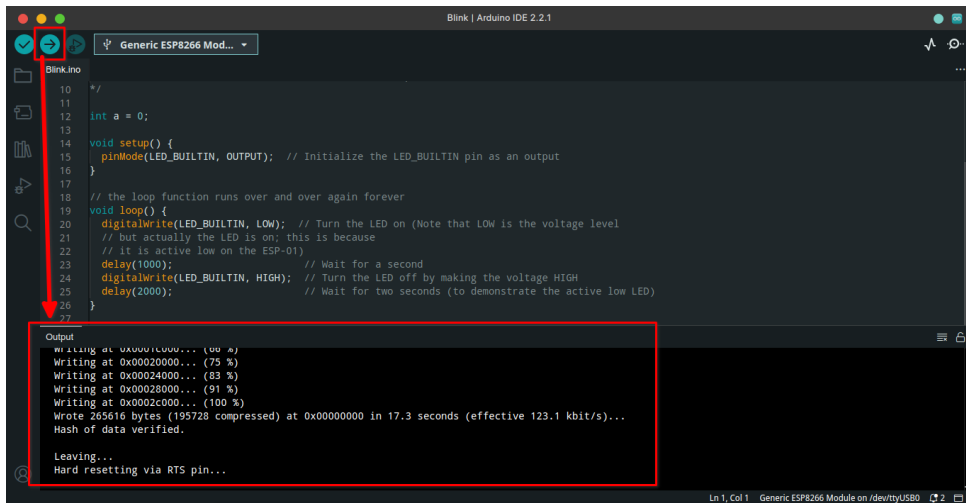
1  /*
2  ESP8266 Blink by Simon Peter
3  Blink the blue LED on the ESP-01 module
4  This example code is in the public domain
5
6  The blue LED on the ESP-01 module is connected to GPIO1
7  (which is also the TXD pin, so we cannot use Serial.print() at the same time)
8
9  Note that this sketch uses LED_BUILTIN to find the pin with the internal LED
10 */
11
12 int a = 0; // PARAMETER GLOBAL
13
14 void setup() { // BAGIAN INISIALISASI, UNTUK SENSOR/WIFI
15   pinMode(LED_BUILTIN, OUTPUT); // Initialize the LED_BUILTIN pin as an output
16 }
17
18 // the loop function runs over and over again forever
19 void loop() { // BAGIAN PERULANGAN, CTH: MEMBACA SENSOR
20   digitalWrite(LED_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage level
21   // but actually the LED is on; this is because
22   // it is active low on the ESP-01)
23   delay(1000); // Wait for a second
24   digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH
25   delay(2000); // Wait for two seconds (to demonstrate the active low LED)
26 }
27

```

14. Tahap berikutnya adalah verifikasi dan upload kode. Verifikasi memastikan kode sudah benar tanpa typo, sedangkan Upload digunakan mengunggah kode ke alat. Sekarang klik **Verify** untuk memastikan kode sudah benar



15. Jika sudah klik **Upload** untuk mengunggah kode ke alat. Alat akan otomatis menjalankan fungsinya sesuai apa yang diprogramkan.



# Bab 2

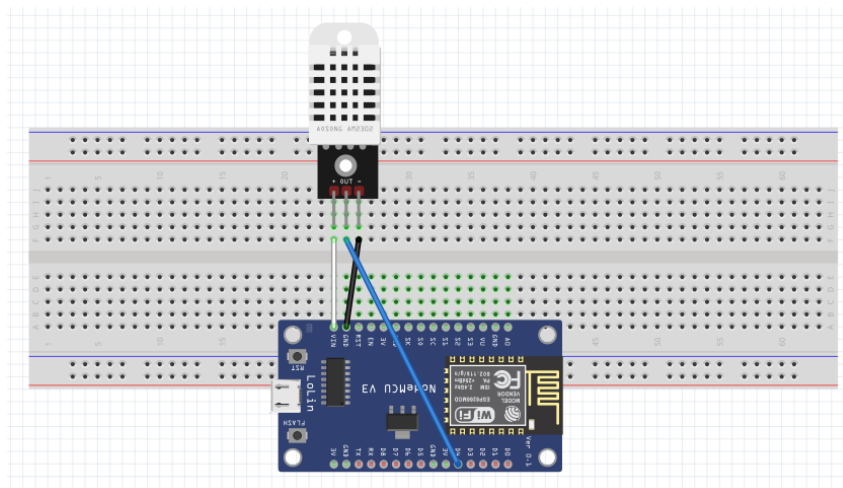
## Praktikum 2

### 2.1 ESP8266, DHT11, dan AdafruitIO

Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke sensor DHT11 dan bagaimana menyimpan data secara daring di layanan AdafruitIO. Mahasiswa diwajibkan memahami **Praktikum 1** yang ada di halaman sebelumnya.

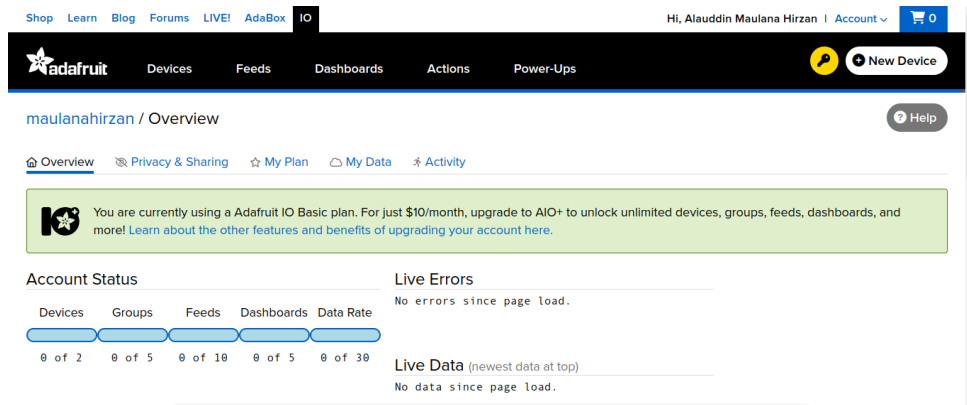
### 2.2 Tutorial

1. Langkah pertama yang perlu dilakukan adalah memasang sensor ke perangkat. Perlu diketahui bahwa dalam memasang sensor harus dalam keadaan **MATI/TIDAK TERTANCAP** untuk menghindari KORSLETING
2. Perhatikan sensor **DHT11**, di bagian kakinya ada tanda **Plus +**, **Minus -**, dan **Out**. Sambungkan sesuai dengan indikator **NodeMCU ESP8266** sebagai berikut:
  - **Plus + / VCC** → **3V / 3.3V**
  - **Minus - / GND** → **G**
  - **OUT** → **D4 / GPIO2**

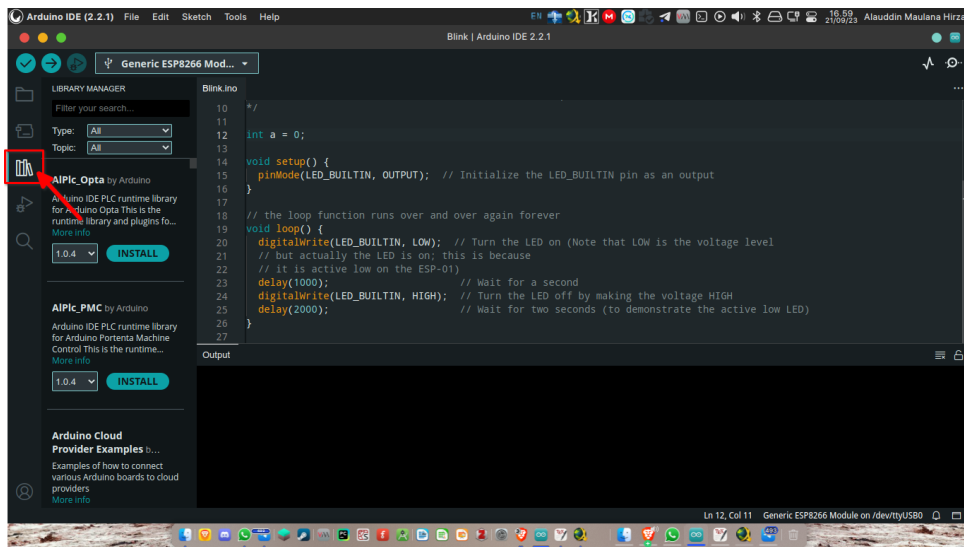




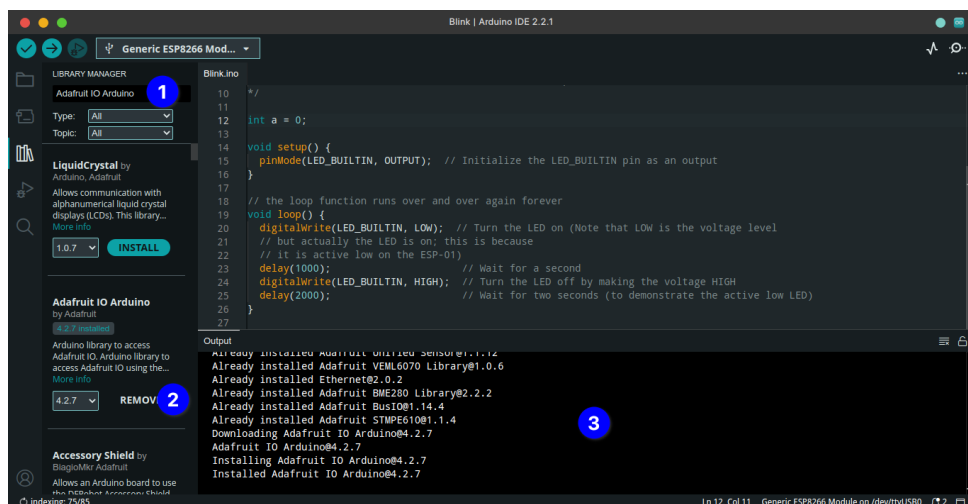
- Setelah selesai menancapkan sensor, berikutnya adalah melakukan registrasi ke website AdafruitIO dengan link : <https://io.adafruit.com/>. Setelah teregistrasi akan terlihat dasbor seperti berikut:



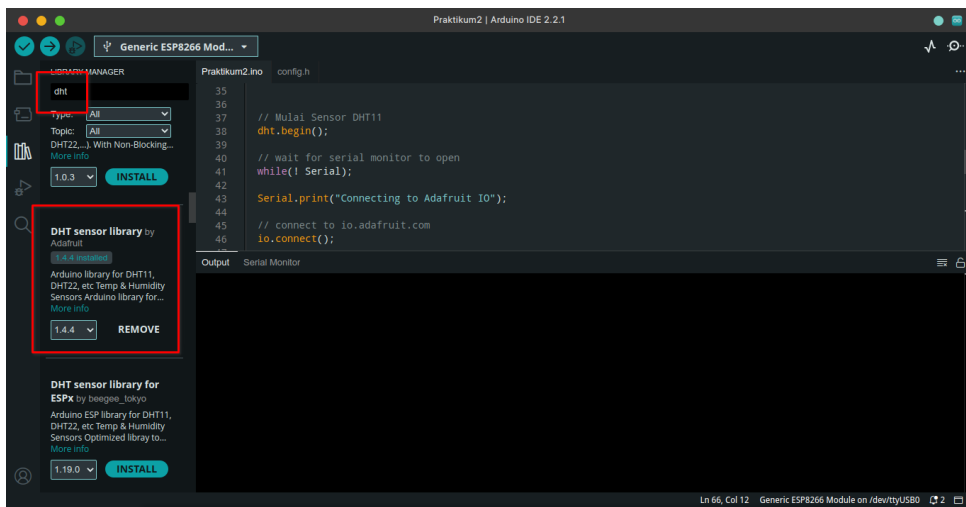
- Kembali ke **Arduino IDE**, dan install **Library** dengan mengakses menu samping atau **Sketch** → **Include Library** → **Manage Libraries**



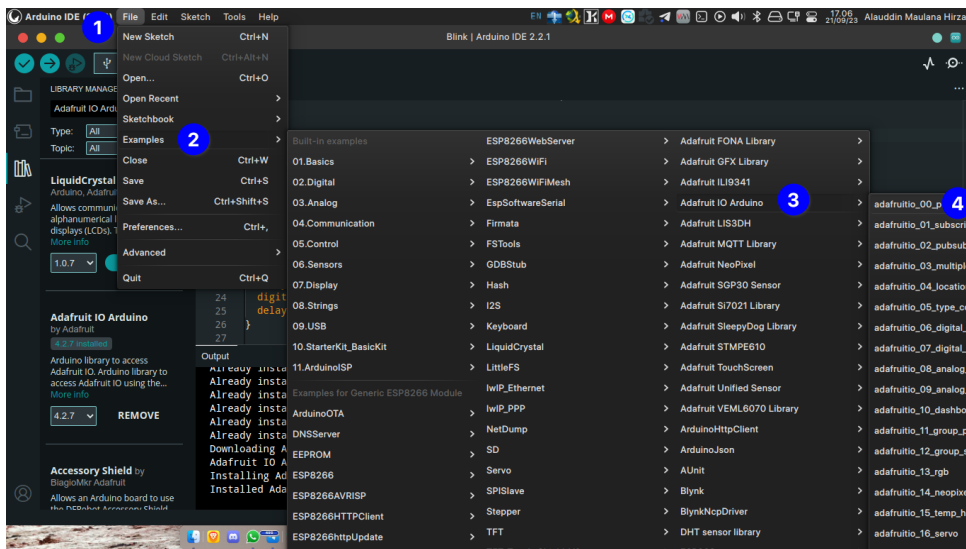
- Cari **Adafruit IO Arduino**, klik **INSTALL**, lalu **INSTALL ALL**



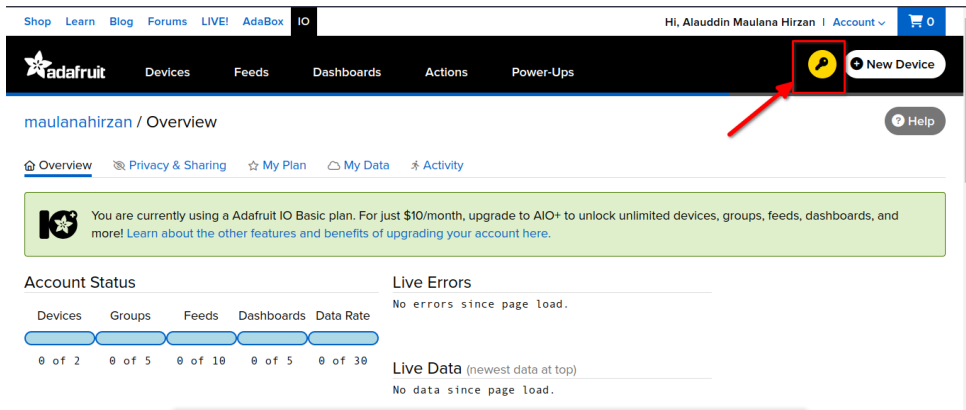
6. Cari DHT sensor Library, klik **INSTALL**, lalu **INSTALL ALL**



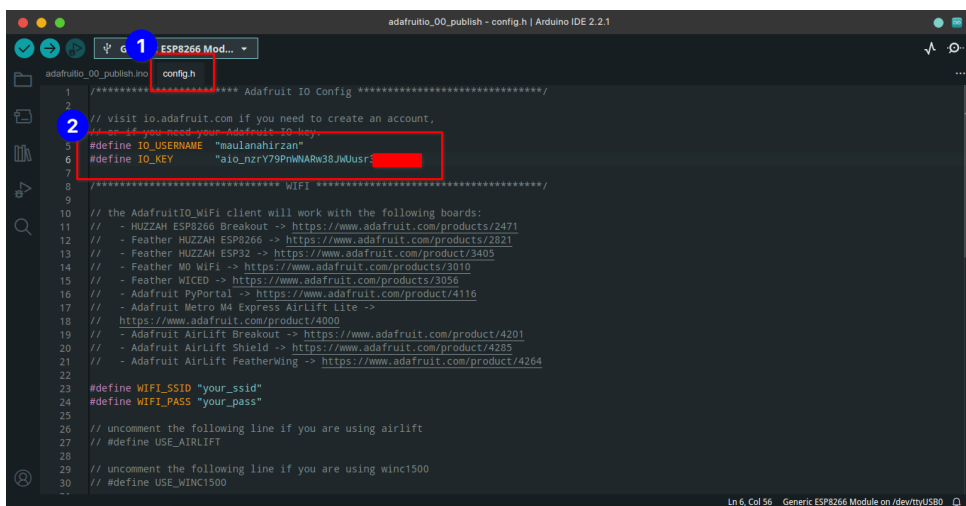
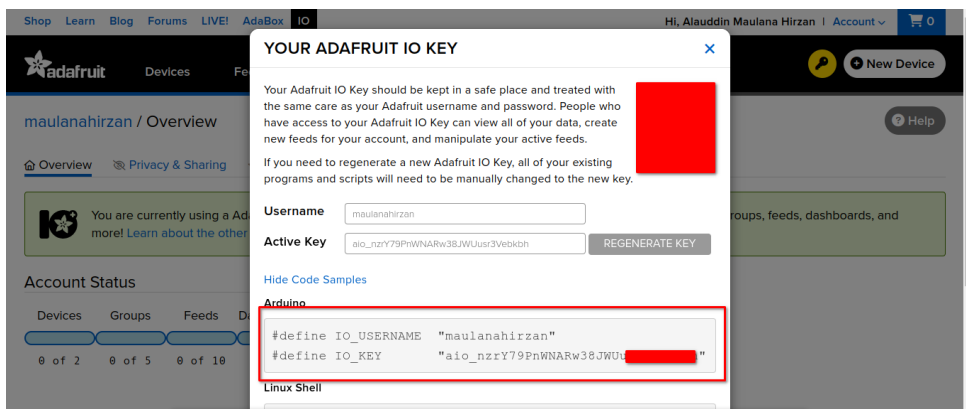
7. Sesudah install, berikutnya adalah membuka **Template Adafuit IO**. Klik menu **File** → **Examples** → **Adafuit IO Arduino** → **adafuit\_00\_publish**. Tutup **Arduino IDE** lain agar fokus



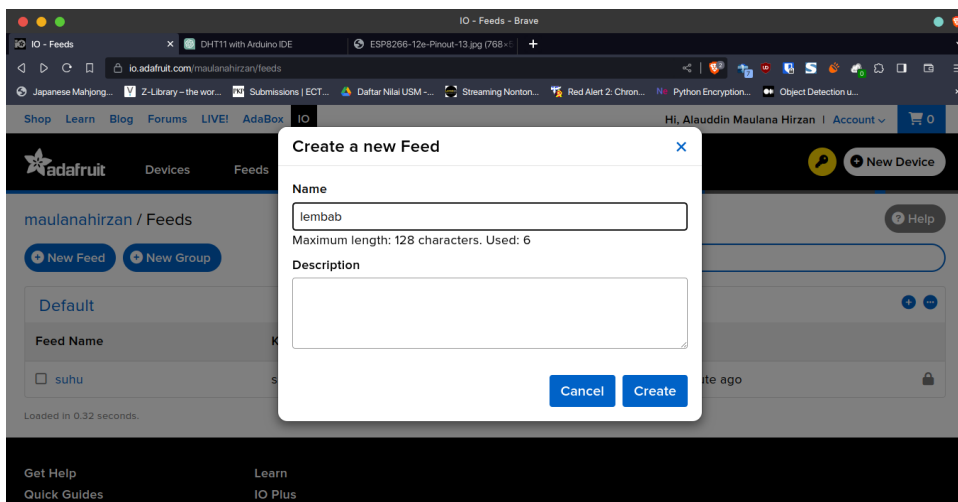
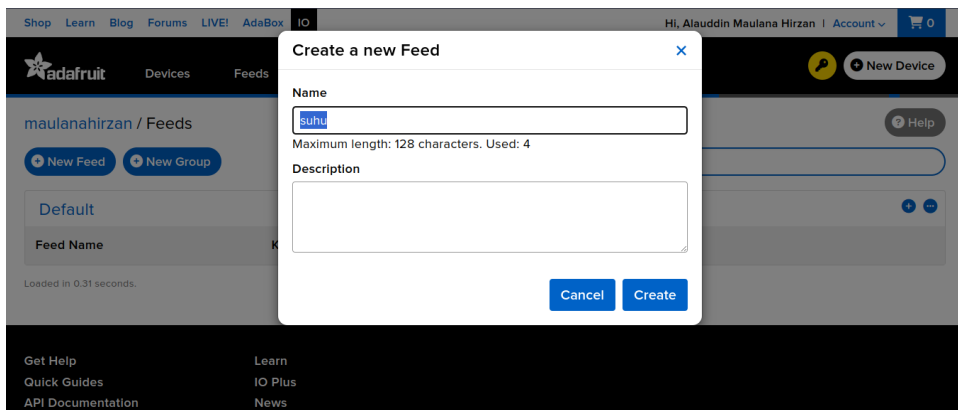
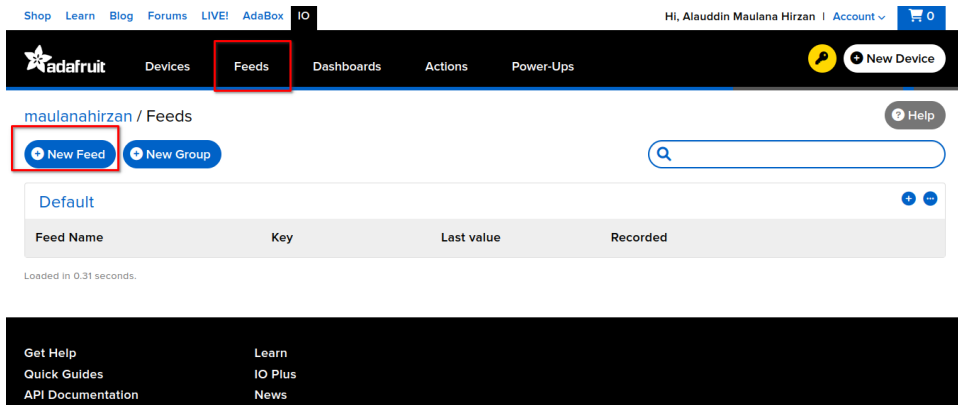
8. Jika sudah terbuka, kembali lagi ke website **Adafuit IO**. Klik **Icon Kunci Kuning** untuk menambahkan perangkat.



9. Adafruit IO akan membuat kunci yang akan dimasukkan ke Sketch Arduino IDE. Lihat bagian yang ditandai dan tempelkan ke file `config.h` di Tab Arduino IDE



10. Jika sudah, buatlah **Feed** terlebih dahulu dengan meng klik **Menu Feeds**. Lalu buat 2 **Feed** baru dengan nama **suhu** dan **lembab**



11. Lalu kembali ke **config.h** dan ubah SSID Wifi dan Passwordnya di bagian bawahnya

```
adafruitio_00_publish - config.h | Arduino IDE 2.2.1
Generic ESP8266 Mod...
adafruitio_00_publish.ino config.h
17 // - Adafruit Metro M4 Express AirLift Lite ->
18 // https://www.adafruit.com/product/4000
19 // - Adafruit AirLift Breakout -> https://www.adafruit.com/product/4201
20 // - Adafruit AirLift Shield -> https://www.adafruit.com/product/4285
21 // - Adafruit AirLift FeatherWing -> https://www.adafruit.com/product/4264
22
23 #define WIFI_SSID "Free Wifi USM 1"
24 #define WIFI_PASS ""
25
26 // uncomment the following line if you are using airlift
27 // #define USE_AIRLIFT
28
29 // uncomment the following line if you are using wincl500
30 // #define USE_WINCL500
31
32 // uncomment the following line if you are using mrk1010 or nano 33 iot
33 // #define ARDUINO_SAMD_MKR1010
34
35 // comment out the following lines if you are using fona or ethernet
36 #include "AdafruitIO_Wifi.h"
37
38 #if defined(USE_AIRLIFT) || defined(ADAFRUIT_METRO_M4_AIRLIFT_LITE) ||
39   defined(ADAFRUIT_PYPORTAL)
40 // Configure the pins used for the ESP32 connection
41 #if !defined(SPIWIFI_SS) // if the wifi definition isnt in the board variant
42 // Don't change the names of these #define's! they match the variant ones
43 #define SPIWIFI_SPI
44 #define SPIWIFI_SS 10 // Chip select pin
45 #define NINA_ACK 9 // a.k.a. BUSY or READY pin
46 #define NINA_RESETN 6 // Reset pin
```

12. Konfigurasi **Adafruit IO** sudah selesai, berikutnya adalah memasukkan kode untuk mengambil data sensor. Kembali ke tab **arduino\_00\_publish.ino**
13. Lalu hapus kode yang ditandai

```
adafruitio_00_publish | Arduino IDE 2.2.1
Generic ESP8266 Mod...
adafruitio_00_publish.ino config.h
***** Example Starts Here *****
22 // this int will hold the current count for our sketch
23 int count = 0;
24
25 // set up the 'counter' feed
26 AdafruitIO_Feed *counter = io.feed("counter");
27
28 void setup() {
29 // start the serial connection
30 Serial.begin(115200);
31 // wait for serial monitor to open
32 while(! Serial);
33 Serial.print("Connecting to Adafruit IO");
34 // connect to io.adafruit.com
35 io.connect();
36 // wait for a connection
37 while(io.status() < AIO_CONNECTED) {
38   Serial.print(".");
39   delay(500);
40 }
41 // we are connected
42 Serial.println();
43 Serial.println(io.statusText());
44 }
```

14. Ubah kode **AdafruitIO\_Feed \*counter = io.feed("counter");** menjadi

**Potongan Kode**

```
AdafruitIO_Feed *suhu = io.feed("suhu");
AdafruitIO_Feed *lembab = io.feed("lembab");
```

```

15 // edit the config.h tab and enter your Adafruit IO credentials
16 // and any additional configuration needed for WiFi, cellular,
17 // or ethernet clients.
18 #include "config.h"
19 #include <ESP8266WiFi.h>
20 #include <DHT.h>
21 ***** Example Starts Here *****
22
23 // set up the 'counter' feed
24 AdafruitIO_Feed *suhu = io.feed("suhu");
25 AdafruitIO_Feed *lembab = io.feed("lembab");
26
27 #define DHTPIN 9
28 #define DHTTYPE DHT11
29 DHT dht(DHTPIN, DHTTYPE);
30
31 void setup() {
32
33 // start the serial connection
34 Serial.begin(115200);
35
36 // wait for serial monitor to open
37 while(! Serial);
38
39 Serial.print("Connecting to Adafruit IO");
40
41 // connect to io.adafruit.com
42 io.connect();
43
44 // wait for a connection
45 while(io.status() < IO_CONNECTED) {

```

15. Berikutnya adalah mengkonfigurasi kode untuk ESP8266 dan DHT11, tambahkan kode berikut tepat di bawah `#include "config.h"`

Potongan Kode

```

#include <ESP8266WiFi.h>
#include <DHT.h>

```

```

15 // edit the config.h tab and enter your Adafruit IO credentials
16 // and any additional configuration needed for WiFi, cellular,
17 // or ethernet clients.
18 #include "config.h"
19 #include <ESP8266WiFi.h>
20 #include <DHT.h>
21 ***** Example Starts Here *****
22
23 // set up the 'counter' feed
24 AdafruitIO_Feed *suhu = io.feed("suhu");
25 AdafruitIO_Feed *lembab = io.feed("lembab");
26
27 #define DHTPIN 9
28 #define DHTTYPE DHT11
29 DHT dht(DHTPIN, DHTTYPE);
30
31 void setup() {
32
33 // start the serial connection
34 Serial.begin(115200);
35
36 // wait for serial monitor to open
37 while(! Serial);
38
39 Serial.print("Connecting to Adafruit IO");
40
41 // connect to io.adafruit.com
42 io.connect();
43
44 // wait for a connection
45 while(io.status() < IO_CONNECTED) {

```

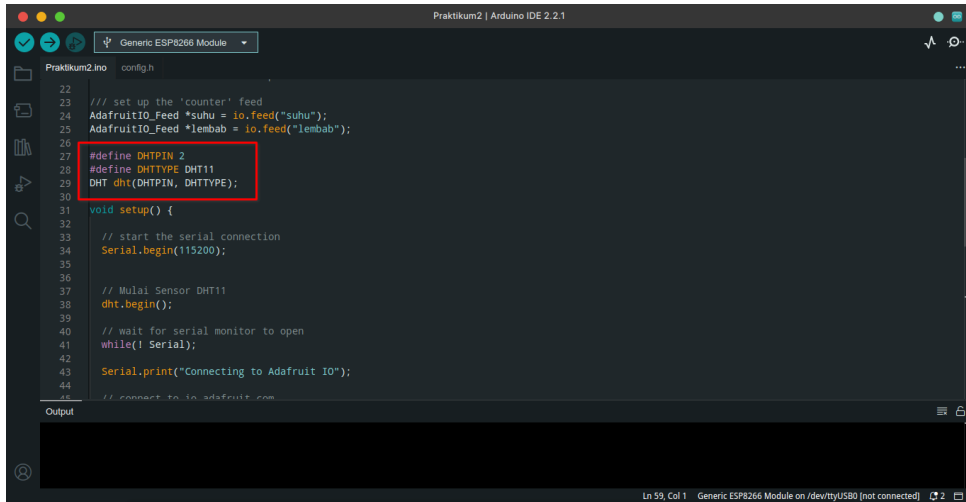
16. Lalu tambahkan kode definisi untuk jenis sensor DHT11. Tambahkan kode berikut tepat di bawah kode `io.feed`. Nomor `DHTPIN` didapatkan dari gambar **Pinout GPIO ESP8266** via **Google**

Potongan Kode

