



UNIVERSITAS SEMARANG
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
TEKNIK INFORMATIKA

Internet of Thing

Modul Praktikum Mahasiswa

Oleh:

Alauddin Maulana Hirzan, S. Kom., M. Kom
NIDN. 0607069401

Daftar Isi

Pendahuluan	2
0.1 Menenal <i>Internet of Things</i>	2
0.2 Perangkat Board IoT	2
Persiapan Praktikum	6
0.3 Perangkat Keras	6
0.4 Perangkat Lunak	6
1 Praktikum 1	7
1.1 Konfigurasi Arduino IDE dan ESP8266	7
1.2 Tutorial	7
2 Praktikum 2	14
2.1 ESP8266, DHT-22, dan AdafruitIO	14
2.2 Tutorial	14
3 Praktikum 3	27
3.1 ESP8266, DHT-22, dan Thingspeak	27
3.2 Tutorial	27
4 Praktikum 4	40
4.1 ESP8266, DHT-22, dan Firebase Realtime	40
4.2 Tutorial	40
5 Praktikum 5	53
5.1 ESP8266, DHT-22, dan Telegram Bot	53
5.2 Tutorial	53
6 Praktikum 6	59
6.1 ESP8266, DHT-22, dan Linear Regression	59
6.2 Tutorial	59
7 Praktikum 7	70
7.1 ESP8266, DHT-22, dan Fuzzy Logic	70
7.2 Tutorial	70
8 Praktikum 8	82
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.

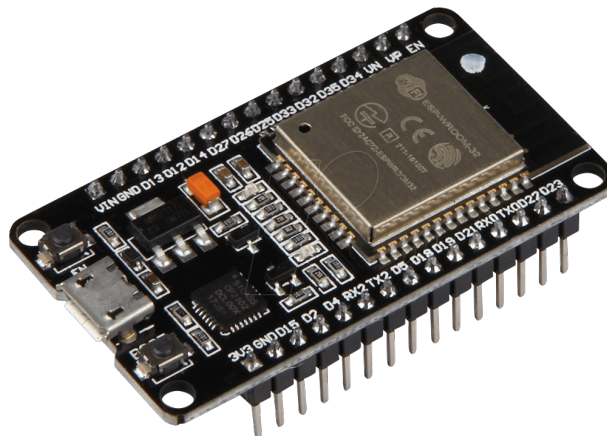
Berikut ini adalah daftar Board yang dapat dibeli dengan harga terjangkau:

1. Arduino



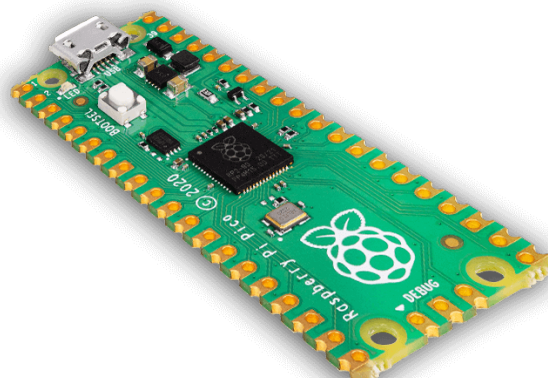
Gambar 2: Board Arduino

2. NodeMCU



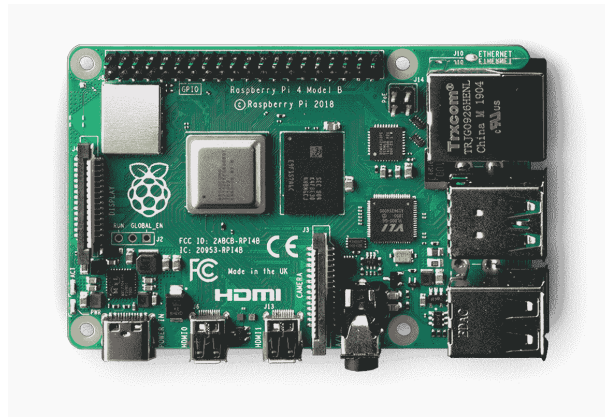
Gambar 3: Board NodeMCU

3. Raspberry Pi Pico



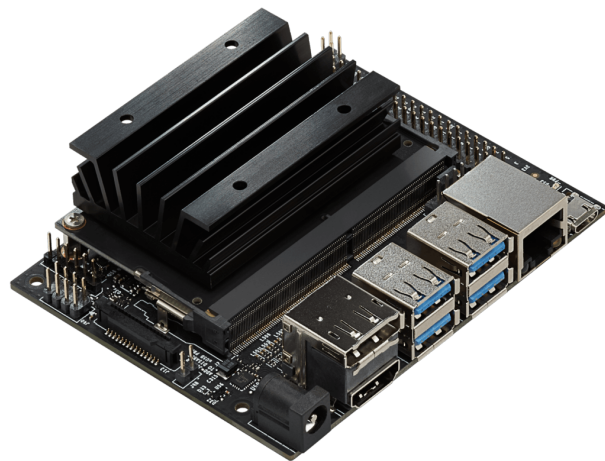
Gambar 4: Board Pico

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



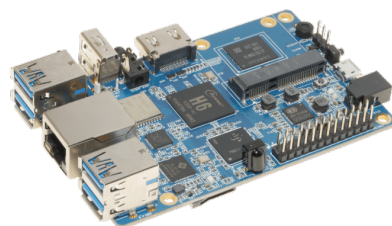
Gambar 5: Board Pi 4B

5. NVidia Jetson



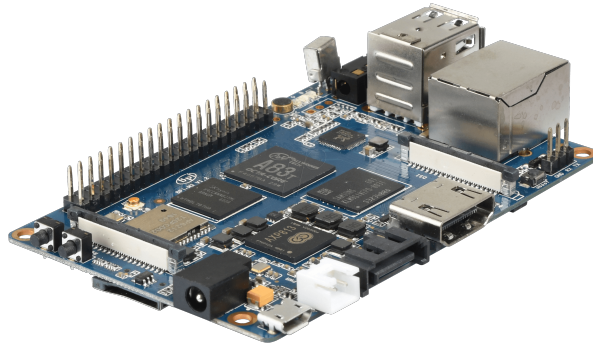
Gambar 6: Board NVidia Jetson

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-22

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
 - CP210X (Model Amica ESP8266MOD) <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>

Bab 1

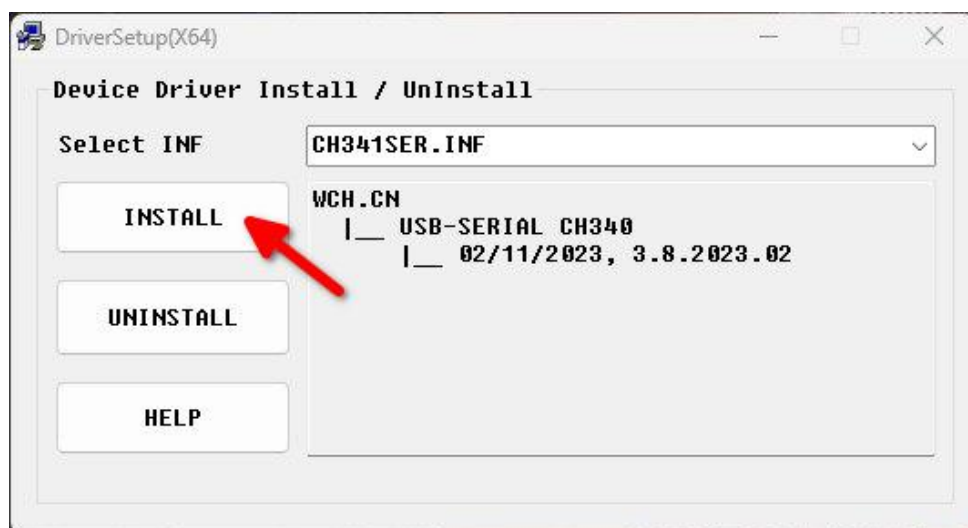
Praktikum 1

1.1 Konfigurasi Arduino IDE dan ESP8266

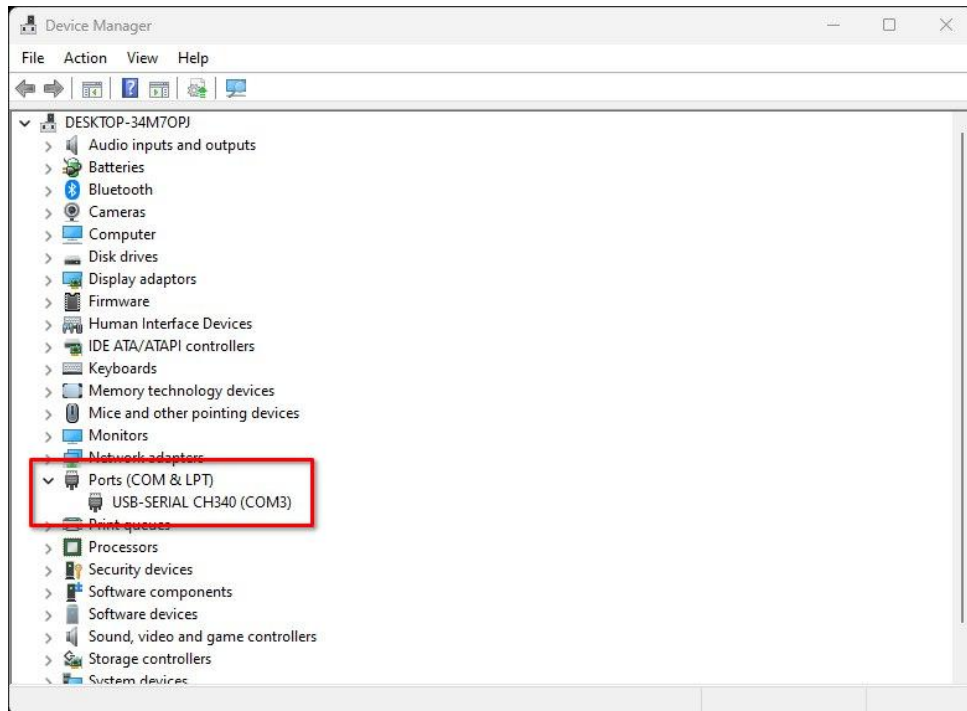
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke komputer beserta konfigurasinya hingga dapat dikenali oleh Arduino IDE. Mahasiswa diharapkan untuk membaca, dan memahami **Persiapan Praktikum** yang ada di halaman sebelumnya.

1.2 Tutorial

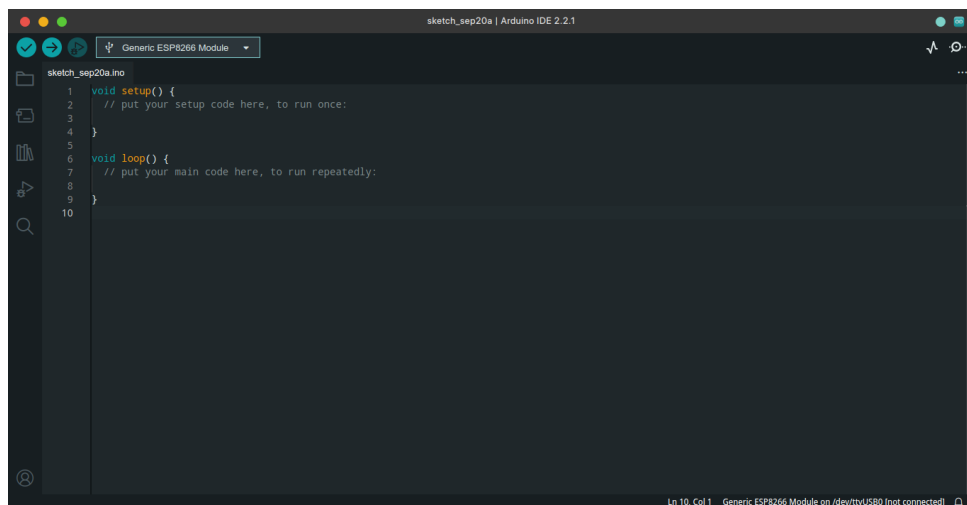
1. Setelah mahasiswa menyiapkan perangkat-perangkat yang diperlukan, maka langkah berikutnya adalah melakukan instalasi driver terlebih dahulu.
2. File driver **CH341SER** yang sudah diunduh, dibuka untuk diinstall. Cukup klik **Install** untuk memasang driver (Windows 10 ke bawah)



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

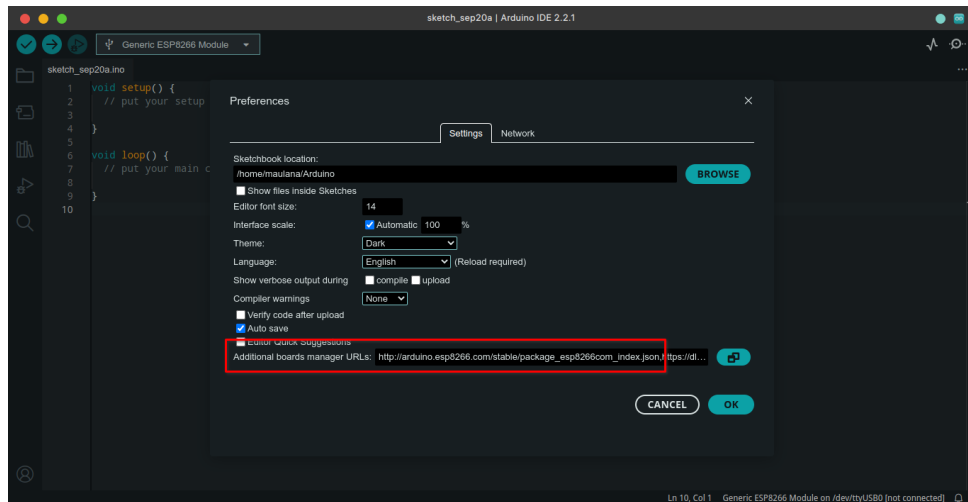


4. Langkah berikutnya adalah mengunduh **Arduino IDE**, usahakan untuk mendapatkan versi terbaru. Setelah unduh, buka aplikasi tersebut

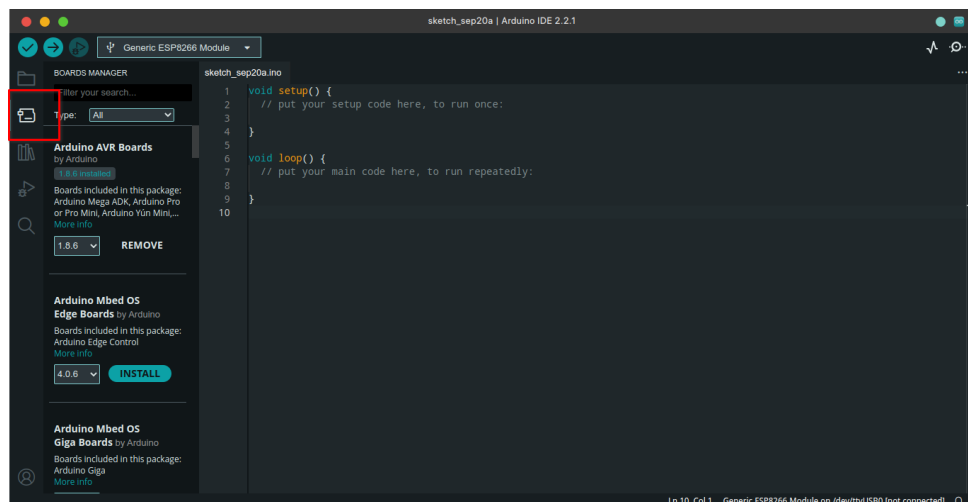


5. Namun **Arduino IDE** ini belum mendukung perangkat yang kita gunakan. Langkah berikutnya buka **File** → **Preferences** →. Tambahkan baris **Alamat URL** berikut ke **Additional board manager URLs**. Klik **OK** untuk mengupdate otomatis.

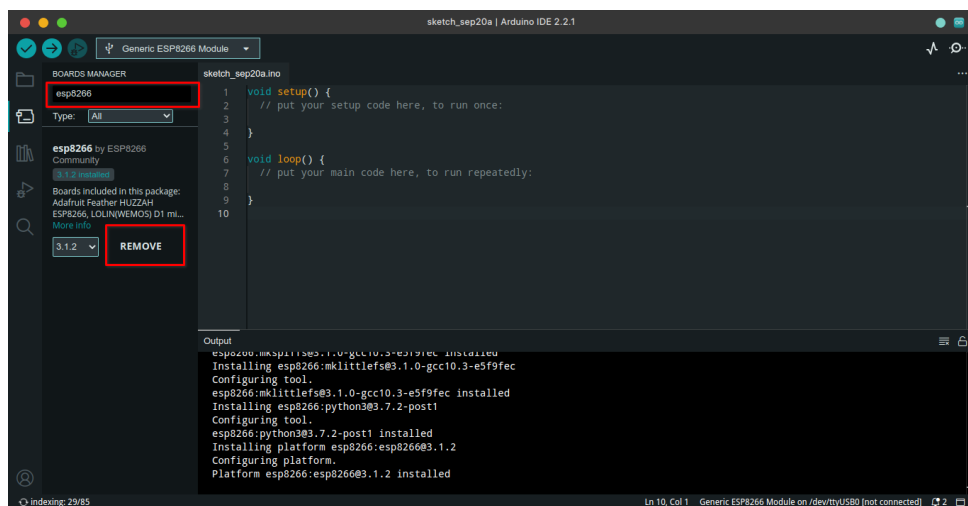
- http://arduino.esp8266.com/stable/package_esp8266com_index.json



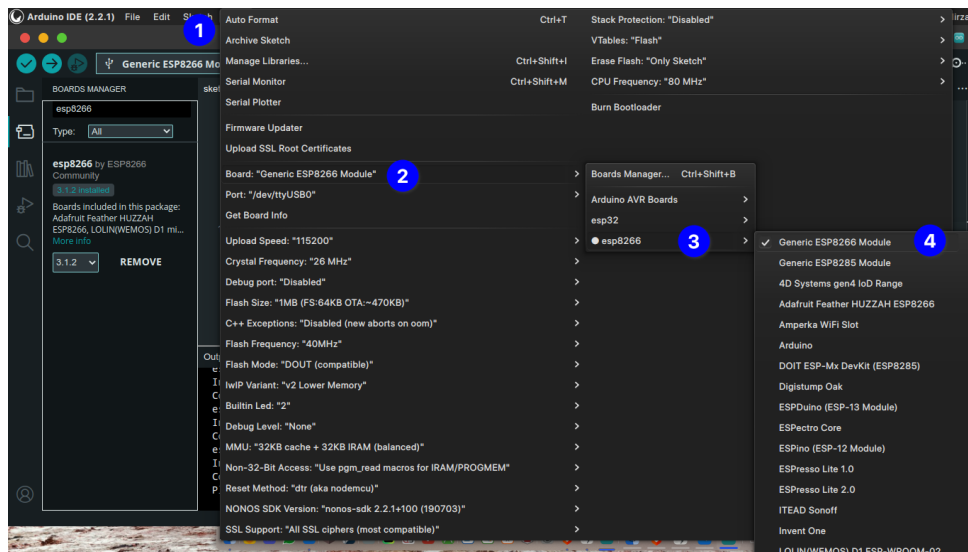
6. Jika sudah, install driver **ESP8266** dengan klik **Boards Manager** di **Sidebar Kiri** atau **Tools** → **Board:** → **Boards Manager**



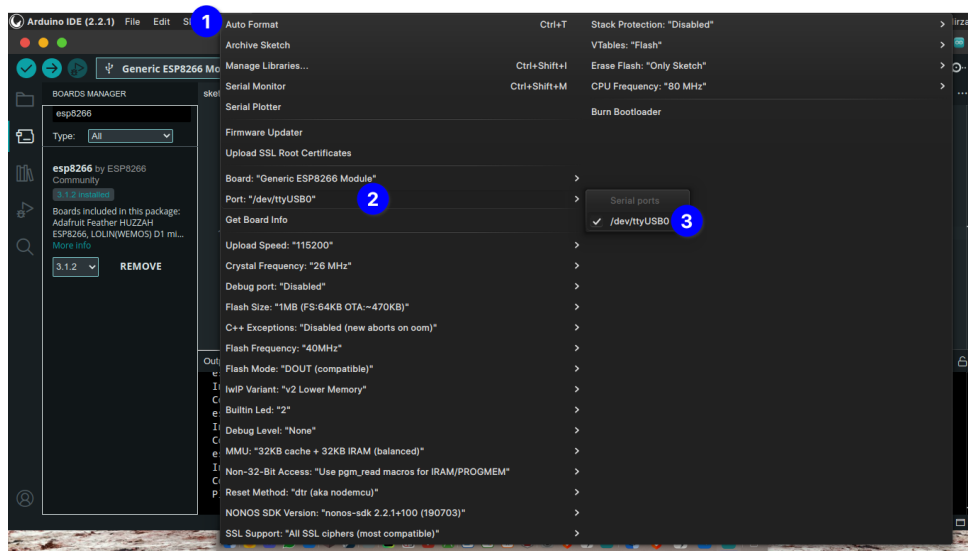
7. Di kolom **Pencarian**, ketik **ESP8266** dan klik **Install**



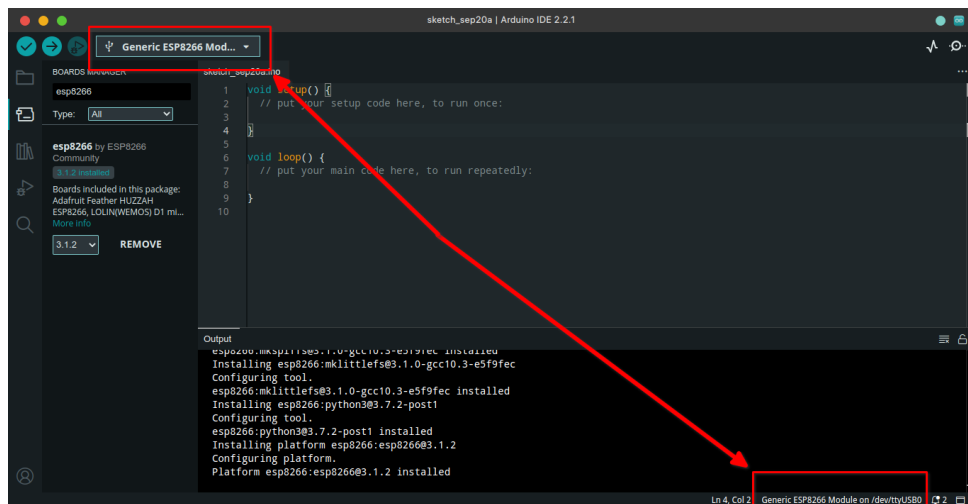
8. **Arduino IDE** sudah siap, namun belum terhubung ke perangkat. Untuk menghubungkan antara **IDE** dengan **ESP8266**, pilih **Tools** → **Board:** → **esp8266** → **Generic ESP8266 Module**



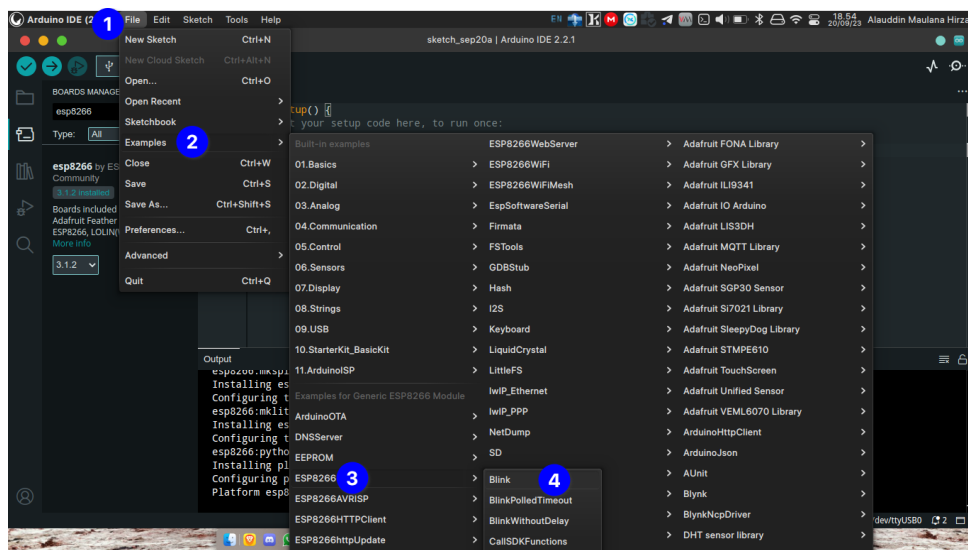
9. Kemudian pastikan **Port Serial** yang digunakan, sama dengan yang ada di **Device Manager**. Cek dengan menu **Tools** → **Port:** → **Pilih COM Sesuai Device Manager**



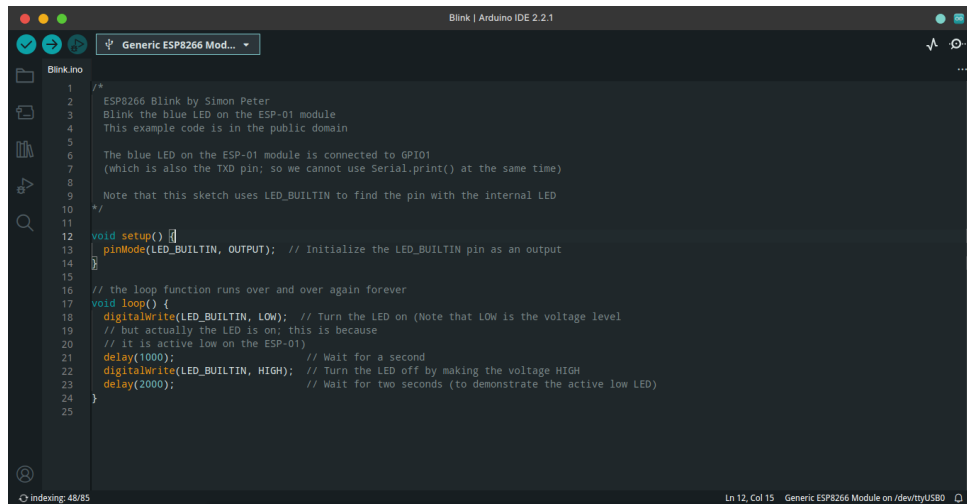
10. Jika sudah terhubung, akan ada tanda tulisan **Generic ESP8266 Module** on **COMXXX** di bawah kanan maupun simbol USB di atas kiri



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**

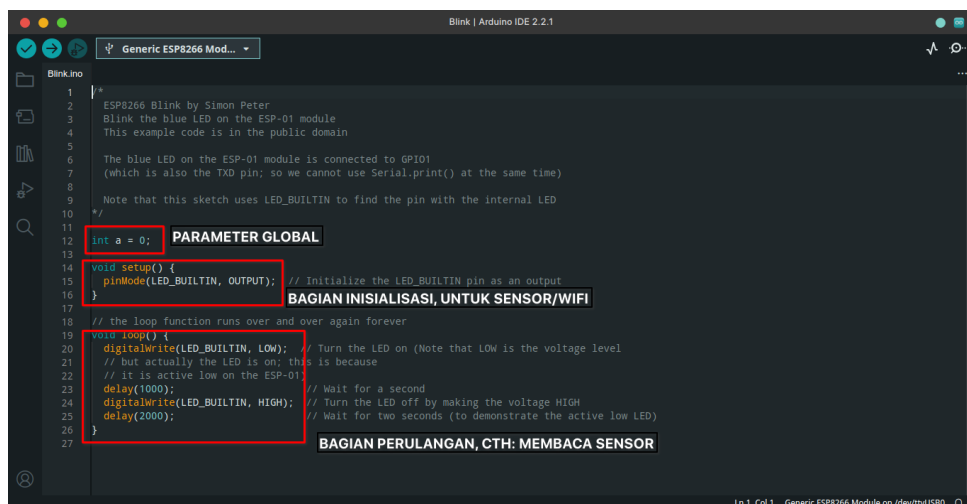


12. **Arduino IDE** akan membuka **Window Baru**. Tutup **Window** sebelumnya agar tidak terganggu.