```

#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

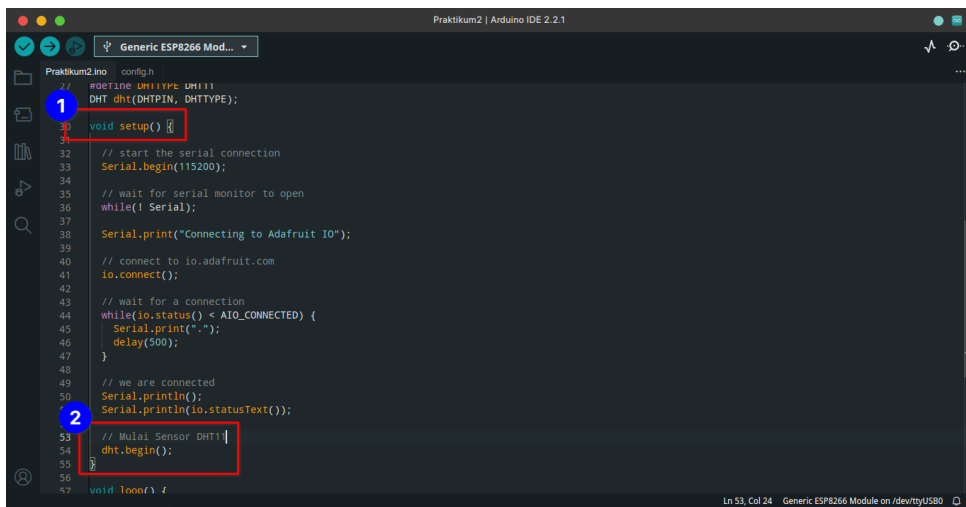
```



17. Parameter global sudah diset. Berikutnya adalah mengatur fungsi **setup** untuk sensor **dht**. Tambahkan kode berikut di bagian akhir fungsi **setup** (BUKAN AKHIR FILE)

Potongan Kode

```
// Mulai Sensor DHT11
dht.begin();
```



18. Lalu tambahkan kode ke fungsi **loop** untuk membaca suhu dan kelembaban. Letakkan di bawa **io.run()**

Potongan Kode

```
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
```

```
Praktikum2.ino config.h
50 Serial.println();
51 Serial.println(io.statusText());
52
53 // Mulai Sensor DHT11
54 dht.begin();
55 }
56
57 void loop() {
58
59 // io.run(); is required for all sketches.
60 // It should always be present at the top of your loop
61 // function. It keeps the client connected to
62 // io.adafruit.com, and processes any incoming data.
63 io.run();
64
65 float temperature = dht.readTemperature();
66 float humidity = dht.readHumidity();
67
68 // save count to the 'counter' feed on Adafruit IO
69 Serial.print("sending -> ");
70 Serial.println(count);
71 counter->save(count);
72
73 // increment the count by 1
74 count++;
75
76 // Adafruit IO is rate limited for publishing, so a delay is required in
77 // between feed->save events. In this example, we will wait three seconds
78 // (1000 milliseconds == 1 second) during each loop.
79 delay(3000);
80 }
```

19. Setelah itu ubah kode `Serial.println(count);` dengan kode berikut:

**Potongan Kode**

```
Serial.print(temperature);
Serial.print(" and ");
Serial.println(humidity);
```

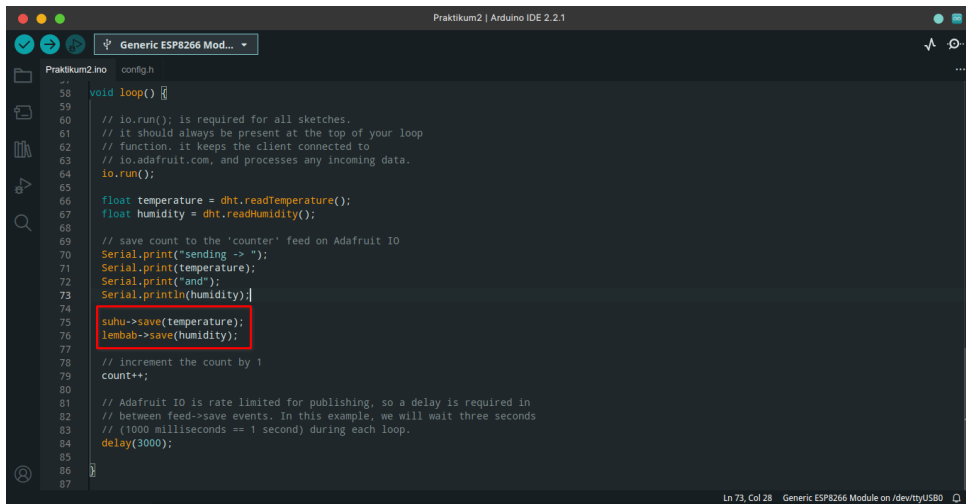
```
Praktikum2.ino config.h
56
57
58 void loop() {
59
60 // io.run(); is required for all sketches.
61 // It should always be present at the top of your loop
62 // function. It keeps the client connected to
63 // io.adafruit.com, and processes any incoming data.
64 io.run();
65
66 float temperature = dht.readTemperature();
67 float humidity = dht.readHumidity();
68
69 // save count to the 'counter' feed on Adafruit IO
70 Serial.print("sending -> ");
71 Serial.print(temperature);
72 Serial.print(" and ");
73 Serial.println(humidity);
74 counter->save(count);
75
76 // increment the count by 1
77 count++;
78
79 // Adafruit IO is rate limited for publishing, so a delay is required in
80 // between feed->save events. In this example, we will wait three seconds
81 // (1000 milliseconds == 1 second) during each loop.
82 delay(3000);
83
84 }
85 }
```

20. Bagian terakhir yang perlu diubah adalah proses unggahnya. Ganti kode `counter->save(count);` menjadi

**Potongan Kode**

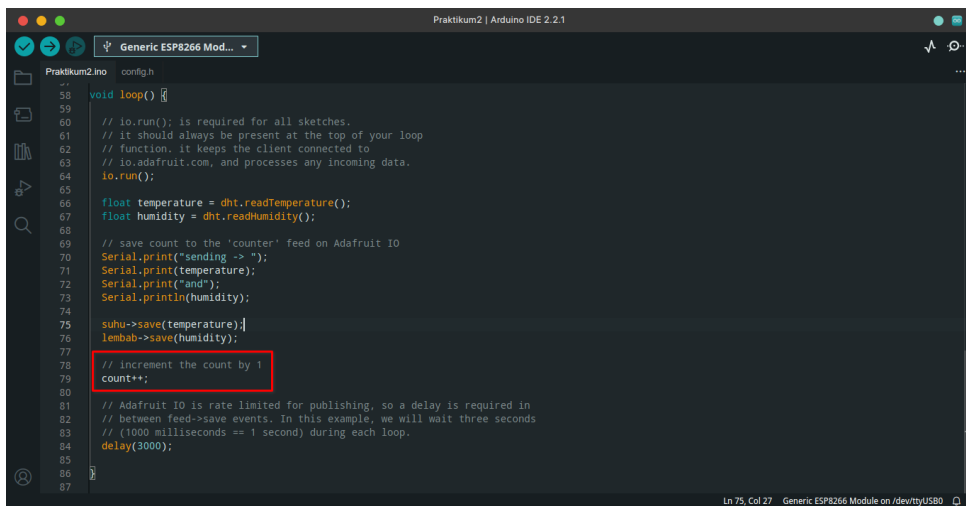
```
suhu->save(temperature);
lembab->save(humidity);
```





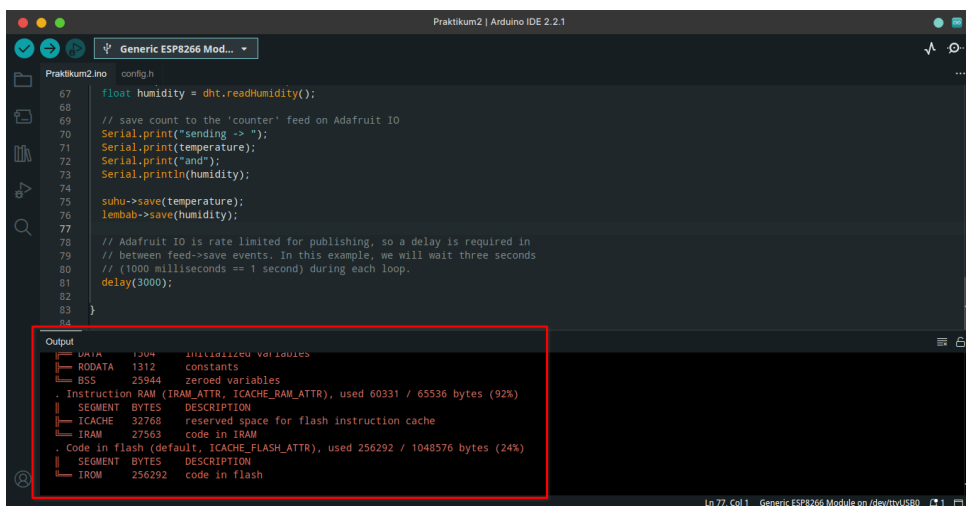
```
Praktikum2.ino config.h
58 void loop() {
59
60 // io.run(); is required for all sketches.
61 // It should always be present at the top of your loop
62 // function. It keeps the client connected to
63 // io.adafruit.com, and processes any incoming data.
64 io.run();
65
66 float temperature = dht.readTemperature();
67 float humidity = dht.readHumidity();
68
69 // save count to the 'counter' feed on Adafruit IO
70 Serial.print("sending -> ");
71 Serial.print(temperature);
72 Serial.print("and");
73 Serial.println(humidity);
74
75 suhu->save(temperature);
76 lembab->save(humidity);
77
78 // increment the count by 1
79 count++;
80
81 // Adafruit IO is rate limited for publishing, so a delay is required in
82 // between feed->save events. In this example, we will wait three seconds
83 // (1000 milliseconds == 1 second) during each loop.
84 delay(3000);
85
86
87
```

21. Terakhir, hapus kode increment `count++`;



```
Praktikum2.ino config.h
58 void loop() {
59
60 // io.run(); is required for all sketches.
61 // It should always be present at the top of your loop
62 // function. It keeps the client connected to
63 // io.adafruit.com, and processes any incoming data.
64 io.run();
65
66 float temperature = dht.readTemperature();
67 float humidity = dht.readHumidity();
68
69 // save count to the 'counter' feed on Adafruit IO
70 Serial.print("sending -> ");
71 Serial.print(temperature);
72 Serial.print("and");
73 Serial.println(humidity);
74
75 suhu->save(temperature);
76 lembab->save(humidity);
77
78 // increment the count by 1
79 count++;
80
81 // Adafruit IO is rate limited for publishing, so a delay is required in
82 // between feed->save events. In this example, we will wait three seconds
83 // (1000 milliseconds == 1 second) during each loop.
84 delay(3000);
85
86
87
```

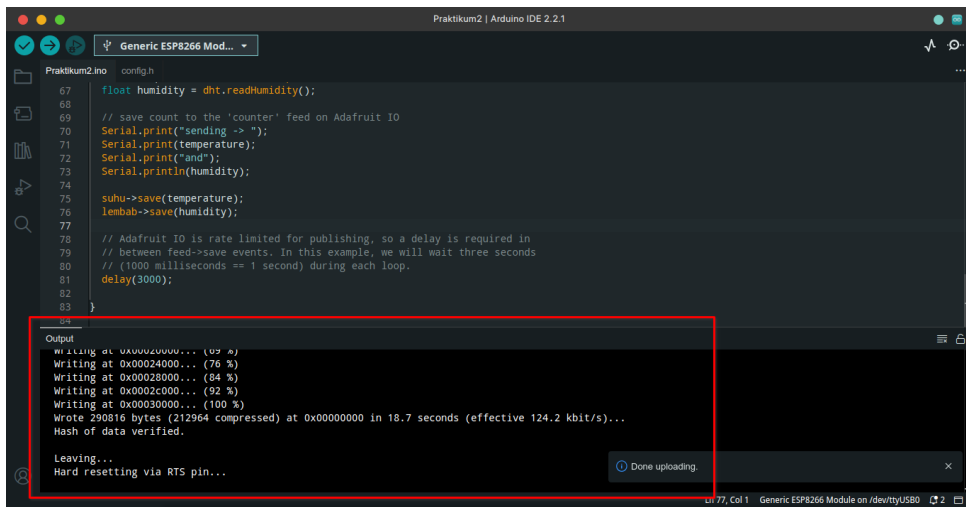
22. Verifikasi kode. Jika tidak ada **Error** seperti digambar. Lanjutkan dengan **Upload**. Pastikan **NodeMCU** tertancap



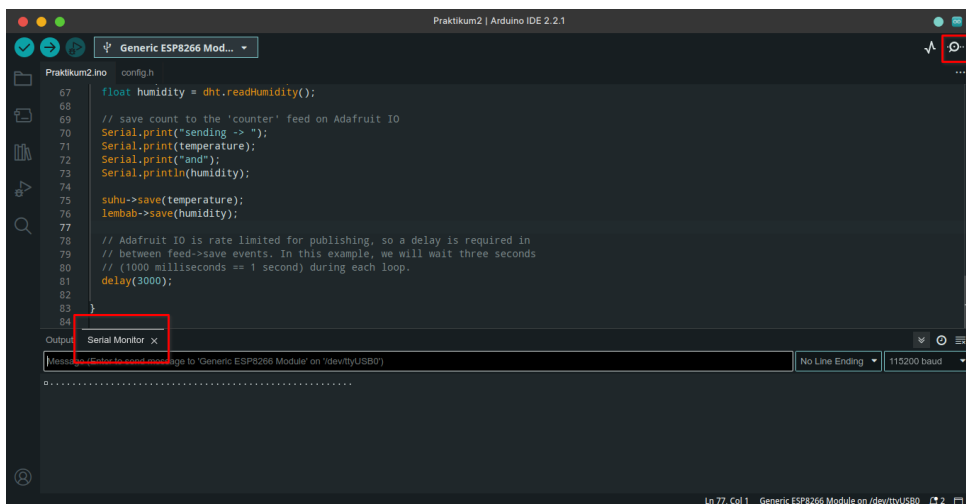
```
Praktikum2.ino config.h
67 float humidity = dht.readHumidity();
68
69 // save count to the 'counter' feed on Adafruit IO
70 Serial.print("sending -> ");
71 Serial.print(temperature);
72 Serial.print("and");
73 Serial.println(humidity);
74
75 suhu->save(temperature);
76 lembab->save(humidity);
77
78 // Adafruit IO is rate limited for publishing, so a delay is required in
79 // between feed->save events. In this example, we will wait three seconds
80 // (1000 milliseconds == 1 second) during each loop.
81 delay(3000);
82
83
84
85
```

```
Output
DATA 1304 initialized variables
RODATA 1312 constants
BSS 28944 zeroed variables
Instruction RAM (IRAM_ATTR, ICACHE_RAM_ATTR), used 60331 / 65536 bytes (92%)
SEGMENT BYTES DESCRIPTION
ICACHE 32768 reserved space for flash instruction cache
IRAM 27563 code in IRAM
Code in flash (default, ICACHE_FLASH_ATTR), used 256292 / 1048576 bytes (24%)
SEGMENT BYTES DESCRIPTION
IROM 256292 code in flash
```

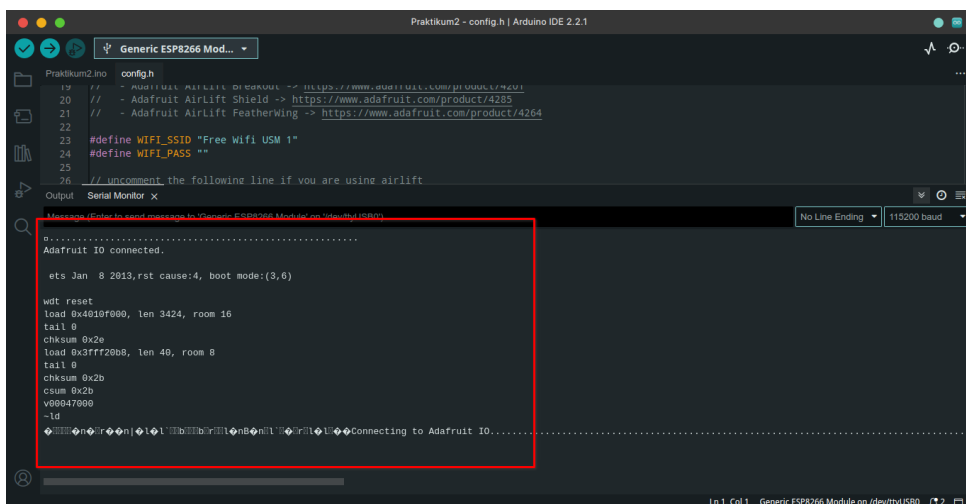
23. Unggah sudah sukses



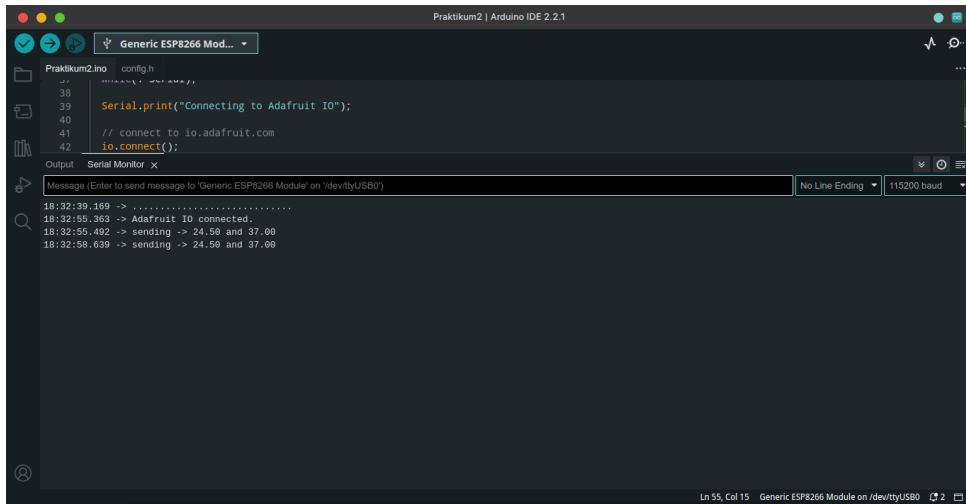
24. Berikutnya adalah mengecek alat. Klik **Tools** → **Serial Monitor**. Pastikan **BAUD** sudah sesuai dengan kode (biasanya 115200 baud)



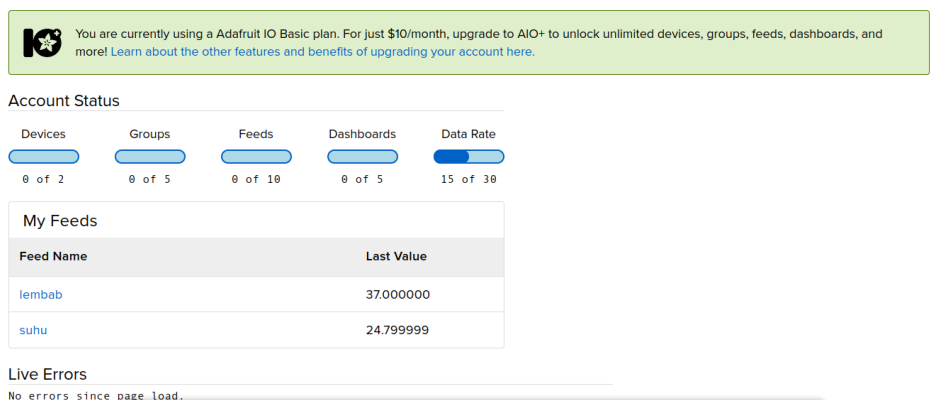
25. Jika proses koneksi lama, cek WiFi SSID apakah sudah benar atau lemot



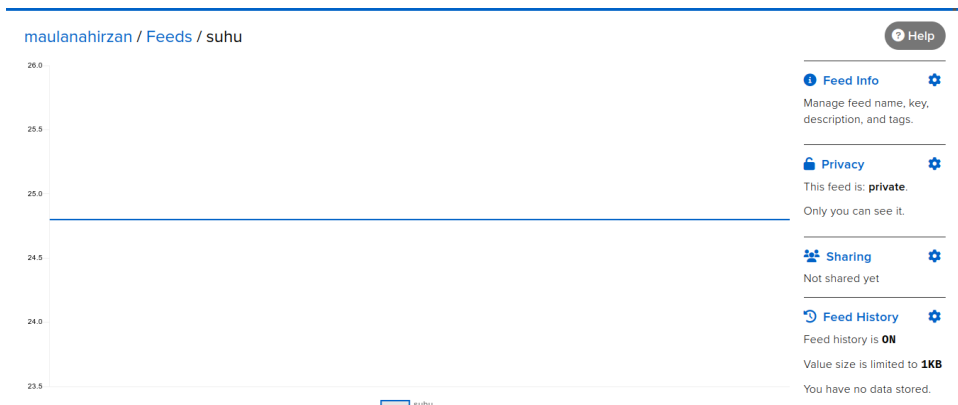
26. Alat terhubung dan berhasil mengirimkan data



27. Hasil di website **Adafruit IO**



28. Klik salah satu **feed** untuk melihat data





### Potongan Kode

suhu->save(temperature);  
 lembab->save(humidity);

```

Praktikum2 | Arduino IDE 2.2.1
Praktikum2.ino  config.h
58 void loop() {
59 // io.run(); is required for all sketches.
60 // it should always be present at the top of your loop
61 // function. it keeps the client connected to
62 // io.adafruit.com, and processes any incoming data.
63 // io.run();
64 io.run();
65 float temperature = dht.readTemperature();
66 float humidity = dht.readHumidity();
67 // save count to the 'counter' feed on Adafruit IO
68 Serial.print("sending -> ");
69 Serial.print(temperature);
70 Serial.print("\n");
71 Serial.println(humidity);}
72 suhu->save(temperature);
73 lembab->save(humidity);
74 // increment the count by 1
75 count++;
76 // Adafruit IO is rate limited for publishing, so a delay is required in
77 // between feed->save events. In this example, we will wait three seconds
78 // (1000 milliseconds == 1 second) during each loop.
79 delay(3000);
80
81
82
83
84
85
86
87
  
```

29. Untuk mengunduh, cukup klik **Download Data** di bagian bawah grafik

Download suhu Data

NOTE: You can download complete feed data once every ten minutes.

Download as JSON Download as CSV

Download All Data

Created at	Value	Location
2023/09/21 06:41:22PM	24.799999	0, 0, 0
2023/09/21 06:41:22PM	24.799999	0, 0, 0
2023/09/21 06:41:15PM	24.799999	0, 0, 0

# Bab 3

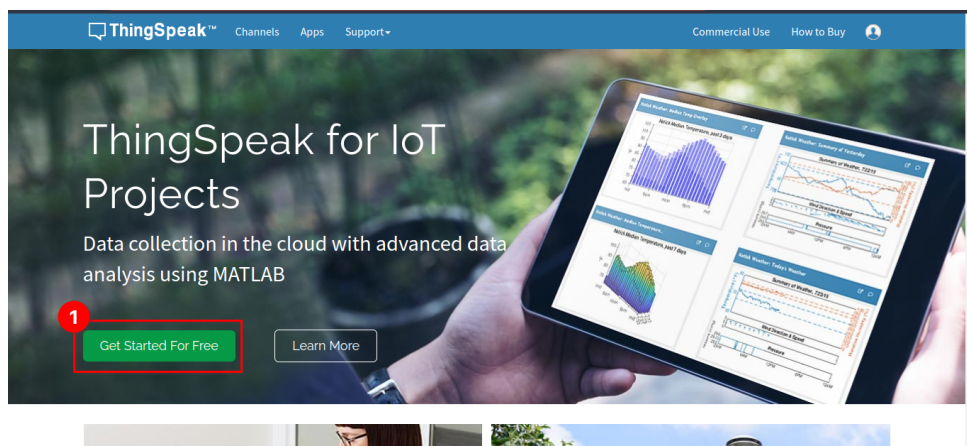
## Praktikum 3

### 3.1 ESP8266, DHT11, dan Thingspeak

Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Thingspeak. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 2** yang ada di halaman sebelumnya.

### 3.2 Tutorial

1. Untuk memulai praktikum ini, mahasiswa diwajibkan untuk membuat akun di <https://thingspeak.com/> secara gratis. Klik **Get started for free**



2. Klik **Create one!**

ThingSpeak™ Channels Apps Support- Commercial Use How to Buy

To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

**1** Email

No account? Create one!

By signing in, you agree to our [privacy policy](#).

Next

### 3. Isi informasi identitas

ThingSpeak™ Channels Apps Support- Commercial Use How to Buy

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

Create MathWorks Account

Email Address

maulanahirzan@gmail.com

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

Location

Indonesia

First Name

Alauddin Maulana

Last Name

Hirzan

Continue

Cancel

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

### 4. Centang untuk menggunakan email pribadi

ThingSpeak™ Channels Apps Support- Commercial Use How to Buy

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

Personal Email Detected

To use your organization's MATLAB, enter your work or university email

**1** Email Address

maulanahirzan@gmail.com

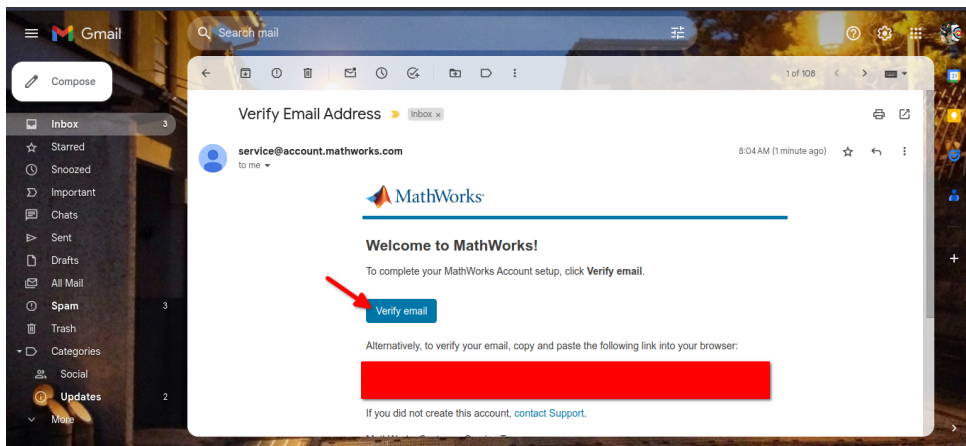
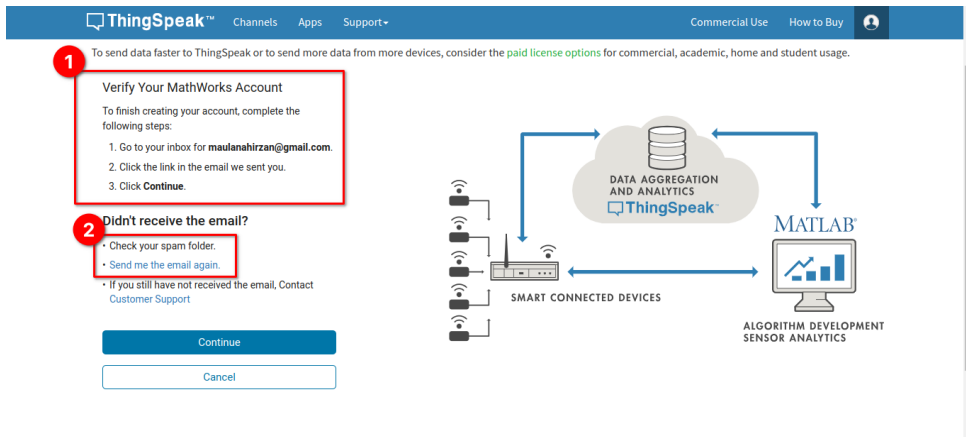
Use this email for my MathWorks Account

Continue

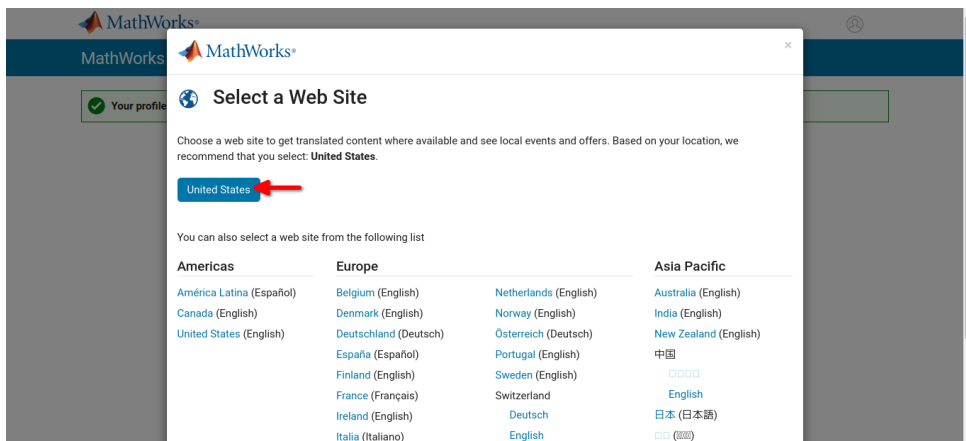
Cancel

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

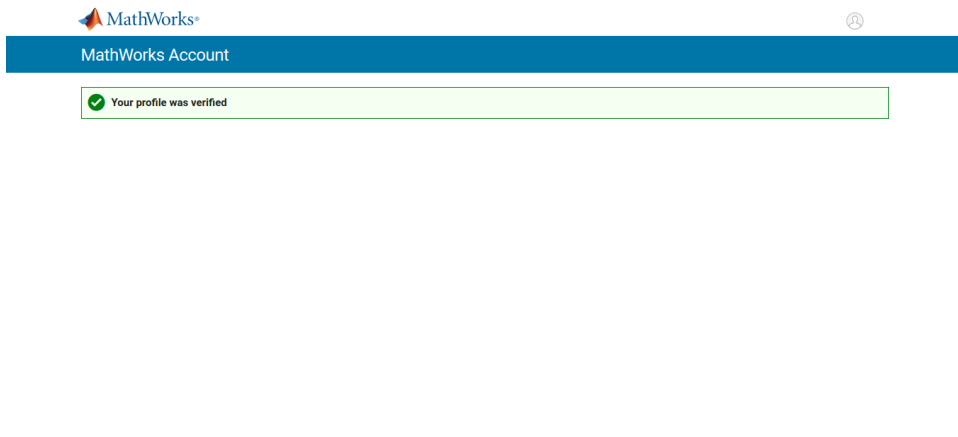
### 5. Cek email anda (termasuk SPAM) untuk verifikasi email. **JANGAN TUTUP WINDOW INI!!!**



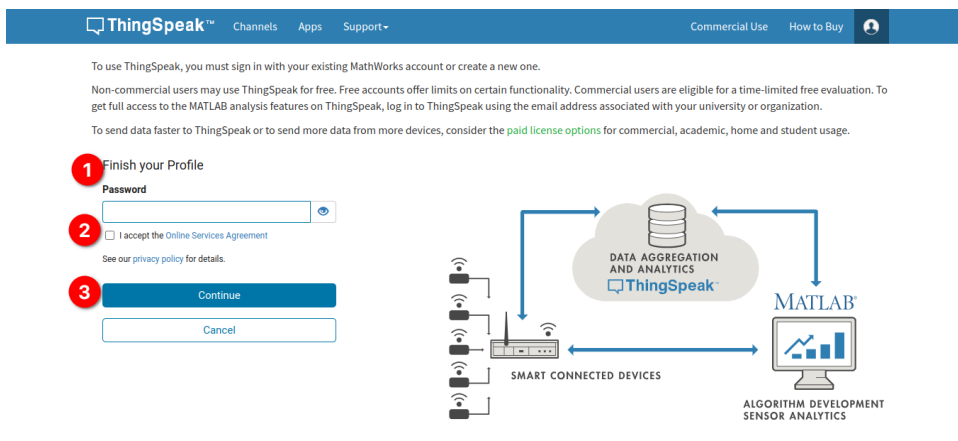
## 6. Pilih negara untuk website



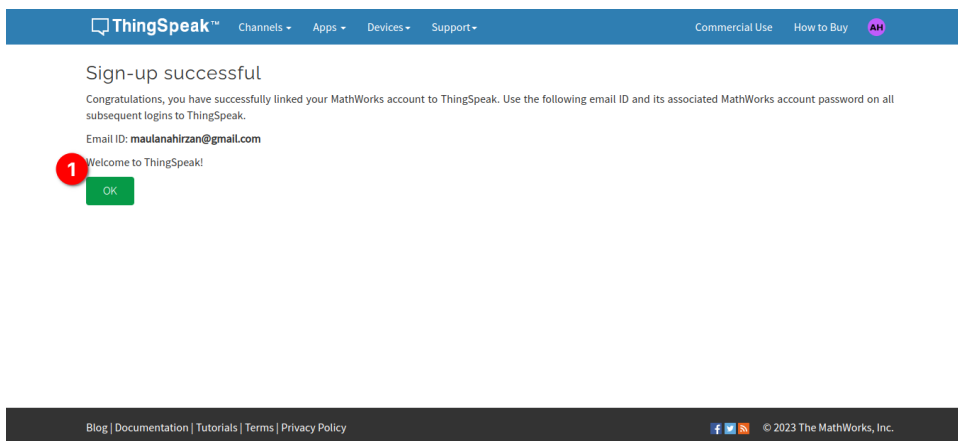
## 7. Akun sudah terverifikasi



8. Ketika sudah selesai, kembali ke **WINDOW** di **Langkah 5.** dan klik **Continue**

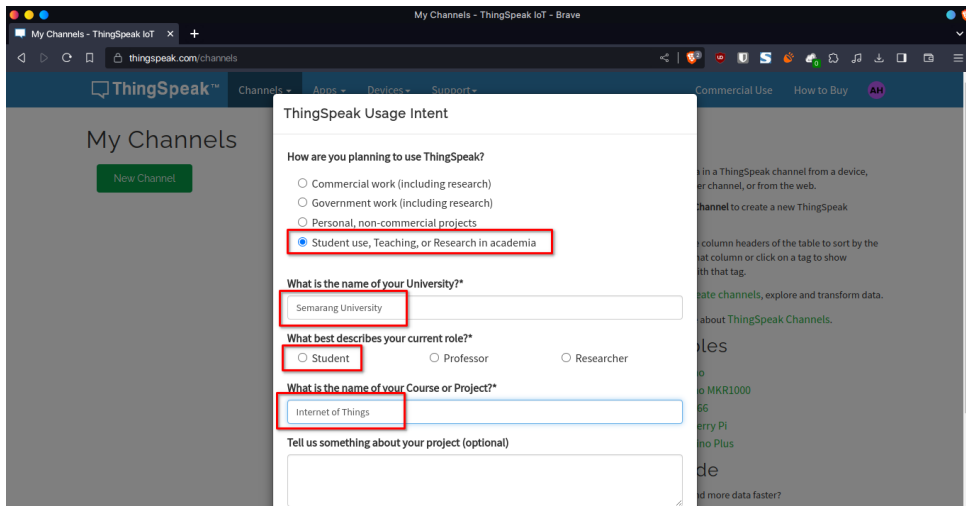


9. Klik **OK** untuk pindah ke **Dasbor**

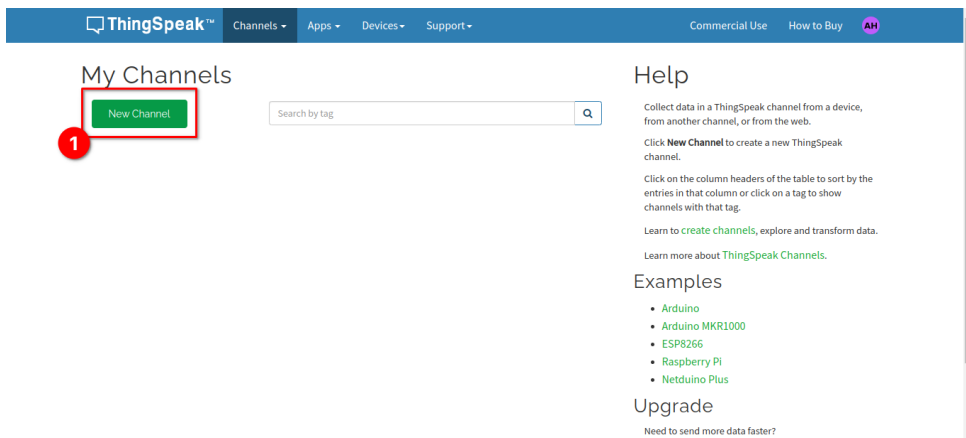


10. Di Dashboar akan ditanya penggunaan **Thingspeak.** Isi sesuai pertanyaan. Jangan lupa untuk klik **OK** atau **Continue**

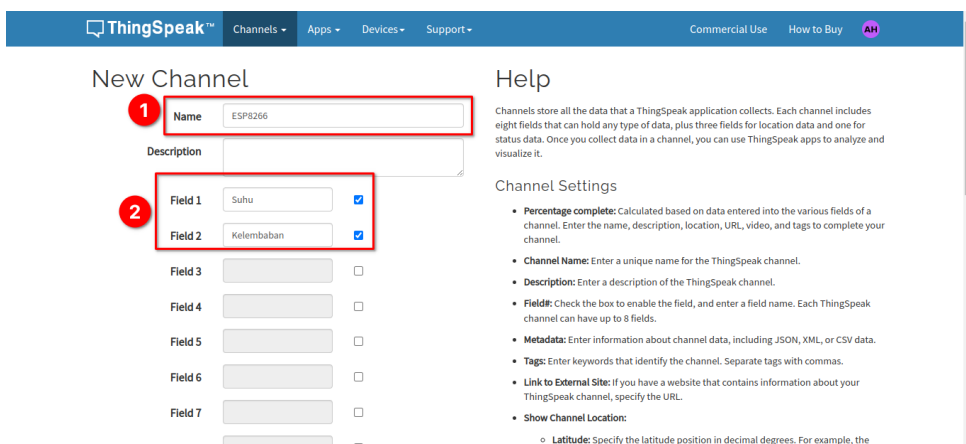




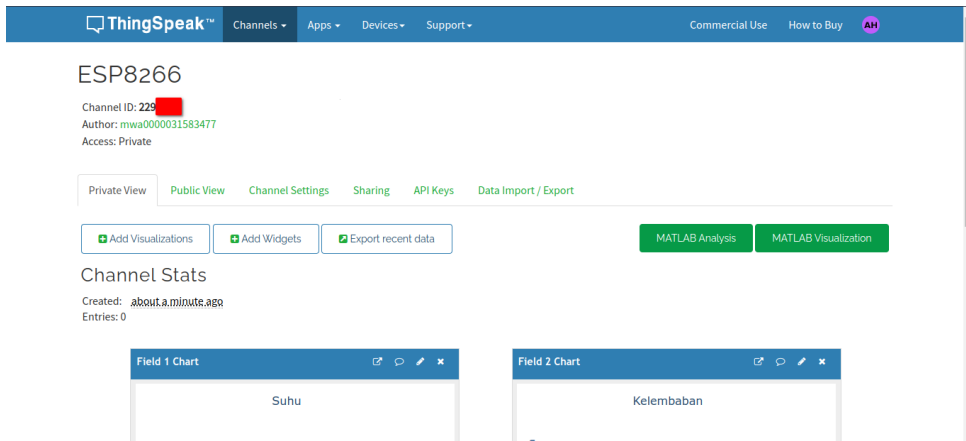
11. Jika sudah, buat **KANAL BARU** dengan klik **New Channel**



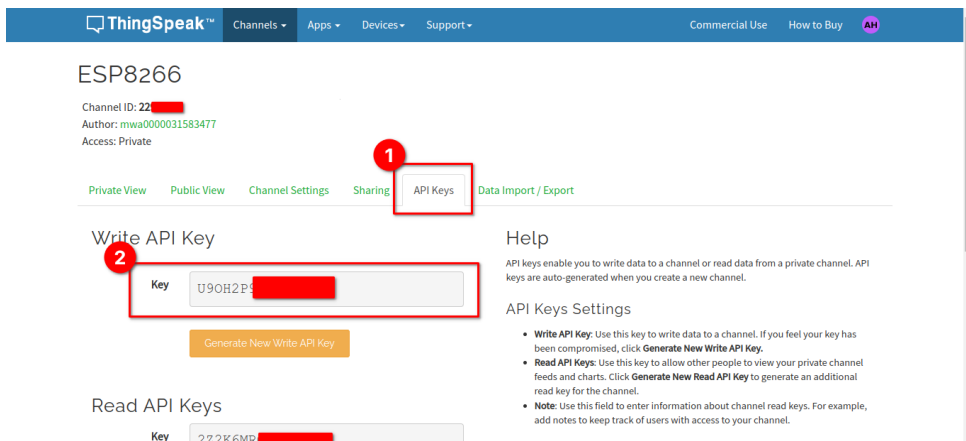
12. Beri nama **KANAL**, dan isi **2 Field** dengan nama **Suhu** dan **Kelembaban**. Klik **Save Channel** di bagian bawah



13. **Kanal** sudah siap dan simpan **Channel ID** untuk digunakan nanti.

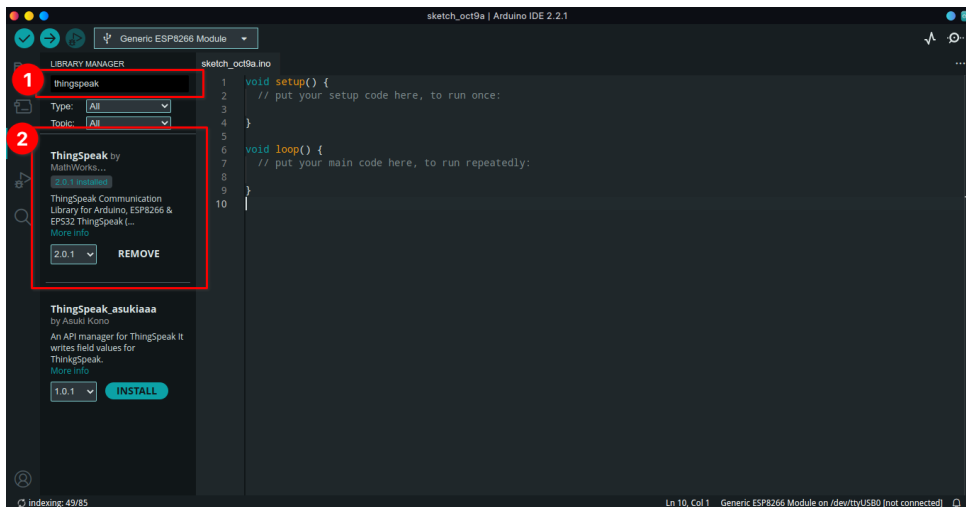


14. Pindah ke tab **API Keys**, dan kopi **Write API Key** untuk **Arduino IDE**



15. Jika **Channel ID** dan **Write API Key** sudah didapatkan. Langkah berikutnya adalah membuka **Arduino IDE**

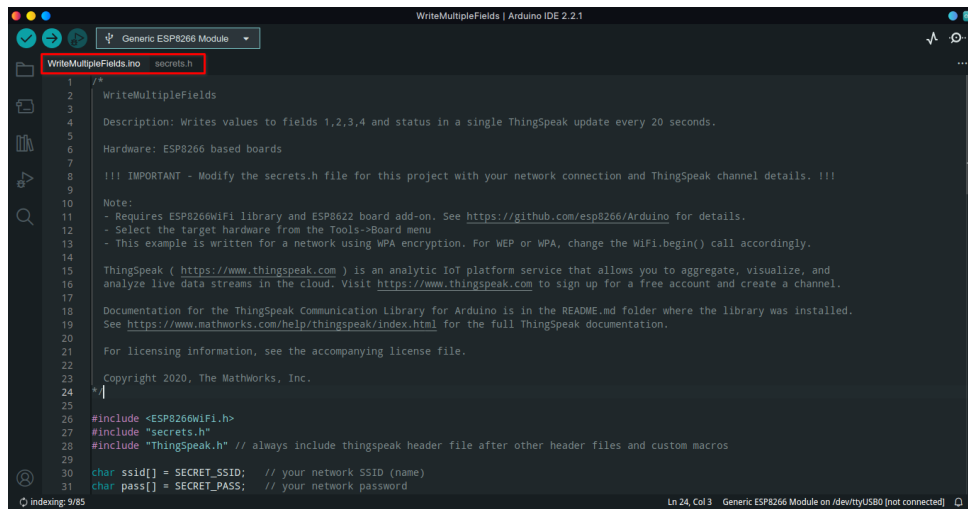
16. **Install Library Thingspeak**



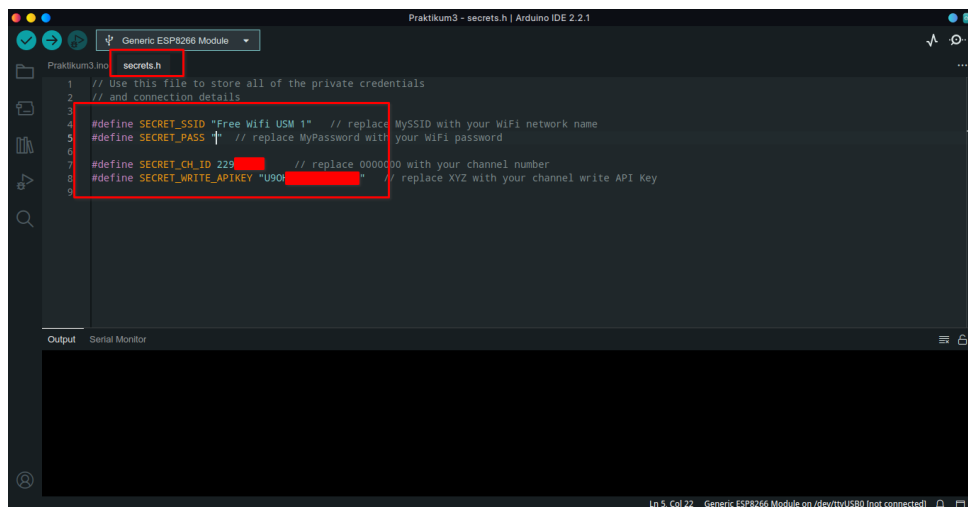
17. Untuk membuat program pengunggah data ke **Thingspeak**, gunakan **Example**

yang sudah disiapkan oleh **Library**. Klik **File** → **Examples** → **ThingSpeak** → **ESP8266** → **Program Board Directly** → **Write Multiple Fields**

18. Jika sudah, simpan proyek sebagai **Praktikum 3**



19. Ketika sudah siap, cukup edit file **secrets.h** melalui tab. Isi sesuai konfigurasi sebelumnya.



20. Kembali ke file **Praktikum3.ino**. Tambahkan **Library DHT** di bawah **ThingSpeak.h**. Lihat gambar

**Potongan Kode**

```
#include <DHT.h>
```

```
Praktikum3.ino secrets.h
9
10 Note:
11 - Requires ESP8266WiFi Library and ESP8262 board add-on. See https://github.com/esp8266/Arduino for details.
12 - Select the target hardware from the Tools->Board menu
13 - This example is written for a network using WPA encryption. For WEP or WPA, change the WiFi.begin() call accordingly.
14
15 ThingSpeak ( https://www.thingspeak.com ) is an analytic IoT platform service that allows you to aggregate, visualize, and
16 analyze live data streams in the cloud. Visit https://www.thingspeak.com to sign up for a free account and create a channel.
17
18 Documentation for the ThingSpeak Communication Library for Arduino is in the README.md folder where the library was installed.
19 See https://www.mathworks.com/help/thingspeak/index.html for the full ThingSpeak documentation.
20
21 For licensing information, see the accompanying license file.
22
23 Copyright 2020, The MathWorks, Inc.
24
25 */
26 #include <ESP8266WiFi.h>
27 #include "secrets.h"
28 #include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros
29 #include <DHT.h>
30
31 char ssid[] = SECRET_SSID; // your network SSID (name)
32 char pass[] = SECRET_PASS; // your network password
33 int keyIndex = 0; // your network key index number (needed only for WEP)
34 WiFiClient client;
35
36 unsigned long myChannelNumber = SECRET_CH_ID;
```

21. Hapus kode berikut

```
Praktikum3.ino secrets.h
25
26 #include <ESP8266WiFi.h>
27 #include "secrets.h"
28 #include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros
29
30 char ssid[] = SECRET_SSID; // your network SSID (name)
31 char pass[] = SECRET_PASS; // your network password
32 int keyIndex = 0; // your network key index number (needed only for WEP)
33 WiFiClient client;
34
35 unsigned long myChannelNumber = SECRET_CH_ID;
36 const char * myWriteAPIKey = SECRET_WRITE_APIKEY;
37
38 // Initialize our values
39 int number1 = 0;
40 int number2 = random(0,100);
41 int number3 = random(0,100);
42 int number4 = random(0,100);
43 String myStatus = "";
44
45 void setup() {
46   Serial.begin(115200); // Initialize serial
47   while (!Serial) {
48     ; // wait for serial port to connect. Needed for Leonardo native USB port only
49   }
50
51   WiFi.mode(WIFI_STA);
52   ThingSpeak.begin(client); // Initialize ThingSpeak
```

22. Ganti kode yang sudah dihapus tadi dengan kode berikut:

```
Potongan Kode
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

```
Praktikum3.ino secrets.h
22 Copyright 2020, The Mathworks, Inc.
23 */
24
25
26 #include <ESP8266WiFi.h>
27 #include "secrets.h"
28 #include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros
29 #include <DHT.h>
30
31 char ssid[] = SECRET_SSID; // your network SSID (name)
32 char pass[] = SECRET_PASS; // your network password
33 int keyIndex = 0; // your network key Index number (needed only for WEP)
34 WiFiClient client;
35
36 unsigned long myChannelNumber = SECRET_CH_ID;
37 const char * myWriteAPIKey = SECRET_WRITE_APIKEY;
38
39 #define DHTPIN 2
40 #define DHTTYPE DHT11
41 DHT dht(DHTPIN, DHTTYPE);
42
43
44 void setup() {
45   Serial.begin(115200); // Initialize serial
46   while (!Serial) {
47     ; // wait for serial port to connect. Needed for Leonardo native USB port only
48   }
49   WiFi.mode(WIFI_STA);
50   ThingSpeak.begin(client); // Initialize ThingSpeak
51 }
```

23. Lalu di dalam **FUNGSI SETUP**, tambahkan kode berikut setelah baris **ThingSpeak.begin()**:

**Potongan Kode**

```
// Mulai Sensor DHT11
dht.begin();
```

```
Praktikum3.ino secrets.h
38 #define DHTPIN 2
39 #define DHTTYPE DHT11
40 #define DHTTYPE DHT11
41 DHT dht(DHTPIN, DHTTYPE);
42
43 void setup() {
44   Serial.begin(115200); // Initialize serial
45   while (!Serial) {
46     ; // wait for serial port to connect. Needed for Leonardo native USB port only
47   }
48
49   WiFi.mode(WIFI_STA);
50   ThingSpeak.begin(client); // Initialize ThingSpeak
51   // Mulai Sensor DHT11
52   dht.begin();
53 }
54
55 void loop() {
56
57   // Connect or reconnect to WiFi
58   if(WiFi.status() != WL_CONNECTED){
59     Serial.print("Attempting to connect to SSID: ");
60     Serial.println(SECRET_SSID);
61     while(WiFi.status() != WL_CONNECTED){
62       WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using open or WEP network
63       Serial.print(".");
64       delay(5000);
65     }
```

24. Di dalam **FUNGSI LOOP** Hapus kode berikut:

```
Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino secrets.h
55 void loop() {
56
57 // Connect or reconnect to WiFi
58 if(WiFi.status() != WL_CONNECTED){
59   Serial.print("Attempting to connect to SSID: ");
60   Serial.println(SECRET_SSID);
61   while(WiFi.status() != WL_CONNECTED){
62     WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using open or WEP network
63     Serial.print(".");
64     delay(5000);
65   }
66   Serial.println("\nConnected.");
67 }
68
69 // set the fields with the values
70 ThingSpeak.setField(1, number1);
71 ThingSpeak.setField(2, number2);
72 ThingSpeak.setField(3, number3);
73 ThingSpeak.setField(4, number4);
74
75 // figure out the status message
76 if(number1 > number2){
77   myStatus = String("field1 is greater than field2");
78 }
79 else if(number1 < number2){
80   myStatus = String("field1 is less than field2");
81 }
82 else{
83   myStatus = String("field1 equals field2");
84 }
85
86 // set the status
87 ThingSpeak.setStatus(myStatus);
88
89 // write to the ThingSpeak channel
90 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
91 if(x == 200){
92   Serial.println("Channel update successful.");
93 }
94 else{
95   Serial.println("Problem updating channel. HTTP error code " + String(x));
96 }
97
98 // change the values
99 number1++;
100 if(number1 > 99){
101   number1 = 0;
102 }
103 number2 = random(0,100);
104 number3 = random(0,100);
105 number4 = random(0,100);
106
107 delay(20000); // Wait 20 seconds to update the channel again
108 }
109 }
```

```
Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino secrets.h
69 // set the fields with the values
70 ThingSpeak.setField(1, number1);
71 ThingSpeak.setField(2, number2);
72 ThingSpeak.setField(3, number3);
73 ThingSpeak.setField(4, number4);
74
75 // figure out the status message
76 if(number1 > number2){
77   myStatus = String("field1 is greater than field2");
78 }
79 else if(number1 < number2){
80   myStatus = String("field1 is less than field2");
81 }
82 else{
83   myStatus = String("field1 equals field2");
84 }
85
86 // set the status
87 ThingSpeak.setStatus(myStatus);
88
89 // write to the ThingSpeak channel
90 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
91 if(x == 200){
92   Serial.println("Channel update successful.");
93 }
94 else{
95   Serial.println("Problem updating channel. HTTP error code " + String(x));
96 }
97
98 // change the values
99 number1++;
100 if(number1 > 99){
101   number1 = 0;
102 }
103 number2 = random(0,100);
104 number3 = random(0,100);
105 number4 = random(0,100);
106
107 delay(20000); // Wait 20 seconds to update the channel again
108 }
109 }
```

```
Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino secrets.h
82 else{
83   myStatus = String("field1 equals field2");
84 }
85
86 // set the status
87 ThingSpeak.setStatus(myStatus);
88
89 // write to the ThingSpeak channel
90 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
91 if(x == 200){
92   Serial.println("Channel update successful.");
93 }
94 else{
95   Serial.println("Problem updating channel. HTTP error code " + String(x));
96 }
97
98 // change the values
99 number1++;
100 if(number1 > 99){
101   number1 = 0;
102 }
103 number2 = random(0,100);
104 number3 = random(0,100);
105 number4 = random(0,100);
106
107 delay(20000); // Wait 20 seconds to update the channel again
108 }
109 }
```

25. Hasil AKHIR SEHARUSNYA:

```
Praktikum3.ino secrets.h
54
55 void loop() {
56
57 // Connect or reconnect to WiFi
58 if(WiFi.status() != WL_CONNECTED){
59 Serial.print("Attempting to connect to SSID: ");
60 Serial.println(SECRET_SSID);
61 while(WiFi.status() != WL_CONNECTED){
62 WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using open or WEP network
63 Serial.print(".");
64 delay(5000);
65 }
66 Serial.println("\nConnected.");
67 }
68
69 // set the fields with the values
70 ThingSpeak.setField(1, number1);
71 ThingSpeak.setField(2, number2);
72
73 // write to the ThingSpeak channel
74 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
75 if(x == 200){
76 Serial.println("Channel update successful.");
77 }
78 else{
79 Serial.println("Problem updating channel. HTTP error code " + String(x));
80 }
81 }
```

26. Jika sudah tambahkan kode berikut tepat di atas **ThingSpeak.setField()**

Potongan Kode

```
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
```

```
Praktikum3.ino secrets.h
54
55 void loop() {
56
57 // Connect or reconnect to WiFi
58 if(WiFi.status() != WL_CONNECTED){
59 Serial.print("Attempting to connect to SSID: ");
60 Serial.println(SECRET_SSID);
61 while(WiFi.status() != WL_CONNECTED){
62 WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using open or WEP network
63 Serial.print(".");
64 delay(5000);
65 }
66 Serial.println("\nConnected.");
67 }
68
69 float temperature = dht.readTemperature();
70 float humidity = dht.readHumidity();
71
72 // set the fields with the values
73 ThingSpeak.setField(1, number1);
74 ThingSpeak.setField(2, number2);
75
76 // write to the ThingSpeak channel
77 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
78 if(x == 200){
79 Serial.println("Channel update successful.");
80 }
81 else{
82 Serial.println("Problem updating channel. HTTP error code " + String(x));
83 }
```

27. Lalu ubah kode di dalam **ThingSpeak.setField** sesuai kode berikut:

Potongan Kode

```
ThingSpeak.setField(1, temperature);
ThingSpeak.setField(2, humidity);
```

```
Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino secrets.h
54
55
56
57
58 // Connect or reconnect to WiFi
59 if(WiFi.status() != WL_CONNECTED){
60   Serial.print("Attempting to connect to SSID: ");
61   Serial.println(SECRET_SSID);
62   while(WiFi.status() != WL_CONNECTED){
63     WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using open or WEP network
64     Serial.print(".");
65     delay(5000);
66   }
67   Serial.println("\nConnected.");
68 }
69
70 float temperature = dht.readTemperature();
71 float humidity = dht.readHumidity();
72
73 // set the fields with the values
74 ThingSpeak.setField(1, temperature);
75 ThingSpeak.setField(2, humidity);
76
77 // write to the ThingSpeak channel
78 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
79 if(x == 200){
80   Serial.println("Channel update successful.");
81 }
82 else{
83   Serial.println("Problem updating channel. HTTP error code " + String(x));
84 }
85
86 delay(20000); // Wait 20 seconds to update the channel again
87
88 }
89
90
Output
```