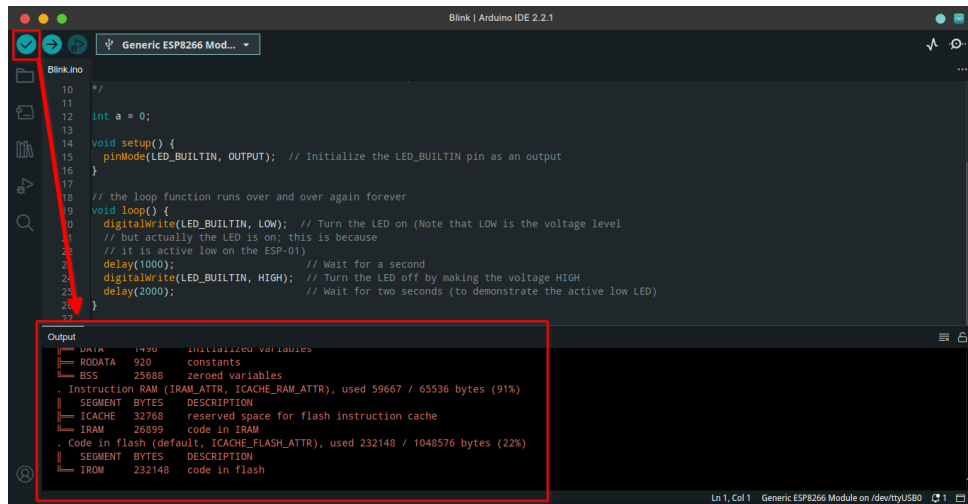
```
1  /*
2  ESP8266 Blink by Simon Peter
3  Blink the blue LED on the ESP-01 module
4  This example code is in the public domain
5
6  The blue LED on the ESP-01 module is connected to GPIO1
7  (which is also the TXD pin, so we cannot use Serial.print() at the same time)
8
9  Note that this sketch uses 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**

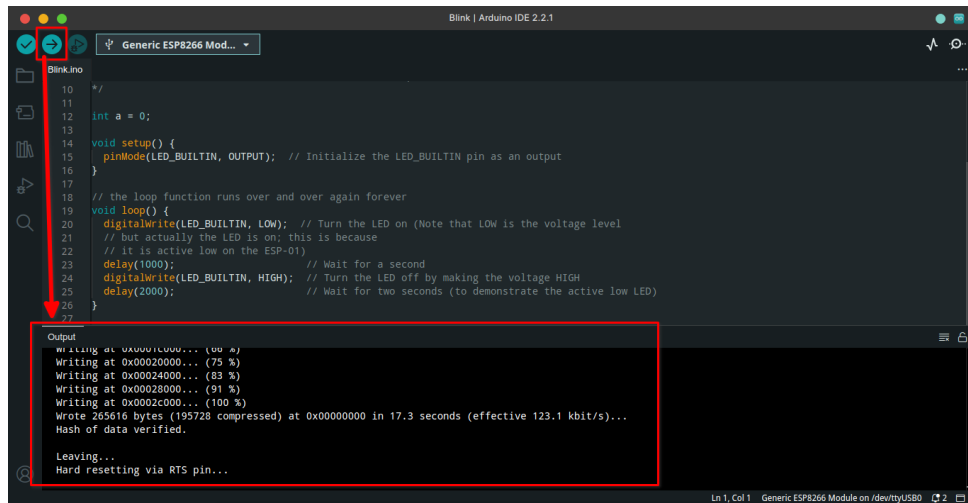


```
1  /*
2  ESP8266 Blink by Simon Peter
3  Blink the blue LED on the ESP-01 module
4  This example code is in the public domain
5
6  The blue LED on the ESP-01 module is connected to GPIO1
7  (which is also the TXD pin, so we cannot use Serial.print() at the same time)
8
9  Note that this sketch uses LED_BUILTIN to find the pin with the internal LED
10 */
11
12 int a = 0; // PARAMETER GLOBAL
13
14 void setup() {
15   pinMode(LED_BUILTIN, OUTPUT); // Initialize the LED_BUILTIN pin as an output
16 } // BAGIAN INISIALISASI, UNTUK SENSOR/WIFI
17
18 // the loop function runs over and over again forever
19 void loop() {
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 } // BAGIAN PERULANGAN, CTH: MEMBACA SENSOR
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

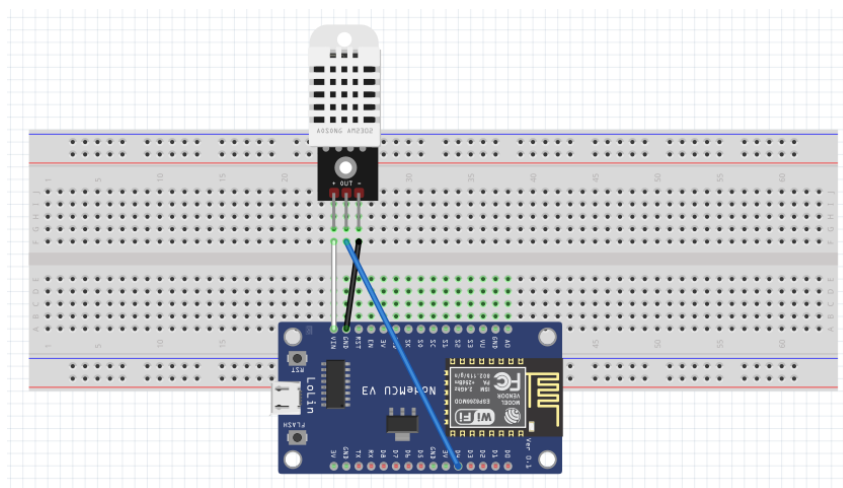
Praktikum 2

2.1 ESP8266, DHT-22, dan AdafruitIO

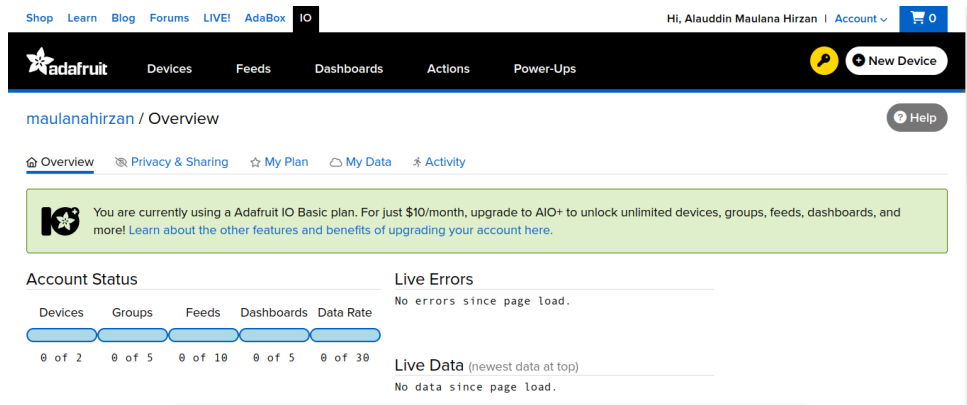
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke sensor DHT-22 dan bagaimana menyimpan data secara daring di layanan AdafruitIO. Mahasiswa diwajibkan memahami **Praktikum 1** yang ada di halaman sebelumnya.

2.2 Tutorial

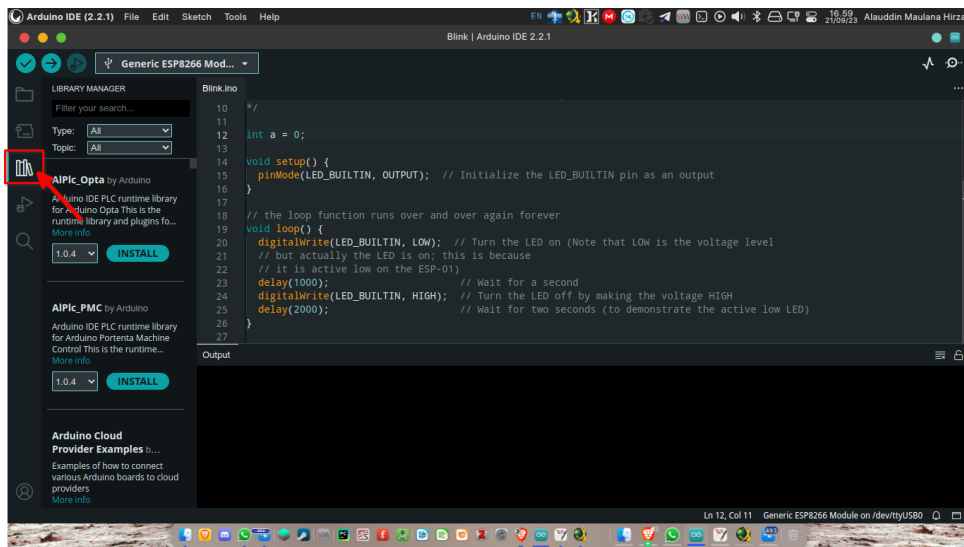
1. Langkah pertama yang perlu dilakukan adalah memasang sensor ke perangkat. Perlu diketahui bahwa dalam memasang sensor harus dalam keadaan **MATI/TI-TAK TERTANCAP** untuk menghindari KORSLETING
2. Perhatikan sensor **DHT-22**, di bagian kakinya ada tanda **Plus +**, **Minus -**, dan **Out**. Sambungkan sesuai dengan indikator **NodeMCU ESP8266** sebagai berikut:
 - **Plus + / VCC** → **3V / 3.3V**
 - **Minus - / GND** → **G**
 - **OUT** → **D4 / GPIO2**



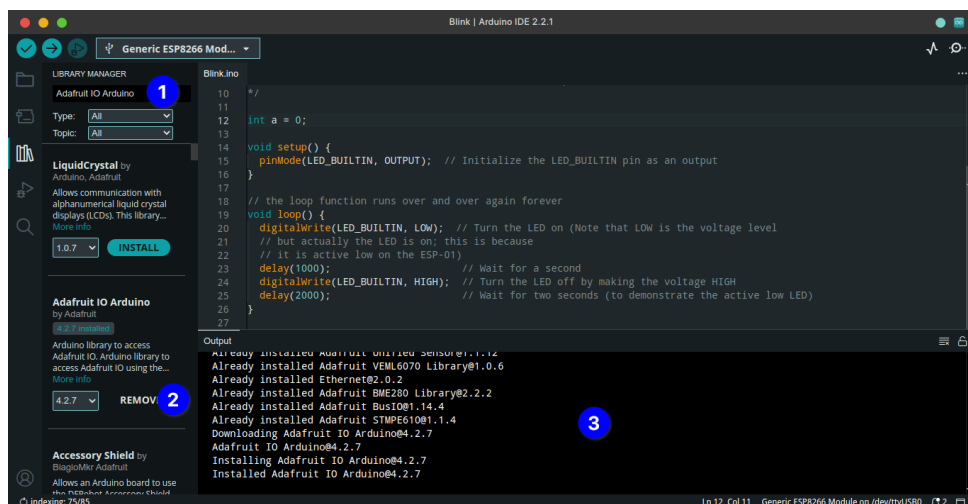
3. Setelah selesai menancapkan sensor, berikutnya adalah melakukan registrasi ke website AdafruitIO dengan link : <https://io.adafruit.com/>. Setelah teregistrasi akan terlihat dasbor seperti berikut:



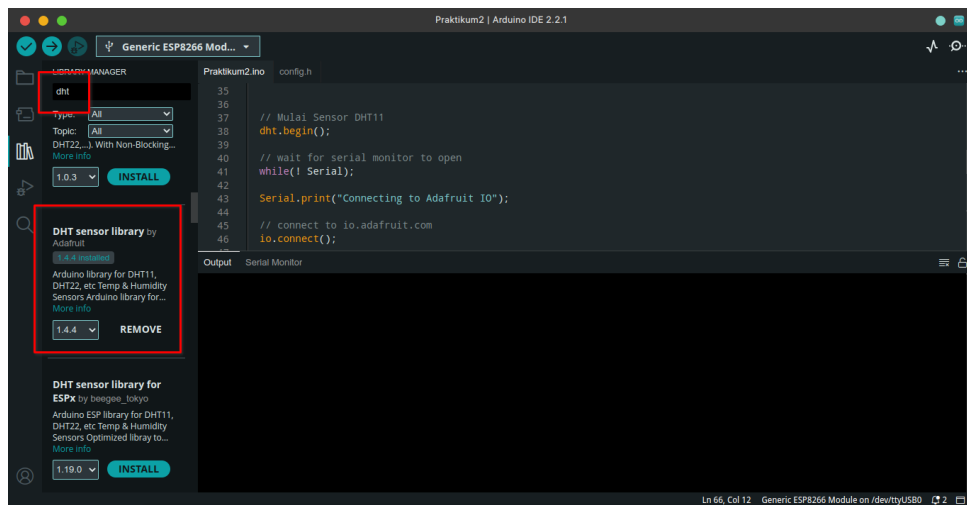
4. Kembali ke **Arduino IDE**, dan install **Library** dengan mengakses menu samping atau **Sketch** → **Include Library** → **Manage Libraries**



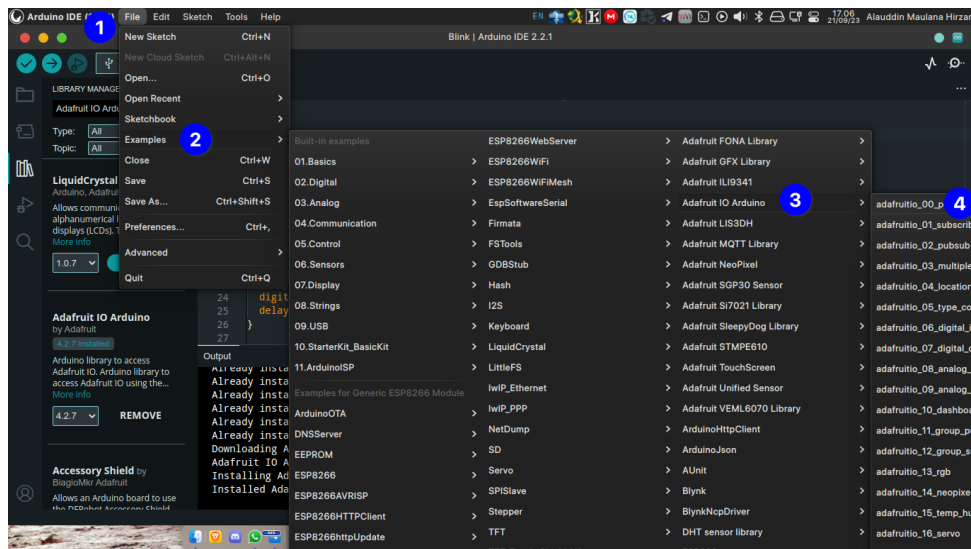
5. Cari Adafruit IO Arduino, klik **INSTALL**, lalu **INSTALL ALL**



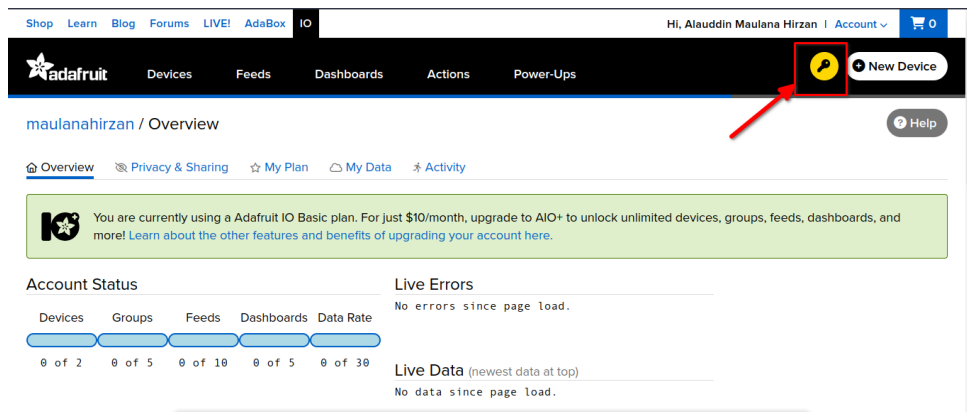
6. Cari **DHT sensor Library**, klik **INSTALL**, lalu **INSTALL ALL**



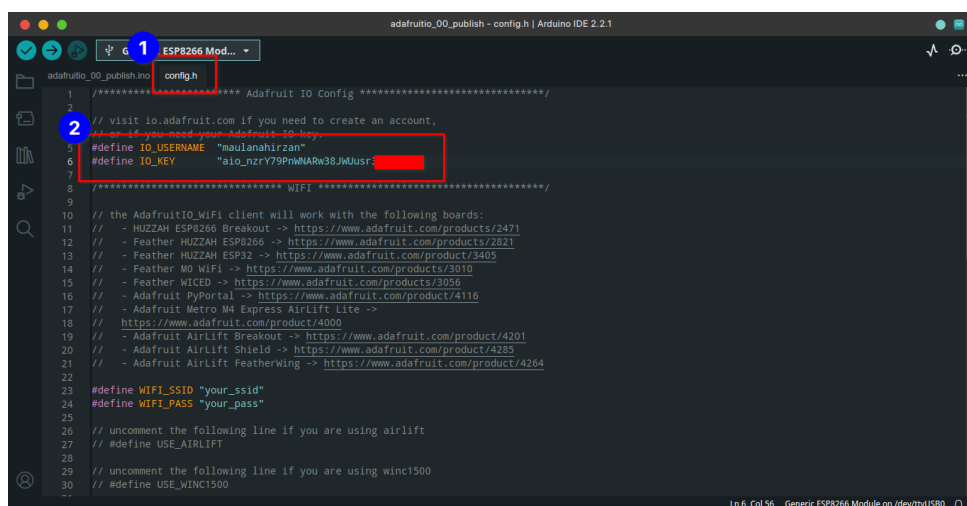
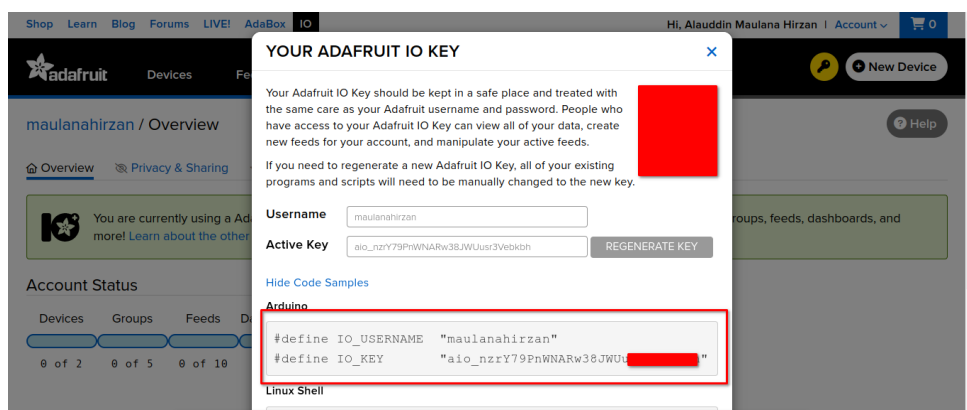
7. Sesudah install, berikutnya adalah membuka **Template Adafuit IO**. Klik menu **File** → **Examples** → **Adafuit IO Arduino** → **adafruit_00_publish**. Tutup **Arduino IDE** lain agar fokus



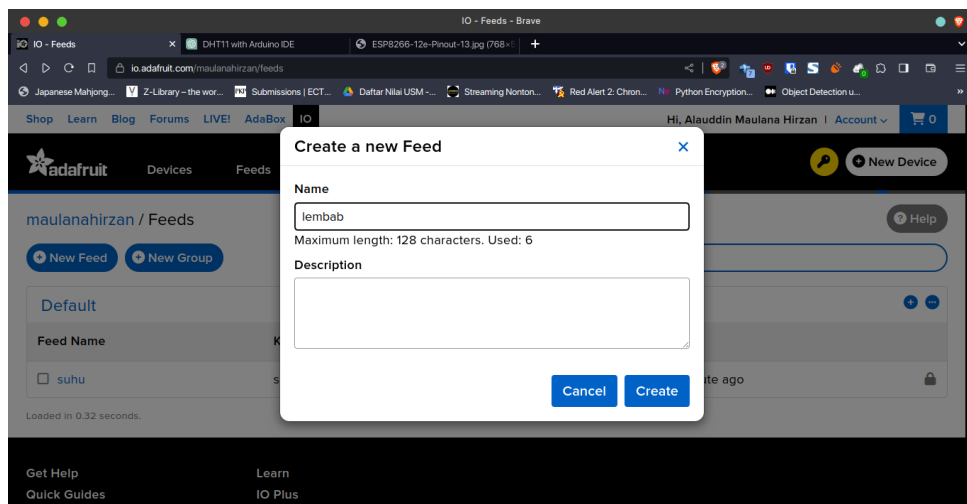
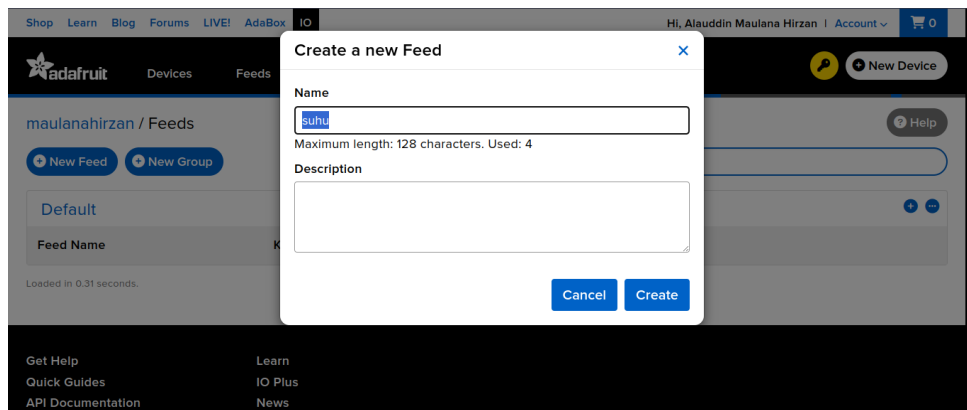
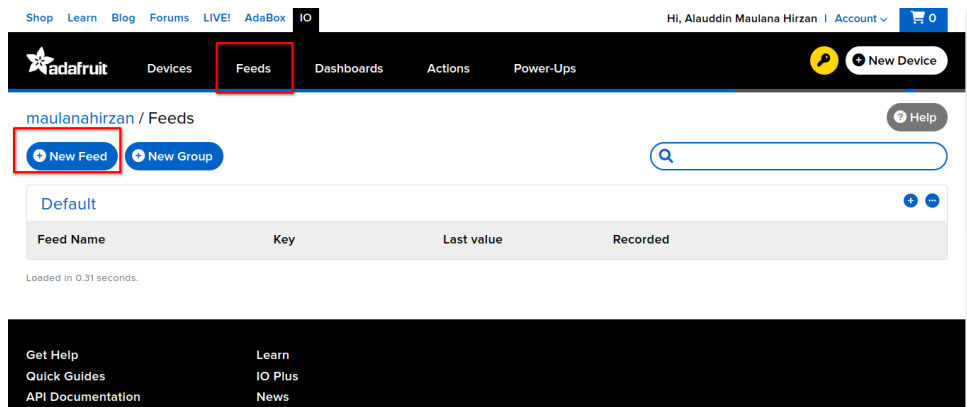
8. Jika sudah terbuka, kembali lagi ke website **Adafuit IO**. Klik **Icon Kunci Kuning** untuk menambahkan perangkat.