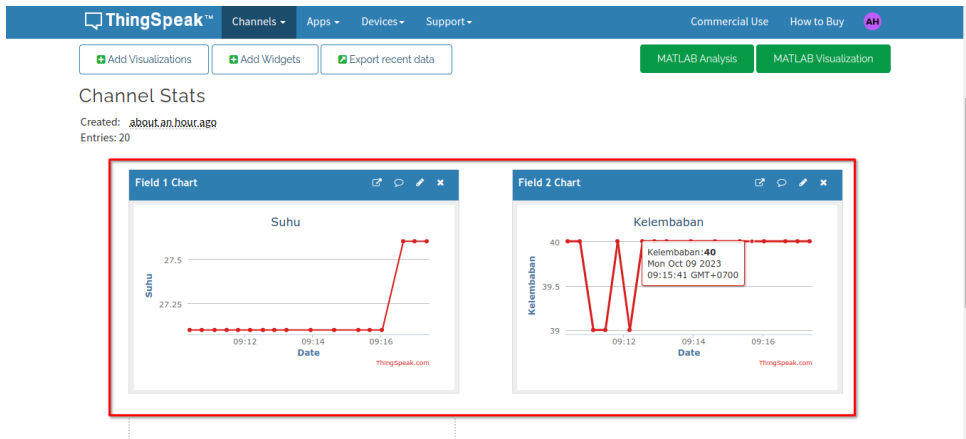
28. Verifikasi untuk memastikan kode sudah benar. Lalu klik Upload

```
Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum3.ino secrets.h
72 // set the fields with the values
73 ThingSpeak.setField(1, temperature);
74 ThingSpeak.setField(2, humidity);
75
76 // write to the ThingSpeak channel
77 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
78 if(x == 200){
79   Serial.println("Channel update successful. ");
80   Serial.print(temperature);
81   Serial.print(" ");
82   Serial.println(humidity);
83 }
84 else{
85   Serial.println("Problem updating channel. HTTP error code " + String(x));
86 }
87
88 delay(20000); // Wait 20 seconds to update the channel again
89
90
Output Serial Monitor
Writing at 0x00024000... (76 %)
Writing at 0x00028000... (84 %)
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 286352 bytes (209997 compressed) at 0x00000000 in 18.5 seconds (effective 123.8 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
Ln 82, Col 30 Generic ESP8266 Module on /dev/ttyUSB0
```

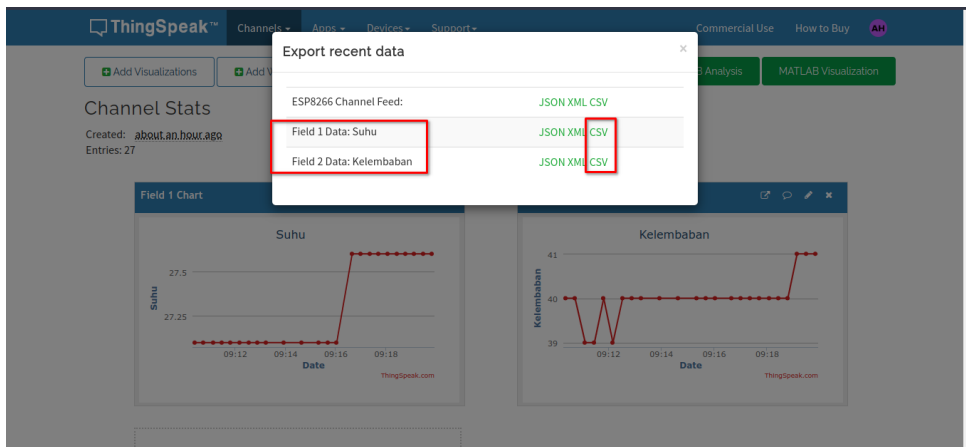
29. Data terkirim dan terunggah

```
Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum3.ino secrets.h
72 // set the fields with the values
73 ThingSpeak.setField(1, temperature);
74 ThingSpeak.setField(2, humidity);
75
76 // write to the ThingSpeak channel
77 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
78 if(x == 200){
79   Serial.println("Channel update successful. ");
80   Serial.print(temperature);
81   Serial.print(" ");
82   Serial.println(humidity);
83 }
84 else{
85   Serial.println("Problem updating channel. HTTP error code " + String(x));
86 }
87
88 delay(20000); // Wait 20 seconds to update the channel again
89
90
Output Serial Monitor x
Message (Enter to send message to Generic ESP8266 Module on /dev/ttyUSB0)
09:17:00.437 -- Channel update successful. 27.69 49.09
Ln 82, Col 30 Generic ESP8266 Module on /dev/ttyUSB0
```





30. Untuk download data, klik **Export recent data** di halaman yang sama. Pilih masing-masing **Field** dengan format **CSV**



# Bab 4

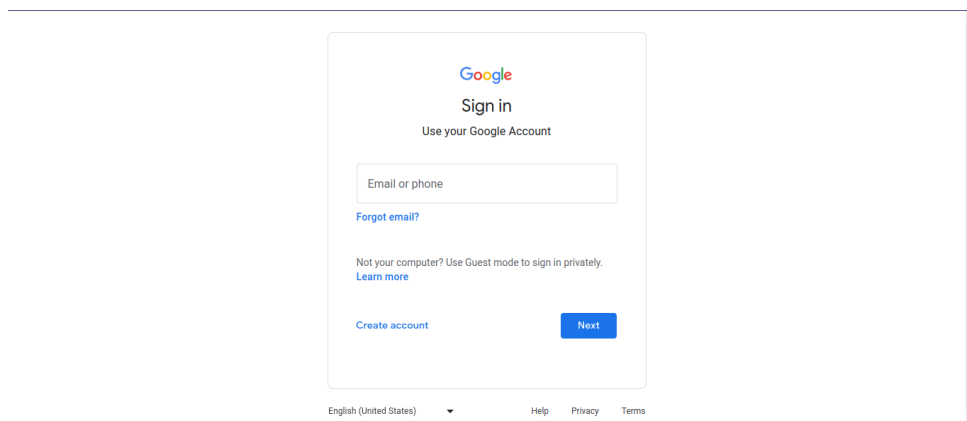
## Praktikum 4

### 4.1 ESP8266, DHT11, dan Firebase Realtime

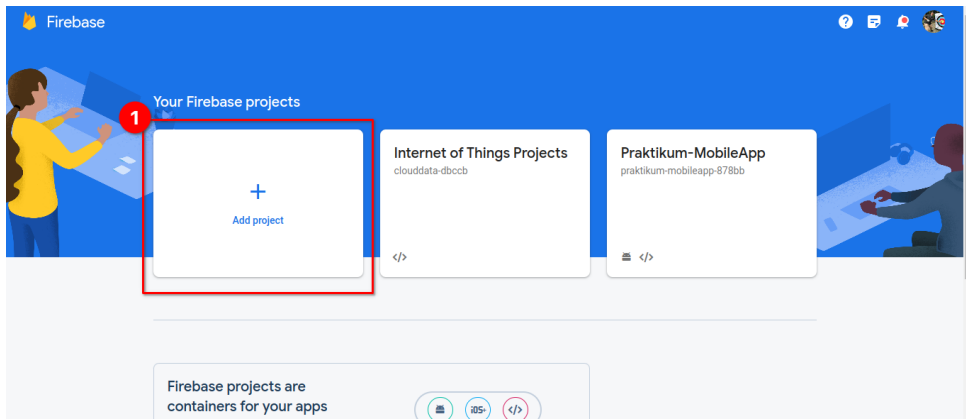
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Firebase Realtime. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 3** yang ada di halaman sebelumnya.

### 4.2 Tutorial

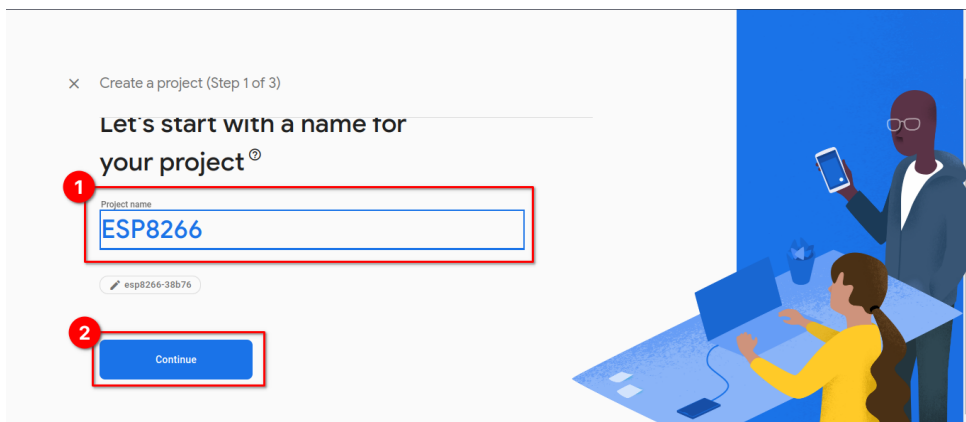
1. Buka browser lalu klik link berikut : <https://console.firebase.google.com/>. Login dengan akun Google dan klik kembali link tersebut.



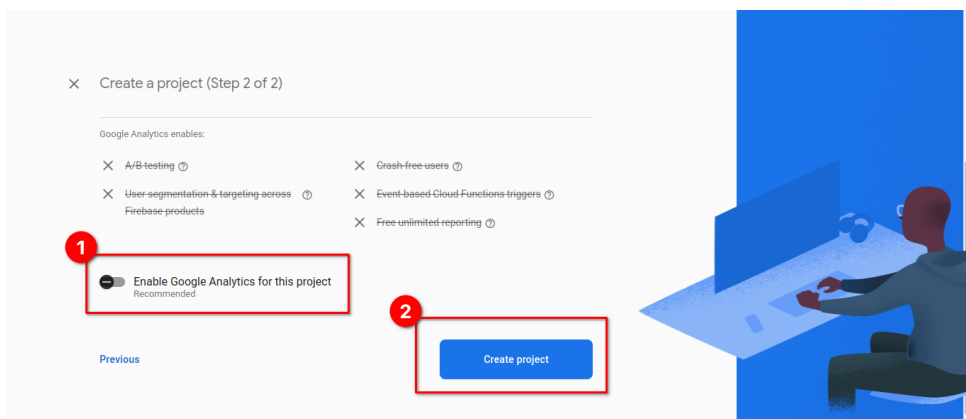
2. Buat projek baru dengan melakukan klik tanda +



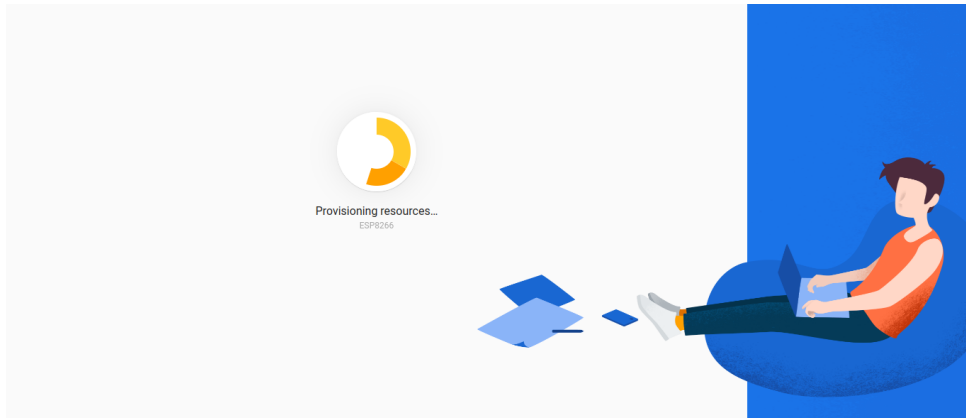
3. Isi nama projek



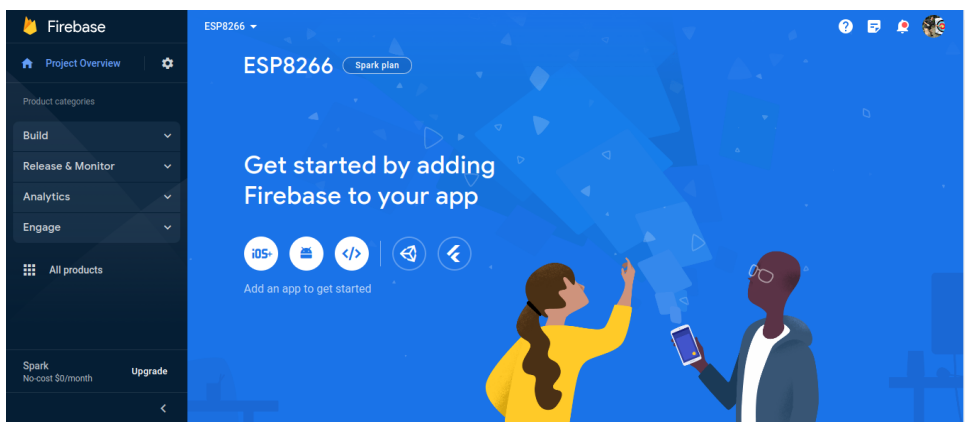
4. Matikan Google Analytic dan klik Create Project



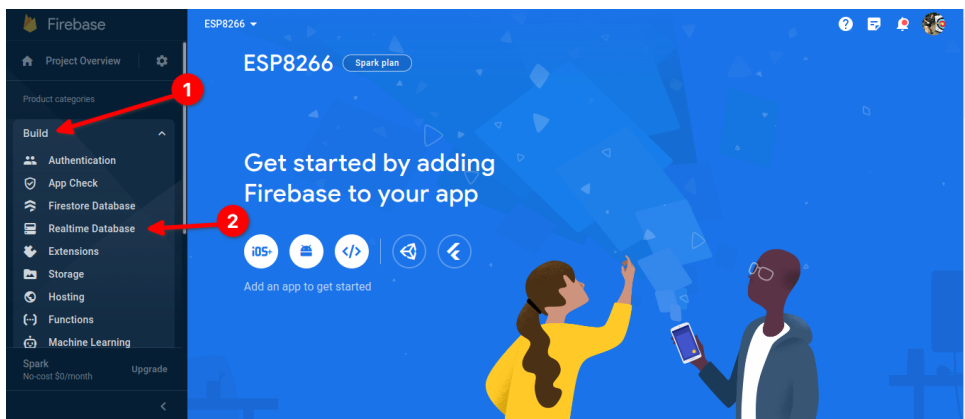
5. Tunggu proses berlangsung dan klik tombol apabila sudah muncul



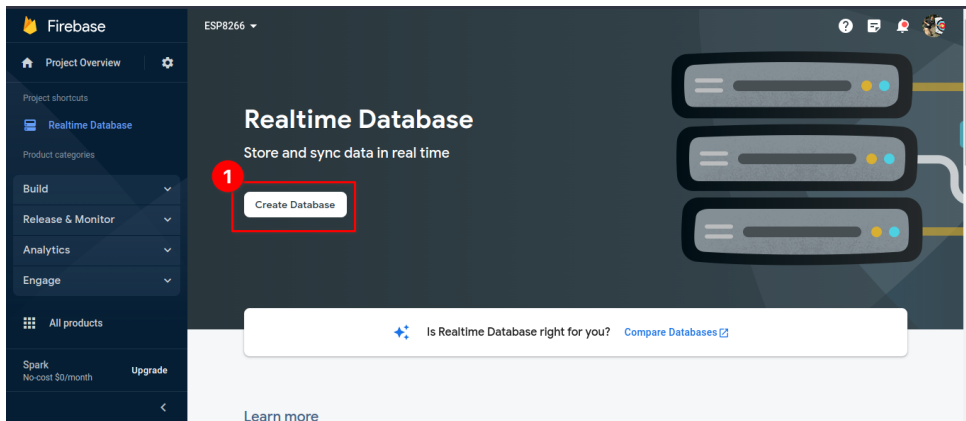
6. Firebase akan menampilkan dasbor sistem



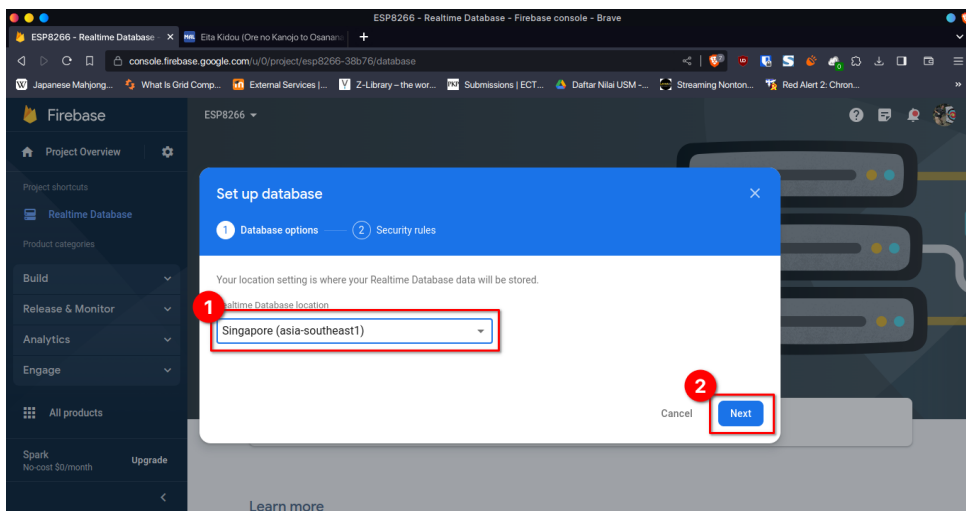
7. Klik **Build** dan pilih **Realtime Database**



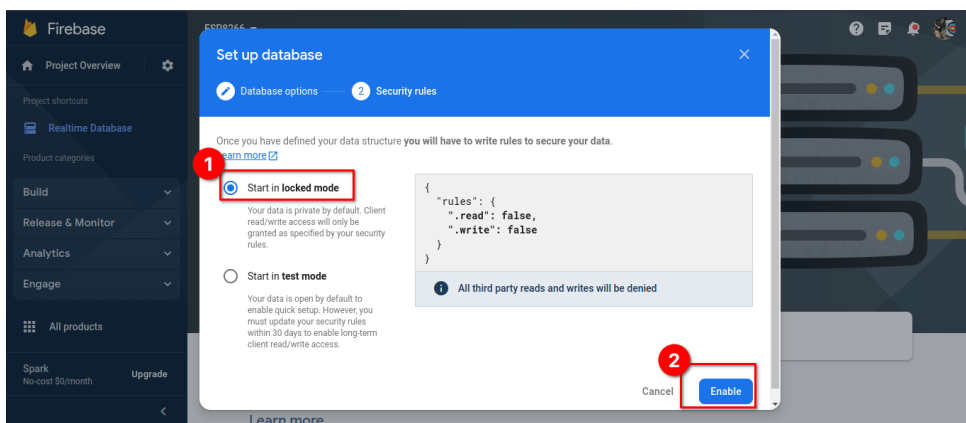
8. Klik **Create Database**



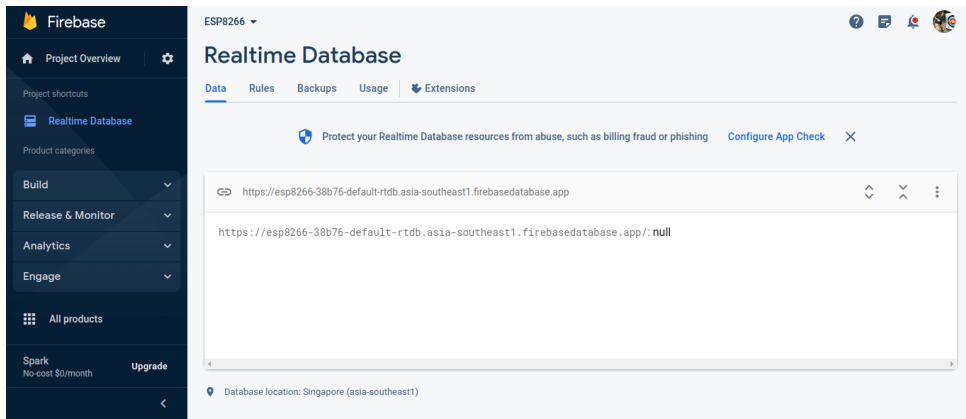
### 9. Pilih Lokasi dan Klik Next



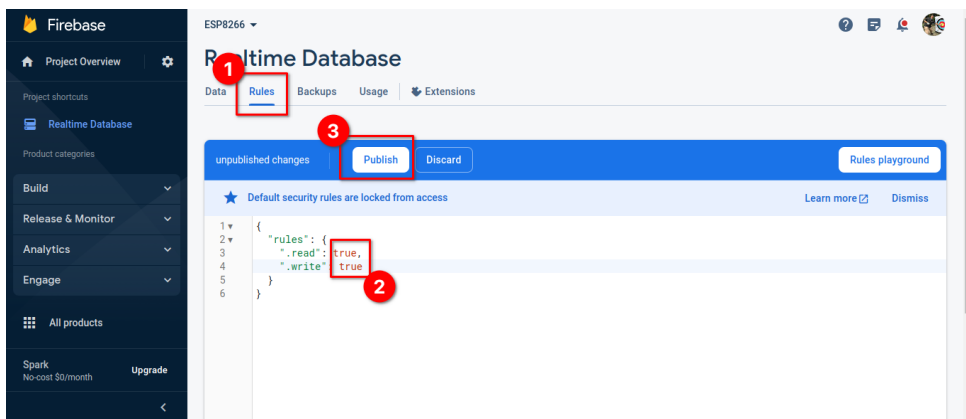
### 10. Pilih Locked Mode dan klik Enable



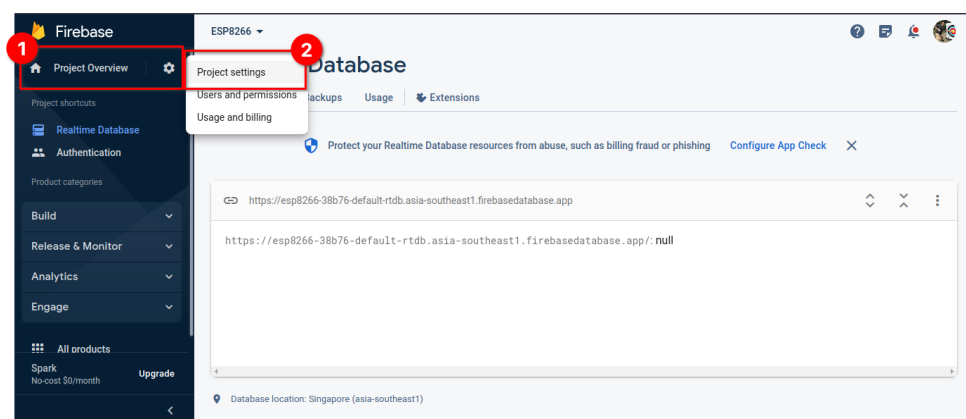
### 11. Database sudah dibuat



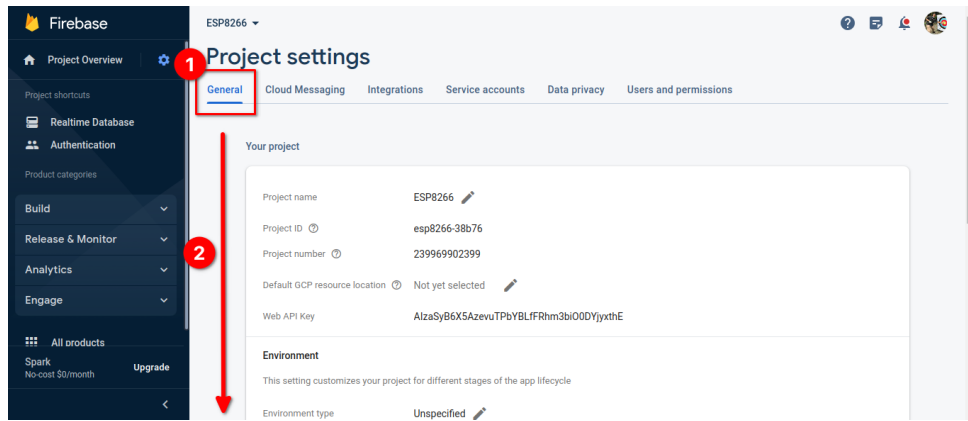
12. Sebelumnya ubah aturan database dengan klik **Rules**, dan ubah kata **false** menjadi **true**. dan klik **Publish**



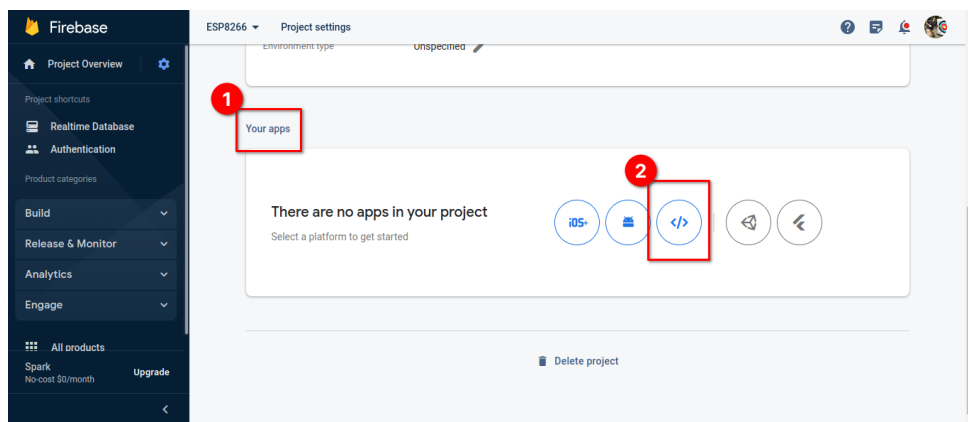
13. Untuk membuat kunci, klik **Roda Gigi Project Overview**, pilih **Project Settings**



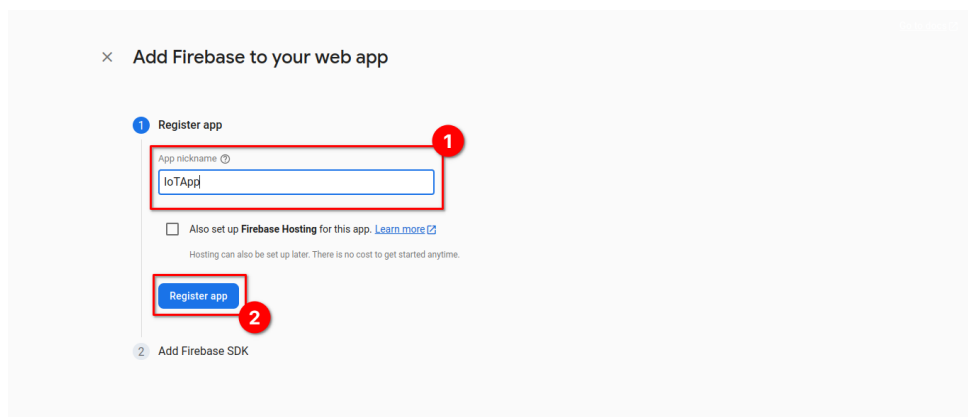
14. Di bagian **General**, scroll turun hingga menemukan **Apps**



15. Di bagian **Your Apps** pilih **Web**



16. Isikan nama app, dan pilih **Register app**



17. Di tahap selanjutnya, sistem akan membuat **API Key** dan **Database URL**. Kopi data ini ke Notepad

```

$ npm install firebase

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyB6XSzreU7DhY",
  authDomain: "esp8266-38b76.firebaseio.com",
  databaseURL: "https://esp8266-38b76-default-rtdb.firebaseio.com",
  projectId: "esp8266-38b76",
  storageBucket: "esp8266-38b76.appspot.com",
  messagingSenderId: "239969902399",
  appId: "1:239969902399:web:8b1411b7b8dcef0252bd8a"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);

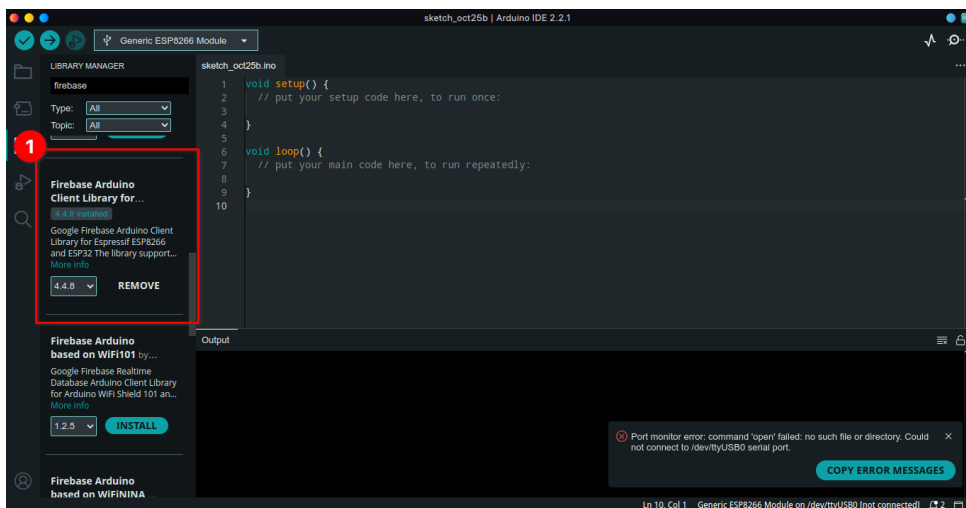
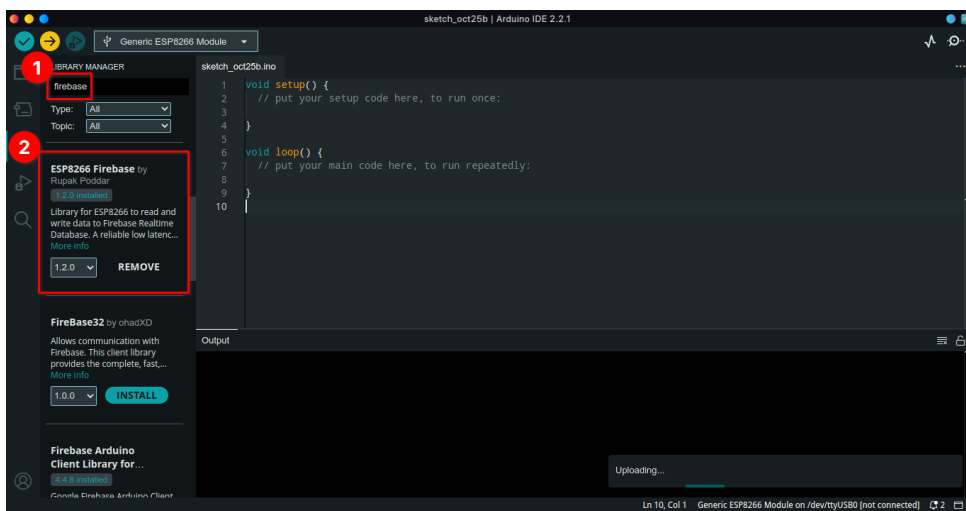
```

Note: This option uses the [modular JavaScript SDK](#), which provides reduced SDK size.

Learn more about Firebase for web: [Get Started](#) | [Web SDK API Reference](#) | [Samples](#)

Kopi ke Notepad

18. Di Arduino IDE, buka Libraries dan install **Firestore Arduino Client Library**



19. Buat proyek baru dengan template yang sudah ada. Klik **File** → **Examples** → **Firestore Arduino Client Library for ESP8266 and ESP32** → **FirestoreJson** → **Client** → **Firestore**



```
Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum4.ino
1
2 /**
3  * Created by K. Suwatchai (Mobizt)
4  *
5  * Email: k_suwatchai@hotmail.com
6  *
7  * Github: https://github.com/mobizt/FirebaseJson
8  *
9  * Copyright (c) 2023 mobizt
10 *
11 */
12
13 #include <Arduino.h>
14 #if defined(ESP32) || defined(ARDUINO_RASPBERRY_PI_PICO_W)
15 #include <WiFi.h>
16 #elif defined(ESP8266)
17 #include <ESP8266WiFi.h>
18 #endif
19
20 Output
Ln 12, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]
```

20. Hapus beberapa bagian kode berikut:

- Bagian 1

```
Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum4.ino
43
44 // Define Firebase Data object
45 FirebaseData fbdo;
46
47 FirebaseAuth auth;
48 FirebaseConfig config;
49
50 unsigned long sendDataPreMillis = 0;
51
52 unsigned long count = 0;
53
54 void setup()
55 {
56
57   Serial.begin(115200);
58
59   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
60   Serial.print("Connecting to Wi-Fi");
61   while (WiFi.status() != WL_CONNECTED)
62     delay(500);
63   Serial.println();
64
65   auth.begin(FirebaseConfig, config);
66   fbdo.begin(auth);
67
68   FirebaseJson jObj;
69   FirebaseJsonArray arr;
70
71   jObj.set("value", count);
72   arr.add("a", "b");
73   arr.add("c", true);
74   arr.add("d", 45);
75   arr.add("e", (float)6.1432);
76   arr.add("f", 123.456789);
77
78   Serial.print("Set array...");
79   fbdo.setArray("test/array", arr);
80   Serial.println();
81
82   Serial.print("Get array...");
83   FirebaseJsonArray arr2 = fbdo.toArray("test/array");
84   Serial.println();
85
86   Serial.print("Push json...");
87   fbdo.pushJSON("test/push", jObj);
88   Serial.println();
89
90   jObj.set("value", count++);
91   Serial.print("Update json...");
92   fbdo.updateNode("test/push", fbdo.pushName(), jObj);
93   Serial.println();
94
95   count++;
96
97 }
98
99 Output
Ln 52, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]
```

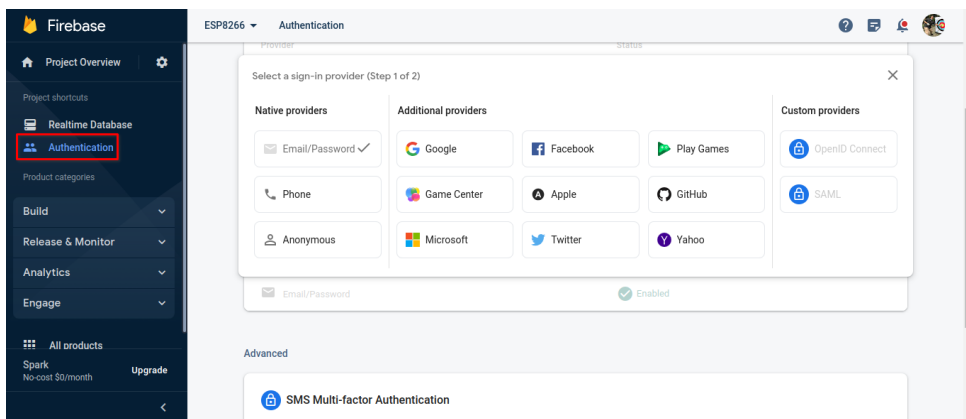
- Bagian 2

```
Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum4.ino
96
97   json.setDoubleDigits(2);
98   json.add("value", count);
99
100   Serial.print("Set json...");
101   fbdo.setJSON("test/json", json);
102   Serial.println();
103
104   Serial.print("Get json...");
105   FirebaseJson jVal = fbdo.toJSON("test/json");
106   Serial.println();
107
108   FirebaseJsonArray arr;
109   arr.setFloatDigits(2);
110   arr.setDoubleDigits(4);
111   arr.add("a", "b");
112   arr.add("c", true);
113   arr.add("d", 45);
114   arr.add("e", (float)6.1432);
115   arr.add("f", 123.456789);
116
117   Serial.print("Set array...");
118   fbdo.setArray("test/array", arr);
119   Serial.println();
120
121   Serial.print("Get array...");
122   FirebaseJsonArray arr2 = fbdo.toArray("test/array");
123   Serial.println();
124
125   Serial.print("Push json...");
126   fbdo.pushJSON("test/push", jObj);
127   Serial.println();
128
129   jObj.set("value", count++);
130   Serial.print("Update json...");
131   fbdo.updateNode("test/push", fbdo.pushName(), jObj);
132   Serial.println();
133
134   count++;
135
136 }
137
138 Output
DATA 1628 initialized variables
Ln 120, Col 17 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]
```

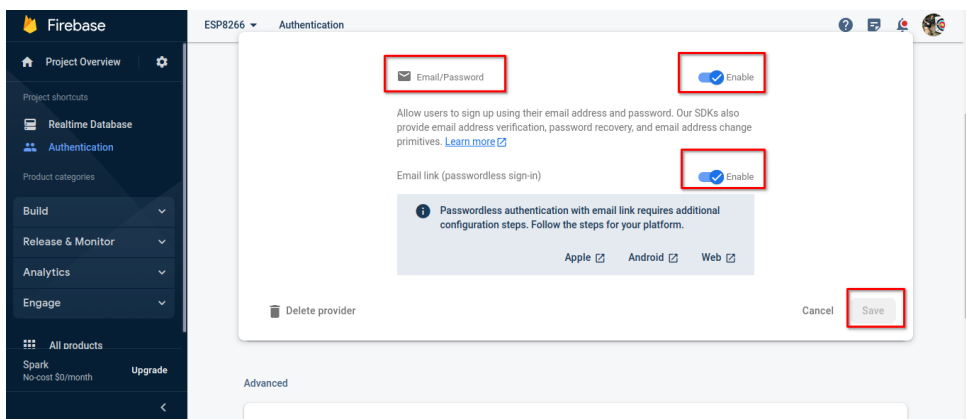
21. Lalu kembali ke bagian atas, dan ubah kode berikut:

```
Praktikum4.ino
1 #include <addons/RTDBHelper.h>
25 /* 1. Define the WiFi credentials */
26 #define WIFI_SSID "MikroTik-Net"
27 #define WIFI_PASSWORD " "
28
29 /* 2. Define the API Key */
30 #define API_KEY "AIzaSyB8X5Azev "
31
32 /* 3. Define the RTDB URL */
33 #define DATABASE_URL "https:
34
35 /* 4. Define the user email and password that already registered or added in your project */
36 #define USER_EMAIL "maulana@rizan@gmail.com"
37 #define USER_PASSWORD "1234567890"
38
39 FirebaseData fbdo;
40
41 FirebaseAuth auth;
42 FirebaseConfig config;
43
44 unsigned long sendDataPrevMillis = 0;
45
46 unsigned long count = 0;
47
48 void setup()
49 {
50 }
```

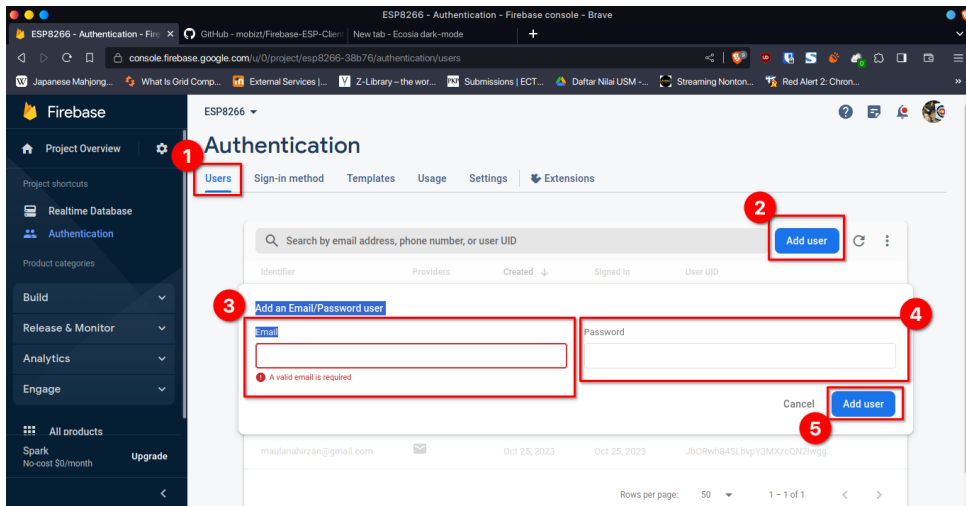
22. Bagian berikutnya adalah akun. Buka kembali **Firestore**, buka menu **Build** lalu **Firestore Authentication**



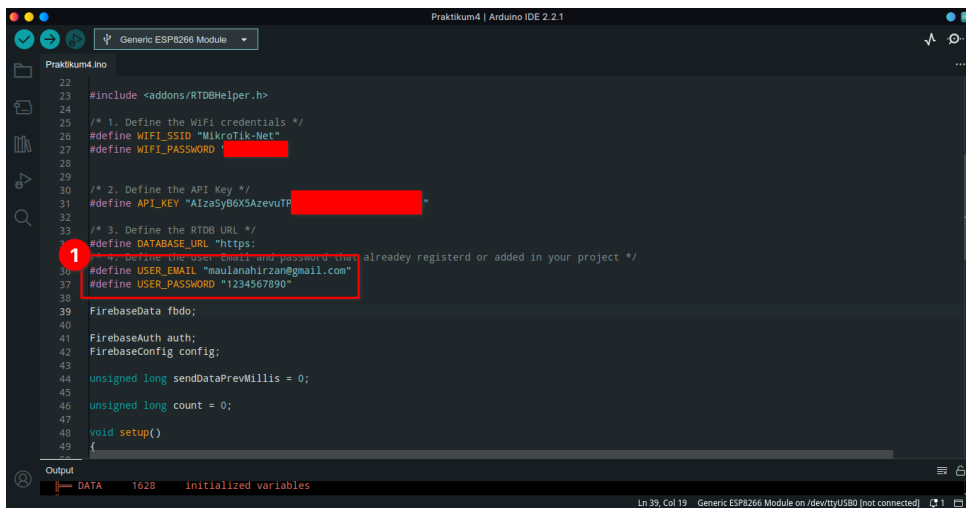
23. Pilih **Email/Password**, klik semua menjadi **Enable**, dan **Save**



24. Kembali ke tab **User**, klik **Add User**, isikan **Email** dan **Password**, klik **Add User**



25. Kembali lagi ke **Arduino IDE** dan ubah bagian **Email** dan **Password**



26. Berikutnya adalah menambahkan kode untuk sensor DHT

**Potongan Kode**

```
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

```
Praktikum4.ino
37 #define USER_EMAIL "user_email"
38
39 FirebaseData fbdo;
40
41 FirebaseAuth auth;
42 FirebaseConfig config;
43
44 unsigned long sendDataPrevMillis = 0;
45
46
47 #include <DHT.h>
48
49 #define DHTPIN 2
50 #define DHTTYPE DHT11
51 DHT dht(DHTPIN, DHTTYPE);
52
53 void setup()
54 {
55
56   Serial.begin(115200);
57
58   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
59   Serial.print("Connecting to Wi-Fi");
60   while (WiFi.status() != WL_CONNECTED)
61   {
62     Serial.print(".");
63     delay(300);
64   }
65   Serial.println();
66   Serial.print("Connected with IP: ");
```

27. Tambahkan di bagian akhir kode `void setup()` dengan kode berikut:

**Potongan Kode**

```
// Mulai Sensor DHT11
dht.begin();
```

```
Praktikum4.ino
71 #define USER_EMAIL = USER_EMAIL;
72 auth.user.password = USER_PASSWORD;
73
74 config.database_url = DATABASE_URL;
75
76
77 config.token_status_callback = tokenStatusCallback;
78 fbdo.setSSLBufferSize(4096 );
79
80
81 Firebase.begin(&config, &auth);
82 Firebase.reconnectNetwork(true);
83
84 // Mulai Sensor DHT11
85 dht.begin();
86
87
88 void loop()
89 {
90
91
92   if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
93   {
94     sendDataPrevMillis = millis();
95
96     FirebaseJson json;
97     json.setDoubleDigits(3);
98     json.add("value", count);
99
```

28. Di dalam kode `void loop()` setelah kode `if`, masukkan kode berikut

**Potongan Kode**

```
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
```

```

77 config.token_status_callback = TokenStatusCallback;
78 fbdo.setBSSLBufferSize(4096 );
79
80
81 Firebase.begin(&config, &auth);
82 Firebase.reconnectNetwork(true);
83
84 // Mulai Sensor DHT11
85 dht.begin();
86
87
88 void loop()
89 {
90   if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
91   {
92     float temperature = dht.readTemperature();
93     float humidity = dht.readHumidity();
94
95     sendDataPrevMillis = millis();
96
97     FirebaseJson json;
98     json.setDoubleDigits(3);
99     json.add("value", count);
100
101     Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo, "/test/json", &json) ? "ok" : fbdo.errorReason().c_str());
102   }
103 }
104

```

29. Setelah itu untuk menyusun query nya, masukkan kode berikut. GANTI baris yang ditandai sesuai dengan kode berikut

**Potongan Kode**

```

FirebaseJson json;
json.setDoubleDigits(3);
json.add("temperature", temperature);
json.add("humidity", humidity);

Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo,
"/livedata", &json) ? "ok" : fbdo.errorReason().c_str());
Serial.printf("Push json... %s\n", Firebase.RTDB.pushJSON(&fbdo,
"/history", &json) ? "ok" : fbdo.errorReason().c_str());

```

```

79
80
81 Firebase.begin(&config, &auth);
82 Firebase.reconnectNetwork(true);
83
84 // Mulai Sensor DHT11
85 dht.begin();
86
87
88 void loop()
89 {
90   if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
91   {
92     float temperature = dht.readTemperature();
93     float humidity = dht.readHumidity();
94
95     sendDataPrevMillis = millis();
96
97     FirebaseJson json;
98     json.setDoubleDigits(3);
99     json.add("temperature", temperature);
100     json.add("humidity", humidity);
101
102     Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo, "/livedata", &json) ? "ok" : fbdo.errorReason().c_str());
103     Serial.printf("Push json... %s\n", Firebase.RTDB.pushJSON(&fbdo, "/history", &json) ? "ok" : fbdo.errorReason().c_str());
104   }
105 }
106

```

30. Verifikasi dan Upload aplikasi

```

Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum4.ino
89 {
90   if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
91   {
92     float temperature = dht.readTemperature();
93     float humidity = dht.readHumidity();
94
95     sendDataPrevMillis = millis();
96
97     FirebaseJson json;
98     json.setDoubleDigits(3);
99     json.add("temperature", temperature);
100    json.add("humidity", humidity);
101
102    Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbd, "/livedata", &json) ? "ok" : fbd.errorReason().c_str());
103    Serial.printf("Push json... %s\n", Firebase.RTDB.pushJSON(&fbd, "/history", &json) ? "ok" : fbd.errorReason().c_str());
104  }
105 }
106
Output Serial Monitor
Writing at 0x00000000... (12 %)
Writing at 0x0000c000... (16 %)
Writing at 0x00010000... (20 %)
Writing at 0x00014000... (25 %)
Writing at 0x00018000... (29 %)
Writing at 0x0001c000... (33 %)
Writing at 0x00020000... (37 %)
Writing at 0x00024000... (41 %)
Writing at 0x00028000... (45 %)
Writing at 0x0002c000... (50 %)
Writing at 0x00030000... (54 %)
Uploading...
Ln 98, Col 33 Generic ESP8266 Module on /dev/ttyUSB0

```

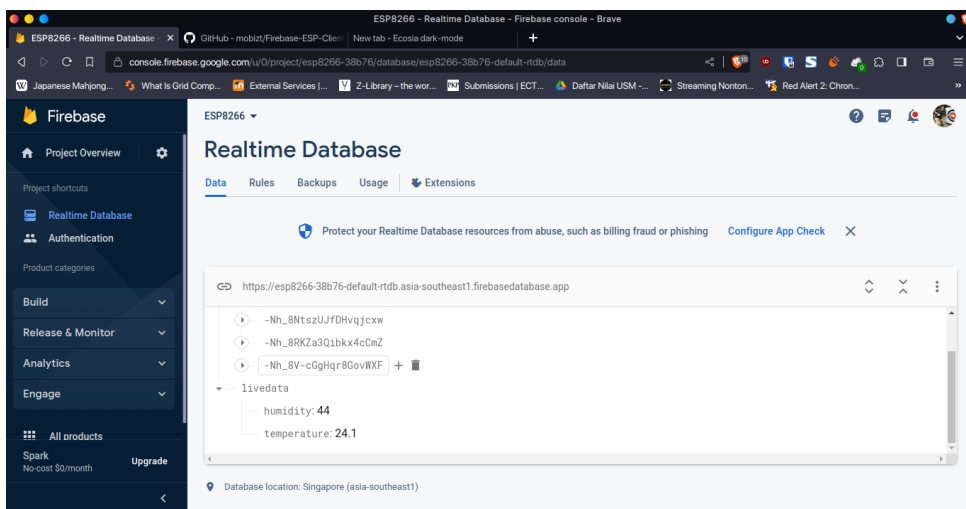
### 31. Data sukses diunggah

```

Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum4.ino
89 {
90   if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
91   {
92     float temperature = dht.readTemperature();
93     float humidity = dht.readHumidity();
94
95     sendDataPrevMillis = millis();
96
97     FirebaseJson json;
98     json.setDoubleDigits(3);
99     json.add("temperature", temperature);
100    json.add("humidity", humidity);
101
102    Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbd, "/livedata", &json) ? "ok" : fbd.errorReason().c_str());
103    Serial.printf("Push json... %s\n", Firebase.RTDB.pushJSON(&fbd, "/history", &json) ? "ok" : fbd.errorReason().c_str());
104  }
105 }
106
Output Serial Monitor x
Message: Click to send message to Device: Generic ESP8266 Module on /dev/ttyUSB0
No Line Ending | 115200 baud
11:44:03.266 -> <-----Kit token, status = ready
11:44:03.266 -> Set json... ok
11:44:03.266 -> Push json... ok
11:44:16.071 -> Set json... ok
11:44:16.167 -> Push json... ok
Ln 98, Col 33 Generic ESP8266 Module on /dev/ttyUSB0

```

### 32. Hasil



# Bab 5

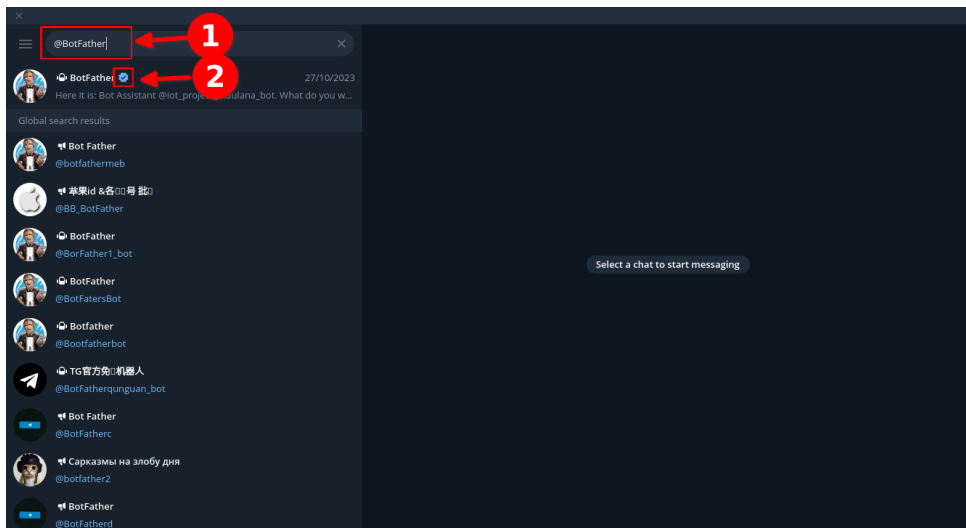
## Praktikum 5

### 5.1 ESP8266, DHT11, dan Telegram Bot

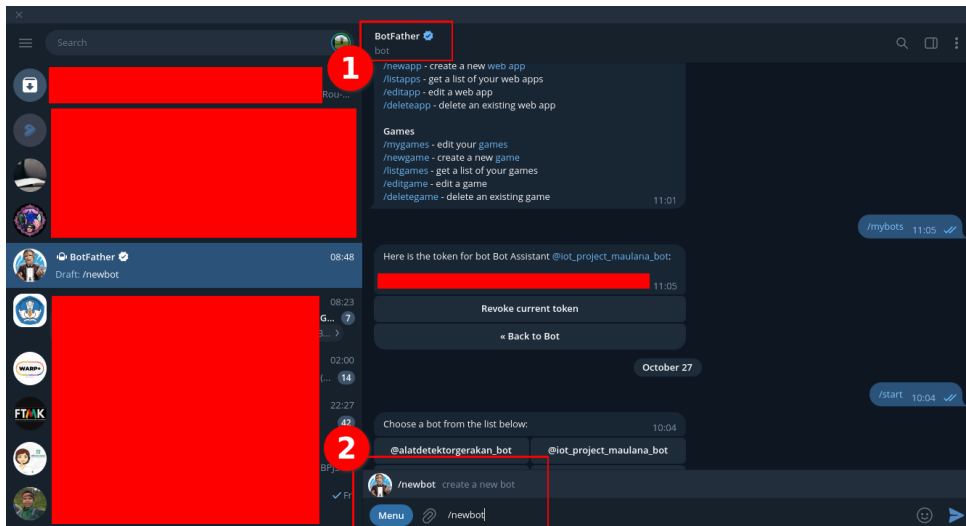
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Telegram Bot. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 4** yang ada di halaman sebelumnya.

### 5.2 Tutorial

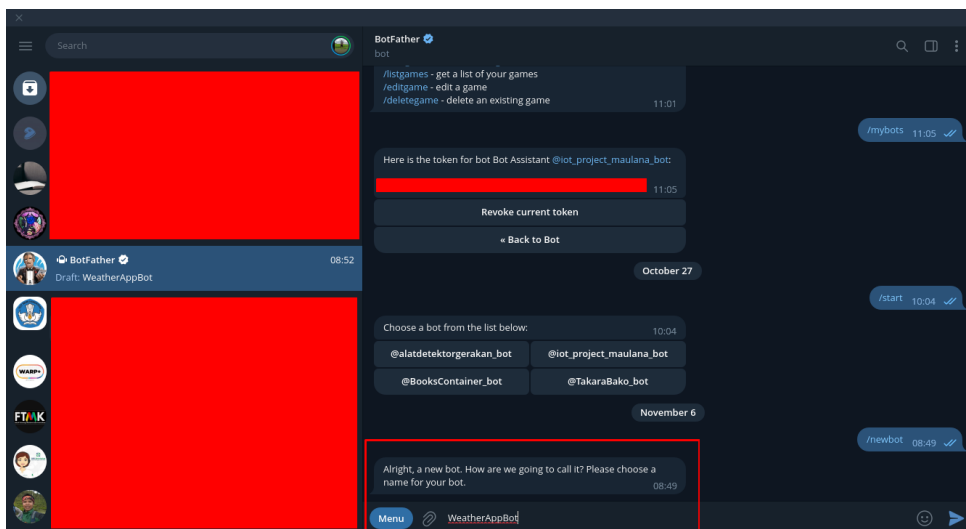
1. Tahap pertama yang dilakukan adalah membuat **Telegram Bot**. Pastikan untuk memiliki Akun Telegram untuk bisa memulai langkah ini
2. Cari **Bog Manager** dengan **@BotFather**



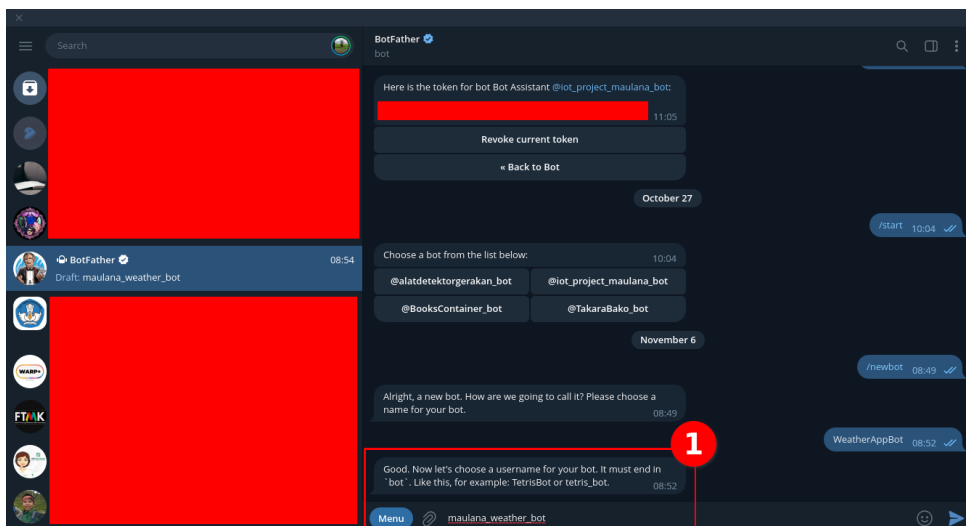
3. Gunakan perintah `/newbot` untuk membuat **Telegram Bot** baru



4. Lalu masukkan nama dari **Telegram Bot**

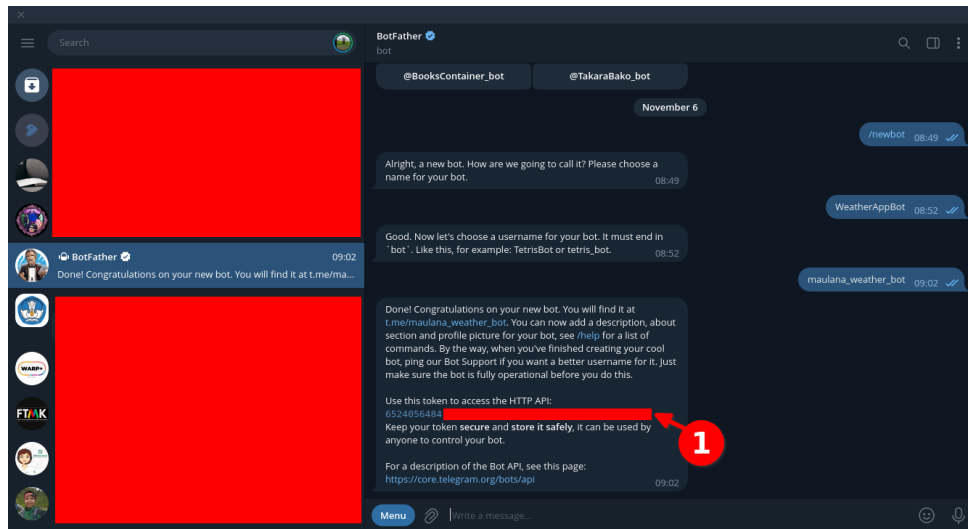


5. Lalu masukkan **username** untuk mempermudah pencarian **Telegram Bot**. Pastikan memiliki akhiran **\_bot**

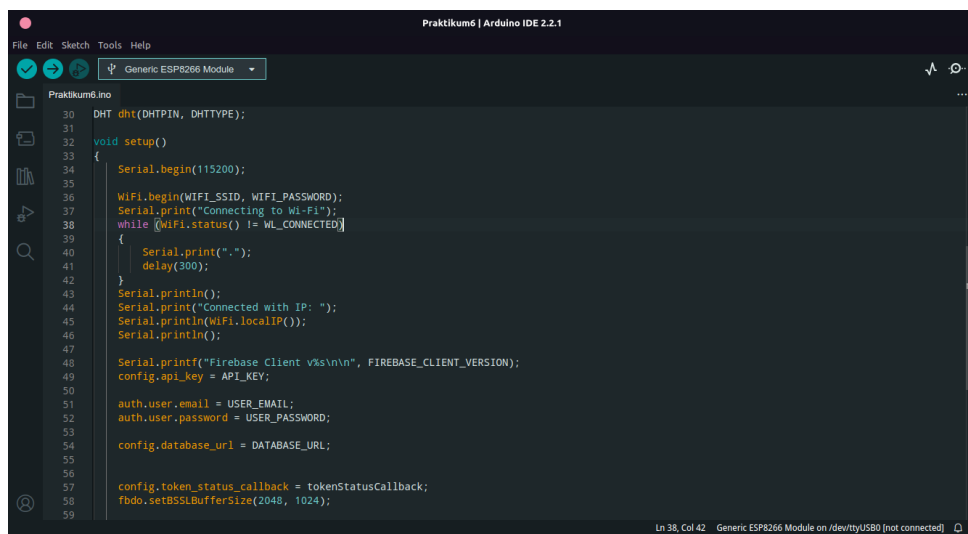




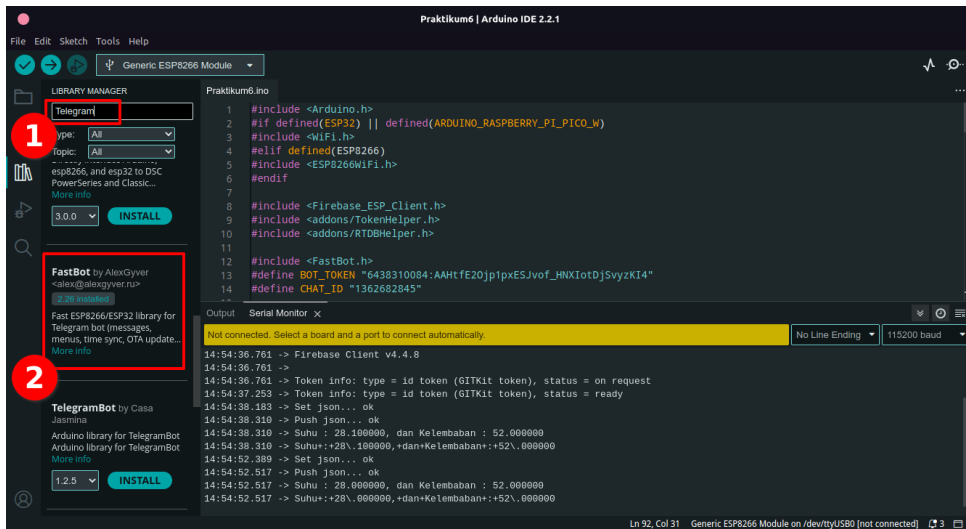
6. **Telegram Bot** sudah jadi dan **Token API** akan ditampilkan. Simpan baik-baik kode tersebut



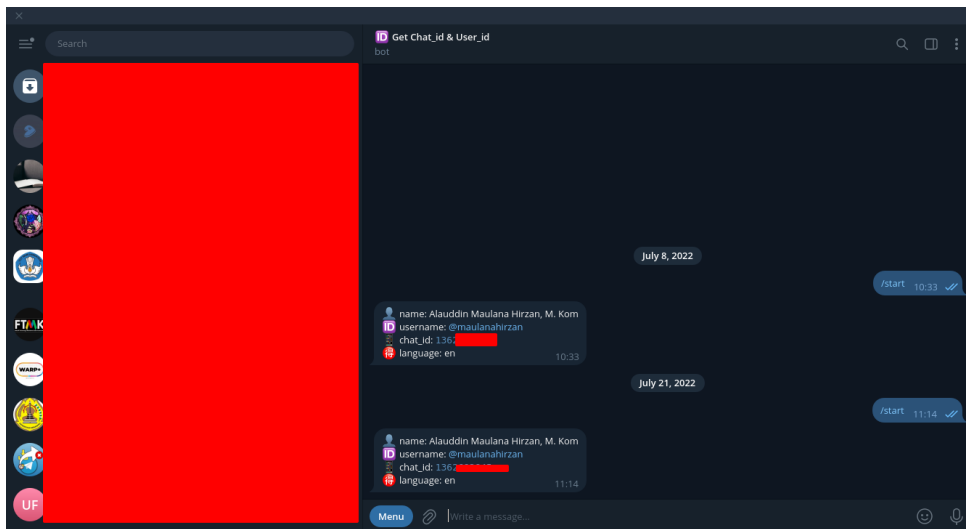
7. Berikutnya adalah membuka kembali **Praktikum 4** dengan menggunakan **Arduino IDE**. Lakukan **Save As** untuk menyimpan sebagai **Praktikum 6**



8. Install Library dengan nama **FastBot**



9. Berikutnya adalah mendapatkan **Chat ID** melalui Bot <https://t.me/chatIDrobot>



10. Setelah itu tambahkan kode berikut tepat setelah **RTDBHelper.h**. Lalu masukkan **TOKEN BOT** dan **Chat ID** di kode berikut

Sesudah