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


```

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 winc1500
30 // #define USE_WINC1500
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 // Dont change the names of these #define's! they match the variant ones
43 #define SPIWIFI_SSPIN
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

```

1 // ***** Example Starts Here *****
2
3 // this int will hold the current count for our sketch
4 int count = 0;
5
6 // set up the 'counter' feed
7 AdafruitIO_Feed *counter = io.feed("counter");
8
9 void setup() {
10 // start the serial connection
11 Serial.begin(115200);
12
13 // wait for serial monitor to open
14 while(! Serial);
15
16 Serial.print("Connecting to Adafruit IO");
17
18 // connect to io.adafruit.com
19 io.connect();
20
21 // wait for a connection
22 while(io.status() < AIO_CONNECTED) {
23   Serial.print(".");
24   delay(500);
25 }
26
27 // we are connected
28 Serial.println();
29 Serial.println(io.statusText());

```

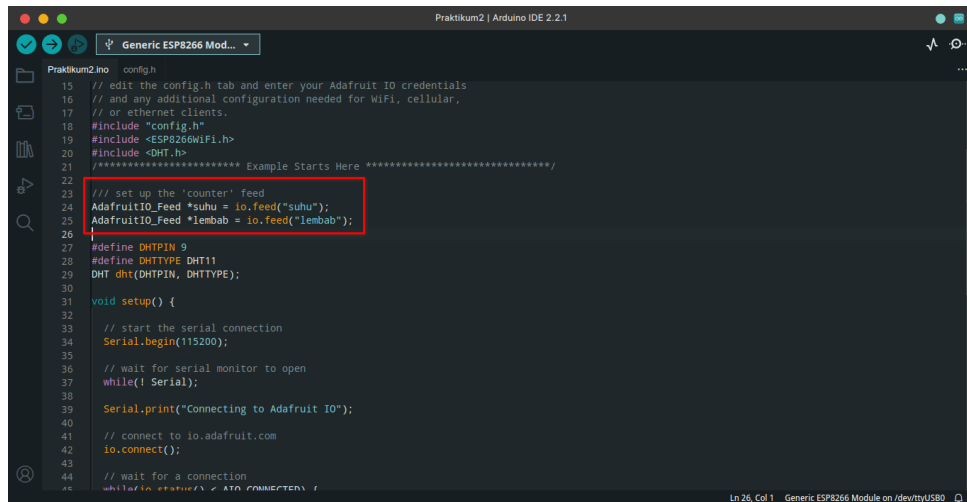
14. Ubah kode **AdafruitIO_Feed *counter = io.feed("counter");** menjadi

Potongan Kode

```

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

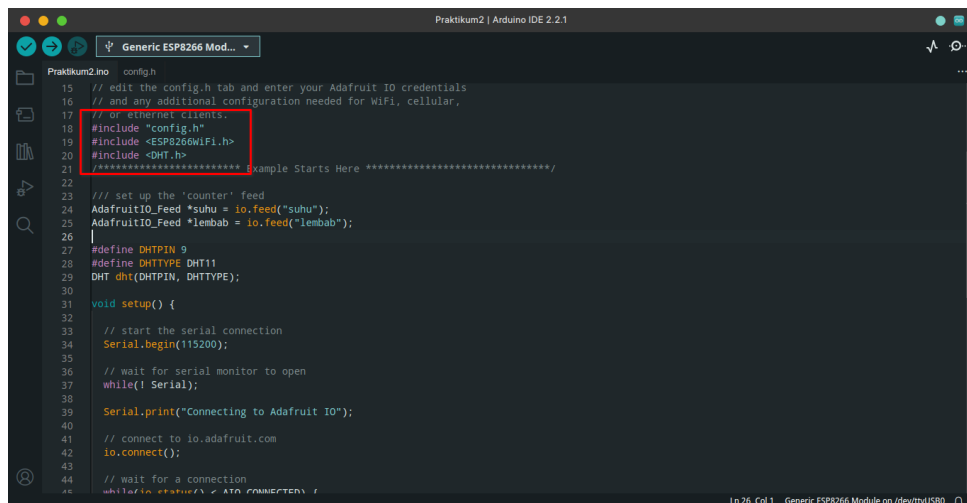
```



15. Berikutnya adalah mengkonfigurasi kode untuk ESP8266 dan DHT-22, tambahkan kode berikut tepat di bawah `#include "config.h"`

Potongan Kode

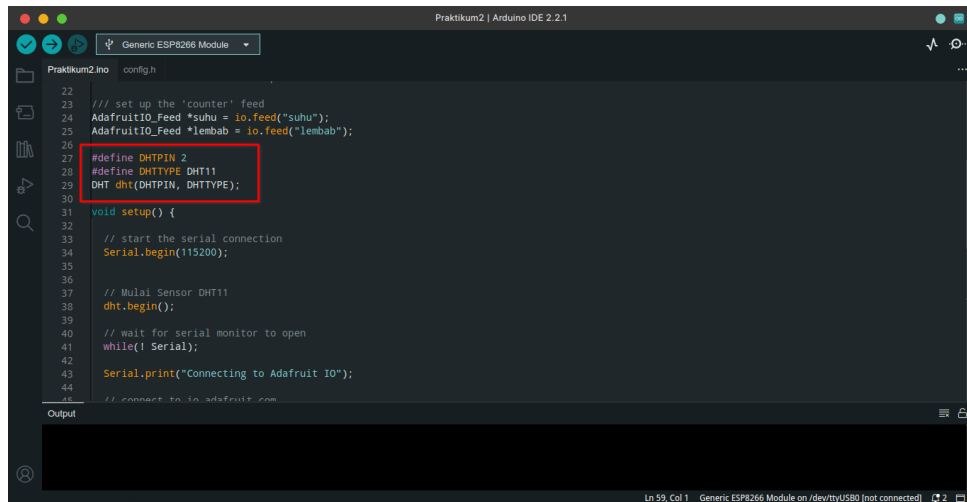
```
#include <ESP8266WiFi.h>
#include <DHT.h>
```



16. Lalu tambahkan kode definisi untuk jenis sensor DHT-22. 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 DHT22
DHT dht(DHTPIN, DHTTYPE);
```



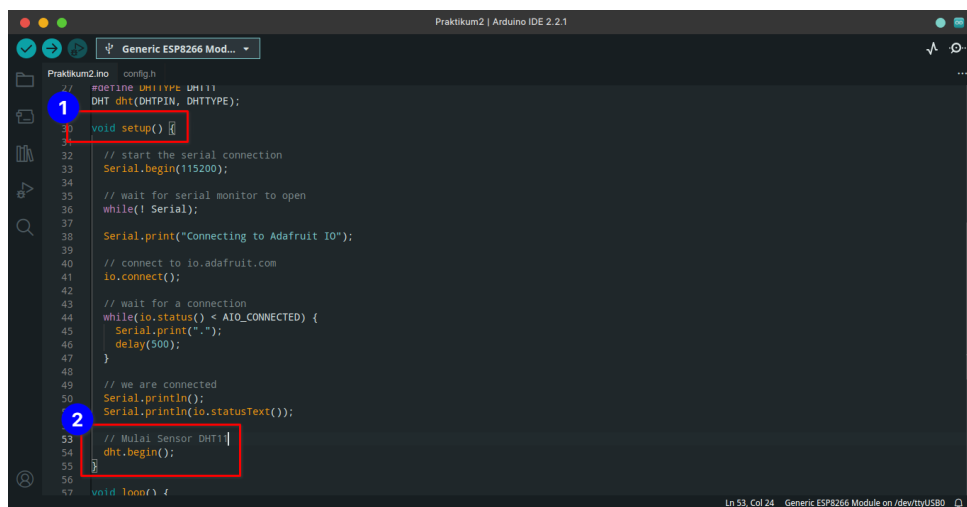
17. Parameter global sudah diset. Berikutnya adalah mengatur fungsi **setup** untuk sensor **dht**. Tambahkan kode berikut di bagian akhir fungsi **setup** (BUKAN AKHIR FILE)

Potongan Kode

```

// Mulai Sensor DHT-22
dht.begin();

```



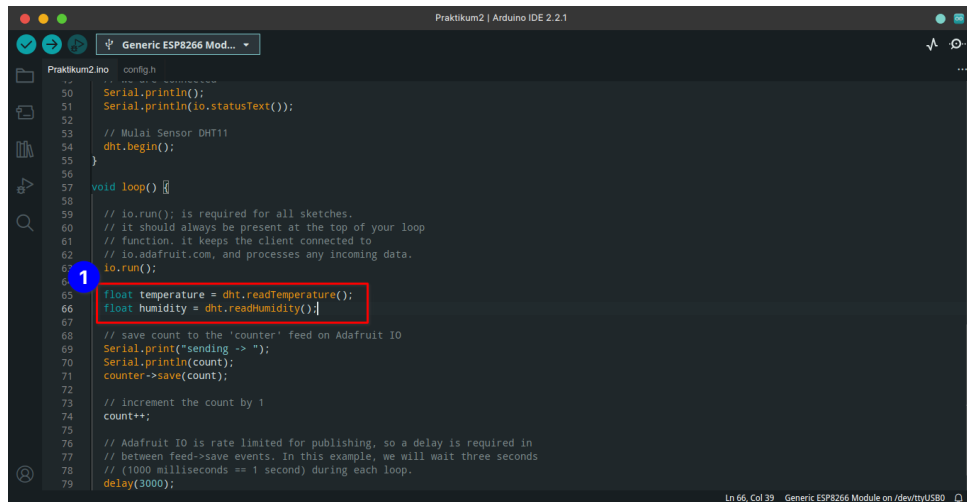
18. Lalu tambahkan kode ke fungsi **loop** untuk membaca suhu dan kelembaban. Letakkan di bawa **io.run()**

Potongan Kode

```

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

```



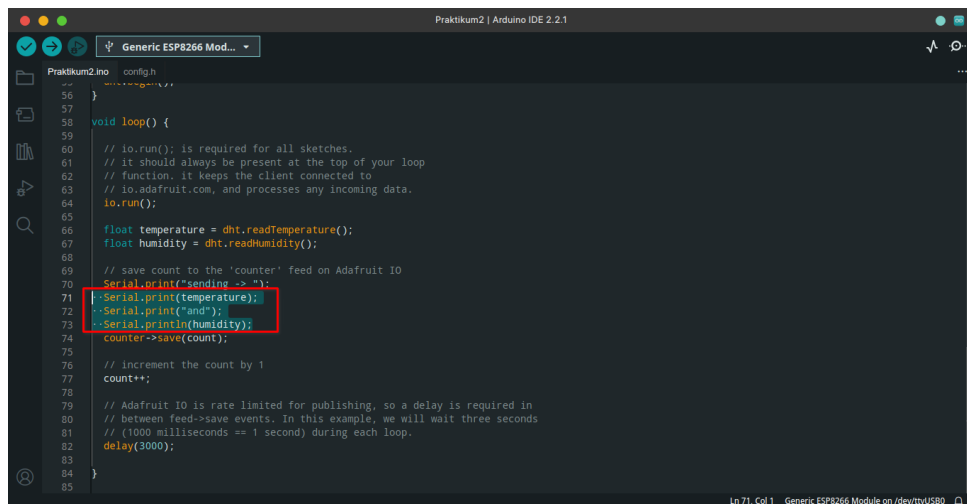
19. Setelah itu ubah kode **Serial.println(count);** dengan kode berikut:

Potongan Kode

```

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

```



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

Potongan Kode

```

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

```

```

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++**;

```

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

```

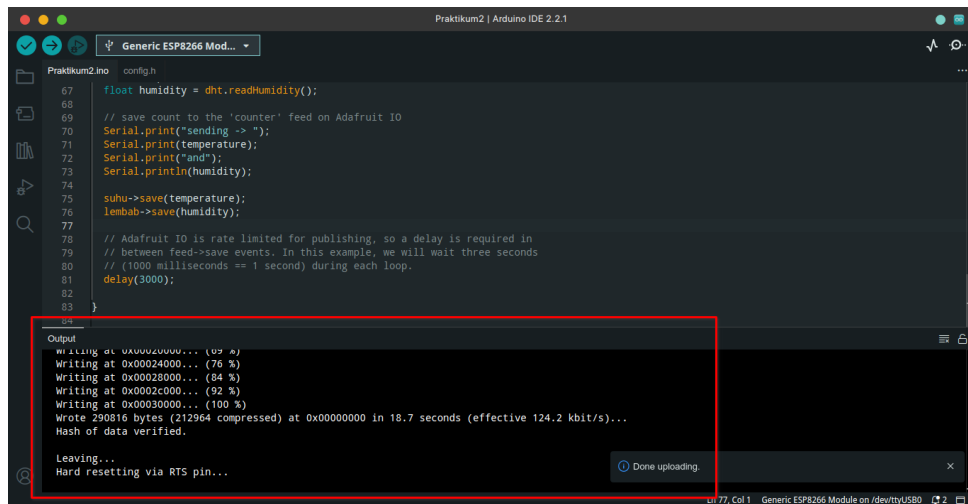
67 float humidity = dht.readHumidity();
68
69 // save count to the 'counter' feed on Adafruit IO
70 Serial.print("sending -> ");
71 Serial.print(temperature);
72 Serial.print("and");
73 Serial.println(humidity);
74
75 suhu->save(temperature);
76 lembab->save(humidity);
77
78 // Adafruit IO is rate limited 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

```

Output

SEGMENT	BYTES	DESCRIPTION
DATA	1304	initialized variables
RODATA	1312	constants
BSS	25944	zeroed variables
. Instruction RAM (IRAM_ATTR, Icache_RAM_ATTR), used 60331 / 65536 bytes (92%)		
. reserved space for flash instruction cache		
IRAM	27563	code in IRAM
. Code in flash (default, Icache_FLASH_ATTR), used 256292 / 1048576 bytes (24%)		
. code in flash		

23. Unggah sudah sukses



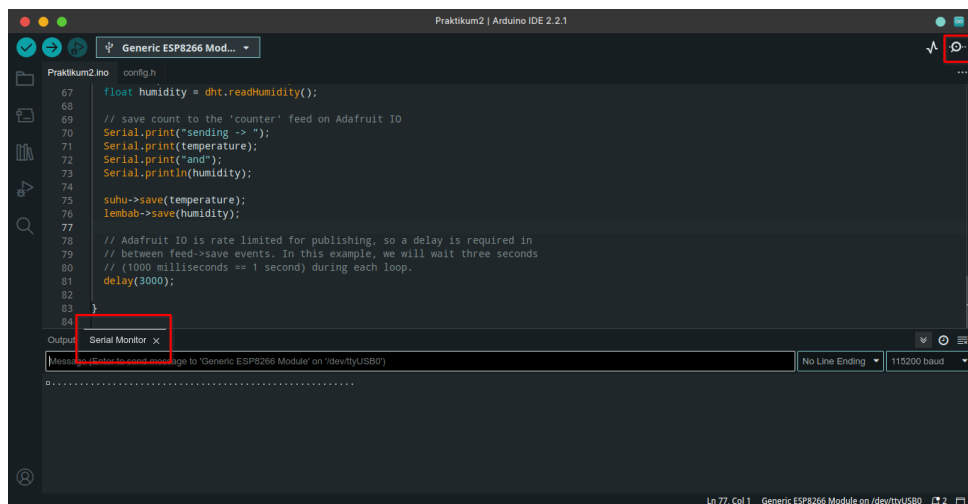
The screenshot shows the Arduino IDE interface with the code for 'Praktikum2.ino' loaded. The code includes comments about saving data to Adafruit IO and a 3-second delay between uploads. The 'Output' window at the bottom shows the upload progress, indicating that the code was successfully written to the module and the hash was verified. A red box highlights the 'Output' window.

```
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 }
```

Output

Writing at 0x00020000... (69 %)
Writing at 0x00024000... (76 %)
Writing at 0x00028000... (84 %)
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 290816 bytes (212964 compressed) at 0x00000000 in 18.7 seconds (effective 124.2 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
Done uploading.

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



The screenshot shows the Arduino IDE interface with the 'Serial Monitor' window open. The 'Tools' menu is highlighted, and the 'Serial Monitor' option is selected. The 'BAUD' rate is set to 115200. A red box highlights the 'Serial Monitor' window.

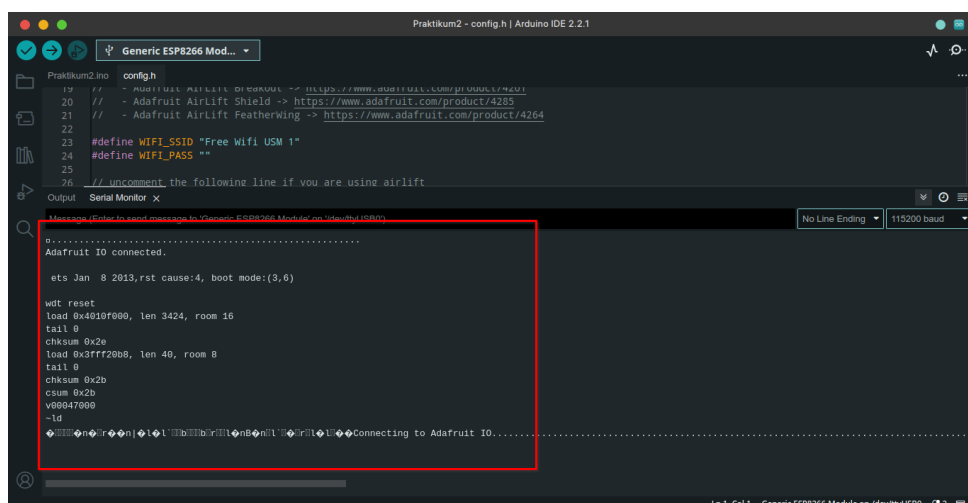
```
Praktikum2.ino config.h
67 float humidity = dht.readHumidity();
68
69 // save count to the 'counter' feed on Adafruit IO
70 Serial.print("sending -> ");
71 Serial.print(temperature);
72 Serial.print("and");
73 Serial.println(humidity);
74
75 suhu->save(temperature);
76 lembab->save(humidity);
77
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 }
```

Serial Monitor

Message (Enter to send message to 'Generic ESP8266 Module' on '/dev/ttyUSB0')

No Line Ending 115200 baud

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



The screenshot shows the Arduino IDE interface with the 'Serial Monitor' window open. The 'Tools' menu is highlighted, and the 'Serial Monitor' option is selected. The 'BAUD' rate is set to 115200. The 'Serial Monitor' window shows the output of the code, including the WiFi connection status. A red box highlights the 'Serial Monitor' window.

```
Praktikum2 - config.h | Arduino IDE 2.2.1
config.h
19 // - Adafruit Airlift FeatherWing -> https://www.adafruit.com/product/4264
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
```

Serial Monitor

Message (Enter to send message to 'Generic ESP8266 Module' on '/dev/ttyUSB0')

No Line Ending 115200 baud

ets Jan 8 2013, rst cause:4, boot mode:(3,0)

wdt reset

load 0x4010f000, len 3424, room 16

tail 0

chksum 0x2e

load 0x3ff720b8, len 40, room 8

tail 0

chksum 0x2b

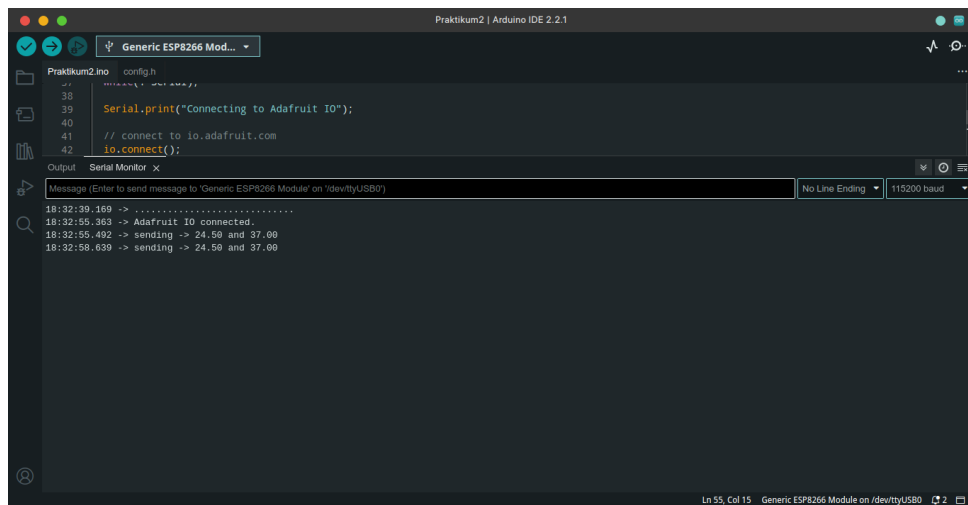
csum 0x2b

v00047000

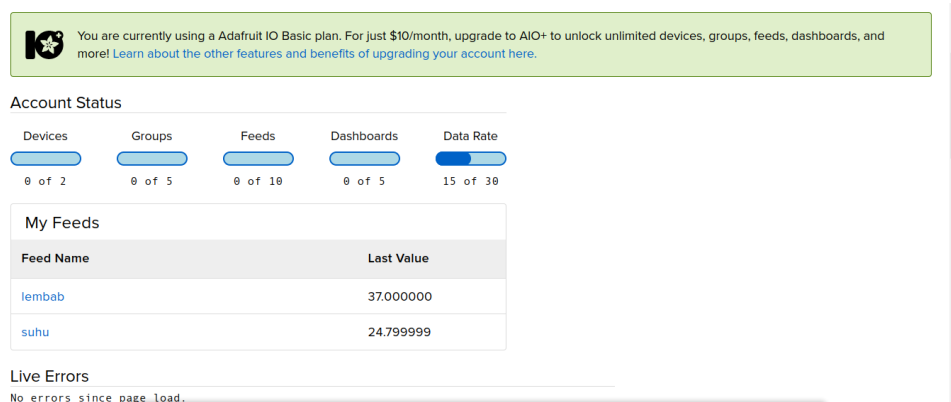
-ld

Connecting to Adafruit IO.....

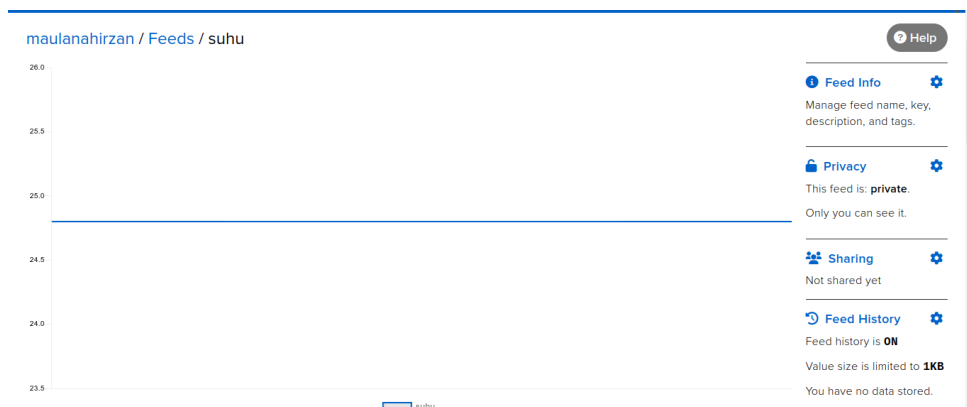
26. Alat terhubung dan berhasil mengirimkan data



27. Hasil di website **Adafruit IO**



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





Potongan Kode

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

```

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

```

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

Bab 3

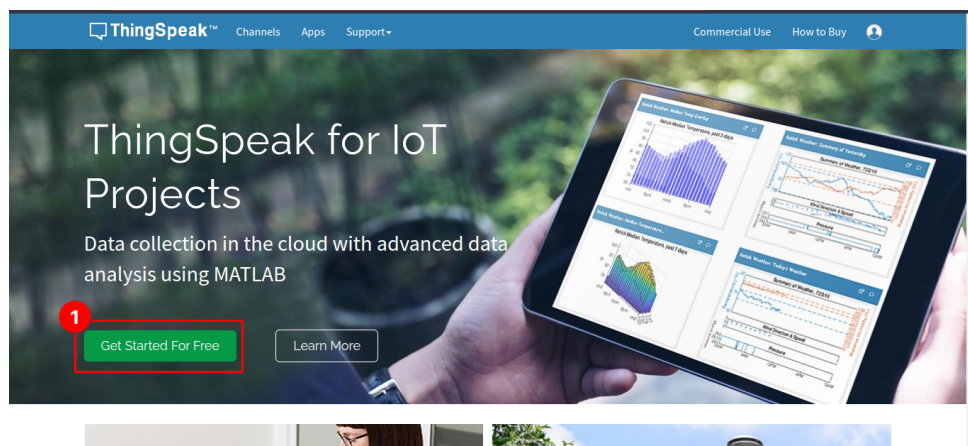
Praktikum 3

3.1 ESP8266, DHT-22, dan Thingspeak

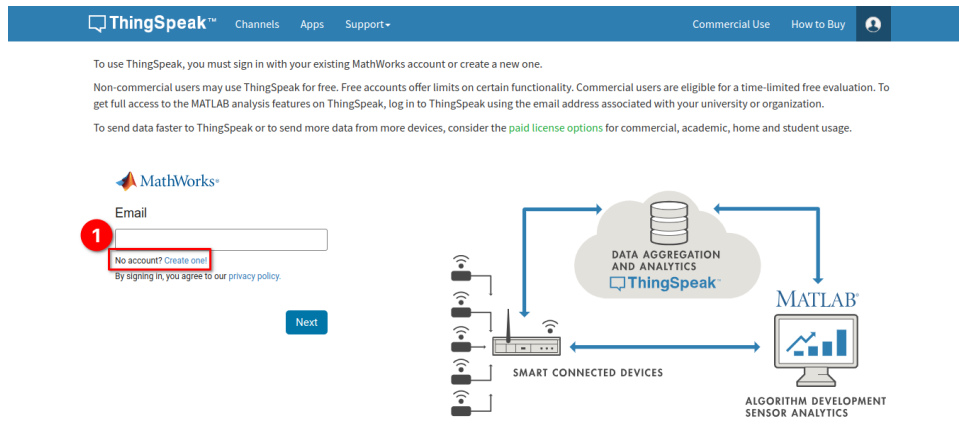
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Thingspeak. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 2** yang ada di halaman sebelumnya.