```
#include <FastBot.h>
#define BOT_TOKEN "<TOKEN BOT>"
#define CHAT_ID "<Chat ID>"
```

```

1 #include <Arduino.h>
2 #if defined(ESP32) || defined(ARDUINO_RASPBERRY_PI_PICO_W)
3 #include <WiFi.h>
4 #elif defined(ESP8266)
5 #include <ESP8266WiFi.h>
6 #endif
7
8 #include <Firebase_ESP_Client.h>
9 #include <addons/TokenHelper.h>
10 #include <addons/RTDHelper.h>
11
12 #include <FastBot.h>
13 #define BOT_TOKEN "6438310084 [REDACTED]"
14 #define CHAT_ID "1362 [REDACTED]"

```

Serial Monitor Output:

```

Writing at 0x00048000... (74 %)
Writing at 0x00048000... (76 %)
Writing at 0x0004c000... (80 %)
Writing at 0x00050000... (84 %)
Writing at 0x00054000... (88 %)
Writing at 0x00058000... (92 %)
Writing at 0x0005c000... (96 %)
Writing at 0x00060000... (100 %)
Wrote 354896 bytes (401648 compressed) at 0x00000000 in 35.3 seconds (effective 125.7 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...

```

11. Setelah itu masukkan kode untuk inialisasi Bot dengan menambahkan kode berikut di atas `void setup()`

Sesudah

```
FastBot bot(BOT_TOKEN);
```

```

27 unsigned long sensorPrevev11115 = 0;
28
29 #include <DHT.h>
30 #define DHTPIN 2
31 #define DHTTYPE DHT11
32 DHT dht(DHTPIN, DHTTYPE);
33
34 FastBot bot(BOT_TOKEN);
35
36 void setup()
37 {
38   Serial.begin(115200);
39
40   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
41   Serial.print("Connecting to Wi-Fi");
42   while (WiFi.status() != WL_CONNECTED)
43   {
44     Serial.print(".");
45     delay(300);
46   }
47   Serial.println();
48   Serial.print("Connected with ID: ");

```

Serial Monitor Output:

```

Writing at 0x00048000... (74 %)
Writing at 0x00054000... (88 %)
Writing at 0x00058000... (92 %)
Writing at 0x0005c000... (96 %)
Writing at 0x00060000... (100 %)
Wrote 354896 bytes (401648 compressed) at 0x00000000 in 35.3 seconds (effective 125.7 kbit/s)...
Hash of data verified.

```

12. Setelah itu, cukup tambahkan kode berikut tepat di akhir fungsi `void loop()`

Sesudah

```

bot.setChatID(CHAT_ID);
bot.setTextMode(FB_MARKDOWN);

char buffer[40];
sprintf(buffer, "Suhu : %f, dan Kelembaban : %f", temperature, humidity);
Serial.println(buffer);

bot.sendMessage(buffer);

```

```
Praktikum6 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Generic ESP8266 Module
Praktikum6.ino
75 sendDataPrevMillis = millis();
76
77 FirebaseJson json;
78 json.setDoubleDigits(3);
79 json.add("temperature", temperature);
80 json.add("humidity", humidity);
81
82 Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo, "/livedata", &json) ? "ok" : fbdo.errorReason().c_str());
83 Serial.printf("Push json... %s\n", Firebase.RTDB.pushJSON(&fbdo, "/history", &json) ? "ok" : fbdo.errorReason().c_str());
84
85 bot.setChatID(CHAT_ID);
86 bot.setTextMode(FB_MARKDOWN);
87
88 char buffer[40];
89 sprintf(buffer, "Suhu: %f, dan Kelembaban: %f", temperature, humidity);
90 Serial.println(buffer);
91
92 bot.sendMessage(buffer);
93
94 }
95
Output Serial Monitor
Writing at 0x00054000... (88 %)
Writing at 0x00058000... (92 %)
Writing at 0x0005c000... (96 %)
Writing at 0x00060000... (100 %)
Wrote 354896 bytes (401648 compressed) at 0x00000000 in 35.3 seconds (effective 125.7 kbit/s)...
Hash of data verified.
Ln 92, Col 33 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]
```

### 13. Verifikasi dan Upload kode ke Perangkat

```
Praktikum6 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Generic ESP8266 Module
Praktikum6.ino
75 sendDataPrevMillis = millis();
76
77 FirebaseJson json;
78 json.setDoubleDigits(3);
79 json.add("temperature", temperature);
80 json.add("humidity", humidity);
81
82 Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo, "/livedata", &json) ? "ok" : fbdo.errorReason().c_str());
83 Serial.printf("Push json... %s\n", Firebase.RTDB.pushJSON(&fbdo, "/history", &json) ? "ok" : fbdo.errorReason().c_str());
84
85 bot.setChatID(CHAT_ID);
86 bot.setTextMode(FB_MARKDOWN);
87
88 char buffer[40];
89 sprintf(buffer, "Suhu: %f, dan Kelembaban: %f", temperature, humidity);
90 Serial.println(buffer);
91
92 bot.sendMessage(buffer);
93
94 }
95
Output Serial Monitor x
14:54:38.761 -> Token Info: type = id token (GITKit token), status = on request
14:54:37.253 -> Token Info: type = id token (GITKit token), status = ready
14:54:38.183 -> Set json... ok
14:54:38.310 -> Push json... ok
14:54:38.310 -> Suhu: 28.100000, dan Kelembaban: 52.000000
14:54:38.310 -> Suhu: +28\,100000, dan+Kelembaban: +52\,000000
14:54:52.389 -> Set json... ok
14:54:52.517 -> Push json... ok
14:54:52.517 -> Suhu: 28.000000, dan Kelembaban: 52.000000
14:54:52.517 -> Suhu: +28\,000000, dan+Kelembaban: +52\,000000
Ln 92, Col 33 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]
```

# Bab 6

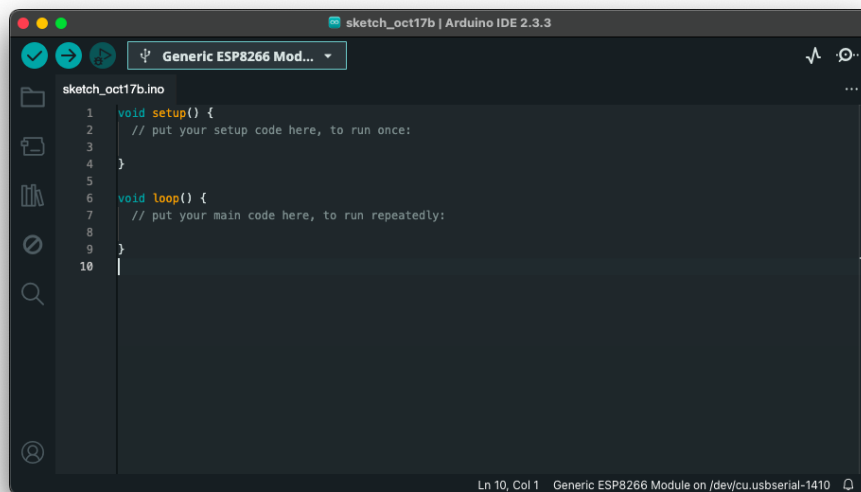
## Praktikum 6

### 6.1 ESP8266, DHT11, dan Linear Regression

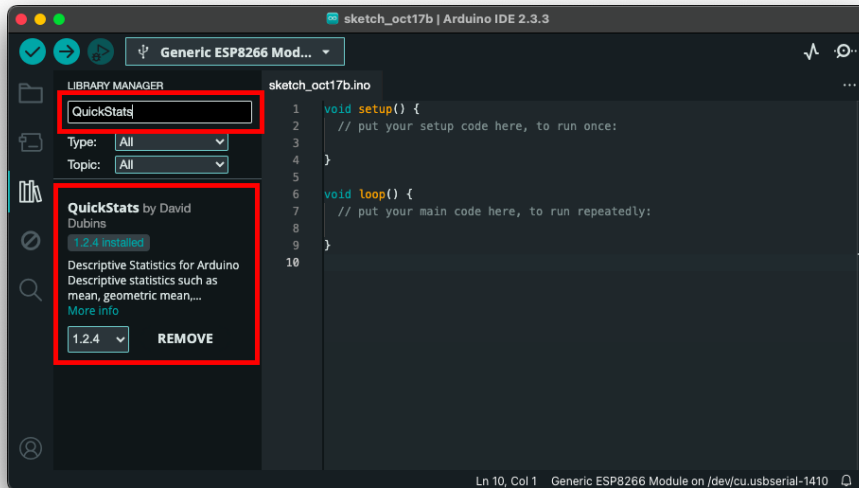
Di bagian ini mahasiswa diajarkan bagaimana mengimplementasikan algoritma *Linear Regression* sederhana dengan perangkat **ESP8266, dan DHT11**. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 5** yang ada di halaman sebelumnya.

### 6.2 Tutorial

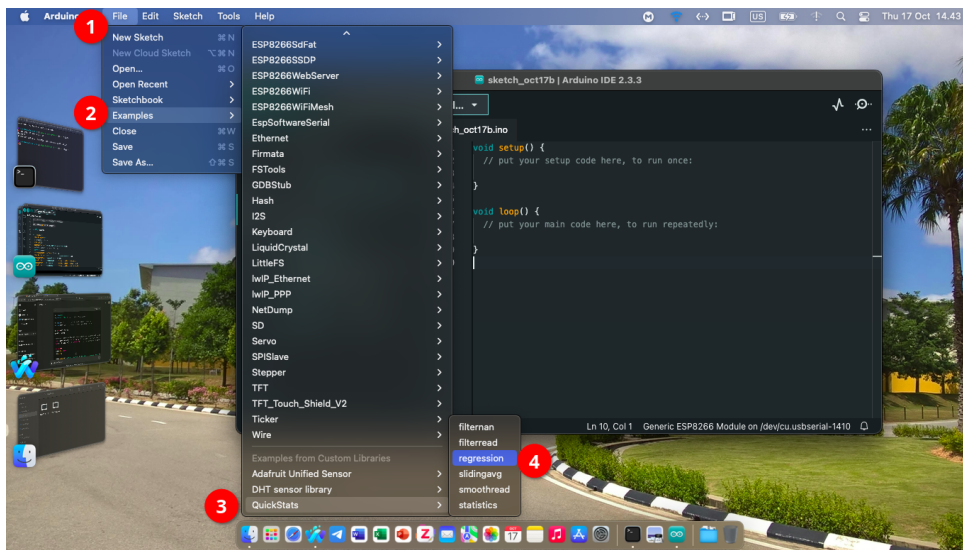
1. Buka kembali Arduino IDE untuk memulai projek baru. ESP8266 tidak harus tercolok.



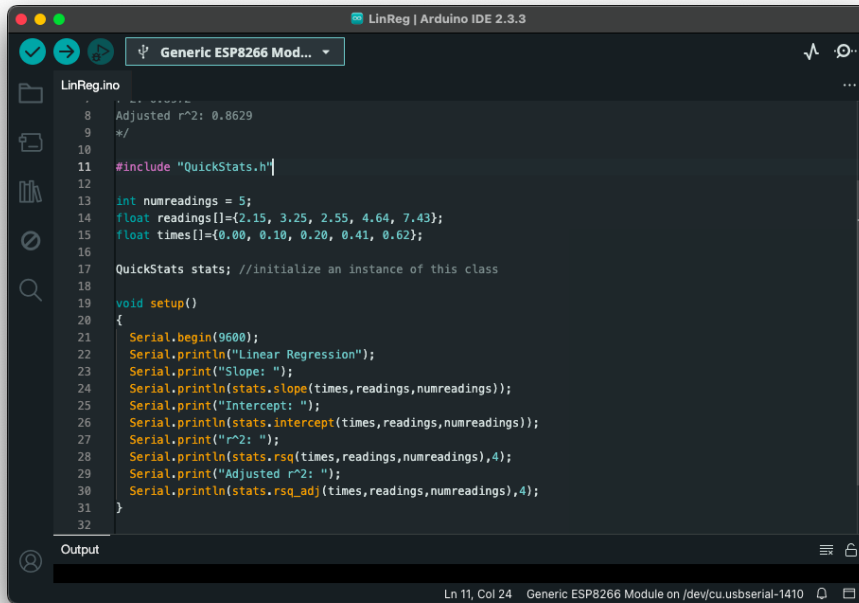
2. Install Library **QuickStats**



3. Buka contoh kode dari **Linear Regression** melalui menu **File** → **Examples** → **QuickStats** → **regresison**

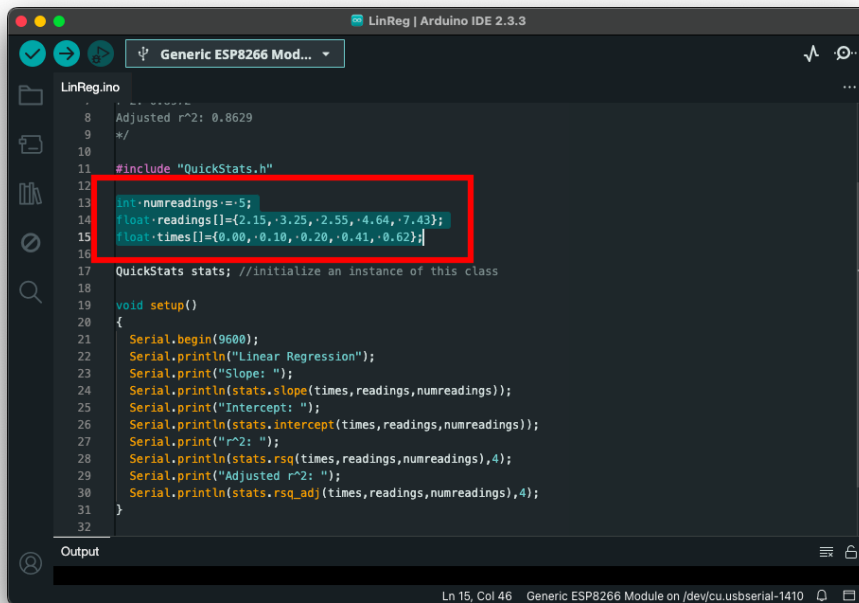


4. Tunggu window contoh kode **QuickStats** muncul

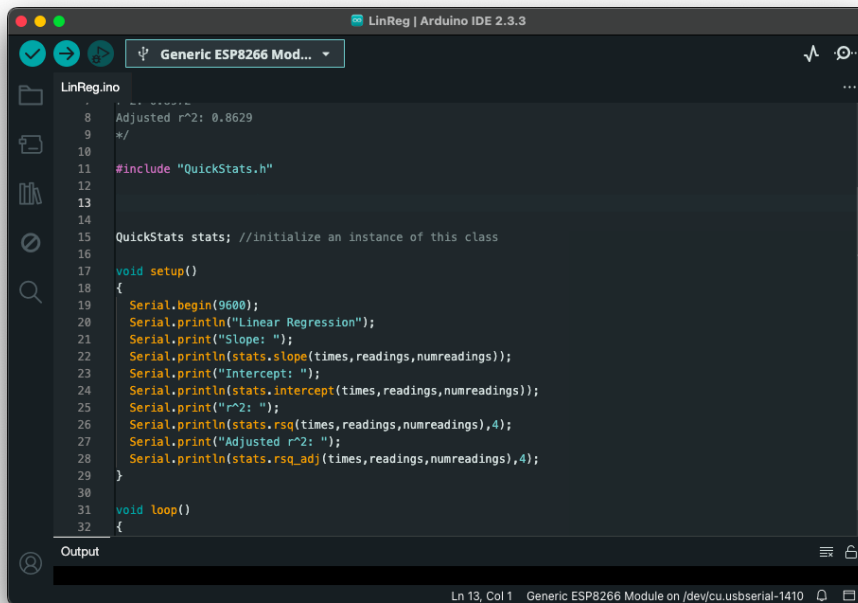


```
LinReg.ino
8 Adjusted r^2: 0.8629
9 */
10
11 #include "QuickStats.h"
12
13 int numreadings = 5;
14 float readings[]={2.15, 3.25, 2.55, 4.64, 7.43};
15 float times[]={0.00, 0.10, 0.20, 0.41, 0.62};
16
17 QuickStats stats; //initialize an instance of this class
18
19 void setup()
20 {
21   Serial.begin(9600);
22   Serial.println("Linear Regression");
23   Serial.print("Slope: ");
24   Serial.println(stats.slope(times, readings, numreadings));
25   Serial.print("Intercept: ");
26   Serial.println(stats.intercept(times, readings, numreadings));
27   Serial.print("r^2: ");
28   Serial.println(stats.rsq(times, readings, numreadings), 4);
29   Serial.print("Adjusted r^2: ");
30   Serial.println(stats.rsq_adj(times, readings, numreadings), 4);
31 }
32
```

5. Hapus kode bagian yang ditandai



```
LinReg.ino
8 Adjusted r^2: 0.8629
9 */
10
11 #include "QuickStats.h"
12
13 int numreadings = 5;
14 float readings[]={2.15, 3.25, 2.55, 4.64, 7.43};
15 float times[]={0.00, 0.10, 0.20, 0.41, 0.62};
16
17 QuickStats stats; //initialize an instance of this class
18
19 void setup()
20 {
21   Serial.begin(9600);
22   Serial.println("Linear Regression");
23   Serial.print("Slope: ");
24   Serial.println(stats.slope(times, readings, numreadings));
25   Serial.print("Intercept: ");
26   Serial.println(stats.intercept(times, readings, numreadings));
27   Serial.print("r^2: ");
28   Serial.println(stats.rsq(times, readings, numreadings), 4);
29   Serial.print("Adjusted r^2: ");
30   Serial.println(stats.rsq_adj(times, readings, numreadings), 4);
31 }
32
```



6. Di bagian bawah dari `#include "QuickStats.h"`, tambahkan kode berikut untuk sensor DHT11:

#### Kode Program

```
// Add DHT Libs
#include <DHT.h>

// Configure Sensor
#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
```



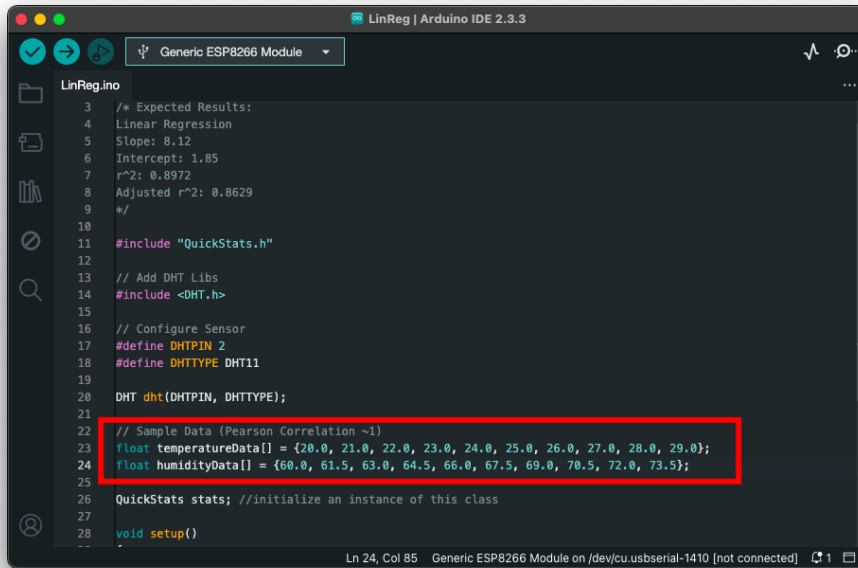
```
LinReg.ino
1 // Example program for use with QuickStats.h
2 // Simple linear regression (y=ax+b)
3 /* Expected Results:
4 Linear Regression
5 Slope: 8.12
6 Intercept: 1.85
7 r^2: 0.8972
8 Adjusted r^2: 0.8629
9 */
10
11 #include "QuickStats.h"
12
13 // Add DHT Libs
14 #include <Adafruit_Sensor.h>
15 #include <DHT.h>
16
17 // Configure Sensor
18 #define DHTPIN 2
19 #define DHTTYPE DHT11
20
21 DHT dht(DHTPIN, DHTTYPE);
22
23
24
25
26
27 QuickStats stats; //initialize an instance of this class
28
29
```

7. DI bagian bawah dari kode `DHT dht(DHTPIN, DHTTYPE);` dan atas kode `QuickStats stats;`, tambahkan sampel data sebanyak 10 unit. Data sampel ini bisa didapatkan dari monitoring mandiri (baik dari Adafruit IO, Thingspeak atau Firebase Realtime), atau gunakan sampel data dari kode berikut:

**Kode Program**

```
// Sample Data (Pearson Correlation ~1)
float temperatureData[] = { 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0};

float humidityData[] = { 60.0, 61.5, 63.0, 64.5, 66.0, 67.5, 69.0,
70.5, 72.0, 73.5};
```

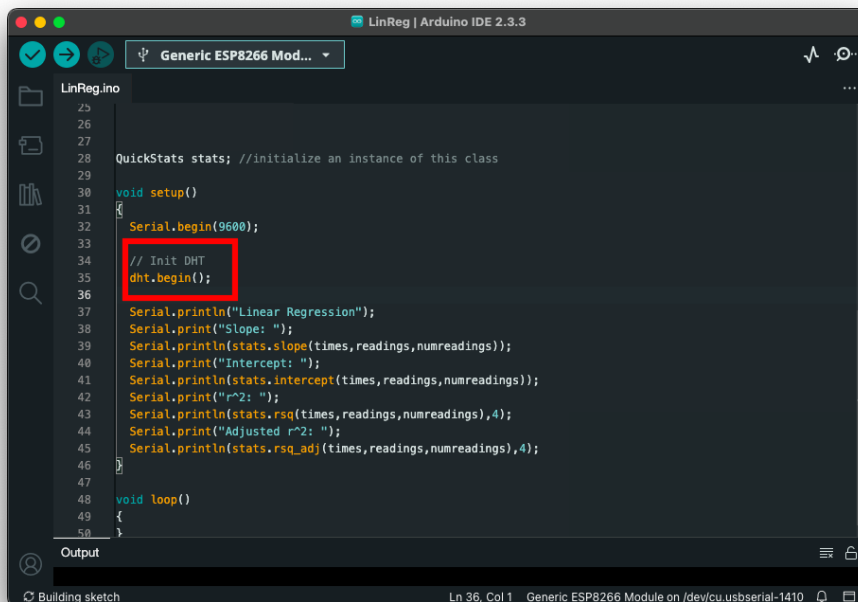


```
LinReg.ino
3  /* Expected Results:
4  Linear Regression
5  Slope: 8.12
6  Intercept: 1.85
7  r^2: 0.8972
8  Adjusted r^2: 0.8629
9  */
10
11 #include "QuickStats.h"
12
13 // Add DHT Libs
14 #include <DHT.h>
15
16 // Configure Sensor
17 #define DHTPIN 2
18 #define DHTTYPE DHT11
19
20 DHT dht(DHTPIN, DHTTYPE);
21
22 // Sample Data (Pearson Correlation ~1)
23 float temperatureData[] = {20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0};
24 float humidityData[] = {60.0, 61.5, 63.0, 64.5, 66.0, 67.5, 69.0, 70.5, 72.0, 73.5};
25
26 QuickStats stats; //initialize an instance of this class
27
28 void setup()
29
```

8. Setelah mendapatkan data *training* untuk **Regresi Linier**, maka langkah berikutnya adalah menambahkan kode inisialisasi untuk sensor **DHT 11** dengan kode berikut setelah kode **Serial.begin(9600);**:

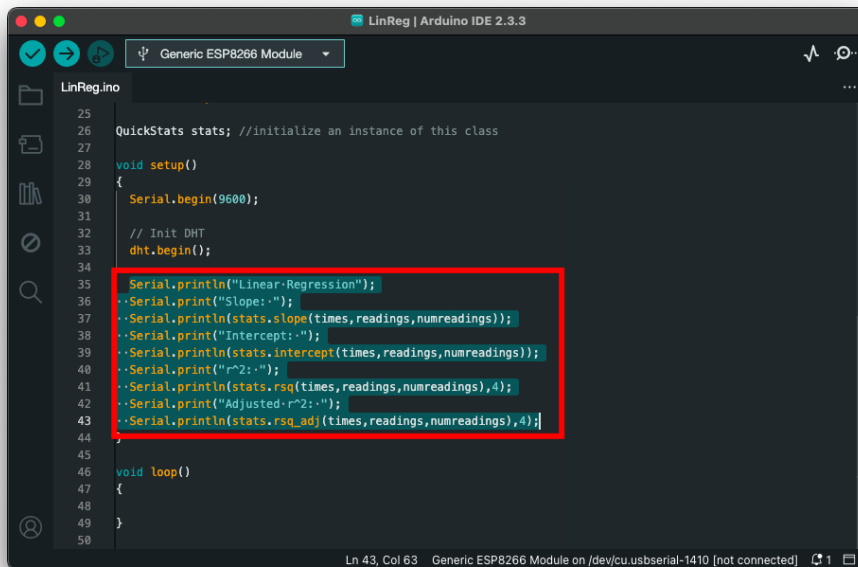
Kode Program