3.2 Tutorial

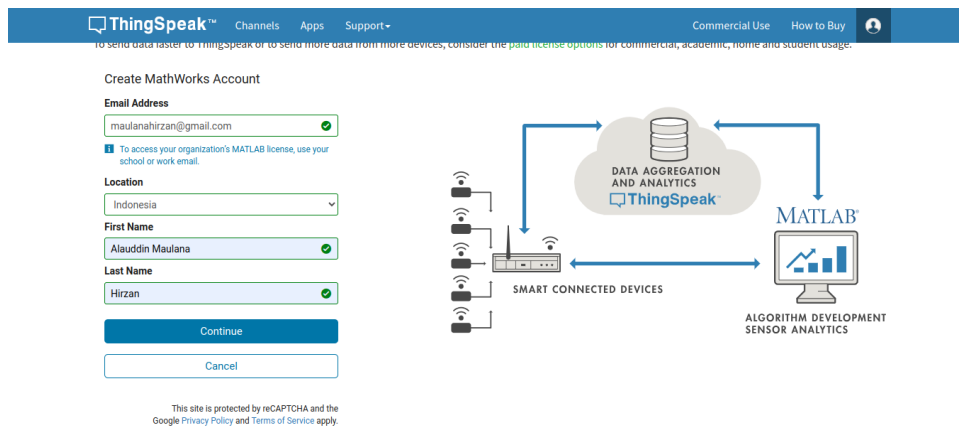
1. Untuk memulai praktikum ini, mahasiswa diwajibkan untuk membuat akun di <https://thingspeak.com/> secara gratis. Klik **Get started for free**



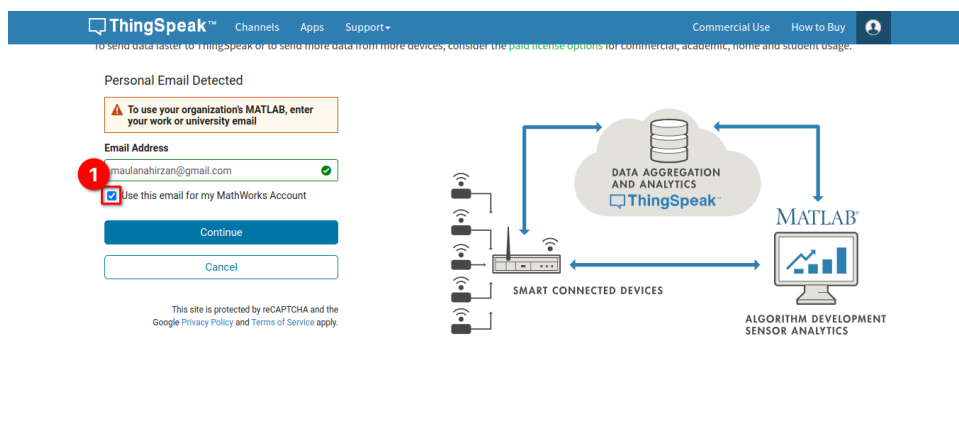
2. Klik **Create one!**



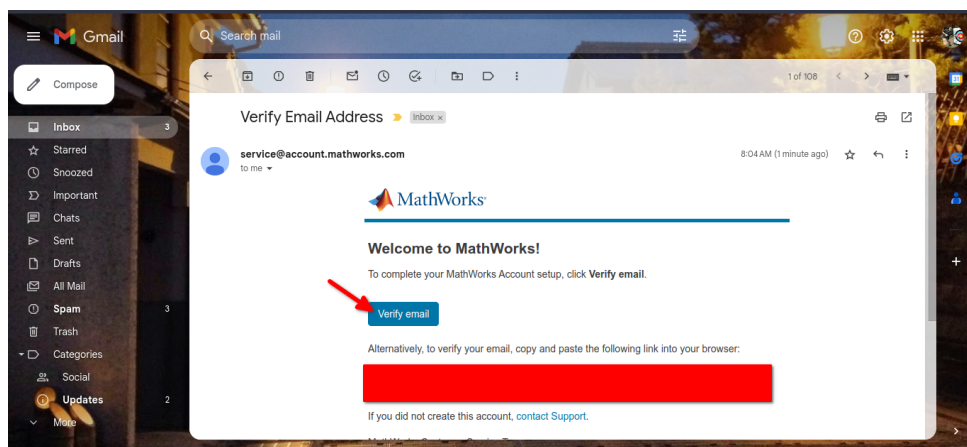
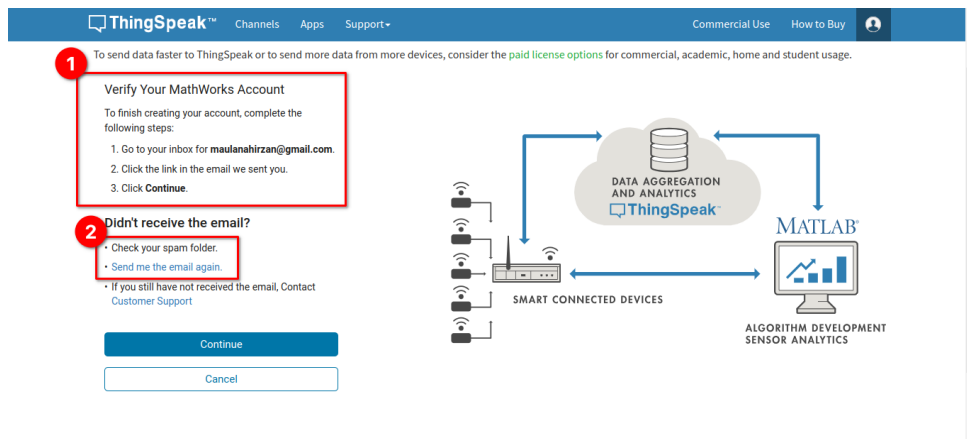
3. Isi informasi identitas



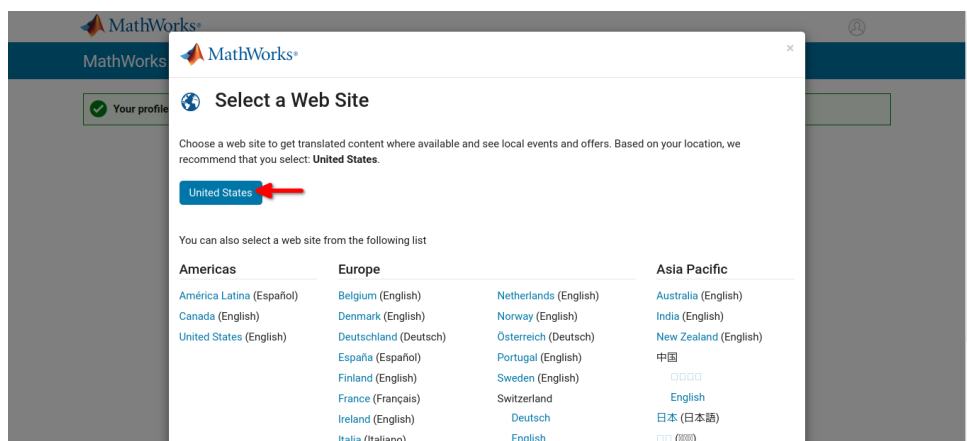
4. Centang untuk menggunakan email pribadi



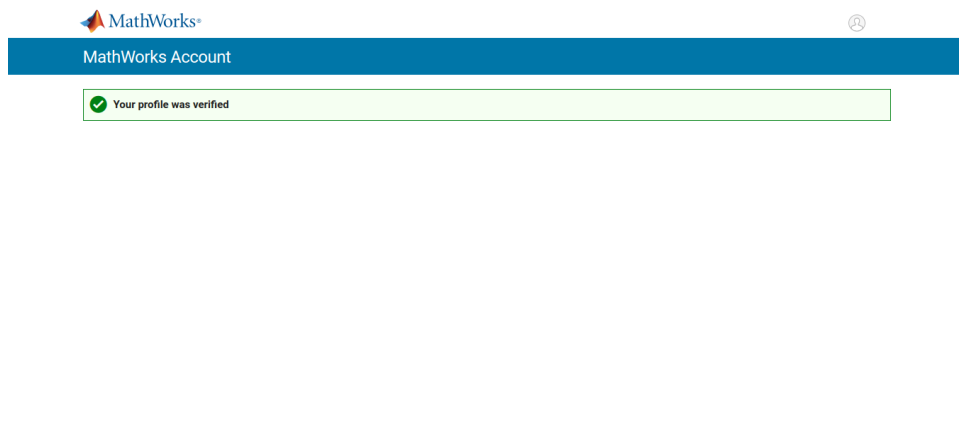
5. Cek email anda (termasuk SPAM) untuk verifikasi email. **JANGAN TUTUP WINDOW INI!!!**



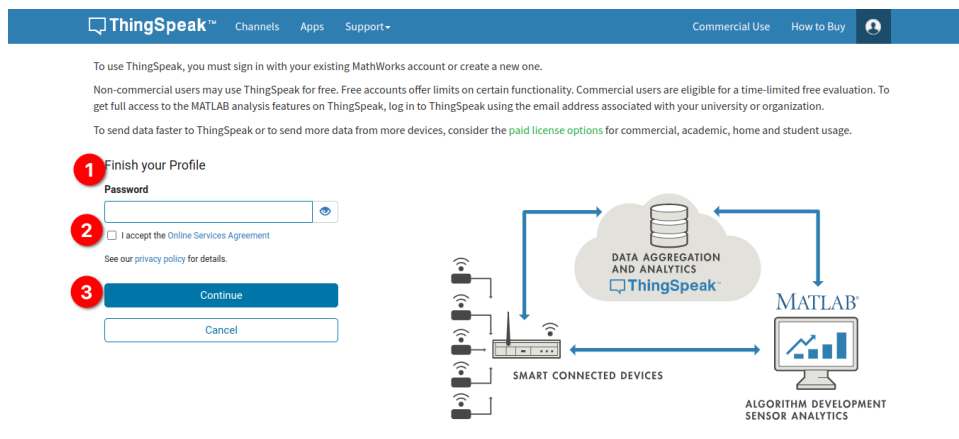
6. Pilih negara untuk website



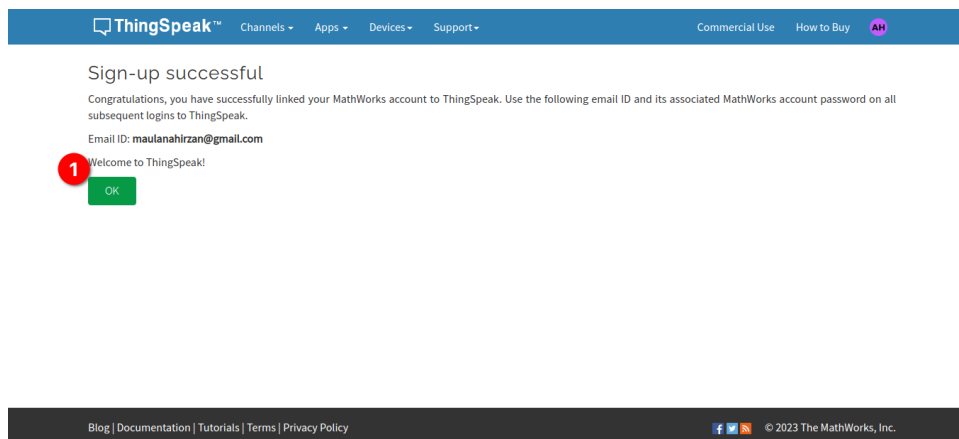
7. Akun sudah terverifikasi



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**

ThingSpeak Usage Intent

How are you planning to use ThingSpeak?

☐ Commercial work (including research)
☐ Government work (including research)
☐ Personal, non-commercial projects
☒ Student use, Teaching, or Research in academia

What is the name of your University?*

Semarang University

What best describes your current role?*

☒ Student
 ☐ Professor
 ☐ Researcher

What is the name of your Course or Project?*

Internet of Things

Tell us something about your project (optional)

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

My Channels

New Channel

Search by tag

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to [create channels](#), explore and transform data.

Learn more about [ThingSpeak Channels](#).

Examples

- Arduino
- Arduino MKR1000
- ESP8266
- Raspberry Pi
- Netduino Plus

Upgrade

Need to send more data faster?

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

New Channel

1 Name ESP8266

Description

2 Field 1 Suhu ☒

Field 2 Kelembaban ☒

Field 3 ☐

Field 4 ☐

Field 5 ☐

Field 6 ☐

Field 7 ☐

Field 8 ☐

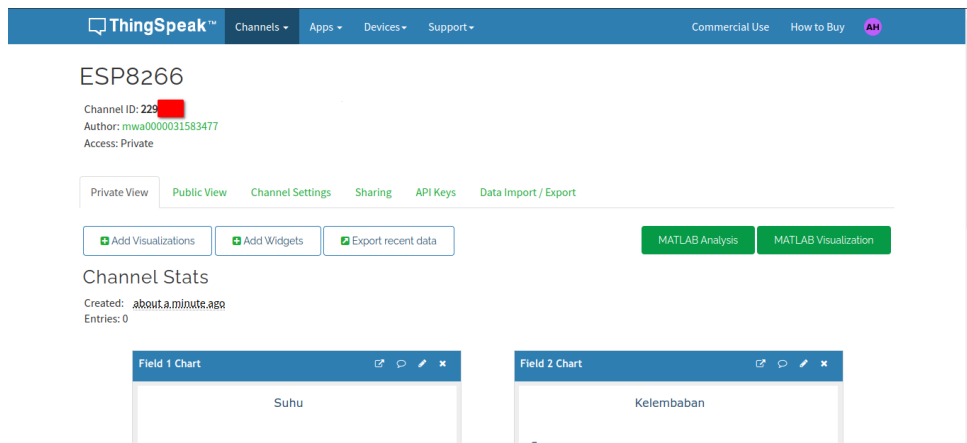
Help

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

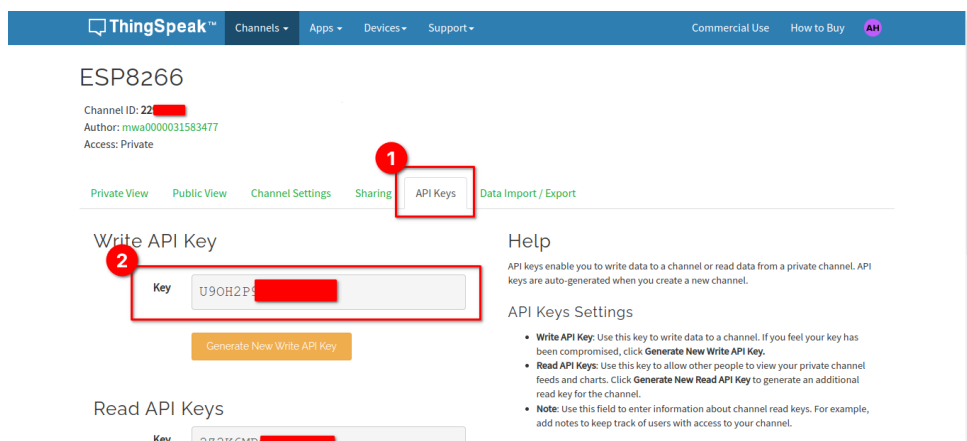
Channel Settings

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

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

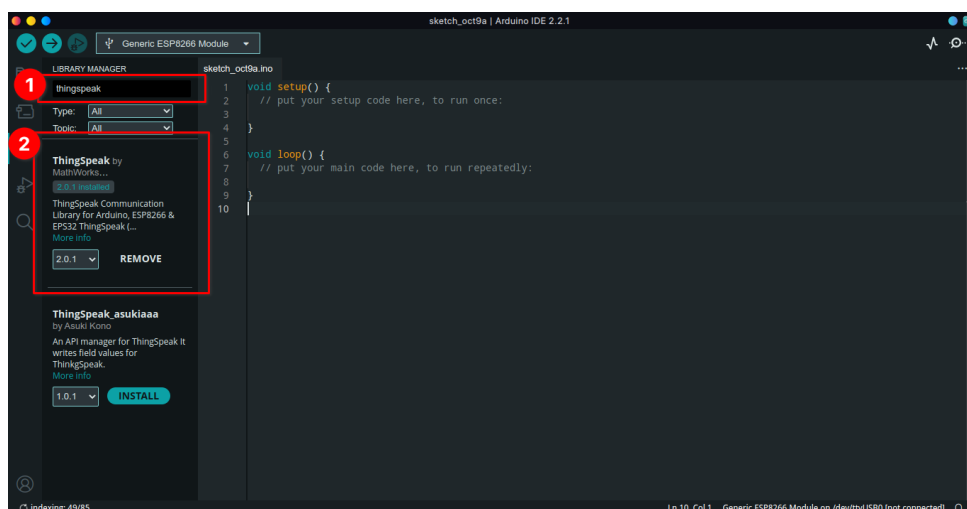


14. Pindah ke tab **API Keys**, dan kopi **Write API Key** untuk **Arduino IDE**



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

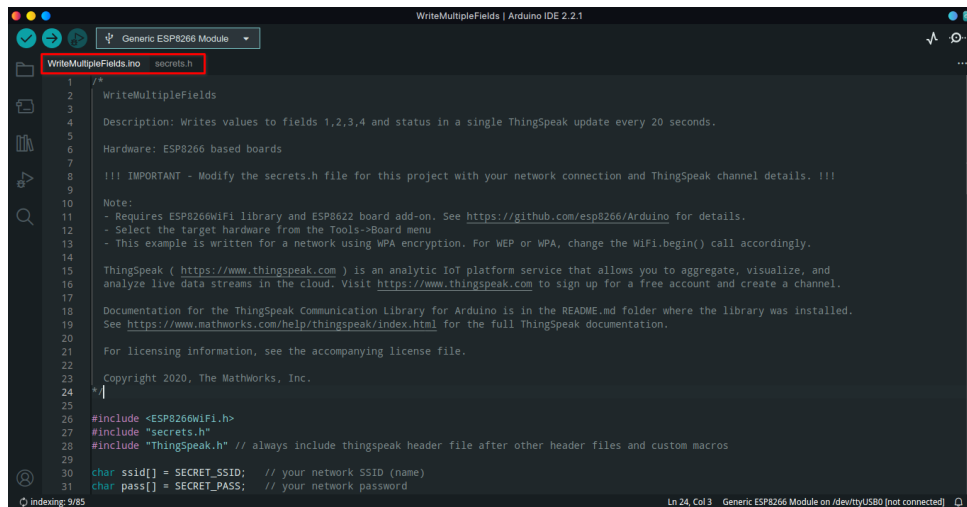
16. **Install Library Thingspeak**



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

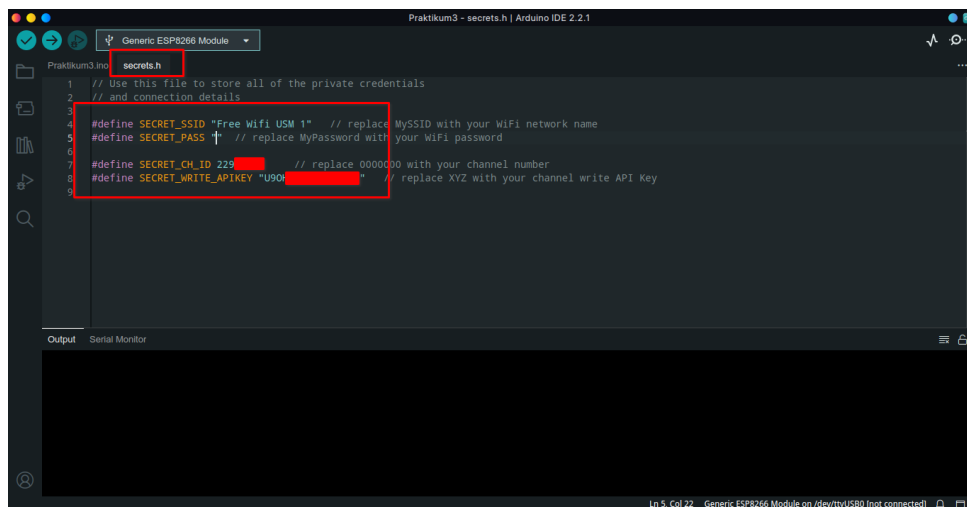
yang sudah disiapkan oleh **Library**. Klik **File** → **Examples** → **ThingSpeak** → **ESP8266** → **Program Board Directly** → **Write Multiple Fields**

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



```
1  /*
2  WriteMultipleFields
3  Description: Writes values to fields 1,2,3,4 and status in a single ThingSpeak update every 20 seconds.
4
5  Hardware: ESP8266 based boards
6
7  !!! IMPORTANT - Modify the secrets.h file for this project with your network connection and ThingSpeak channel details. !!!
8
9  Note:
10 - Requires ESP8266WiFi library and ESP8262 board add-on. See https://github.com/esp8266/Arduino for details.
11 - Select the target hardware from the Tools->Board menu
12 - This example is written for a network using WPA encryption. For WEP or WPA, change the WiFi.begin() call accordingly.
13
14 ThingSpeak ( https://www.thingspeak.com ) is an analytic IoT platform service that allows you to aggregate, visualize, and
15 analyze live data streams in the cloud. Visit https://www.thingspeak.com to sign up for a free account and create a channel.
16
17 Documentation for the ThingSpeak Communication Library for Arduino is in the README.md folder where the library was installed.
18 See https://www.mathworks.com/help/thingspeak/index.html for the full ThingSpeak documentation.
19
20 For licensing information, see the accompanying license file.
21
22 Copyright 2020, The MathWorks, Inc.
23 */
24
25 #include <ESP8266WiFi.h>
26 #include "secrets.h"
27 #include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros
28
29 char ssid[] = SECRET_SSID; // your network SSID (name)
30 char pass[] = SECRET_PASS; // your network password
```

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

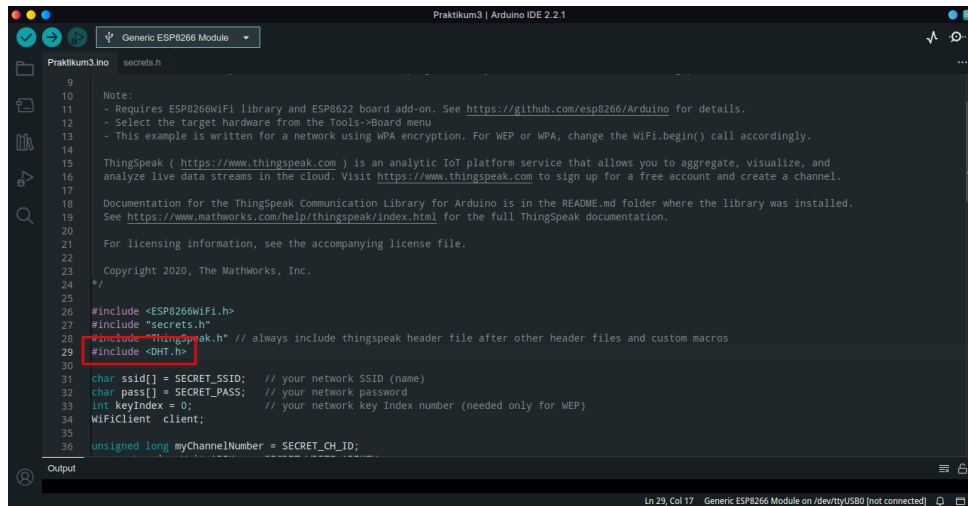


```
1 // Use this file to store all of the private credentials
2 // and connection details
3
4 #define SECRET_SSID "Free Wifi USM 1" // replace MySSID with your WiFi network name
5 #define SECRET_PASS "1" // replace MyPassword with your WiFi password
6
7 #define SECRET_CH_ID 229 // replace 00000000 with your channel number
8 #define SECRET_WRITE_APIKEY "U90K" // replace XYZ with your channel write API Key
9
```

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

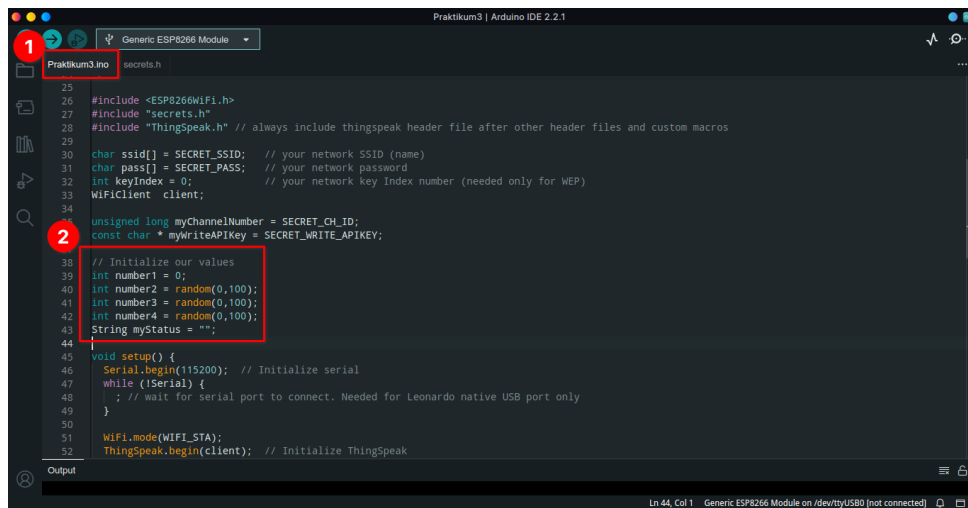
Potongan Kode

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



```
9
10
11 Note:
12 - Requires ESP8266WiFi library and ESP8222 board add-on. See https://github.com/esp8266/Arduino for details.
13 - Select the target hardware from the tools->Board menu
14 - This example is written for a network using WPA encryption. For WEP or WPA, change the WiFi.begin() call accordingly.
15
16 ThingSpeak ( https://www.thingspeak.com ) is an analytic IoT platform service that allows you to aggregate, visualize, and
17 analyze live data streams in the cloud. Visit https://www.thingspeak.com to sign up for a free account and create a channel.
18
19 Documentation for the ThingSpeak Communication Library for Arduino is in the README.md folder where the library was installed.
20 See https://www.mathworks.com/help/thingspeak/index.html for the full ThingSpeak documentation.
21
22 For licensing information, see the accompanying license file.
23
24 Copyright 2020, The MathWorks, Inc.
25
26 */
27 #include <ESP8266WiFi.h>
28 #include "secrets.h"
29 #include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros
30 #include <DHT.h>
31
32 char ssid[] = SECRET_SSID; // your network SSID (name)
33 char pass[] = SECRET_PASS; // your network password
34 int keyIndex = 0; // your network key index number (needed only for WEP)
35 WiFiClient client;
36 unsigned long myChannelNumber = SECRET_CH_ID;
```

21. Hapus kode berikut

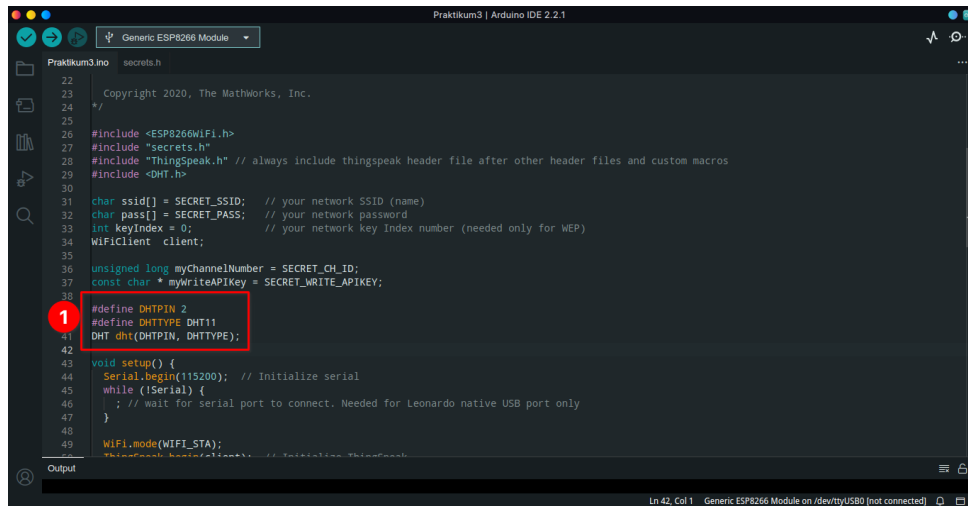


```
25
26 #include <ESP8266WiFi.h>
27 #include "secrets.h"
28 #include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros
29
30 char ssid[] = SECRET_SSID; // your network SSID (name)
31 char pass[] = SECRET_PASS; // your network password
32 int keyIndex = 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 DHT22
DHT dht(DHTPIN, DHTTYPE);
```

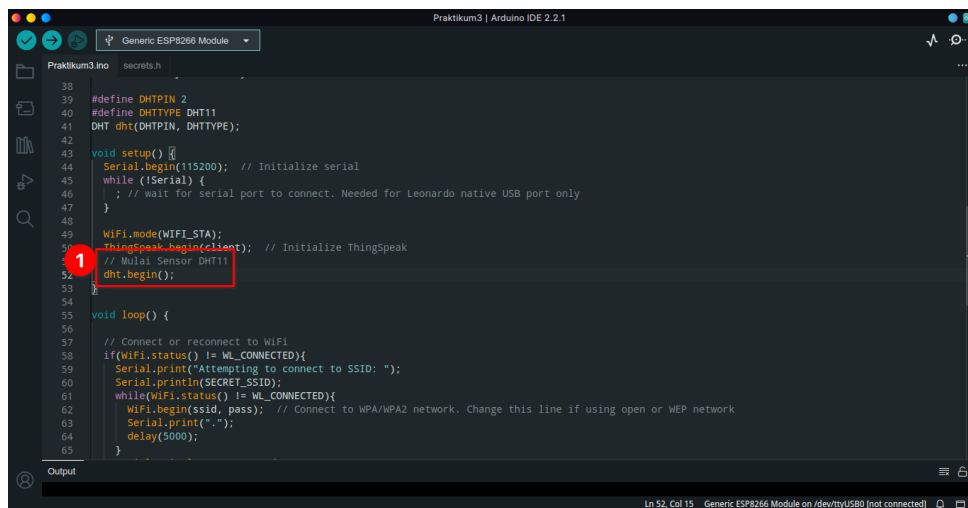



```
Praktikum3.ino secrets.h
22 Copyright 2020, The Mathworks, Inc.
23 */
24
25
26 #include <ESP8266WiFi.h>
27 #include "secrets.h"
28 #include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros
29 #include <DHT.h>
30
31 char ssid[] = SECRET_SSID; // your 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 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 }
```

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

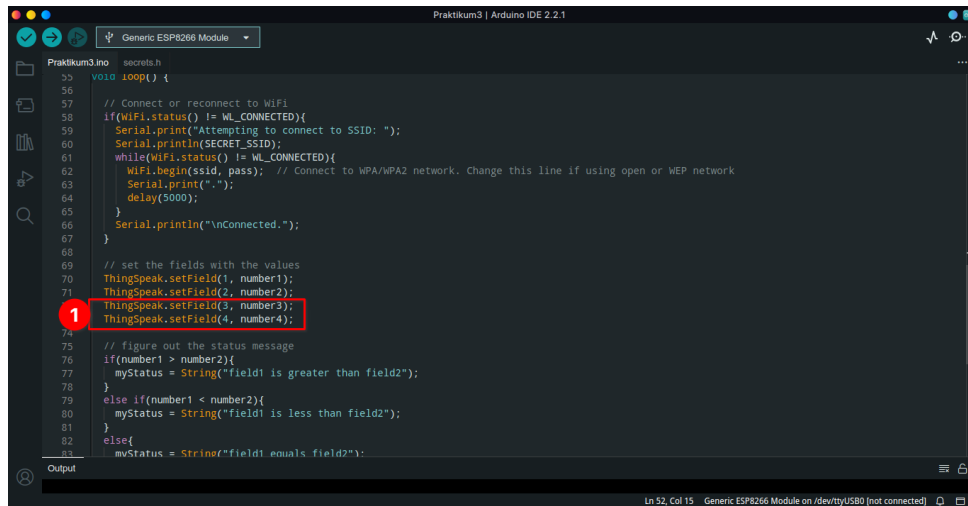
Potongan Kode

```
// Mulai Sensor DHT-22
dht.begin();
```

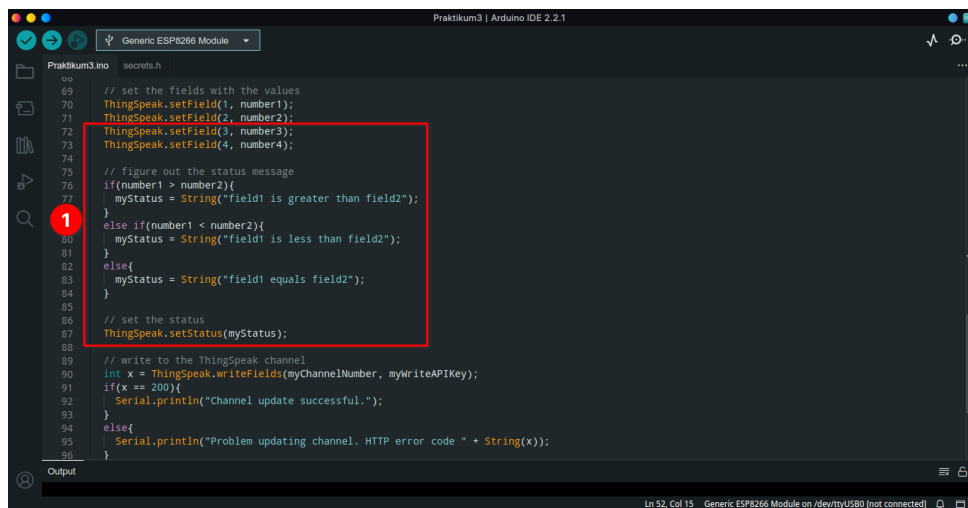


```
Praktikum3.ino secrets.h
38 #define DHTPIN 2
39 #define DHTTYPE DHT11
40 DHT dht(DHTPIN, DHTTYPE);
41
42 void setup() {
43   Serial.begin(115200); // Initialize serial
44   while (!Serial) {
45     ; // wait for serial port to connect. Needed for Leonardo native USB port only
46   }
47
48   WiFi.mode(WIFI_STA);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

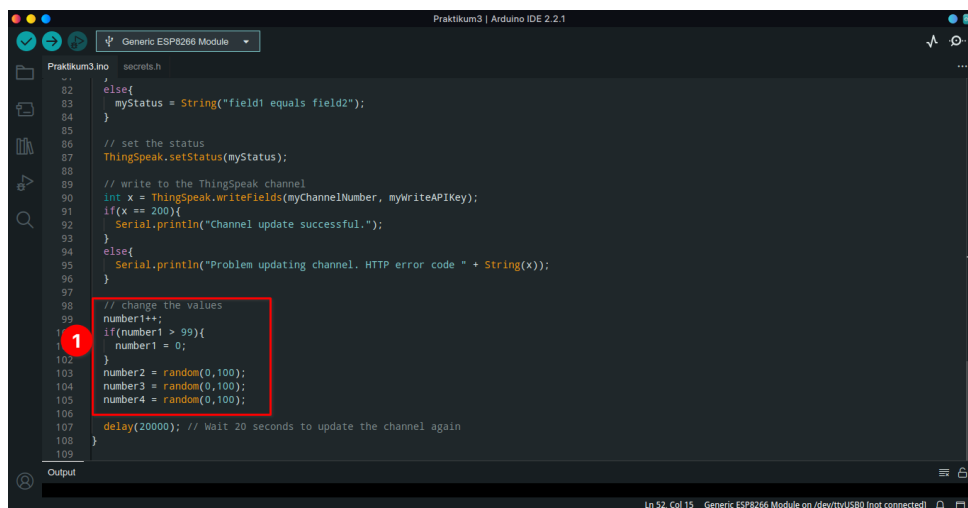
24. Di dalam **FUNGSI LOOP** Hapus kode berikut:



```
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 }
```



```
56 // set the fields with the values
57 ThingSpeak.setField(1, number1);
58 ThingSpeak.setField(2, number2);
59 ThingSpeak.setField(3, number3);
60 ThingSpeak.setField(4, number4);
61
62 // figure out the status message
63 if(number1 > number2){
64   myStatus = String("field1 is greater than field2");
65 }
66 else if(number1 < number2){
67   myStatus = String("field1 is less than field2");
68 }
69 else{
70   myStatus = String("field1 equals field2");
71 }
72
73 // set the status
74 ThingSpeak.setStatus(myStatus);
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 }
84 }
```



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

25. Hasil AKHIR SEHARUSNYA:

The screenshot shows the Arduino IDE interface with the file 'Praktikum3.ino' open. The code is for a Generic ESP8266 Module. It includes a `void loop()` function that handles WiFi connection, sets fields for ThingSpeak, and writes data to the channel. The code is as follows:

```
54
55
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();
```

The screenshot shows the same Arduino IDE interface, but the code has been updated. The two lines `float temperature = dht.readTemperature();` and `float humidity = dht.readHumidity();` have been added to the code, highlighted with a red box. The code is as follows:

```
54
55
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 }
84
```

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

Potongan Kode

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

```
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 // 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 }  
81 else{  
82   Serial.println("Problem updating channel. HTTP error code: " + String(x));  
83 }  
84  
85 delay(20000); // Wait 20 seconds to update the channel again  
86 }  
87 }  
88 }  
89 }  
90 }
```

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

```
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...

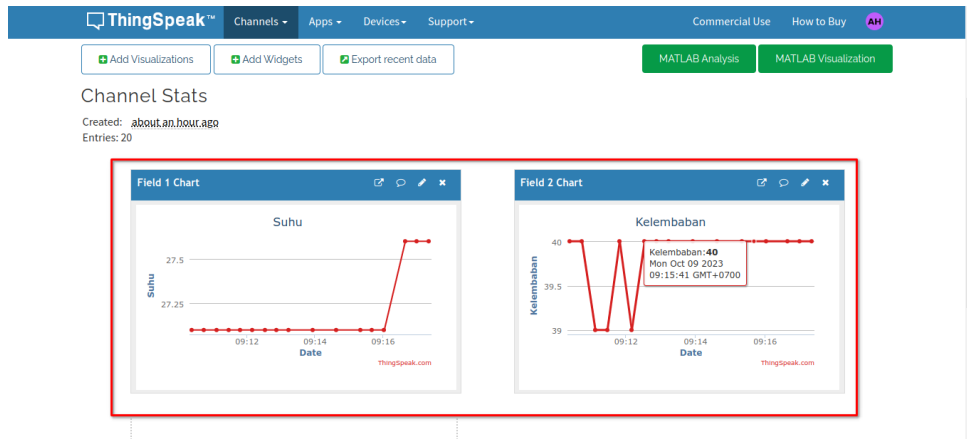
29. Data terkirim dan terunggah

```
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 }
```

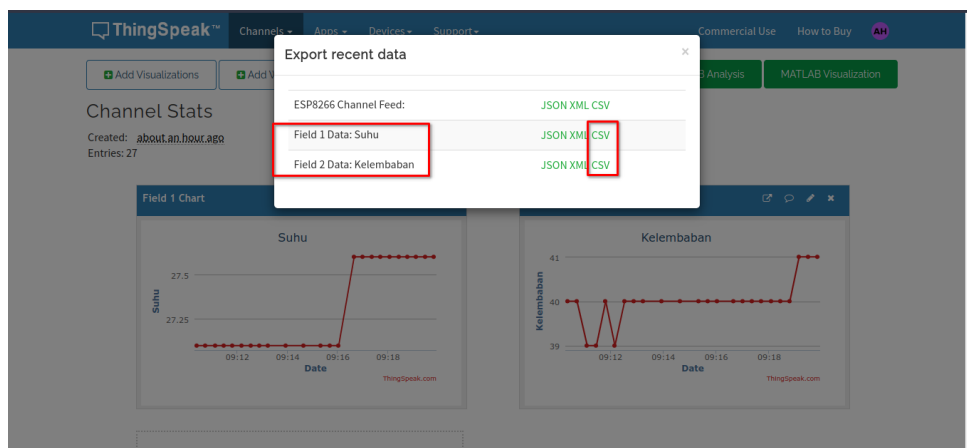
Output Serial Monitor

Message (Enter to send message to "Generic ESP8266 Module on /dev/ttyUSB0")

09:17:00.437 -> Channel update successful. 27.69 40.60



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



Bab 4

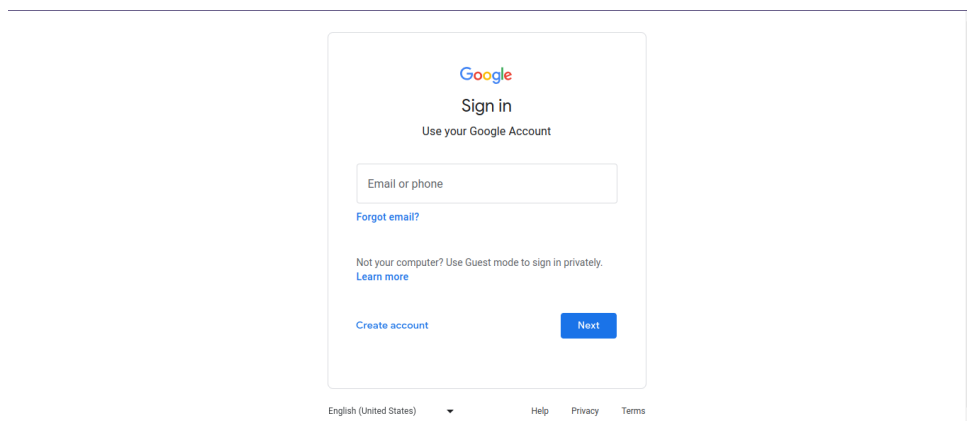
Praktikum 4

4.1 ESP8266, DHT-22, dan Firebase Realtime

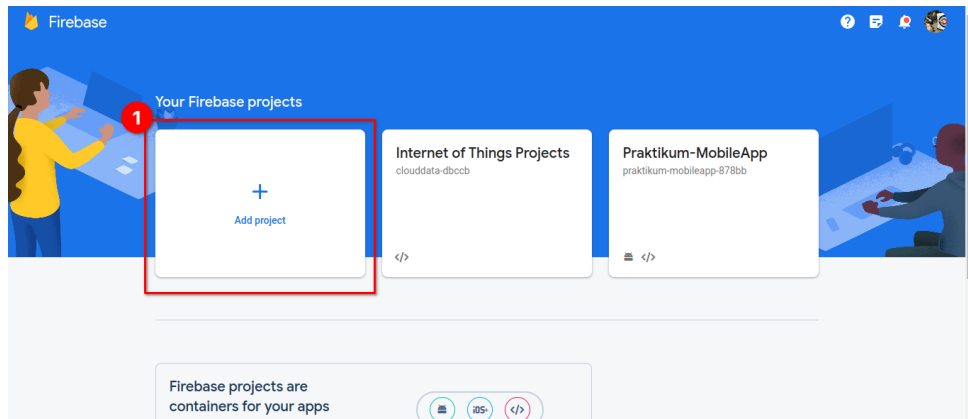
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Firebase Realtime. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 3** yang ada di halaman sebelumnya.

4.2 Tutorial

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



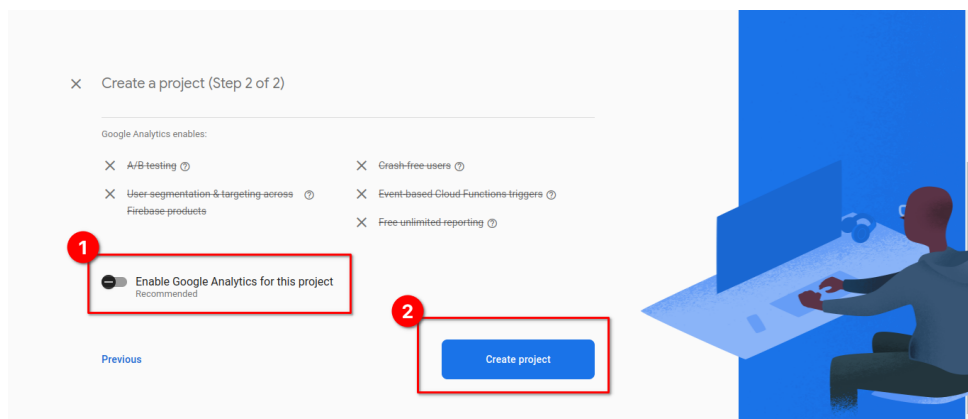
2. Buat projek baru dengan melakukan klik tanda +



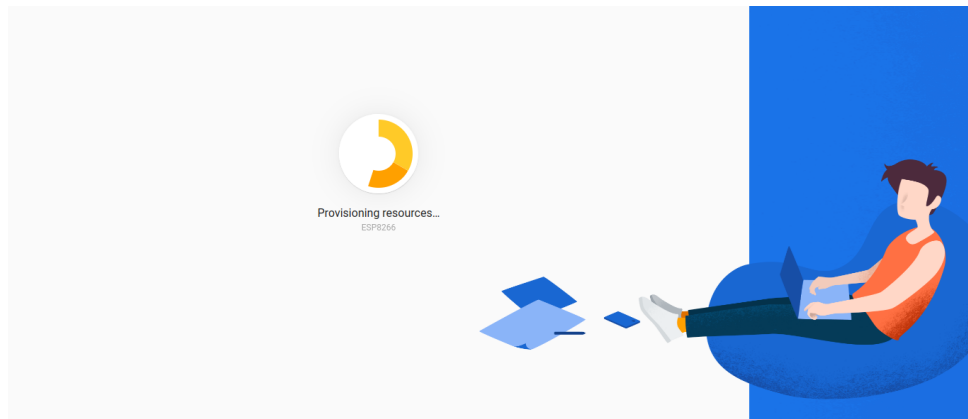
3. Isi nama projek



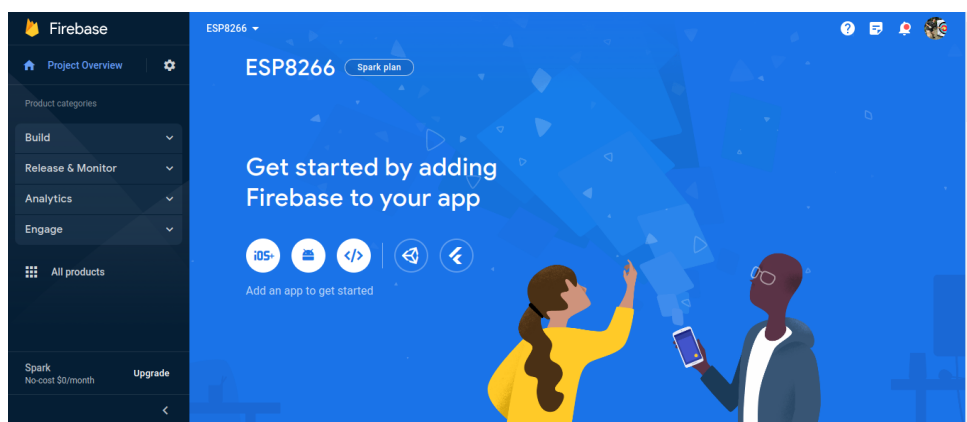
4. Matikan **Google Analytic** dan klik **Create Project**



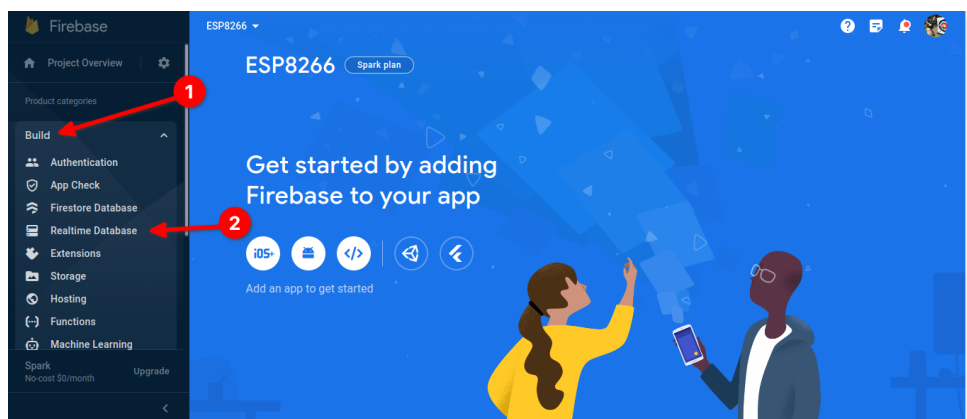
5. Tunggu proses berlangsung dan klik tombol apabila sudah muncul



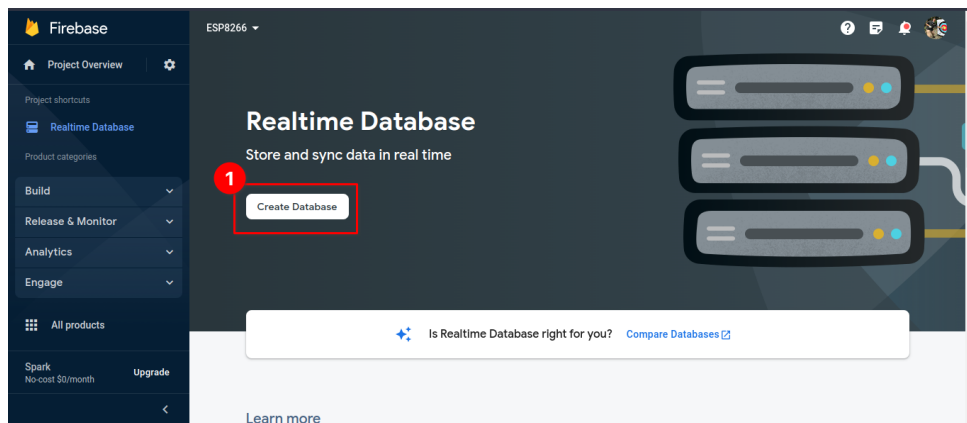
6. Firebase akan menampilkan dasbor sistem



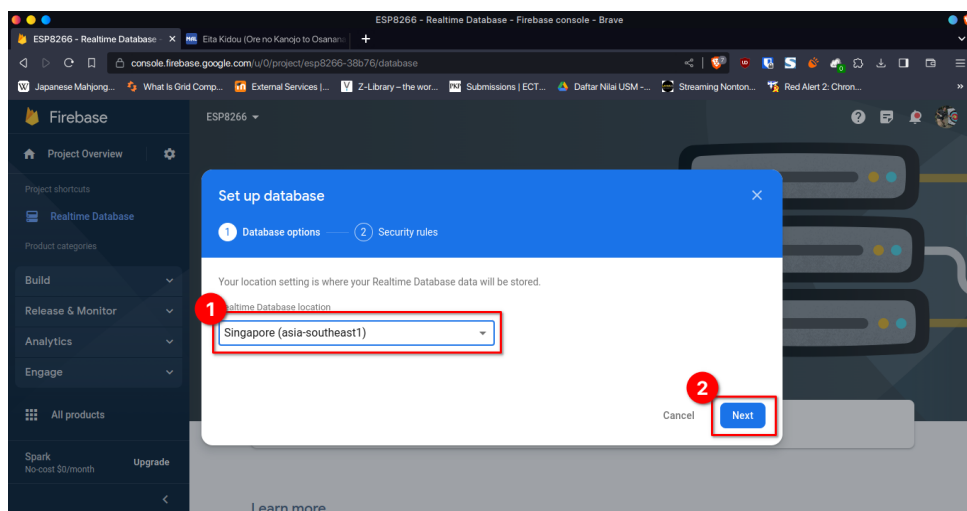
7. Klik **Build** dan pilih **Realtime Database**



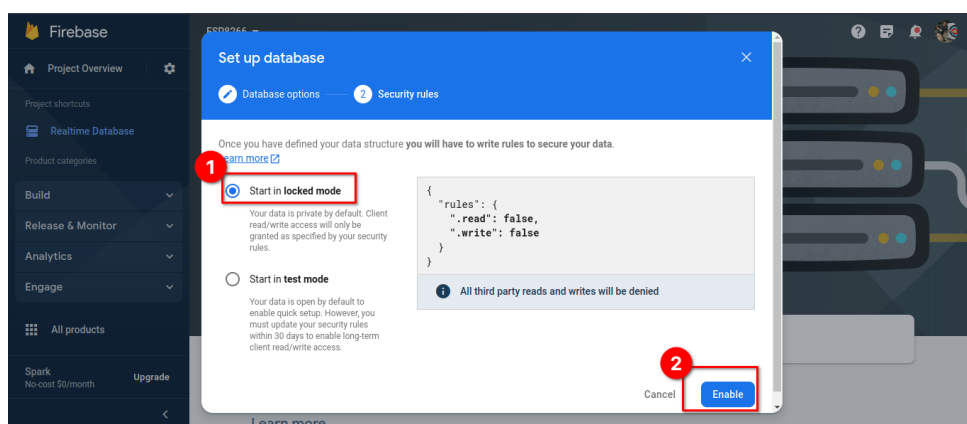
8. Klik **Create Database**



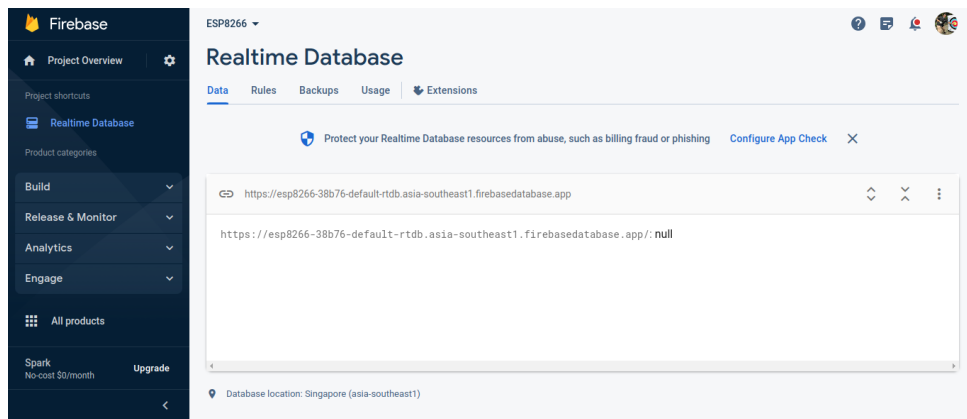
9. Pilih Lokasi dan Klik **Next**



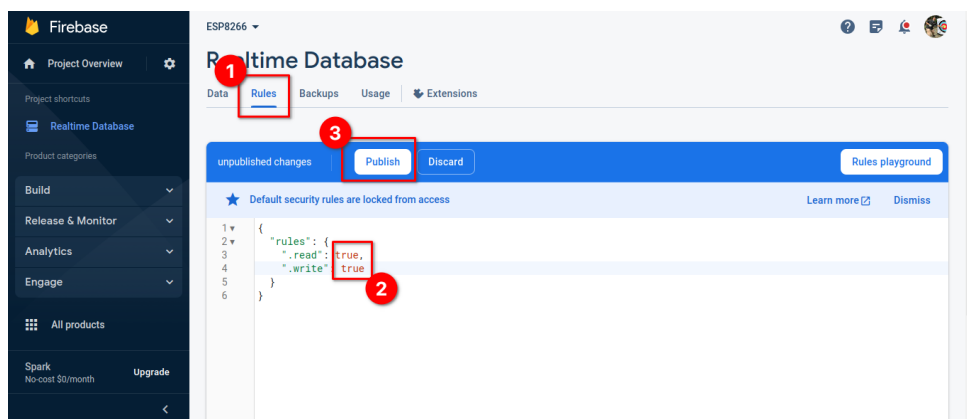
10. Pilih **Locked Mode** dan klik **Enable**