```
// Init DHT
dht.begin();
```

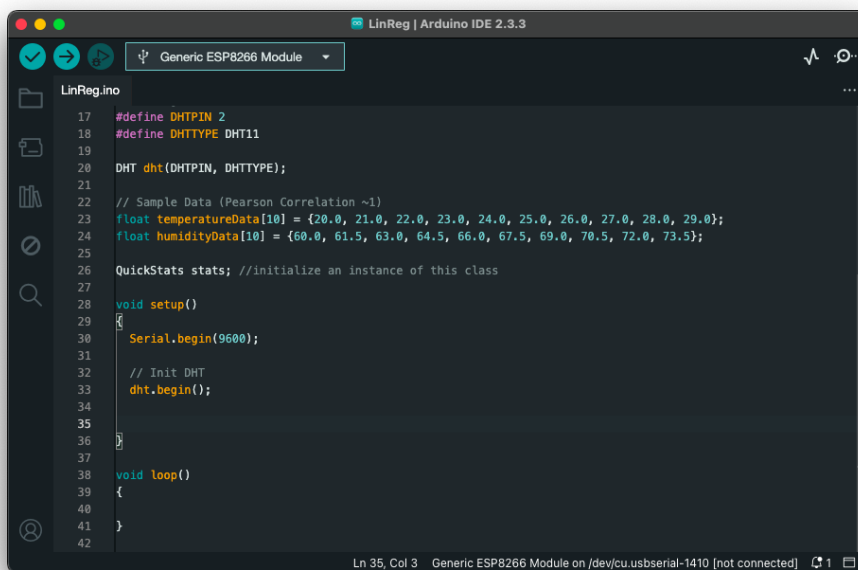


```
LinReg.ino
25
26
27
28 QuickStats stats; //initialize an instance of this class
29
30 void setup()
31 {
32   Serial.begin(9600);
33
34   // Init DHT
35   dht.begin();
36
37   Serial.println("Linear Regression");
38   Serial.print("Slope: ");
39   Serial.println(stats.slope(times, readings, numreadings));
40   Serial.print("Intercept: ");
41   Serial.println(stats.intercept(times, readings, numreadings));
42   Serial.print("r^2: ");
43   Serial.println(stats.rsq(times, readings, numreadings),4);
44   Serial.print("Adjusted r^2: ");
45   Serial.println(stats.rsq_adj(times, readings, numreadings),4);
46
47
48 void loop()
49 {
50
```

9. Hapus kode sisa dari fungsi **setup()** karena tidak terpakai



```
25 QuickStats stats; //initialize an instance of this class
26
27
28 void setup()
29 {
30   Serial.begin(9600);
31
32   // Init DHT
33   dht.begin();
34
35   Serial.println("Linear Regression");
36   Serial.print("Slope:");
37   Serial.println(stats.slope(times, readings, numreadings));
38   Serial.print("Intercept:");
39   Serial.println(stats.intercept(times, readings, numreadings));
40   Serial.print("r^2:");
41   Serial.println(stats.rsq(times, readings, numreadings), 4);
42   Serial.print("Adjusted r^2:");
43   Serial.println(stats.rsq_adj(times, readings, numreadings), 4);
44
45
46 void loop()
47 {
48
49
50
```



```
17 #define DHTPIN 2
18 #define DHTTYPE DHT11
19
20 DHT dht(DHTPIN, DHTTYPE);
21
22 // Sample Data (Pearson Correlation ~1)
23 float temperatureData[10] = {20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0};
24 float humidityData[10] = {60.0, 61.5, 63.0, 64.5, 66.0, 67.5, 69.0, 70.5, 72.0, 73.5};
25
26 QuickStats stats; //initialize an instance of this class
27
28 void setup()
29 {
30   Serial.begin(9600);
31
32   // Init DHT
33   dht.begin();
34
35
36
37
38 void loop()
39 {
40
41
42
```

10. Kembali ke bagian atas, tambahkan kode setelah kode **QuickStats stats;**

**Kode Program**

```
float slope = 0.0;
float intercept = 0.0;
```

```

11
12
13 // Add DHT Libs
14 #include <DHT.h>
15
16 // Configure Sensor
17 #define DHTPIN 2
18 #define DHTTYPE DHT11
19
20 DHT dht(DHTPIN, DHTTYPE);
21
22 // Sample Data (Pearson Correlation ~1)
23 float temperatureData[] = {20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0};
24 float humidityData[] = {60.0, 61.5, 63.0, 64.5, 66.0, 67.5, 69.0, 70.5, 72.0, 73.5};
25
26 QuickStats stats; //initialize an instance of this class
27
28 float slope = 0.0;
29 float intercept = 0.0;
30
31 void setup()
32 {
33   Serial.begin(9600);
34
35   // Init DHT
36   dht.begin();
37

```

11. Kembali lagi ke fungsi `setup()`, dan tambahkan kode berikut setelah kode `dht.begin()`;  
Rumus ini digunakan untuk mencari garis miring dari data dan nilai intersepsi nya

Kode Program

```

// Calculate Slope of Temperature using Humidity Data
slope = stats.slope(humidityData, temperatureData, 10);
intercept = stats.intercept(humidityData, temperatureData, 10);

```

```

22 // Sample Data (Pearson Correlation ~1)
23 float temperatureData[] = {20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0};
24 float humidityData[] = {60.0, 61.5, 63.0, 64.5, 66.0, 67.5, 69.0, 70.5, 72.0, 73.5};
25
26 QuickStats stats; //initialize an instance of this class
27
28 float slope = 0.0;
29 float intercept = 0.0;
30
31 void setup()
32 {
33   Serial.begin(9600);
34
35   // Init DHT
36   dht.begin();
37
38   // Calculate Slope of Temperature using Humidity Data
39   slope = stats.slope(humidityData, temperatureData, 10);
40   intercept = stats.intercept(humidityData, temperatureData, 10);
41 }
42
43 void loop()
44 {
45
46
47

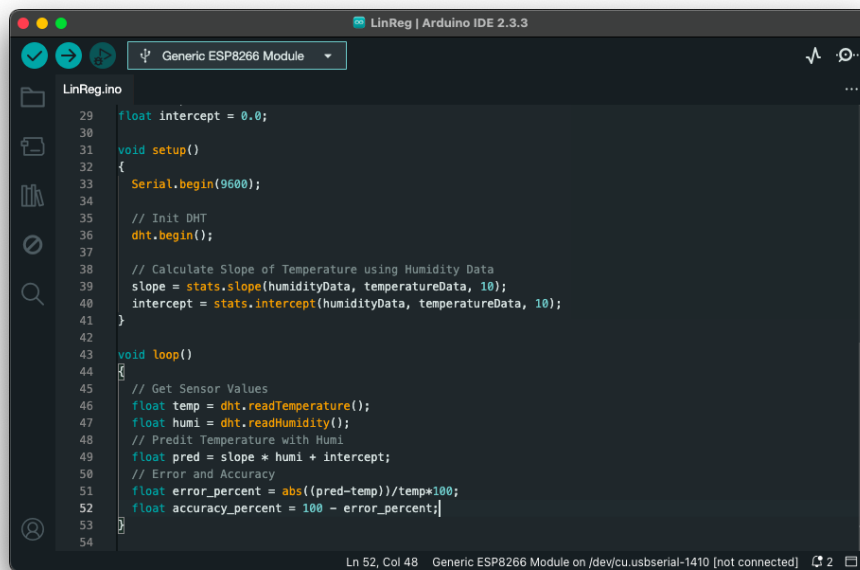
```

12. Bagian terakhir untuk kode `loop()` ada dua kode yang ditambahkan:

- (a) Kode untuk melakukan penarikan data terbaru beserta penghitungan prediksi, persentase akurasi dan galat nya

### Kode Program

```
// Get Sensor Values
float temp = dht.readTemperature();
float humi = dht.readHumidity();
// Predit Temperature with Humi
float pred = slope * humi + intercept;
// Error and Accuracy
float error_percent = abs((pred-temp))/temp*100;
float accuracy_percent = 100 - error_percent;
```

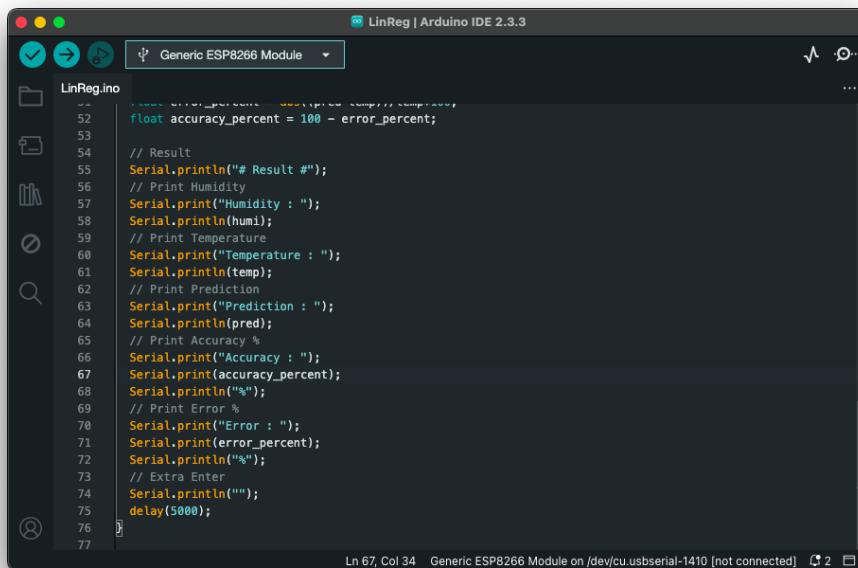


```
LinReg.ino
29 float intercept = 0.0;
30
31 void setup()
32 {
33   Serial.begin(9600);
34
35   // Init DHT
36   dht.begin();
37
38   // Calculate Slope of Temperature using Humidity Data
39   slope = stats.slope(humidityData, temperatureData, 10);
40   intercept = stats.intercept(humidityData, temperatureData, 10);
41 }
42
43 void loop()
44 {
45   // Get Sensor Values
46   float temp = dht.readTemperature();
47   float humi = dht.readHumidity();
48   // Predit Temperature with Humi
49   float pred = slope * humi + intercept;
50   // Error and Accuracy
51   float error_percent = abs((pred-temp))/temp*100;
52   float accuracy_percent = 100 - error_percent;
53 }
54
```

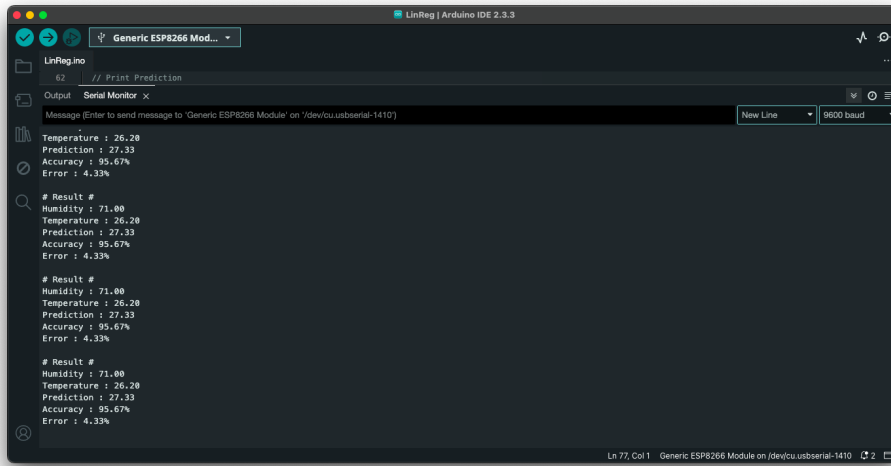
(b) Kode untuk menampilkan hasilnya, beserta delay 5 detik

### Kode Program

```
// Result
Serial.println("# Result #");
// Print Humidity
Serial.print("Humidity : ");
Serial.println(humi);
// Print Temperature
Serial.print("Temperature : ");
Serial.println(temp);
// Print Prediction
Serial.print("Prediction : ");
Serial.println(pred);
// Print Accuracy %
Serial.print("Accuracy : ");
Serial.print(accuracy_percent);
Serial.println("%");
// Print Error %
Serial.print("Error : ");
Serial.print(error_percent);
Serial.println("%");
// Extra Enter
Serial.println("");
delay(5000);
```



13. Tancapkan alat, **Upload Kode**, dan lihat hasil akhir melalui **Serial Monitor**



# Bab 7

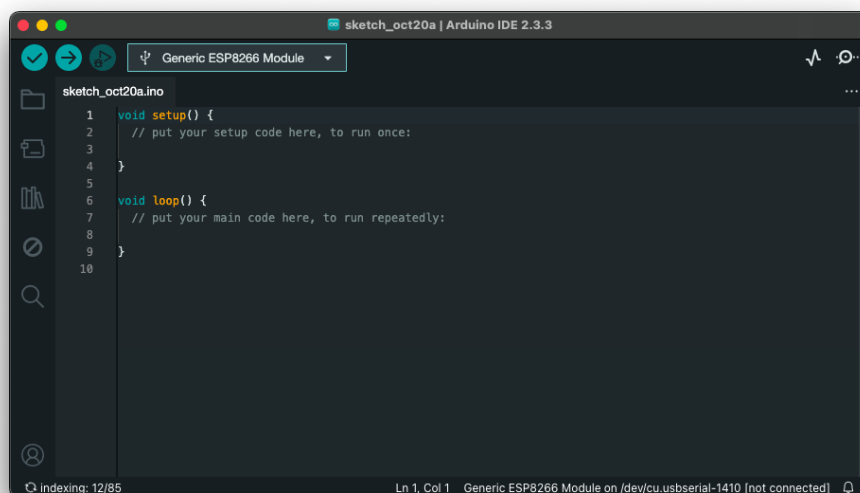
## Praktikum 7

### 7.1 ESP8266, DHT11, dan Fuzzy Logic

Di bagian ini mahasiswa diajarkan bagaimana mengimplementasikan algoritma *Fuzzy Logic* sederhana dengan perangkat **ESP8266**, dan **DHT11**. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 6** yang ada di halaman sebelumnya.

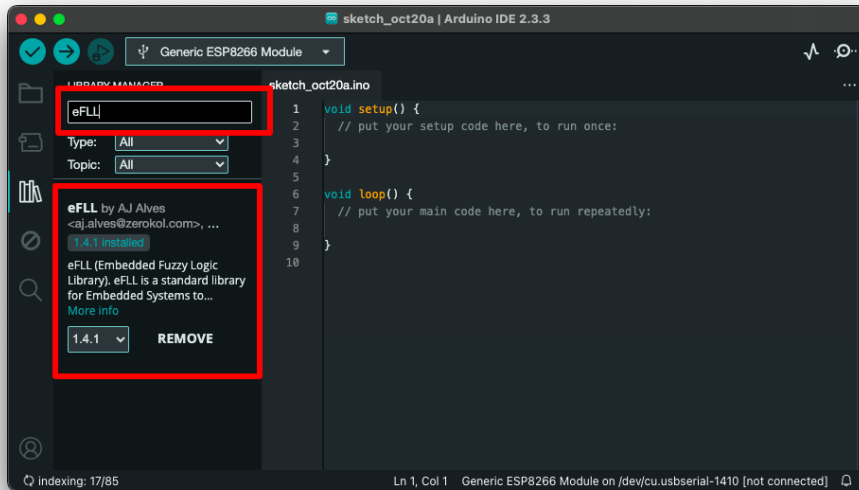
### 7.2 Tutorial

1. Buka Arduino IDE kembali

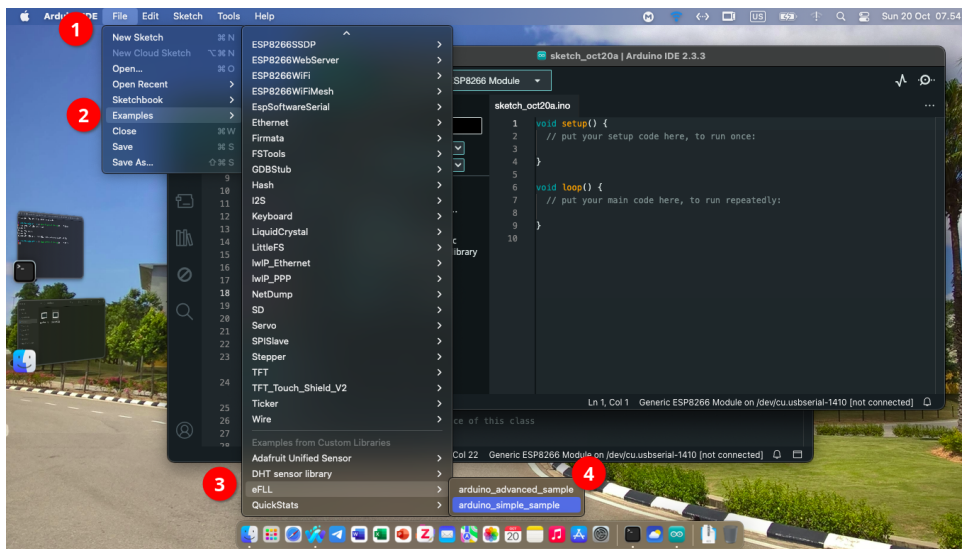


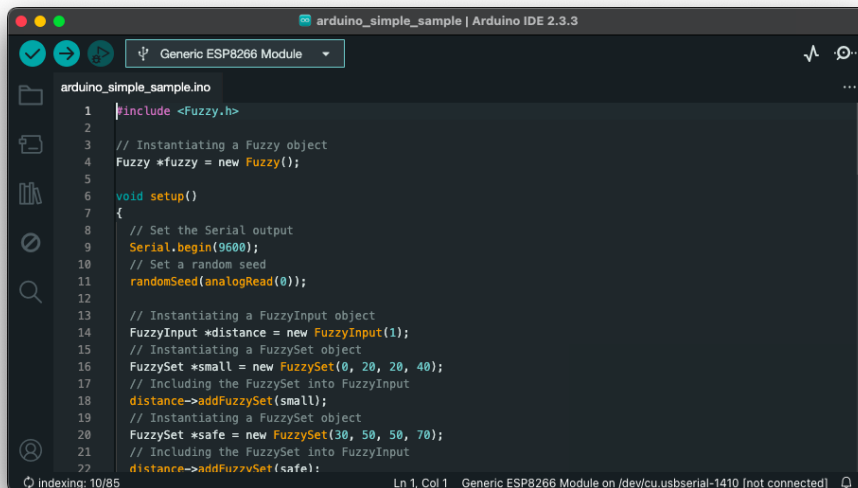
2. Install Library **eFLL**





3. Buka kode example dari eFLL dari menu **File** → **Examples** → **eFLL** → **arduino\_simple\_sample**.





```
arduino_simple_sample | Arduino IDE 2.3.3
Generic ESP8266 Module
arduino_simple_sample.ino
1 #include <Fuzzy.h>
2
3 // Instantiating a Fuzzy object
4 Fuzzy *fuzzy = new Fuzzy();
5
6 void setup()
7 {
8   // Set the Serial output
9   Serial.begin(9600);
10  // Set a random seed
11  randomSeed(analogRead(0));
12
13  // Instantiating a FuzzyInput object
14  FuzzyInput *distance = new FuzzyInput(1);
15  // Instantiating a FuzzySet object
16  FuzzySet *small = new FuzzySet(0, 20, 20, 40);
17  // Including the FuzzySet into FuzzyInput
18  distance->addFuzzySet(small);
19  // Instantiating a FuzzySet object
20  FuzzySet *safe = new FuzzySet(30, 50, 50, 70);
21  // Including the FuzzySet into FuzzyInput
22  distance->addFuzzySet(safe);
indexing: 10/85 Ln 1, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]
```

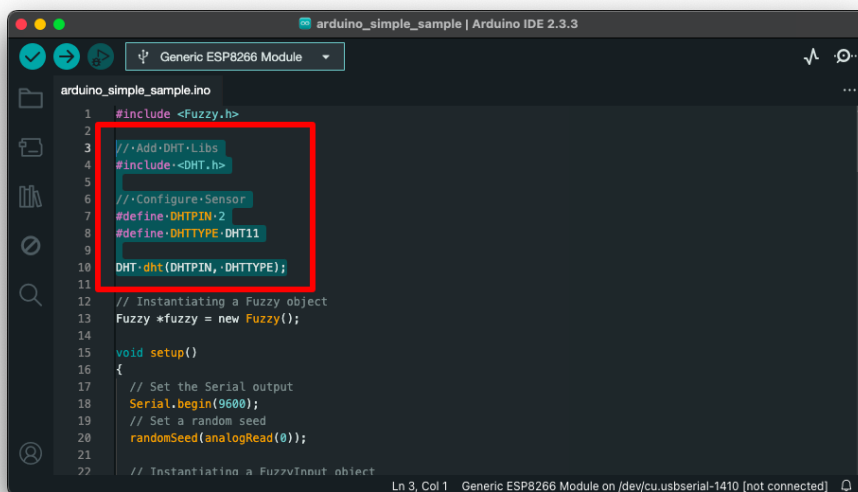
4. Di bagian bawah kode `#include <Fuzzy.h>`, tambahkan kode library DHT11

**Kode Program**

```
// Add DHT Libs
#include <DHT.h>

// Configure Sensor
#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
```



```
arduino_simple_sample | Arduino IDE 2.3.3
Generic ESP8266 Module
arduino_simple_sample.ino
1 #include <Fuzzy.h>
2
3 // Add DHT Libs
4 #include <DHT.h>
5
6 // Configure Sensor
7 #define DHTPIN 2
8 #define DHTTYPE DHT11
9
10 DHT dht(DHTPIN, DHTTYPE);
11
12 // Instantiating a Fuzzy object
13 Fuzzy *fuzzy = new Fuzzy();
14
15 void setup()
16 {
17   // Set the Serial output
18   Serial.begin(9600);
19   // Set a random seed
20   randomSeed(analogRead(0));
21
22   // Instantiating a FuzzyInput object
indexing: 10/85 Ln 3, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]
```

5. Pindah ke fungsi `setup()`. Di bagian bawah dari kode `Serial.begin(9600);`. Hapus dua baris kode.

```

14 void setup()
15 {
16   // Set the Serial output
17   Serial.begin(9600);
18   //Set a random seed
19   randomSeed(analogRead(0));
20
21
22   // Instantiating a FuzzyInput object
23   FuzzyInput *distance = new FuzzyInput(1);
24   // Instantiating a FuzzySet object
25   FuzzySet *small = new FuzzySet(0, 20, 20, 40);
26   // Including the FuzzySet into FuzzyInput
27   distance->addFuzzySet(small);
28   // Instantiating a FuzzySet object
29   FuzzySet *safe = new FuzzySet(30, 50, 50, 70);
30   // Including the FuzzySet into FuzzyInput
31   distance->addFuzzySet(safe);
32   // Instantiating a FuzzySet object
33   FuzzySet *big = new FuzzySet(60, 80, 80, 80);
34   // Including the FuzzySet into FuzzyInput
35   distance->addFuzzySet(big);
36   // Including the FuzzyInput into Fuzzy

```

6. Kemudian hapus lagi bagian kode yang ditunjuk oleh gambar

```

16 {
17   // Set the Serial output
18   Serial.begin(9600);
19
20   //Instantiating a FuzzyInput object
21   FuzzyInput *distance = new FuzzyInput(1);
22   //Instantiating a FuzzySet object
23   FuzzySet *small = new FuzzySet(0, 20, 20, 40);
24   //Including the FuzzySet into FuzzyInput
25   distance->addFuzzySet(small);
26   //Instantiating a FuzzySet object
27   FuzzySet *safe = new FuzzySet(30, 50, 50, 70);
28   //Including the FuzzySet into FuzzyInput
29   distance->addFuzzySet(safe);
30   //Instantiating a FuzzySet object
31   FuzzySet *big = new FuzzySet(60, 80, 80, 80);
32   //Including the FuzzySet into FuzzyInput
33   distance->addFuzzySet(big);
34   //Including the FuzzyInput into Fuzzy
35   fuzzy->addFuzzyInput(distance);
36
37   // Instantiating a FuzzyInput object

```

7. Di bawah baris kode `Serial.begin(9600);` yang sudah bersih tadi, tambahkan persis di bawahnya **Kode Parameter Input Fuzzy**, dengan contoh Suhu/Temperature (Dingin, Hangat, dan Panas):

**Kode Program**

```

// Define Temperature Range (Cold, Warm, and Hot)
FuzzyInput *temp = new FuzzyInput(1);
FuzzySet *cold = new FuzzySet(0, 0, 15, 20);
temp->addFuzzySet(cold);
FuzzySet *warm = new FuzzySet(15, 20, 30, 35);
temp->addFuzzySet(warm);
FuzzySet *hot = new FuzzySet(30, 35, 100, 100);
temp->addFuzzySet(hot);
fuzzy->addFuzzyInput(temp);

```

```

13 Fuzzy *fuzzy = new Fuzzy();
14
15 void setup()
16 {
17   // Set the Serial output
18   Serial.begin(9600);
19
20   // Define Temperature Range (Cold, Warm, and Hot)
21   FuzzyInput *temp = new FuzzyInput(1);
22   FuzzySet *cold = new FuzzySet(0, 0, 15, 20);
23   temp->addFuzzySet(cold);
24   FuzzySet *warm = new FuzzySet(15, 20, 30, 35);
25   temp->addFuzzySet(warm);
26   FuzzySet *hot = new FuzzySet(30, 35, 100, 100);
27   temp->addFuzzySet(hot);
28   fuzzy->addFuzzyInput(temp);
29
30   // Instantiating a FuzzyOutput objects
31   FuzzyOutput *speed = new FuzzyOutput(1);
32   // Instantiating a FuzzySet object
33   FuzzySet *slow = new FuzzySet(0, 10, 10, 20);
34   // Including the FuzzySet into FuzzyOutput

```

8. Setelah itu, hapus bagian kode yang ditunjuk gambar. Kode yang dihapus tepat berada di bawah kode yang baru dimasukkan.