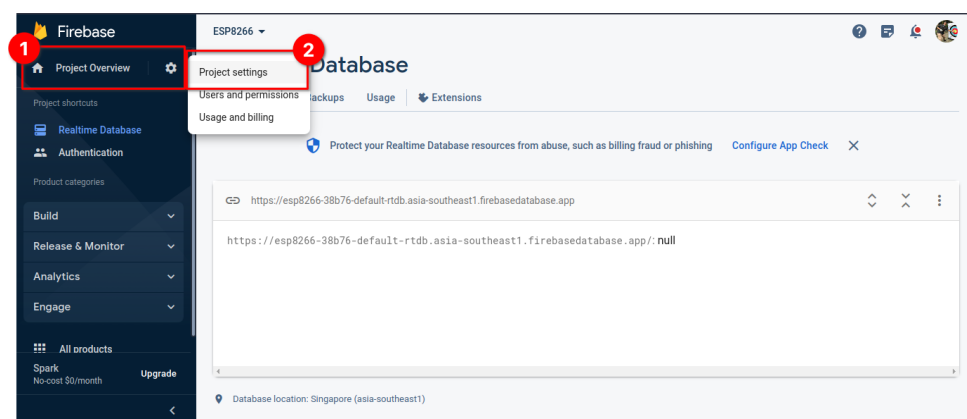
11. Database sudah dibuat



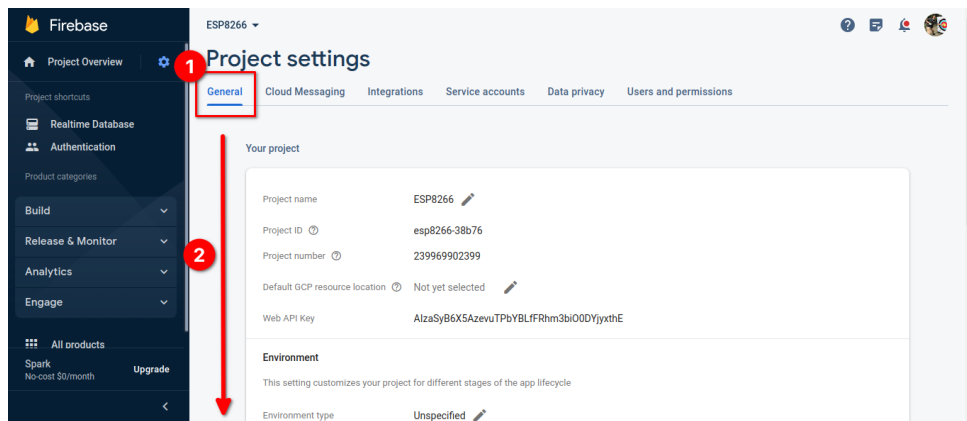
12. Sebelumnya ubah aturan database dengan klik **Rules**, dan ubah kata **false** menjadi **true**. dan klik **Publish**



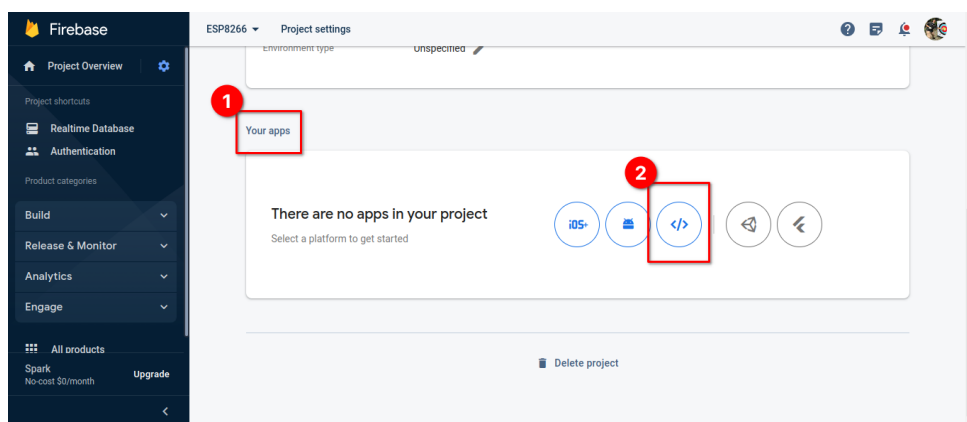
13. Untuk membuat kunci, klik **Roda Gigi Project Overview**, pilih **Project Settings**



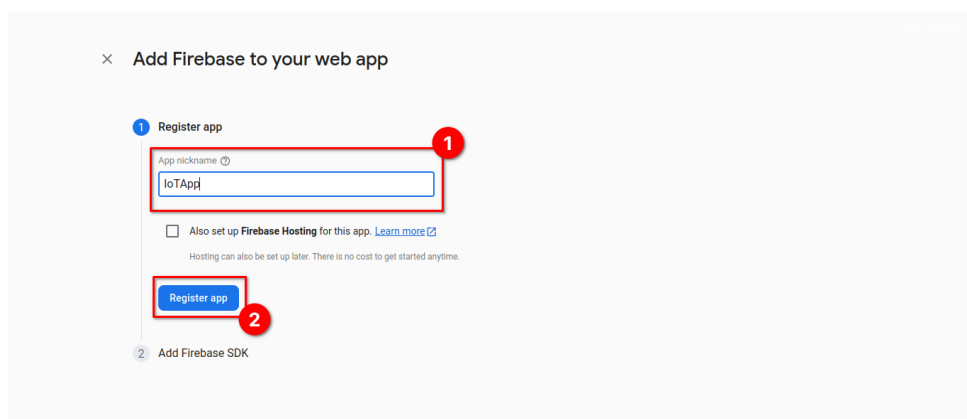
14. Di bagian **General**, scroll turun hingga menemukan **Apps**



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



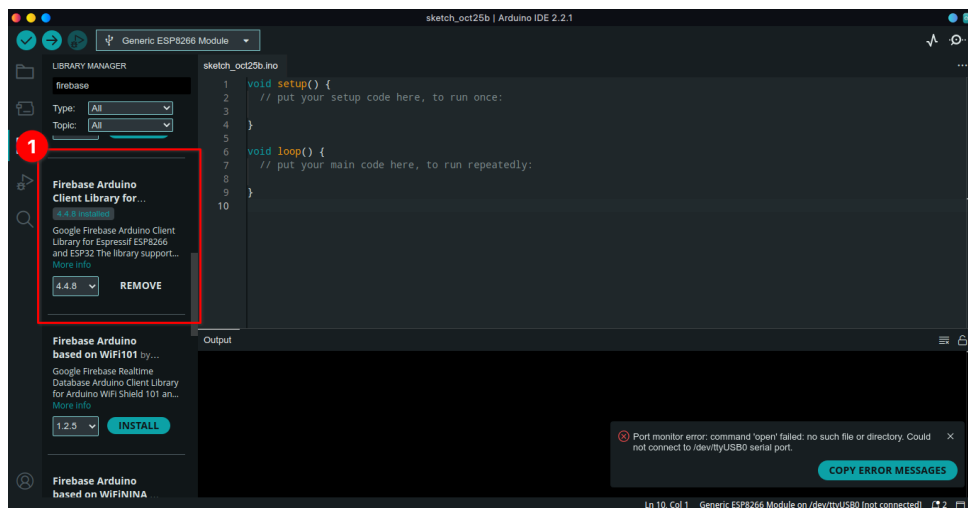
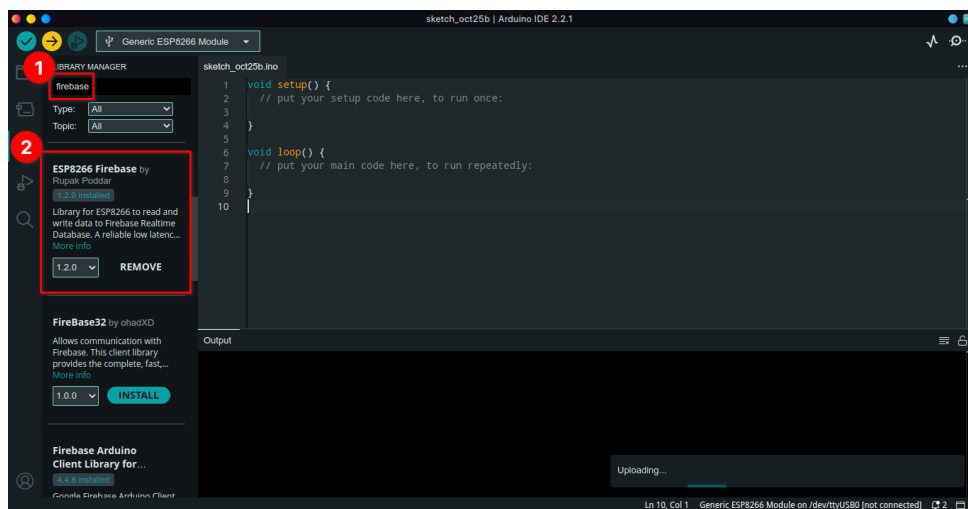
16. Isikan nama app, dan pilih **Register app**



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

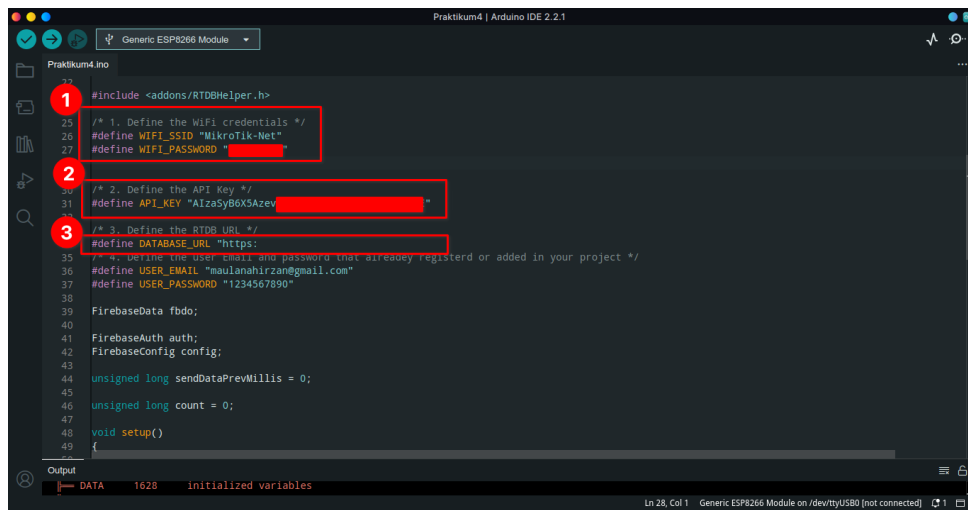


18. Di **Arduino IDE**, buka **Libraries** dan install **Firebase Arduino Client Library**



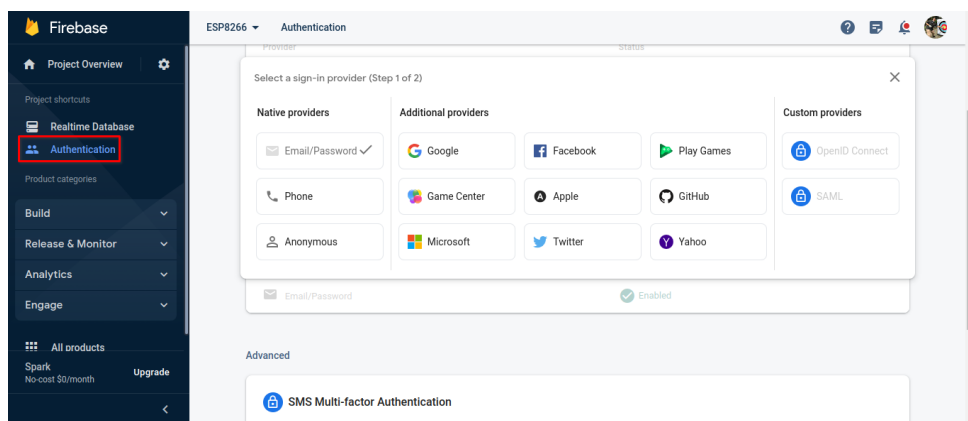
19. Buat proyek baru dengan template yang sudah ada. Klik **File** → **Examples** → **Firebase Arduino Client Library for ESP8266 and ESP32** → **FirebaseJson** → **Client** → **Firebase**

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

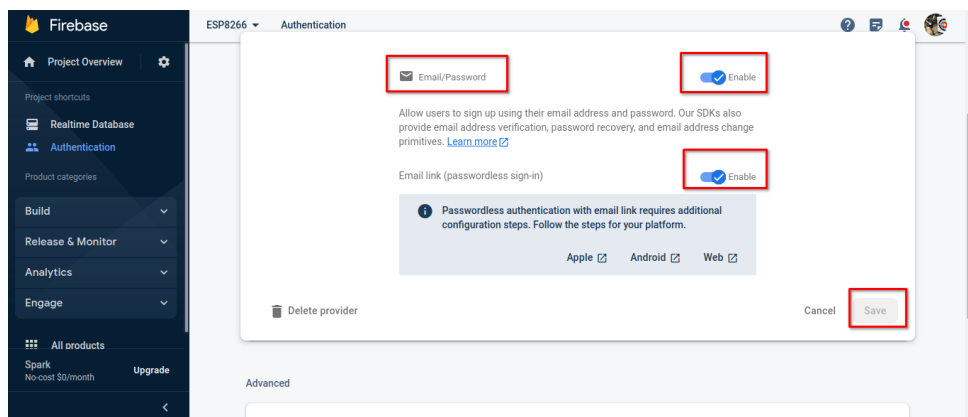


```
Praktikum4.ino
1 #include <addons/RTDBHelper.h>
25 /* 1. Define the WiFi credentials */
26 #define WIFI_SSID "MikroTik-Net"
27 #define WIFI_PASSWORD "..."
31 /* 2. Define the API Key */
32 #define API_KEY "AlzaSyB8x5Azev..."
33 /* 3. Define the RTDB URL */
34 #define DATABASE_URL "https:"
35 /* 4. Define the user email and password that already registered or added in your project */
36 #define USER_EMAIL "maulana@irzan@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 }
```

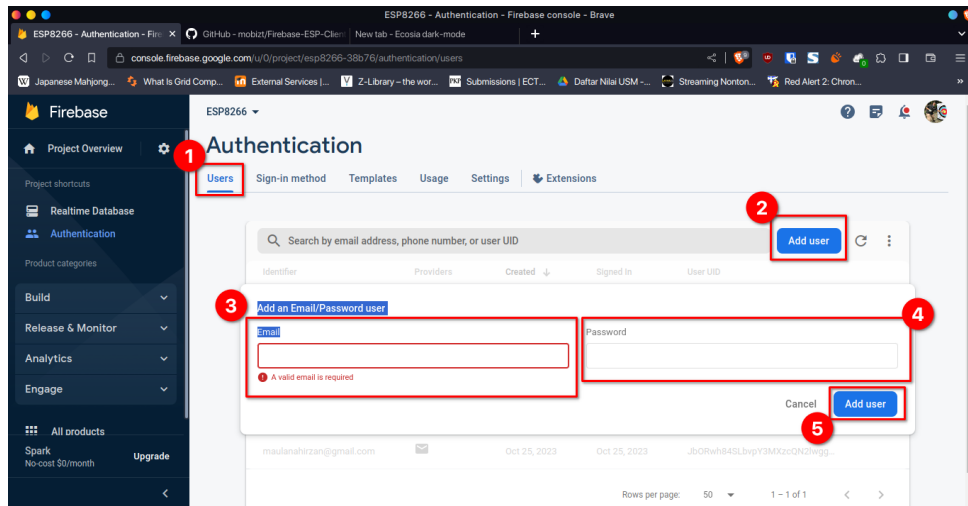
22. Bagian berikutnya adalah akun. Buka kembali **Firestore**, buka menu **Build** lalu **Firestore Authentication**



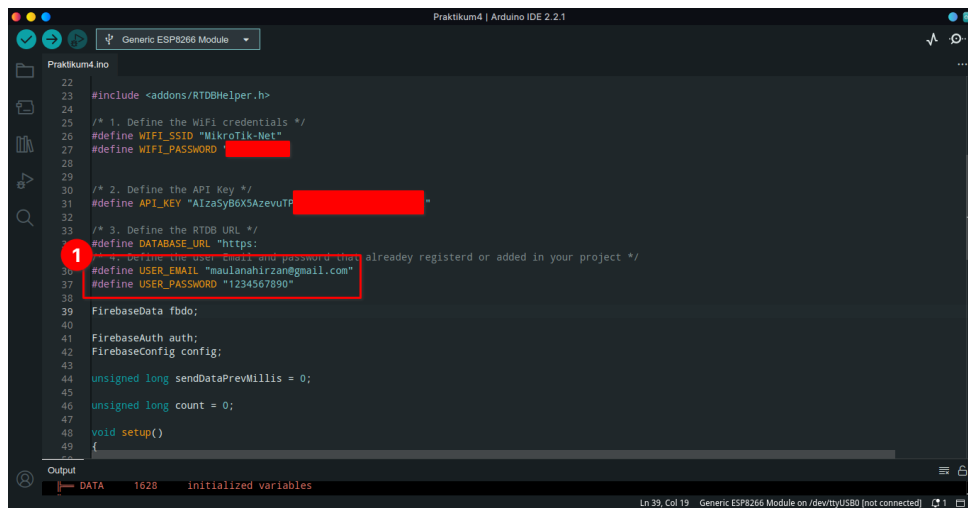
23. Pilih **Email/Password**, klik semua menjadi **Enable**, dan **Save**



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**

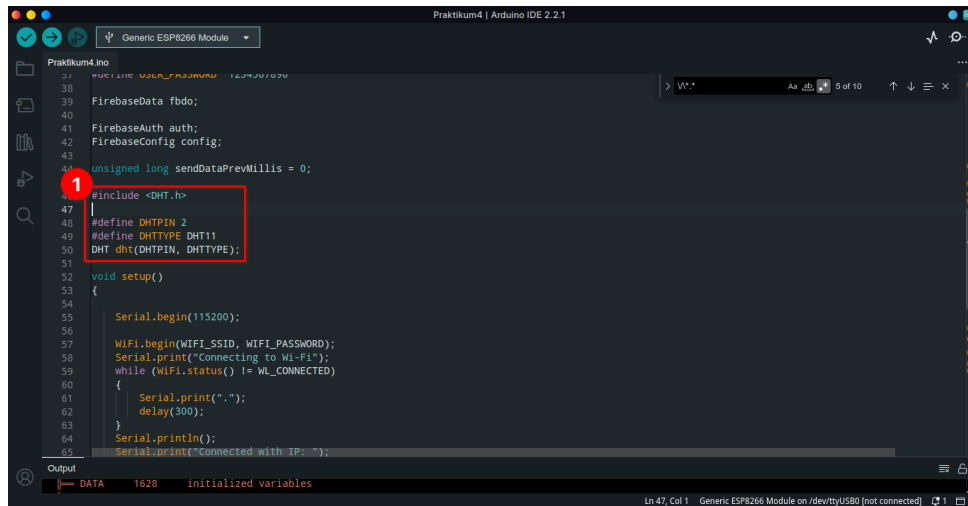


26. Berikutnya adalah menambahkan kode untuk sensor DHT

Potongan Kode

```
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
```



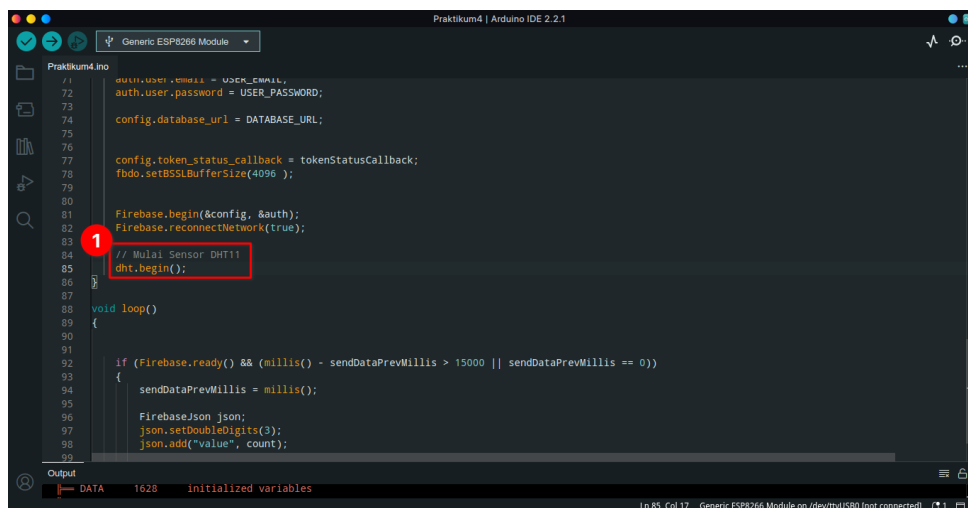
27. Tambahkan di bagian akhir kode **void setup()** dengan kode berikut:

Potongan Kode

```

// Mulai Sensor DHT-22
dht.begin();

```



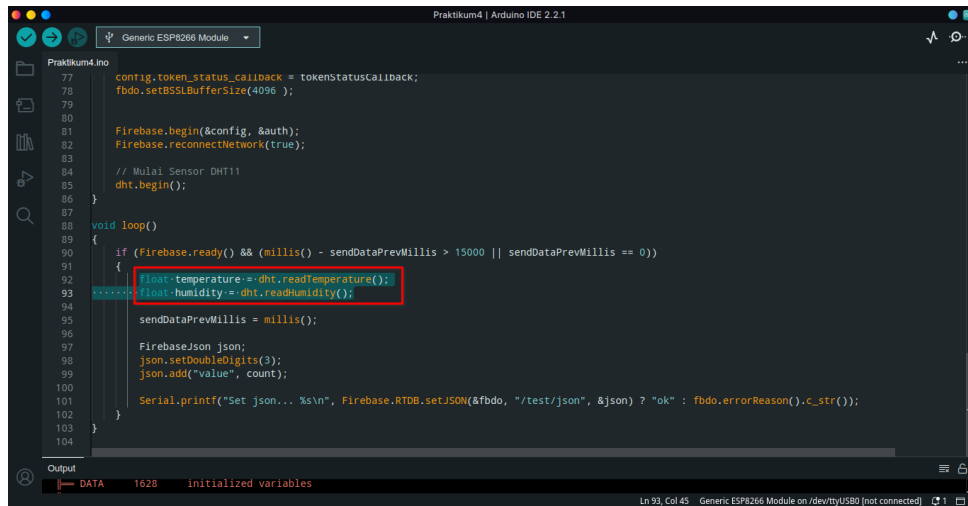
28. Di dalam kode **void loop()** setelah kode **if**, masukkan kode berikut

Potongan Kode

```

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

```

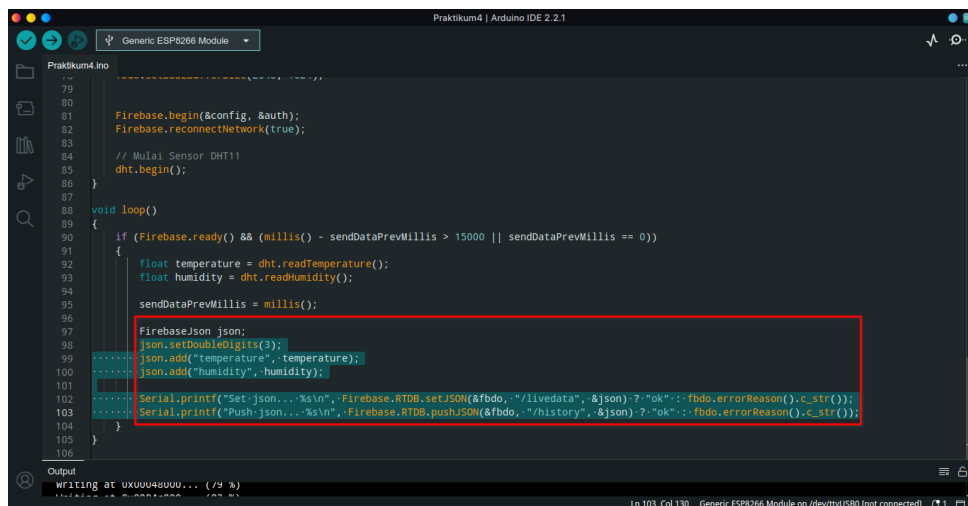
```
Praktikum4.ino
77 config.token_status_callback = tokenStatusCallback;
78 fbdo.setSSLBufferSize(4096);
79
80
81 Firebase.begin(&config, &auth);
82 Firebase.reconnectNetwork(true);
83
84 // Mulai Sensor DHT11
85 dht.begin();
86
87
88 void loop()
89 {
90   if (Firebase.ready() && (millis() - sendDataPrevMillis > 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

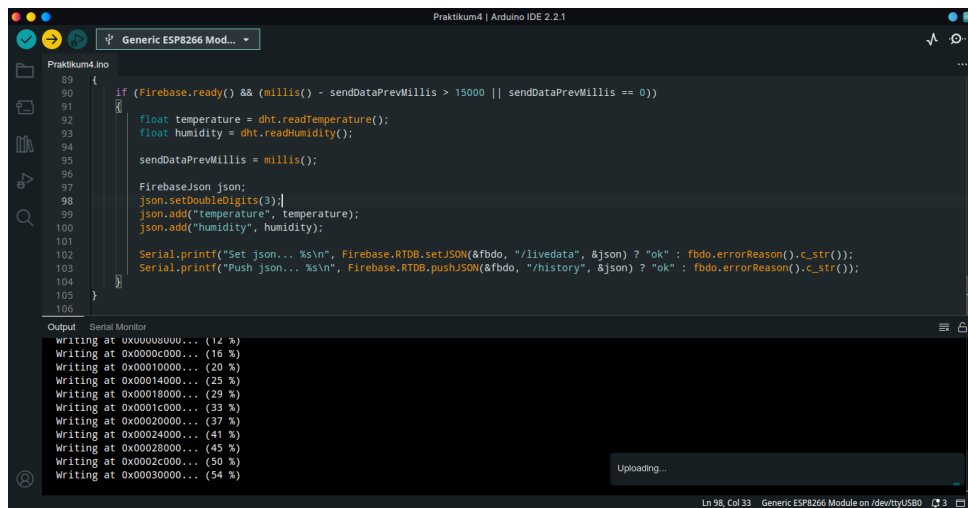
```
FirebaseJson json;
json.setDoubleDigits(3);
json.add("temperature", temperature);
json.add("humidity", humidity);