```

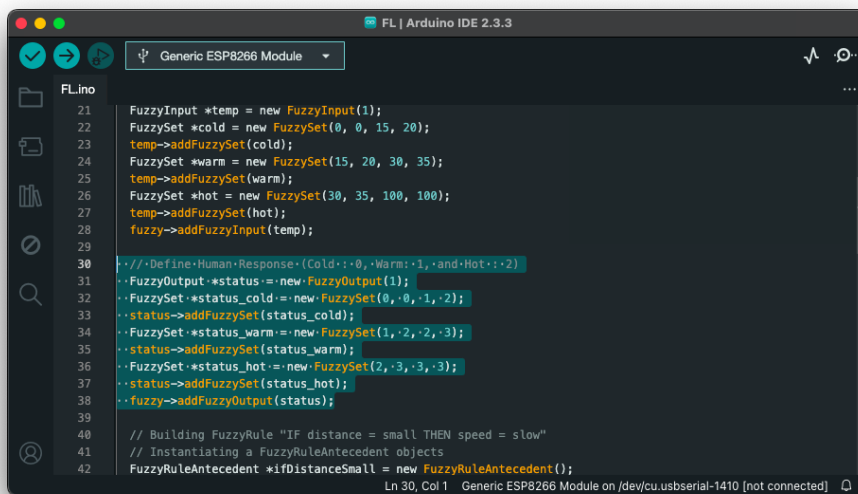
26 FuzzySet *hot = new FuzzySet(30, 35, 100, 100);
27 temp->addFuzzySet(hot);
28 fuzzy->addFuzzyInput(temp);
29
30 // Instantiating a FuzzyOutput objects
31 FuzzyOutput *speed = new FuzzyOutput(1);
32 // Instantiating a FuzzySet object
33 FuzzySet *slow = new FuzzySet(0, 10, 10, 20);
34 // Including the FuzzySet into FuzzyOutput
35 speed->addFuzzySet(slow);
36 // Instantiating a FuzzySet object
37 FuzzySet *average = new FuzzySet(10, 20, 30, 40);
38 // Including the FuzzySet into FuzzyOutput
39 speed->addFuzzySet(average);
40 // Instantiating a FuzzySet object
41 FuzzySet *fast = new FuzzySet(30, 40, 40, 50);
42 // Including the FuzzySet into FuzzyOutput
43 speed->addFuzzySet(fast);
44 // Including the FuzzyOutput into Fuzzy
45 fuzzy->addFuzzyOutput(speed);
46

```

9. Lalu tambahkan Kode Parameter Output Fuzzy yang dibuat agar mudah dipahami oleh manusia (Dingin = 1, Hangat = 2, dan Panas = 3) tepat di bawah Kode Parameter Input tadi.

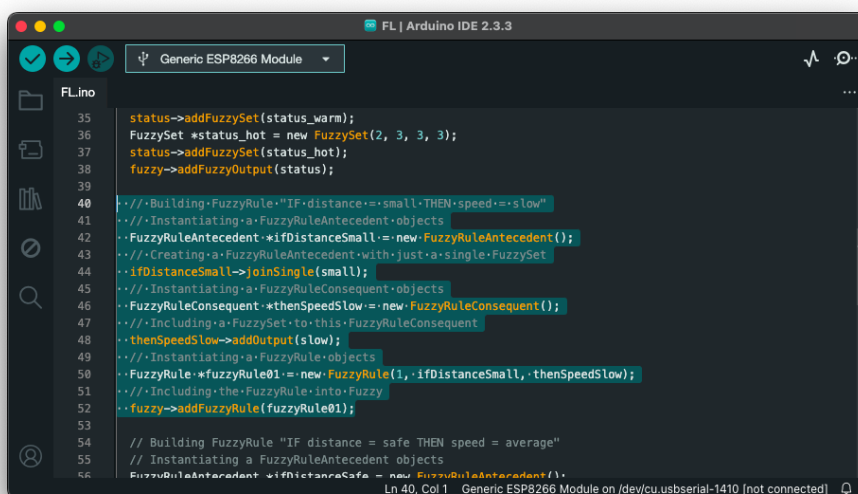
### Kode Program

```
// Define Human Response (Cold : 1, Warm: 2, and Hot : 3)
FuzzyOutput *status = new FuzzyOutput(1);
FuzzySet *status_cold = new FuzzySet(0, 1, 1, 2);
status->addFuzzySet(status_cold);
FuzzySet *status_warm = new FuzzySet(1, 2, 2, 3);
status->addFuzzySet(status_warm);
FuzzySet *status_hot = new FuzzySet(2, 3, 3, 4);
status->addFuzzySet(status_hot);
fuzzy->addFuzzyOutput(status);
```



```
FL.ino
21 FuzzyInput *temp = new FuzzyInput(1);
22 FuzzySet *cold = new FuzzySet(0, 0, 15, 20);
23 temp->addFuzzySet(cold);
24 FuzzySet *warm = new FuzzySet(15, 20, 30, 35);
25 temp->addFuzzySet(warm);
26 FuzzySet *hot = new FuzzySet(30, 35, 100, 100);
27 temp->addFuzzySet(hot);
28 fuzzy->addFuzzyInput(temp);
29
30 // Define Human Response (Cold : 1, Warm: 2, and Hot : 3)
31 FuzzyOutput *status = new FuzzyOutput(1);
32 FuzzySet *status_cold = new FuzzySet(0, 0, 1, 2);
33 status->addFuzzySet(status_cold);
34 FuzzySet *status_warm = new FuzzySet(1, 2, 2, 3);
35 status->addFuzzySet(status_warm);
36 FuzzySet *status_hot = new FuzzySet(2, 3, 3, 3);
37 status->addFuzzySet(status_hot);
38 fuzzy->addFuzzyOutput(status);
39
40 // Building FuzzyRule "IF distance = small THEN speed = slow"
41 // Instantiating a FuzzyRuleAntecedent objects
42 FuzzyRuleAntecedent *ifDistanceSmall = new FuzzyRuleAntecedent();
Ln 30, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]
```

10. Berikutnya, hapus tiga (3) bagian kode yang ditunjukkan oleh gambar



```
FL.ino
35 status->addFuzzySet(status_warm);
36 FuzzySet *status_hot = new FuzzySet(2, 3, 3, 3);
37 status->addFuzzySet(status_hot);
38 fuzzy->addFuzzyOutput(status);
39
40 // Building FuzzyRule "IF distance = small THEN speed = slow"
41 // Instantiating a FuzzyRuleAntecedent objects
42 FuzzyRuleAntecedent *ifDistanceSmall = new FuzzyRuleAntecedent();
43 // Creating a FuzzyRuleAntecedent with just a single FuzzySet
44 *ifDistanceSmall->joinSingle(small);
45 // Instantiating a FuzzyRuleConsequent objects
46 FuzzyRuleConsequent *thenSpeedSlow = new FuzzyRuleConsequent();
47 // Including a FuzzySet to this FuzzyRuleConsequent
48 *thenSpeedSlow->addOutput(sLow);
49 // Instantiating a FuzzyRule objects
50 FuzzyRule *fuzzyRule01 = new FuzzyRule(1, *ifDistanceSmall, *thenSpeedSlow);
51 // Including the FuzzyRule into Fuzzy
52 *fuzzy->addFuzzyRule(fuzzyRule01);
53
54 // Building FuzzyRule "IF distance = safe THEN speed = average"
55 // Instantiating a FuzzyRuleAntecedent objects
56 FuzzyRuleAntecedent *ifDistanceSafe = new FuzzyRuleAntecedent();
Ln 40, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]
```

```

50 FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifDistanceSmall, thenSpeedSlow);
51 // Including the FuzzyRule into Fuzzy
52 fuzzy->addFuzzyRule(fuzzyRule01);
53
54 // Building FuzzyRule "IF distance = safe THEN speed = average"
55 // Instantiating a FuzzyRuleAntecedent objects
56 FuzzyRuleAntecedent *ifDistanceSafe = new FuzzyRuleAntecedent();
57 // Creating a FuzzyRuleAntecedent with just a single FuzzySet
58 ifDistanceSafe->joinSingle(safe);
59 // Instantiating a FuzzyRuleConsequent objects
60 FuzzyRuleConsequent *thenSpeedAverage = new FuzzyRuleConsequent();
61 // Including a FuzzySet to this FuzzyRuleConsequent
62 thenSpeedAverage->addOutput(average);
63 // Instantiating a FuzzyRule objects
64 FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifDistanceSafe, thenSpeedAverage);
65 // Including the FuzzyRule into Fuzzy
66 fuzzy->addFuzzyRule(fuzzyRule02);
67
68 // Building FuzzyRule "IF distance = big THEN speed = high"
69 // Instantiating a FuzzyRuleAntecedent objects
70 FuzzyRuleAntecedent *ifDistanceBig = new FuzzyRuleAntecedent();

```

```

65 // Including the FuzzyRule into Fuzzy
66 fuzzy->addFuzzyRule(fuzzyRule02);
67
68 // Building FuzzyRule "IF distance = big THEN speed = high"
69 // Instantiating a FuzzyRuleAntecedent objects
70 FuzzyRuleAntecedent *ifDistanceBig = new FuzzyRuleAntecedent();
71 // Creating a FuzzyRuleAntecedent with just a single FuzzySet
72 ifDistanceBig->joinSingle(big);
73 // Instantiating a FuzzyRuleConsequent objects
74 FuzzyRuleConsequent *thenSpeedFast = new FuzzyRuleConsequent();
75 // Including a FuzzySet to this FuzzyRuleConsequent
76 thenSpeedFast->addOutput(fast);
77 // Instantiating a FuzzyRule objects
78 FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifDistanceBig, thenSpeedFast);
79 // Including the FuzzyRule into Fuzzy
80 fuzzy->addFuzzyRule(fuzzyRule03);
81 }
82
83 void loop()
84 {
85 // Getting a random value

```

11. Setelah dihapus, masukkan kode **Fuzzy Rules** baru tepat di bawah kode **Fuzzy Output** yang telah dibuat sebelumnya

(a) **Fuzzy Rule 1**

Kode Program

```

// Define Fuzzy Rule #1. IF Temp = COLD, THEN Status = COLD
FuzzyRuleAntecedent *ifTempCold = new FuzzyRuleAntecedent();
ifTempCold->joinSingle(cold);
FuzzyRuleConsequent *thenStatusCold = new FuzzyRuleConsequent();
thenStatusCold->addOutput(status_cold);
FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifTempCold, thenStatusCold);
fuzzy->addFuzzyRule(fuzzyRule01);

```

```

32 FuzzySet *status_cold = new FuzzySet(0, 0, 1, 2);
33 status->addFuzzySet(status_cold);
34 FuzzySet *status_warm = new FuzzySet(1, 2, 2, 3);
35 status->addFuzzySet(status_warm);
36 FuzzySet *status_hot = new FuzzySet(2, 3, 3, 3);
37 status->addFuzzySet(status_hot);
38 fuzzy->addFuzzyOutput(status);
39
40 // Define Fuzzy Rule #1. IF Temp = COLD, THEN Status = COLD
41 FuzzyRuleAntecedent *ifTempCold = new FuzzyRuleAntecedent();
42 ifTempCold->joinSingle(cold);
43 FuzzyRuleConsequent *thenStatusCold = new FuzzyRuleConsequent();
44 thenStatusCold->addOutput(status_cold);
45 FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifTempCold, thenStatusCold);
46 fuzzy->addFuzzyRule(fuzzyRule01);
47
48 // Define Fuzzy Rule #2. IF Temp = WARM, THEN Status = WARM
49 FuzzyRuleAntecedent *ifTempWarm = new FuzzyRuleAntecedent();
50 ifTempWarm->joinSingle(warm);
51 FuzzyRuleConsequent *thenStatusWarm = new FuzzyRuleConsequent();
52 thenStatusWarm->addOutput(status_warm);
53 FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifTempWarm, thenStatusWarm);

```

(b) Fuzzy Rule 2

Kode Program

```

// Define Fuzzy Rule #2. IF Temp = WARM, THEN Status = WARM
FuzzyRuleAntecedent *ifTempWarm = new FuzzyRuleAntecedent();
ifTempWarm->joinSingle(warm);
FuzzyRuleConsequent *thenStatusWarm = new FuzzyRuleConsequent();
thenStatusWarm->addOutput(status_warm);
FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifTempWarm, thenStatusWarm);
fuzzy->addFuzzyRule(fuzzyRule02);

```

```

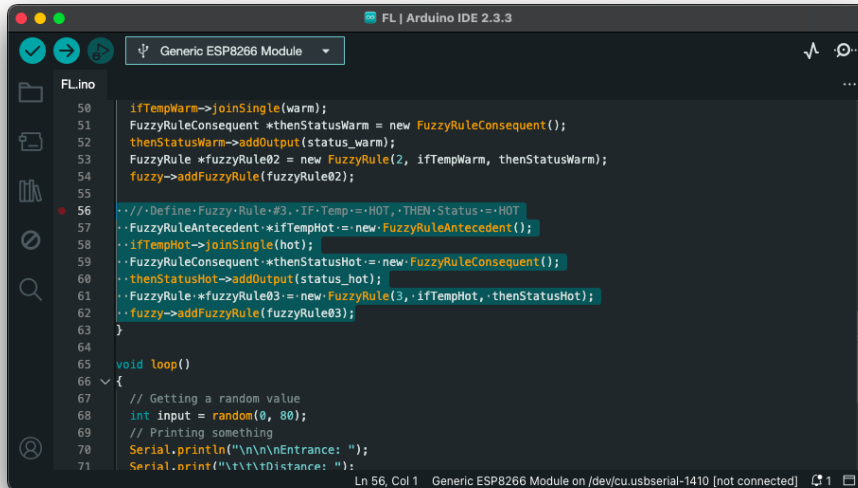
42 ifTempCold->joinSingle(cold);
43 FuzzyRuleConsequent *thenStatusCold = new FuzzyRuleConsequent();
44 thenStatusCold->addOutput(status_cold);
45 FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifTempCold, thenStatusCold);
46 fuzzy->addFuzzyRule(fuzzyRule01);
47
48 // Define Fuzzy Rule #2. IF Temp = WARM, THEN Status = WARM
49 FuzzyRuleAntecedent *ifTempWarm = new FuzzyRuleAntecedent();
50 ifTempWarm->joinSingle(warm);
51 FuzzyRuleConsequent *thenStatusWarm = new FuzzyRuleConsequent();
52 thenStatusWarm->addOutput(status_warm);
53 FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifTempWarm, thenStatusWarm);
54 fuzzy->addFuzzyRule(fuzzyRule02);
55
56 // Define Fuzzy Rule #3. IF Temp = HOT, THEN Status = HOT
57 FuzzyRuleAntecedent *ifTempHot = new FuzzyRuleAntecedent();
58 ifTempHot->joinSingle(hot);
59 FuzzyRuleConsequent *thenStatusHot = new FuzzyRuleConsequent();
60 thenStatusHot->addOutput(status_hot);
61 FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifTempHot, thenStatusHot);
62 fuzzy->addFuzzyRule(fuzzyRule03);
63

```

(c) Fuzzy Rule 3

### Kode Program

```
// Define Fuzzy Rule #3. IF Temp = HOT, THEN Status = HOT
FuzzyRuleAntecedent *ifTempHot = new FuzzyRuleAntecedent();
ifTempHot->joinSingle(hot);
FuzzyRuleConsequent *thenStatusHot = new FuzzyRuleConsequent();
thenStatusHot->addOutput(status_hot);
FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifTempHot, thenStatusHot);
fuzzy->addFuzzyRule(fuzzyRule03);
```

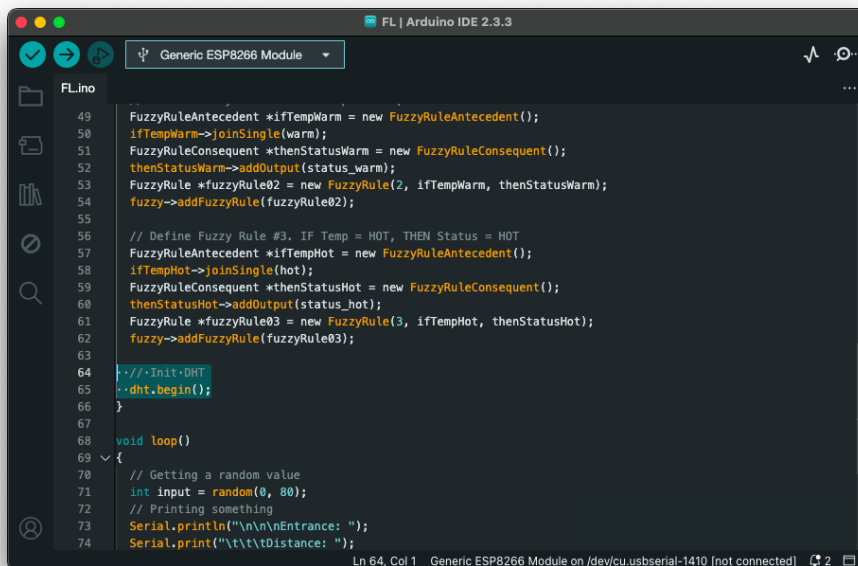


The screenshot shows the Arduino IDE interface with the code from the previous block highlighted in blue. The code is part of a larger program in 'FL.ino'. The IDE title bar shows 'FL | Arduino IDE 2.3.3' and the board is set to 'Generic ESP8266 Module'. The status bar at the bottom indicates 'Ln 56, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]'.

12. Bagian terakhir dari fungsi `setup()` adalah kode inisialisasi dht

### Kode Program

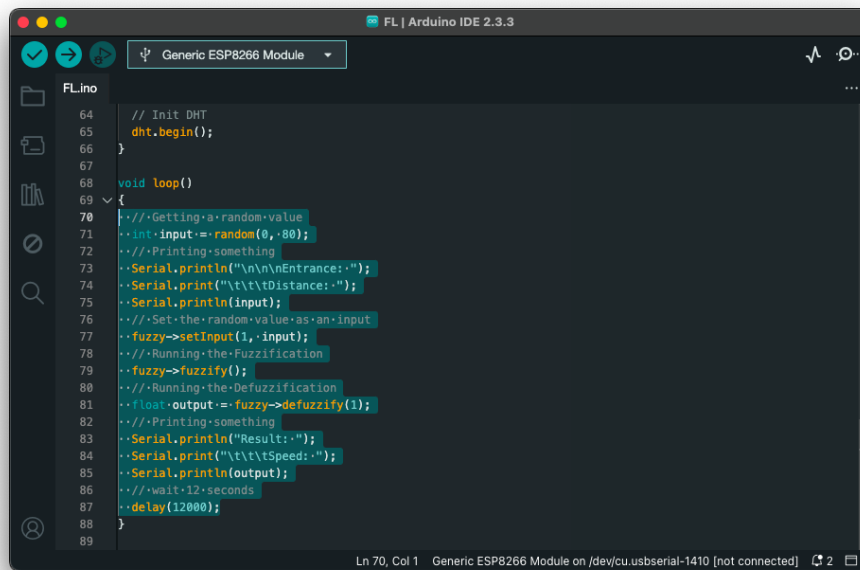
```
// Init DHT
dht.begin();
```



The screenshot shows the Arduino IDE interface with the code for initializing the DHT sensor. The code is part of the same program in 'FL.ino'. The IDE title bar shows 'FL | Arduino IDE 2.3.3' and the board is set to 'Generic ESP8266 Module'. The status bar at the bottom indicates 'Ln 64, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]'.



13. Masuk ke fungsi `loop()`, dan hapus semua kode di dalam fungsi ini



The screenshot shows the Arduino IDE interface with the following code in the `loop()` function:

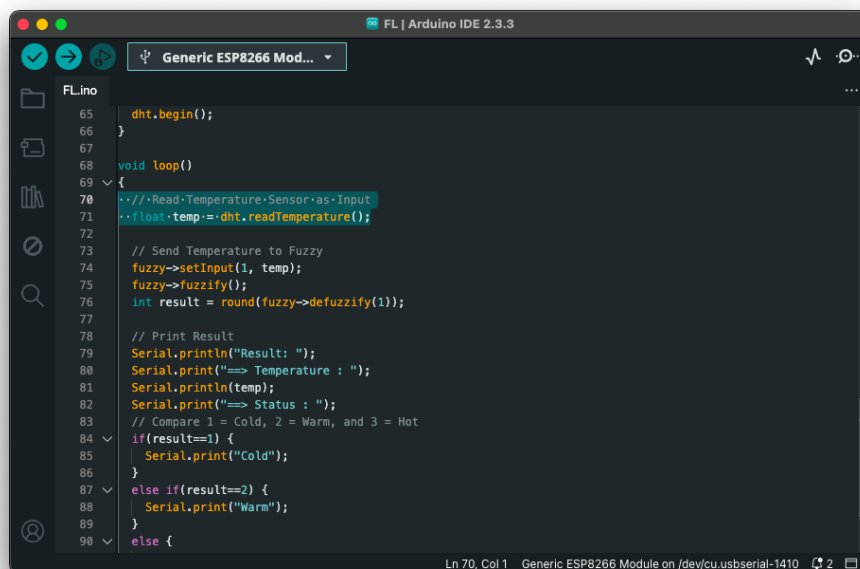
```
64 // Init DHT
65 dht.begin();
66 }
67
68 void loop()
69 {
70 //Getting a random value
71 int input = random(0, 80);
72 //Printing something
73 Serial.println("\n\nEntrance:");
74 Serial.print("\t\tDistance:");
75 Serial.println(input);
76 //Set the random value as an input
77 fuzzy->setInput(1, input);
78 //Running the Fuzzification
79 fuzzy->fuzzify();
80 //Running the Defuzzification
81 float output = fuzzy->defuzzify(1);
82 //Printing something
83 Serial.println("Result:");
84 Serial.print("\t\t\tSpeed:");
85 Serial.println(output);
86 //wait 12 seconds
87 delay(12000);
88 }
89
```

14. Lalu masukkan satu per satu kode berikut:

(a) Baris kode pembaca sensor suhu

Kode Program

```
// Read Temperature Sensor as Input
float temp = dht.readTemperature();
```



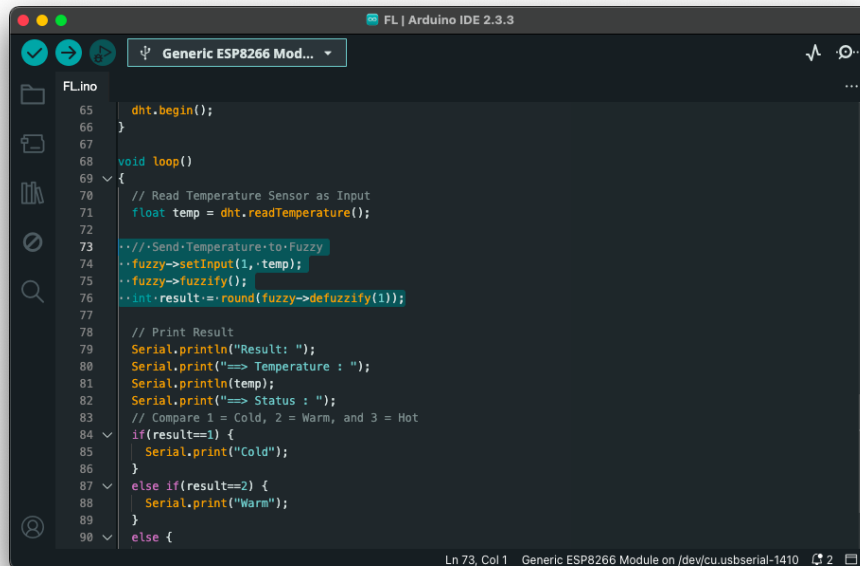
The screenshot shows the updated `loop()` function in the Arduino IDE:

```
65 dht.begin();
66 }
67
68 void loop()
69 {
70 //Read Temperature Sensor as Input
71 float temp = dht.readTemperature();
72
73 // Send Temperature to Fuzzy
74 fuzzy->setInput(1, temp);
75 fuzzy->fuzzify();
76 int result = round(fuzzy->defuzzify(1));
77
78 // Print Result
79 Serial.println("Result: ");
80 Serial.print("=> Temperature : ");
81 Serial.println(temp);
82 Serial.print("=> Status : ");
83 // Compare 1 = Cold, 2 = Warm, and 3 = Hot
84 if(result==1) {
85   Serial.print("Cold");
86 }
87 else if(result==2) {
88   Serial.print("Warm");
89 }
90 else {
```

(b) Berikutnya kode Fuzzifikasi dan Defuzzifikasi

### Kode Program

```
// Send Temperature to Fuzzy
fuzzy->setInput(1, temp);
fuzzy->fuzzify();
int result = round(fuzzy->defuzzify(1));
```



The screenshot shows the Arduino IDE interface with the following code visible in the main editor window:

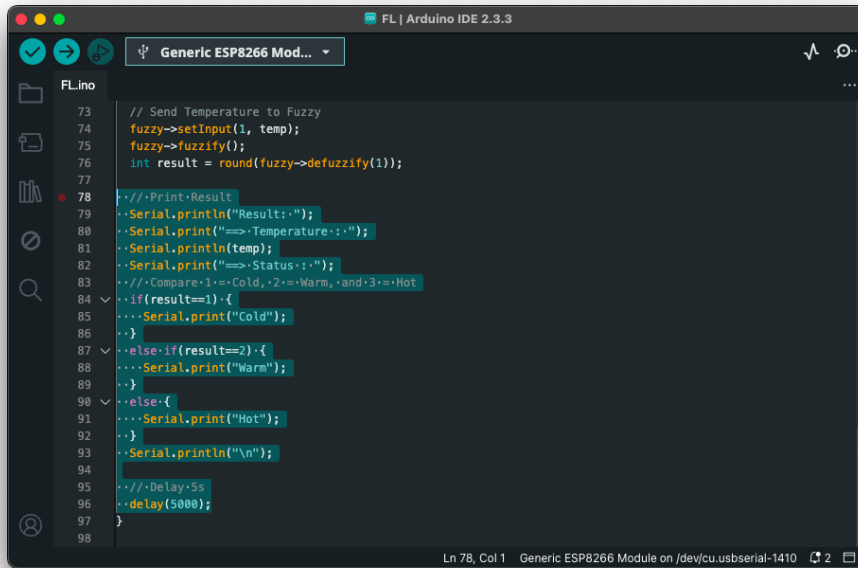
```
65 dht.begin();
66 }
67
68 void loop()
69 {
70 // Read Temperature Sensor as Input
71 float temp = dht.readTemperature();
72
73 //Send Temperature to Fuzzy
74 fuzzy->setInput(1, temp);
75 fuzzy->fuzzify();
76 int result = round(fuzzy->defuzzify(1));
77
78 // Print Result
79 Serial.println("Result: ");
80 Serial.print("==> Temperature : ");
81 Serial.println(temp);
82 Serial.print("==> Status : ");
83 // Compare 1 = Cold, 2 = Warm, and 3 = Hot
84 if(result==1) {
85   Serial.print("Cold");
86 }
87 else if(result==2) {
88   Serial.print("Warm");
89 }
90 else {
```

(c) Terakhir, kode cetak hasil dan delay

### Kode Program

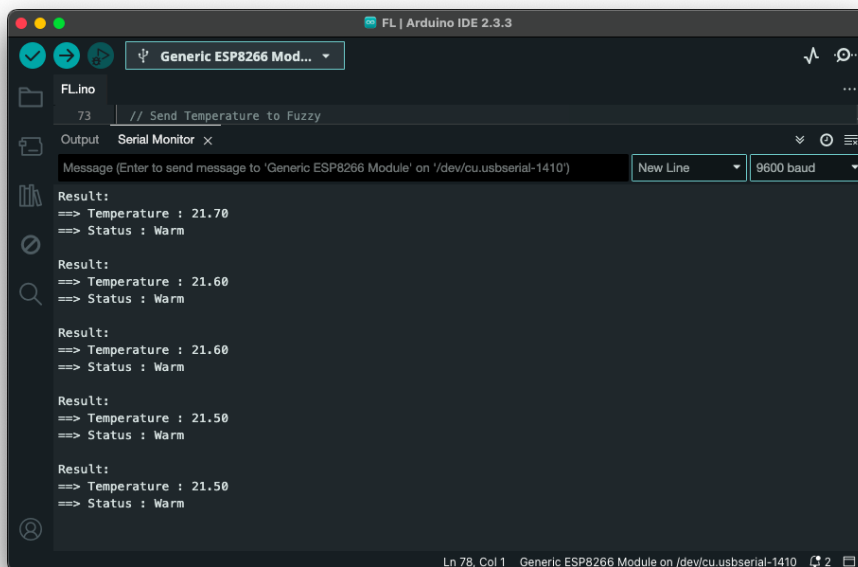
```
// Print Result
Serial.println("Result: ");
Serial.print("==> Temperature : ");
Serial.println(temp);
Serial.print("==> Status : ");
// Compare 1 = Cold, 2 = Warm, and 3 = Hot
if(result==1) {
    Serial.print("Cold");
}
else if(result==2) {
    Serial.print("Warm");
}
else {
    Serial.print("Hot");
}
Serial.println("\n");

// Delay 5s
delay(5000);
```



```
73 // Send Temperature to Fuzzy
74 fuzzy->setInput(1, temp);
75 fuzzy->fuzzify();
76 int result = round(fuzzy->defuzzify(1));
77
78 //Print Result
79 ..Serial.println("Result:");
80 ..Serial.print("==> Temperature:");
81 ..Serial.println(temp);
82 ..Serial.print("==> Status:");
83 // Compare 1=Cold, 2=Warm, and 3=Hot
84 ..if(result==1){
85 ..Serial.print("Cold");
86 ..}
87 ..else-if(result==2){
88 ..Serial.print("Warm");
89 ..}
90 ..else{
91 ..Serial.print("Hot");
92 ..}
93 ..Serial.println("\n");
94
95 //Delay 5s
96 ..delay(5000);
97
98
```

15. Colokkan ESP8266 dan DHT11, dan unggah kode. Lihat hasil seperti berikut



```
73 // Send Temperature to Fuzzy
Output Serial Monitor x
Message (Enter to send message to 'Generic ESP8266 Module' on '/dev/cu.usbserial-1410') New Line 9600 baud
Result:
==> Temperature : 21.70
==> Status : Warm
Result:
==> Temperature : 21.60
==> Status : Warm
Result:
==> Temperature : 21.60
==> Status : Warm
Result:
==> Temperature : 21.50
==> Status : Warm
Result:
==> Temperature : 21.50
==> Status : Warm
```

16. Parameter **Fuzzy Input**, **Fuzzy Output** maupun **Fuzzy Rule** dapat diubah sesuai dengan jenis data yang ingin disederhanakan. Contoh data **Humidity**

# Bab 8

## Praktikum 8

Di bagian ini mahasiswa diminta melakukan pengambilan data lingkungan dengan menggunakan Internet of Things. Mahasiswa diharapkan telah menyelesaikan semua praktikum yang ada di halaman sebelumnya.

### 8.1 Tugas Akhir Praktikum

- Mahasiswa perlu menyiapkan perlengkapan berupa:
  - Perangkat yang sudah berjalan dengan baik
    1. **Wajib** → ESP8266
    2. **Wajib** → DHT11 (DHT22 jika paham)
    3. **Opsional** → Breakout Board ESP8266
  - Wajib menggunakan **Regresi Linier**
  - Platform bebas memilih antara:
    1. Adafruit IO
    2. Thingspeak
  - Charger HP dan Kabel MicroUSB/USB-C
  - Akses Internet
- Pastikan Akses Poin sudah sesuai dengan kode perangkat Internet of Things
- Lakukan pengambilan data di lingkungan bebas.
- Setelah satu jam atau lebih, unduh data yang didapatkan dalam format **CSV/Excel**
- Buat laporan sesuai format seperti berikut:
  1. Cover Laporan dengan nama tim lengkap
  2. Halaman Daftar Isi
  3. Spesifikasi Model (Jelaskan komponen-komponen yang digunakan)

4. Proses Observasi (Jelaskan proses observasi dengan alatnya)
  5. Hasil Observasi #1 dalam bentuk tabel berisikan sampel data (15 baris)
  6. Hasil Observasi #2 (dalam bentuk grafik):
    - (a) Suhu
    - (b) Prediksi Suhu (dari Regresi Linier, cek Praktikum 6)
    - (c) Kelembaban
  7. Analisis Hasil Observasi (Jelaskan hasil observasi yang didapatkan)
  8. Kesimpulan
- Laporan dan Hasil Data (CSV/Excel) dikirimkan ke Praktikum 8
  - Format File hanya **PDF** dan Hasil Data (.csv/.xlsx)