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());
```

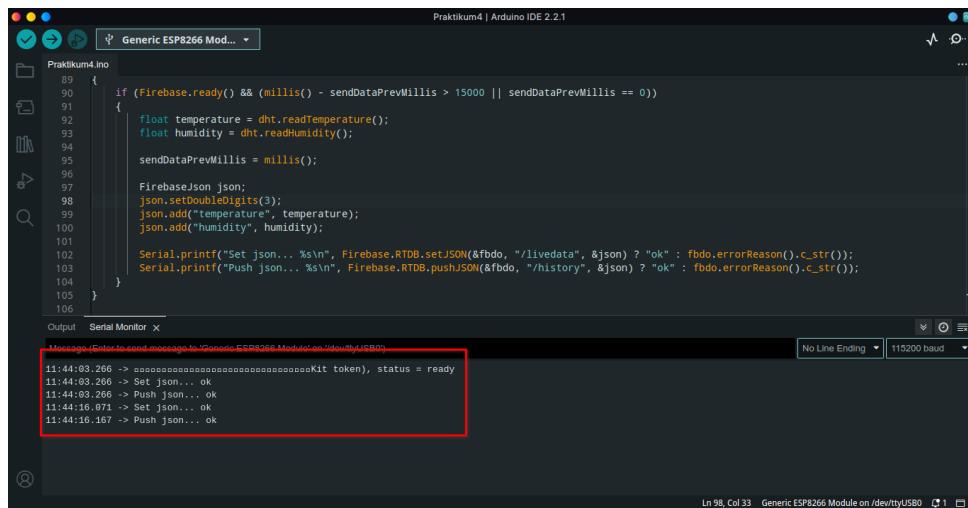


```
Praktikum4.ino
79
80
81 Firebase.begin(&config, &auth);
82 Firebase.reconnectNetwork(true);
83
84 // Mulai Sensor DHT11
85 dht.begin();
86
87
88 void loop()
89 {
90   if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
91   {
92     float temperature = dht.readTemperature();
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 }
```

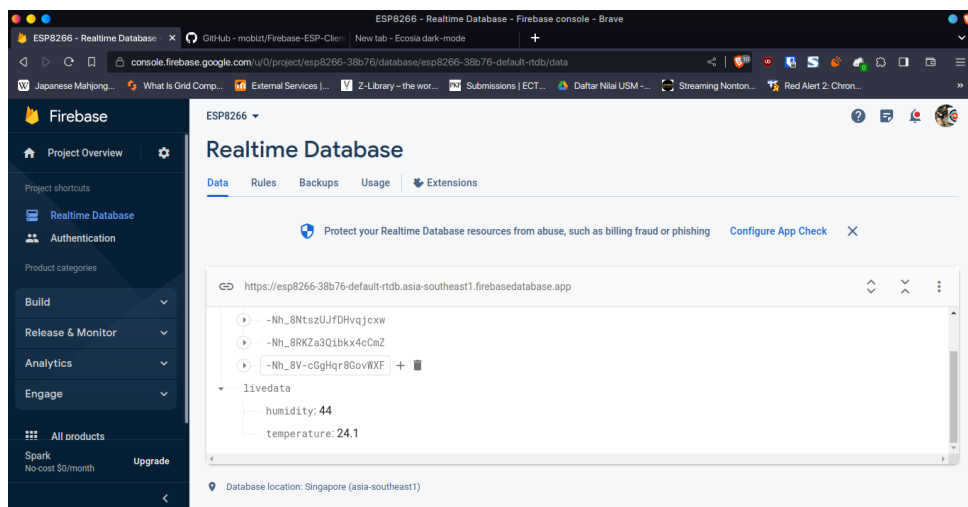
30. Verifikasi dan Upload aplikasi



31. Data sukses diunggah



32. Hasil



Bab 5

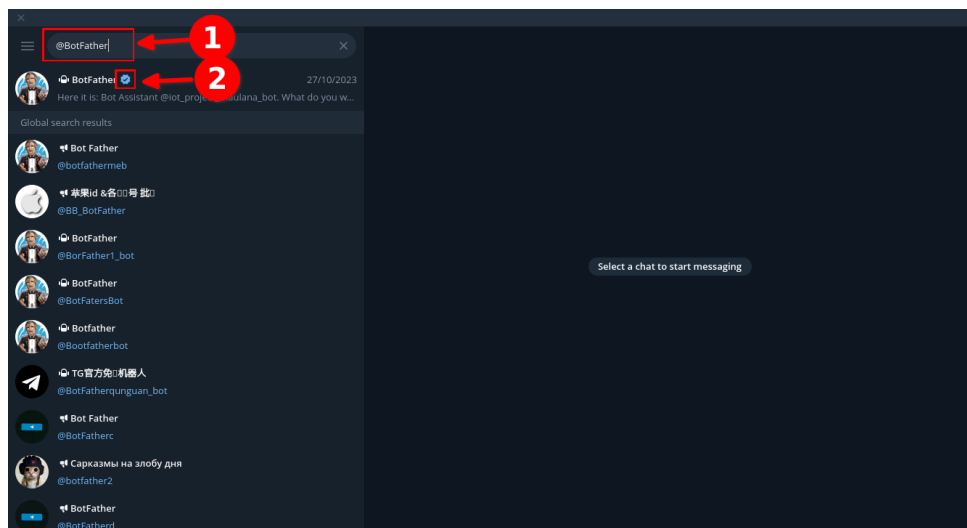
Praktikum 5

5.1 ESP8266, DHT-22, dan Telegram Bot

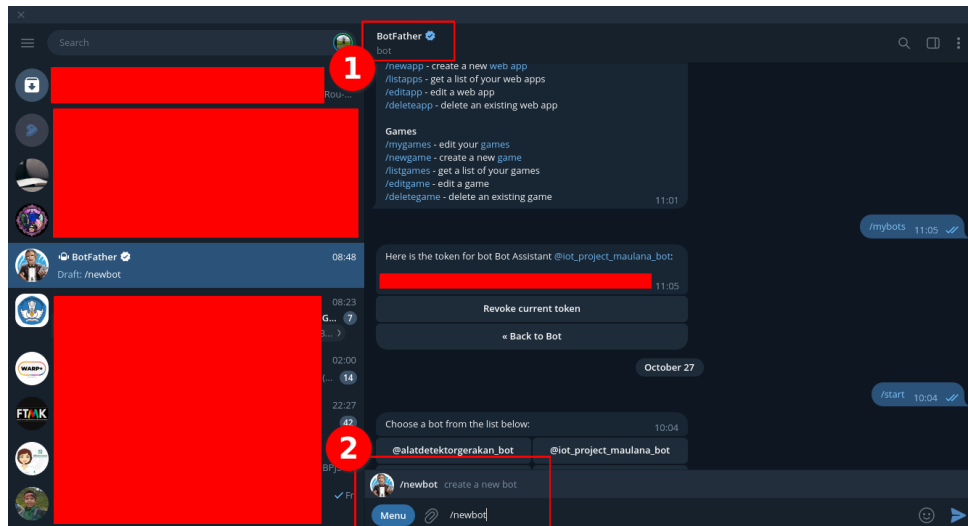
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Telegram Bot. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 4** yang ada di halaman sebelumnya.

5.2 Tutorial

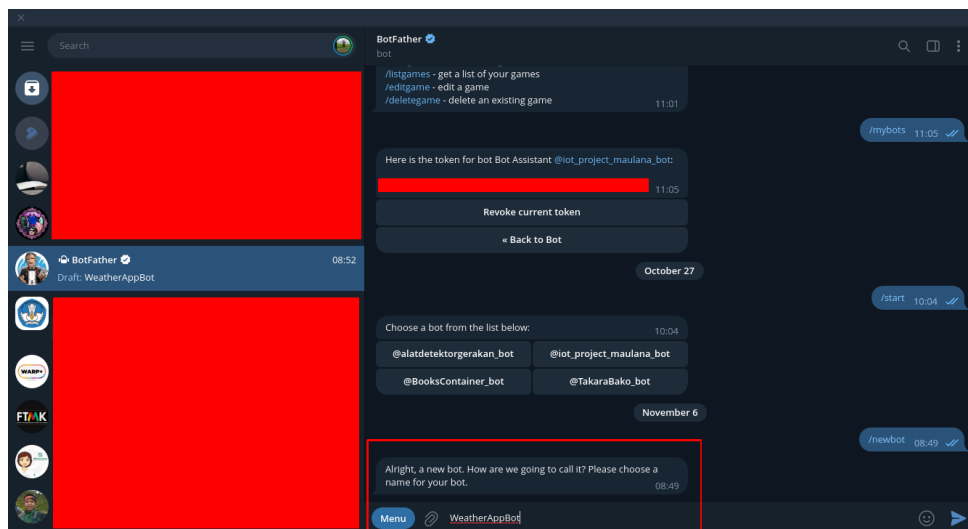
1. Tahap pertama yang dilakukan adalah membuat **Telegram Bot**. Pastikan untuk memiliki Akun Telegram untuk bisa memulai langkah ini
2. Cari **Bog Manager** dengan **@BotFather**



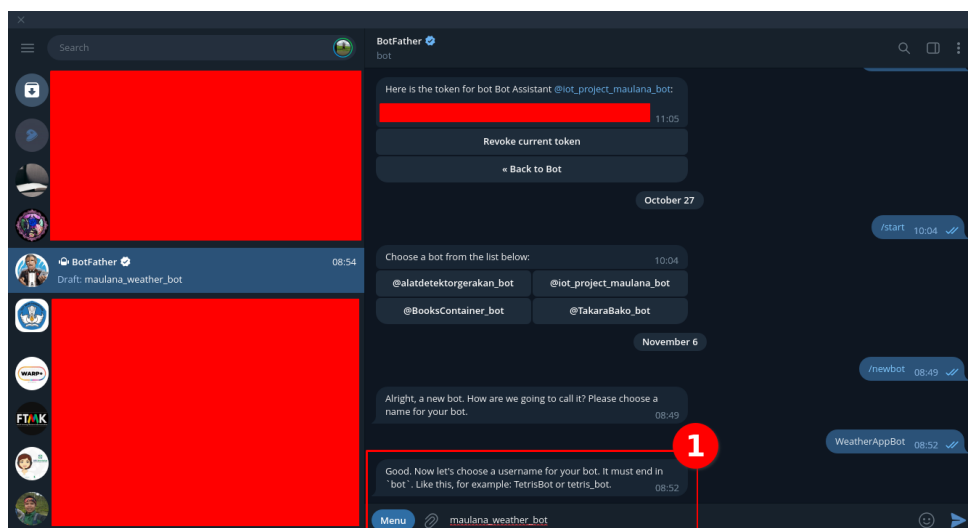
3. Gunakan perintah **/newbot** untuk membuat **Telegram Bot** baru



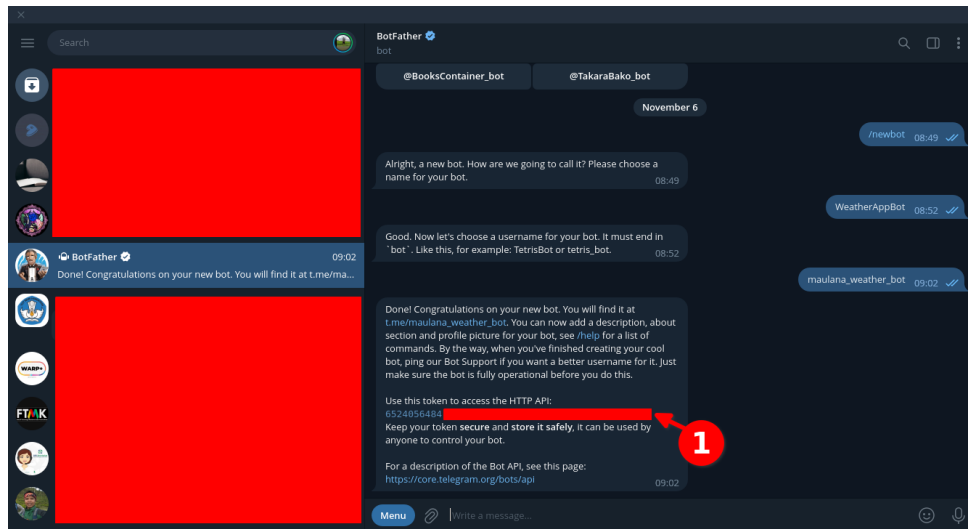
4. Lalu masukkan nama dari **Telegram Bot**



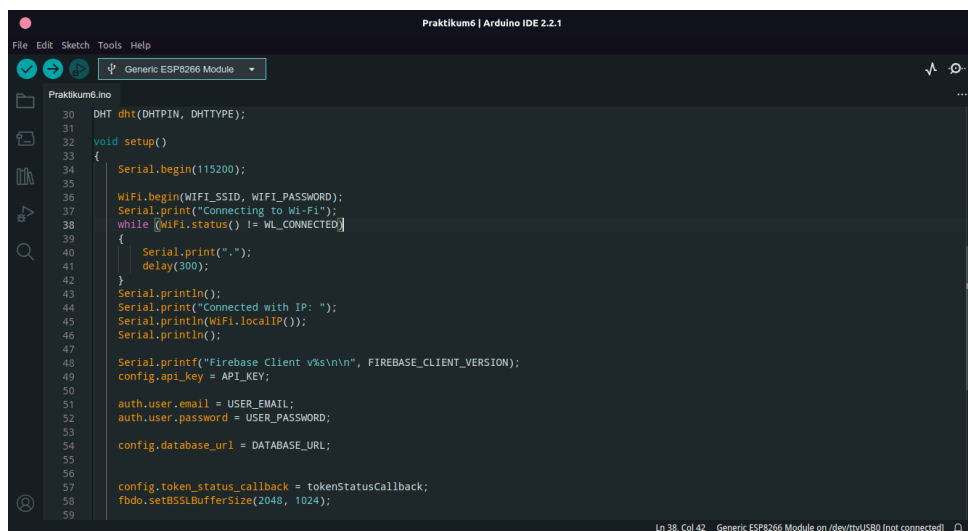
5. Lalu masukkan **username** untuk mempermudah pencarian **Telegram Bot**. Pastikan memiliki akhiran **_bot**



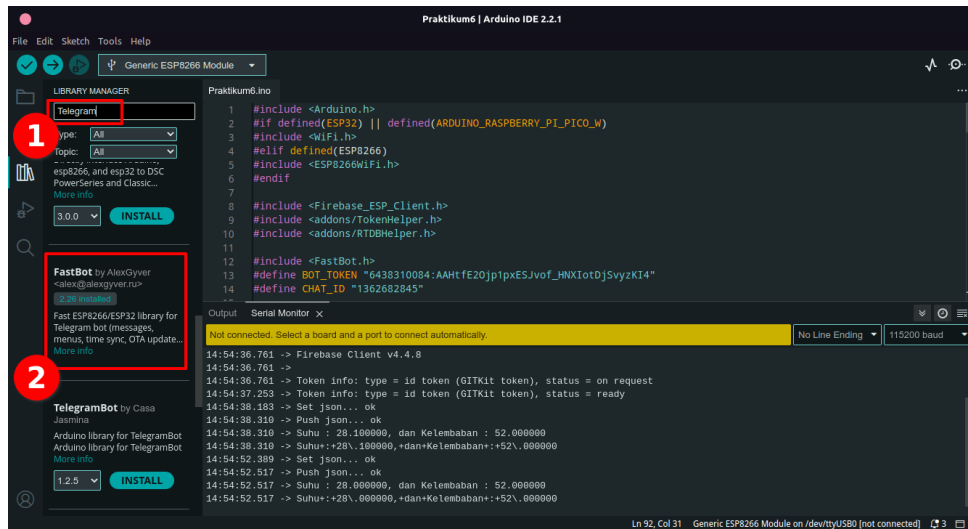
6. **Telegram Bot** sudah jadi dan **Token API** akan ditampilkan. Simpan baik-baik kode tersebut



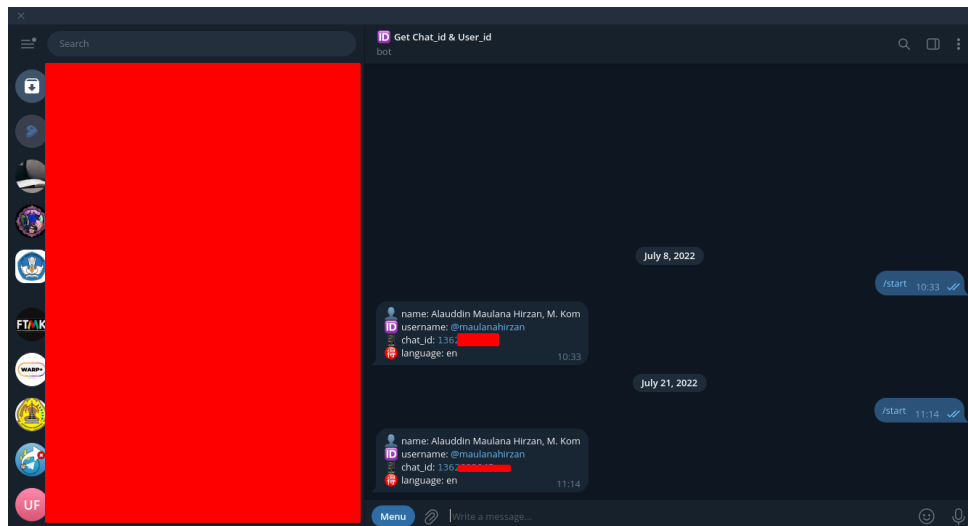
7. Berikutnya adalah membuka kembali **Praktikum 4** dengan menggunakan **Arduino IDE**. Lakukan **Save As** untuk menyimpan sebagai **Praktikum 6**



8. Install Library dengan nama **FastBot**



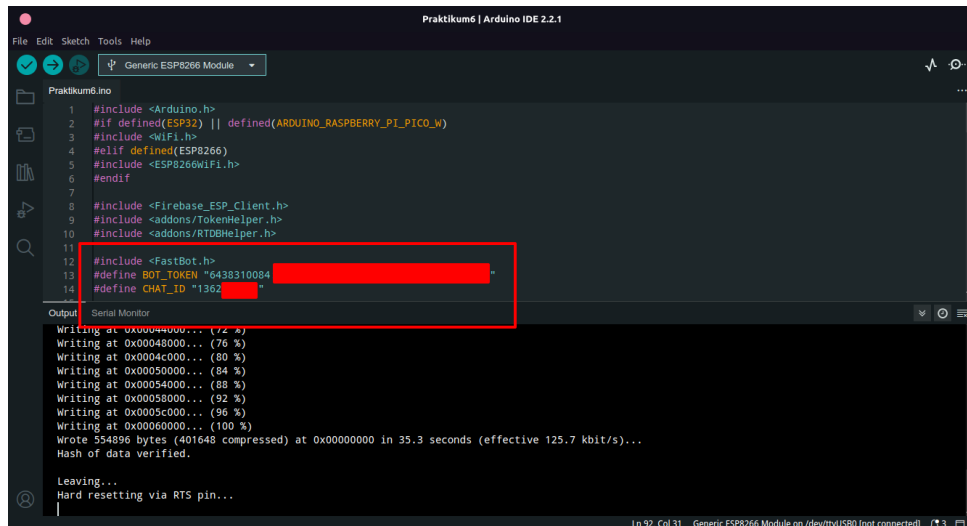
9. Berikutnya adalah mendapatkan **Chat ID** melalui Bot <https://t.me/chatIDrobot>



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

Sesudah

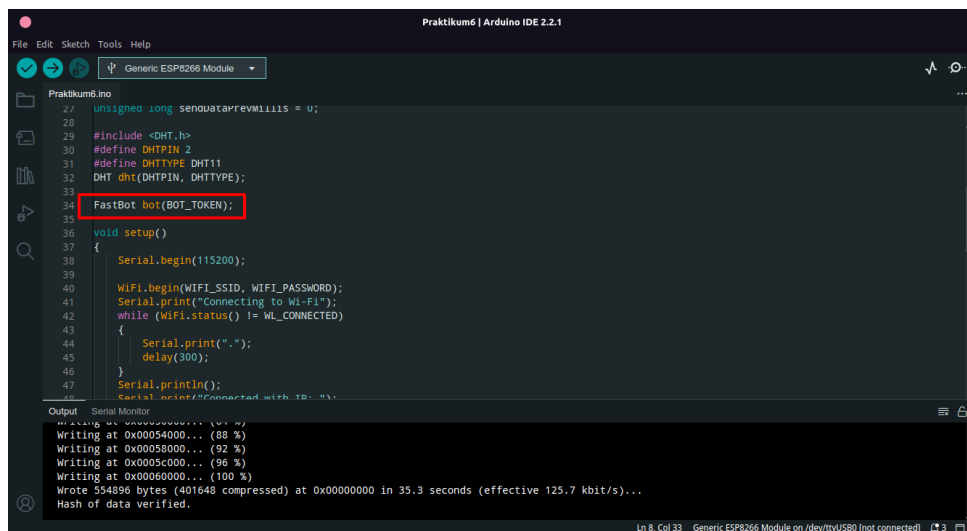
```
#include <FastBot.h>
#define BOT_TOKEN "<TOKEN BOT>"
#define CHAT_ID "<Chat ID>"
```



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

Sesudah

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



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.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 554896 bytes (401648 compressed) at 0x00000000 in 35.3 seconds (effective 125.7 kbit/s)...
Hash of data verified.
```

13. Verifikasi dan Upload kode ke Perangkat

```
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

```
14:54:30.761 -> 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
```


Bab 6

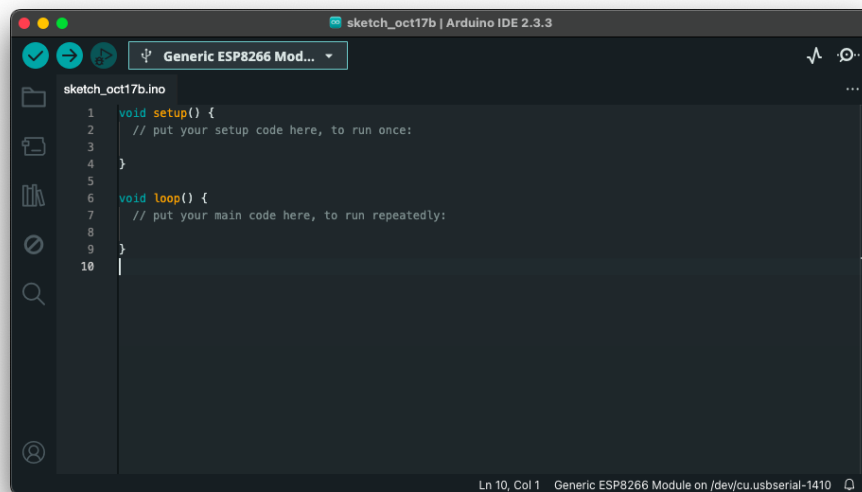
Praktikum 6

6.1 ESP8266, DHT-22, dan Linear Regression

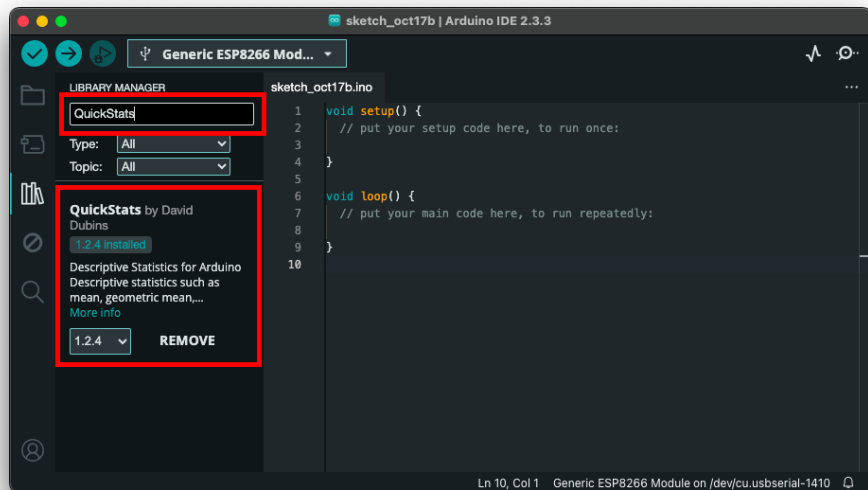
Di bagian ini mahasiswa diajarkan bagaimana mengimplementasikan algoritma *Linear Regression* sederhana dengan perangkat **ESP8266**, dan **DHT-22**. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 5** yang ada di halaman sebelumnya.

6.2 Tutorial

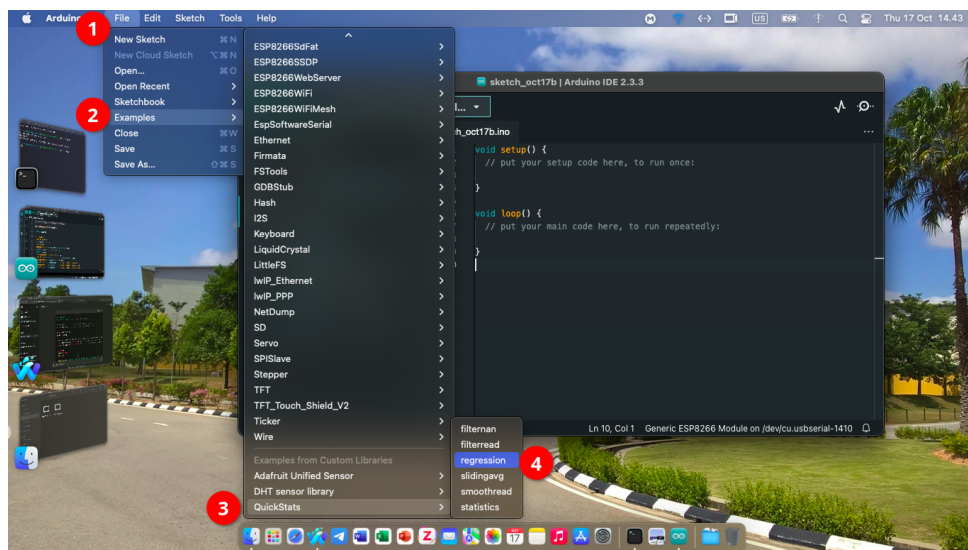
1. Buka kembali Arduino IDE untuk memulai proyek baru. ESP8266 tidak harus tercolok.



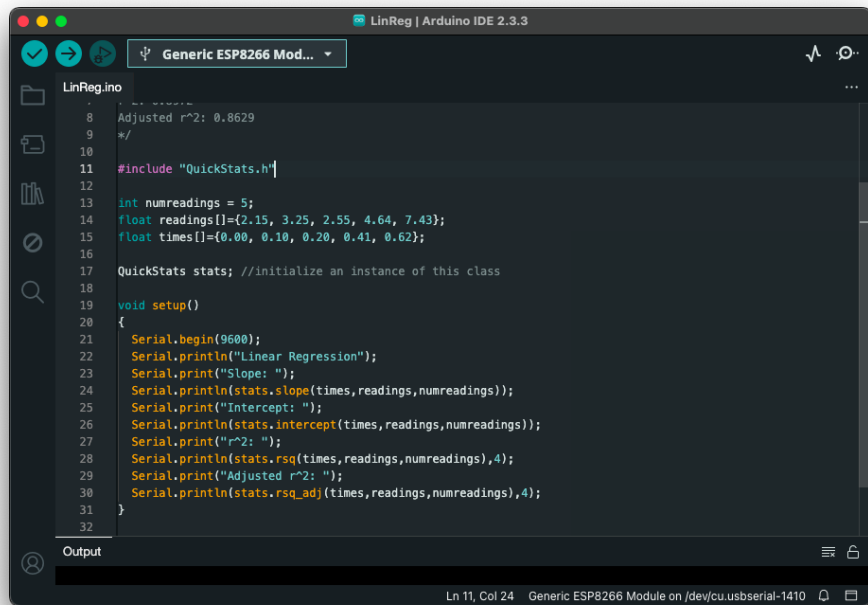
2. Install Library **QuickStats**



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

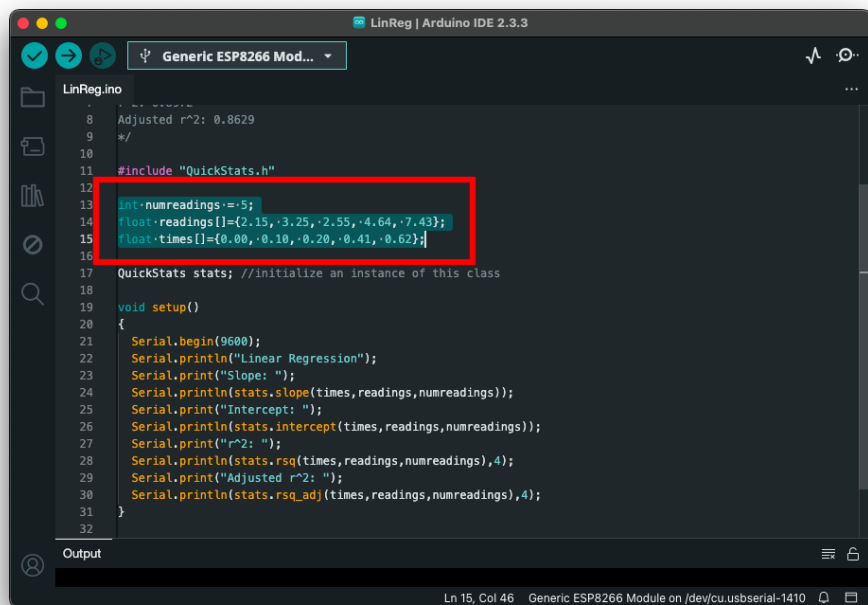


4. Tunggu window contoh kode **QuickStats** muncul

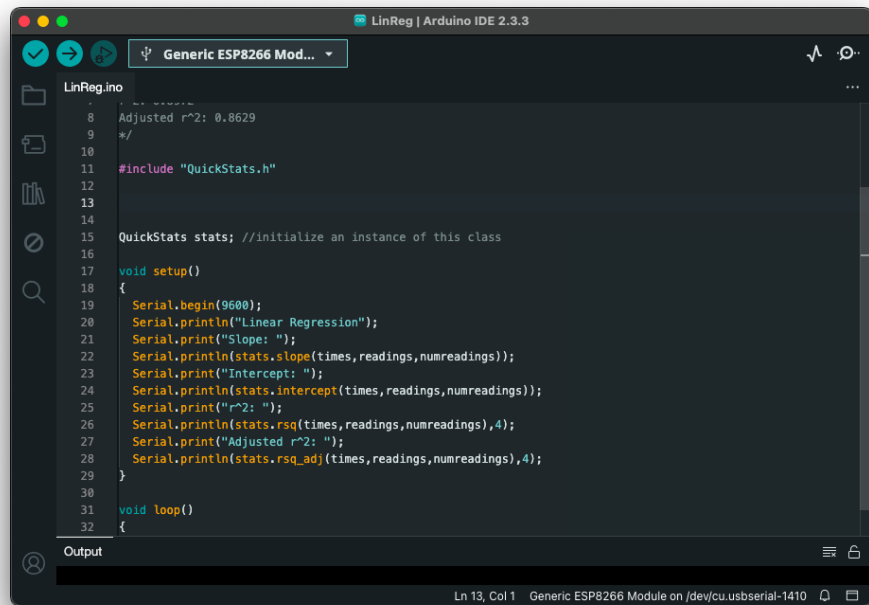


```
LinReg.ino
8 Adjusted r^2: 0.8629
9 */
10
11 #include "QuickStats.h"
12
13 int numreadings = 5;
14 float readings[]={2.15, 3.25, 2.55, 4.64, 7.43};
15 float times[]={0.00, 0.10, 0.20, 0.41, 0.62};
16
17 QuickStats stats; //initialize an instance of this class
18
19 void setup()
20 {
21   Serial.begin(9600);
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
```



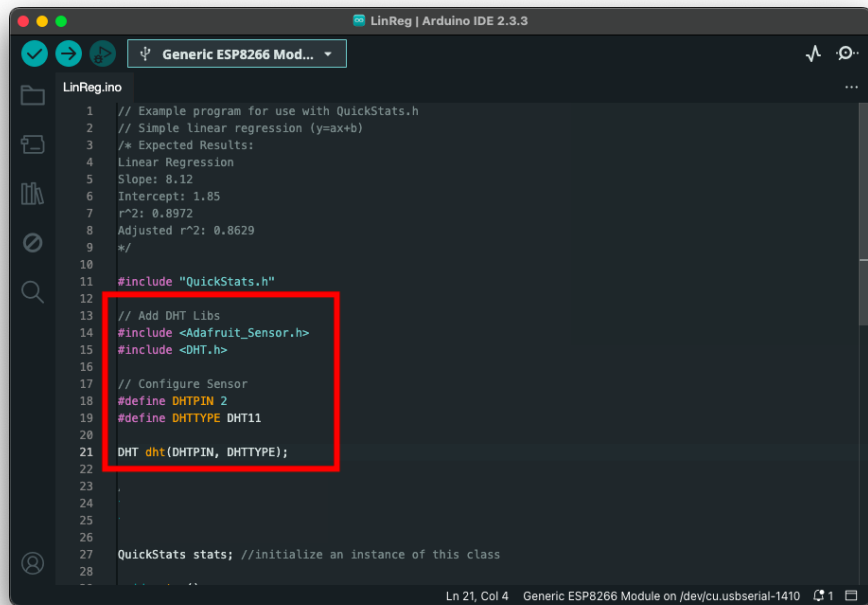
6. Di bagian bawah dari `#include "QuickStats.h"`, tambahkan kode berikut untuk sensor DHT-22:

Kode Program

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

// Configure Sensor
#define DHTPIN 2
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
```



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:

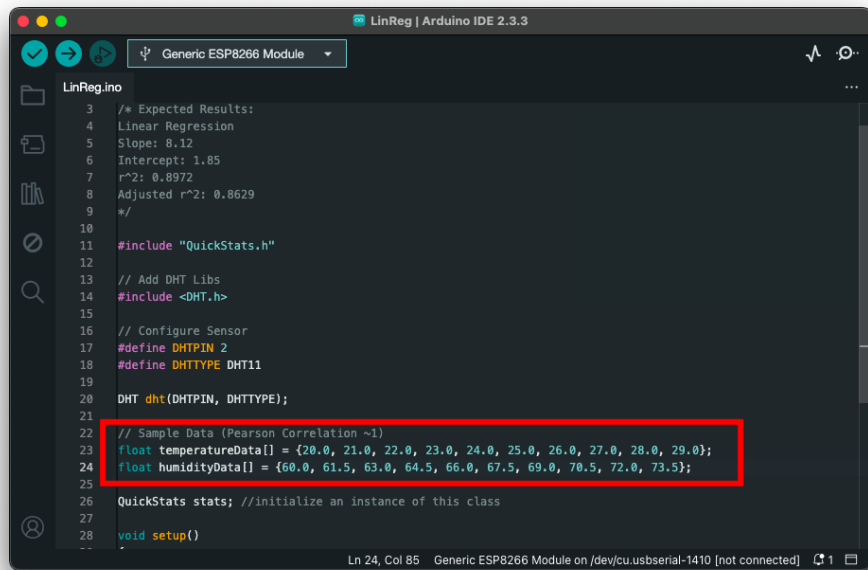
Kode Program

```

// 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};

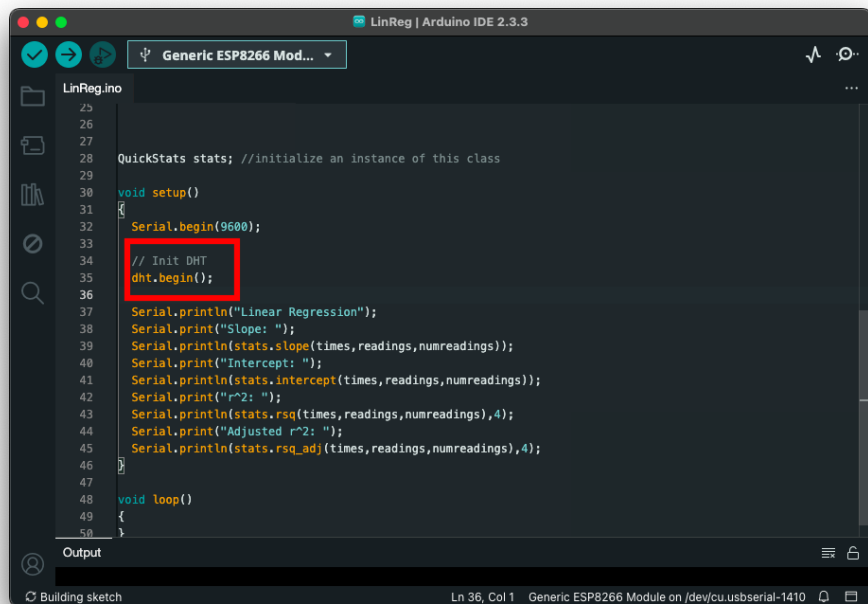
```



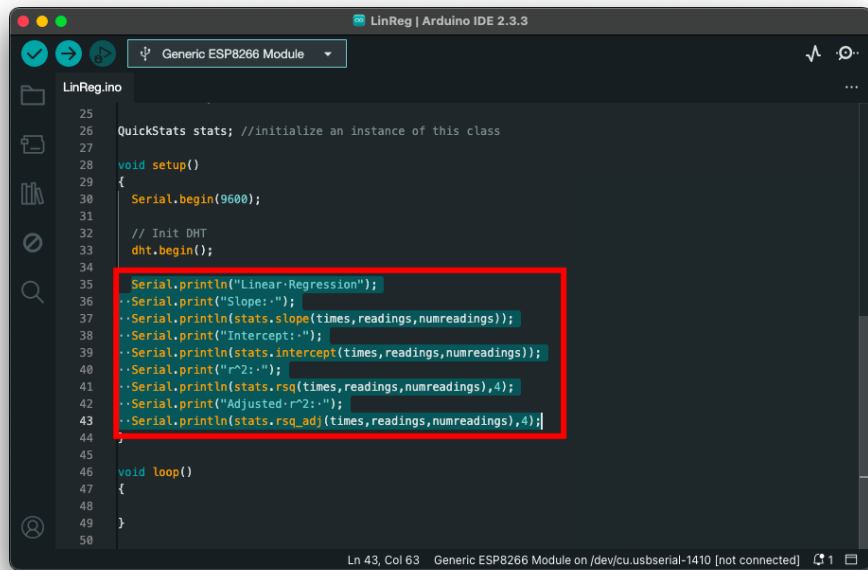
8. Setelah mendapatkan data *training* untuk **Regresi Linier**, maka langkah berikutnya adalah menambahkan kode inisialisasi untuk sensor **DHT 11** dengan kode berikut setelah kode **Serial.begin(9600);**:

Kode Program

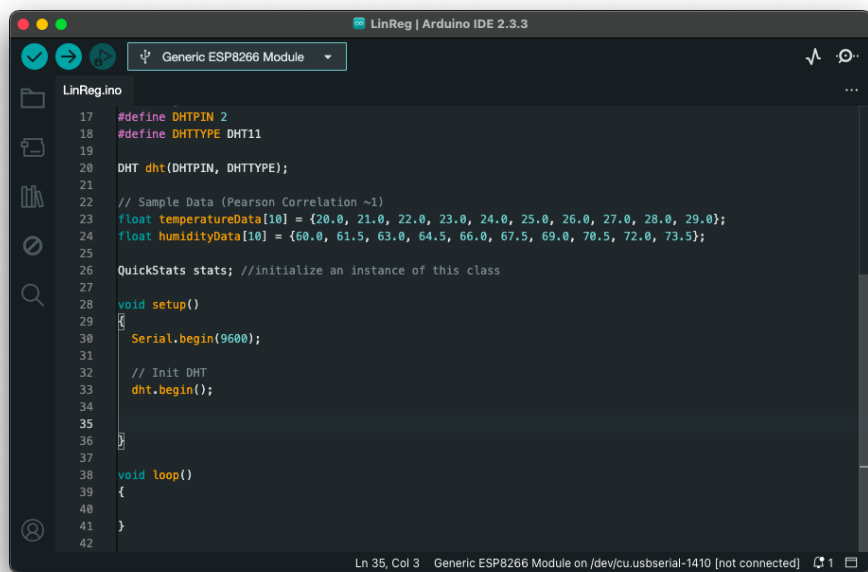
```
// Init DHT
dht.begin();
```



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 }
```

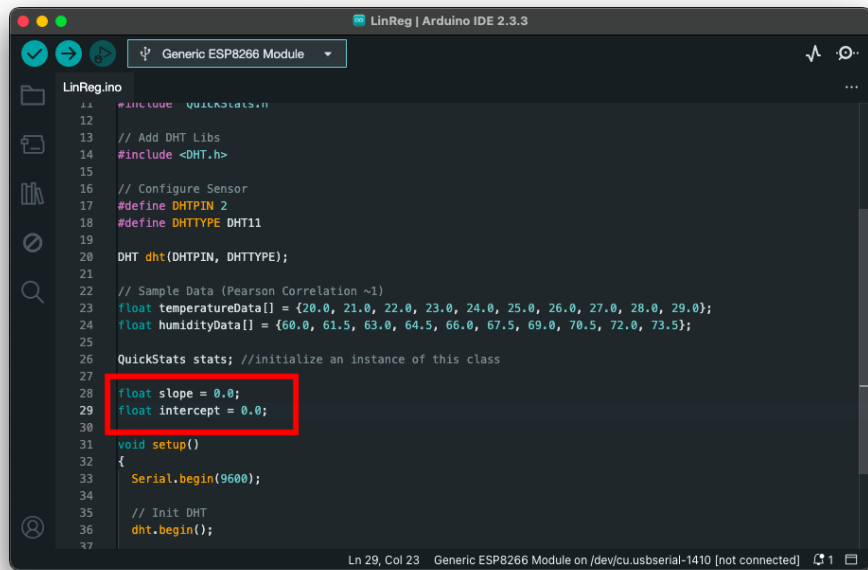


```
17 #define DHTPIN 2
18 #define DHTTYPE DHT11
19
20 DHT dht(DHTPIN, DHTTYPE);
21
22 // Sample Data (Pearson Correlation ~1)
23 float temperatureData[10] = {20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0};
24 float humidityData[10] = {60.0, 61.5, 63.0, 64.5, 66.0, 67.5, 69.0, 70.5, 72.0, 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;
```



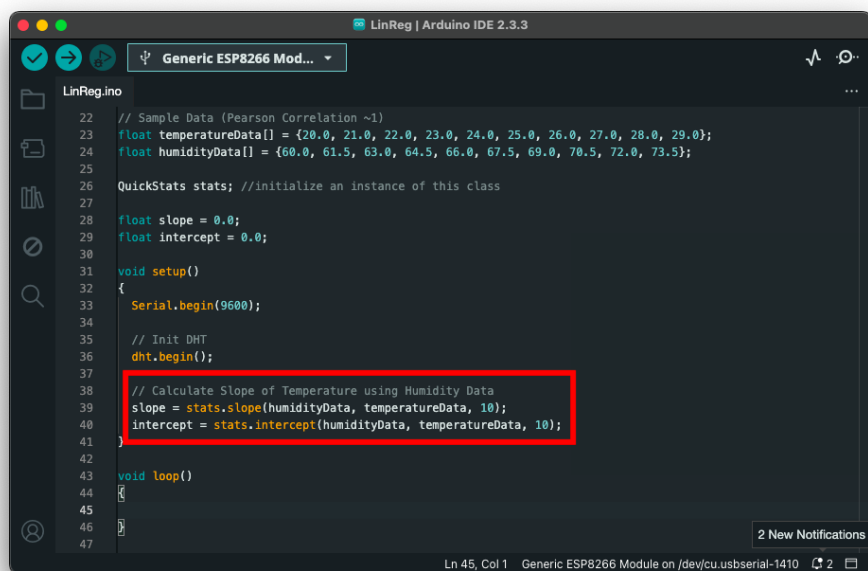
11. Kembali lagi ke fungsi **setup()**, dan tambahkan kode berikut setelah kode **dht.begin()**;
Rumus ini digunakan untuk mencari garis miring dari data dan nilai intersepsi nya

Kode Program

```

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

```

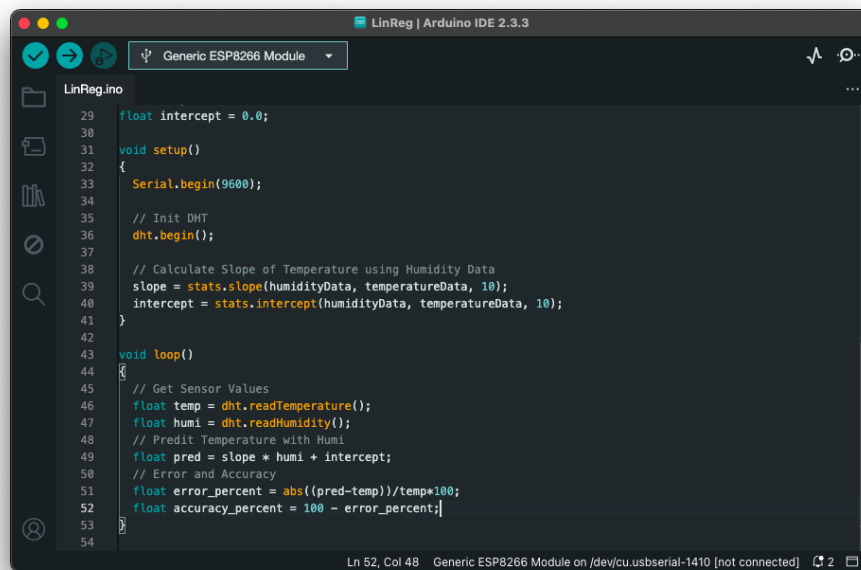


12. Bagian terakhir untuk kode **loop()** ada dua kode yang ditambahkan:

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

Kode Program

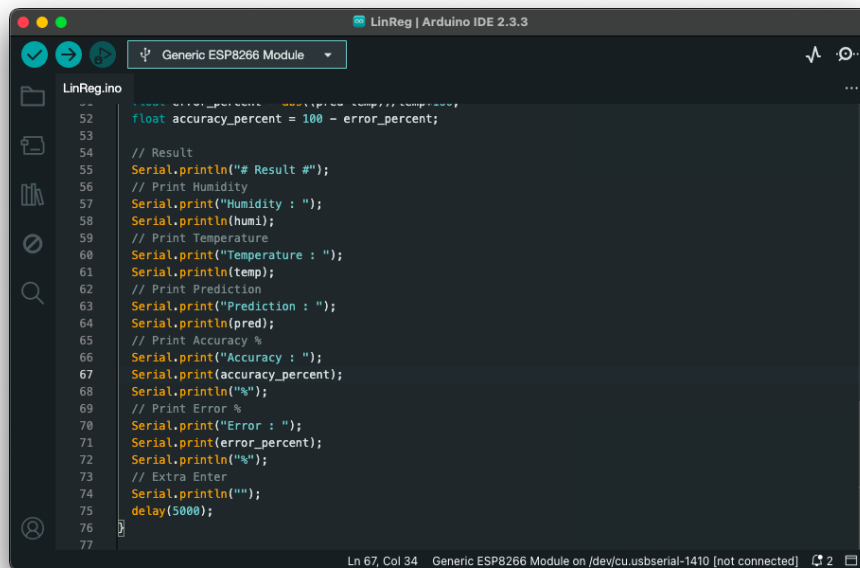
```
// Get Sensor Values
float temp = dht.readTemperature();
float humi = dht.readHumidity();
// Predit Temperature with Humi
float pred = slope * humi + intercept;
// Error and Accuracy
float error_percent = abs((pred-temp))/temp*100;
float accuracy_percent = 100 - error_percent;
```



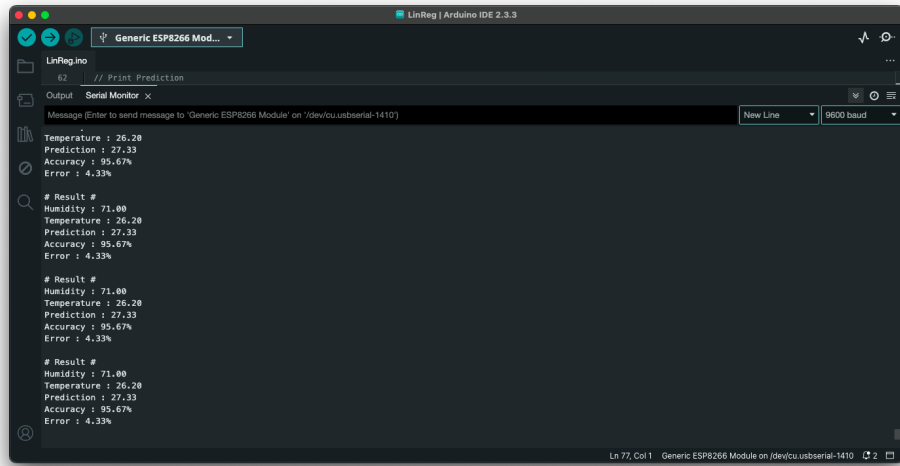
(b) Kode untuk menampilkan hasilnya, beserta delay 5 detik

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 lihast hasil akhir melalui **Serial Monitor**



Bab 7

Praktikum 7

7.1 ESP8266, DHT-22, dan Fuzzy Logic

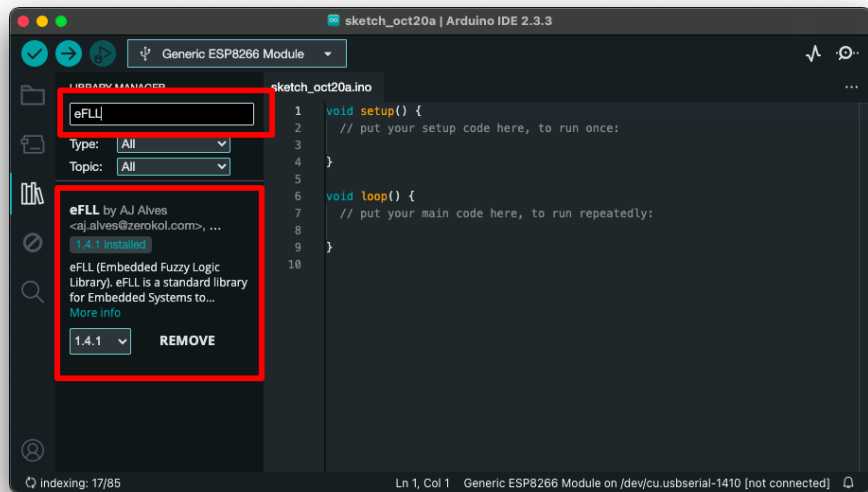
Di bagian ini mahasiswa diajarkan bagaimana mengimplementasikan algoritma *Fuzzy Logic* sederhana dengan perangkat **ESP8266**, dan **DHT-22**. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 6** yang ada di halaman sebelumnya.

7.2 Tutorial

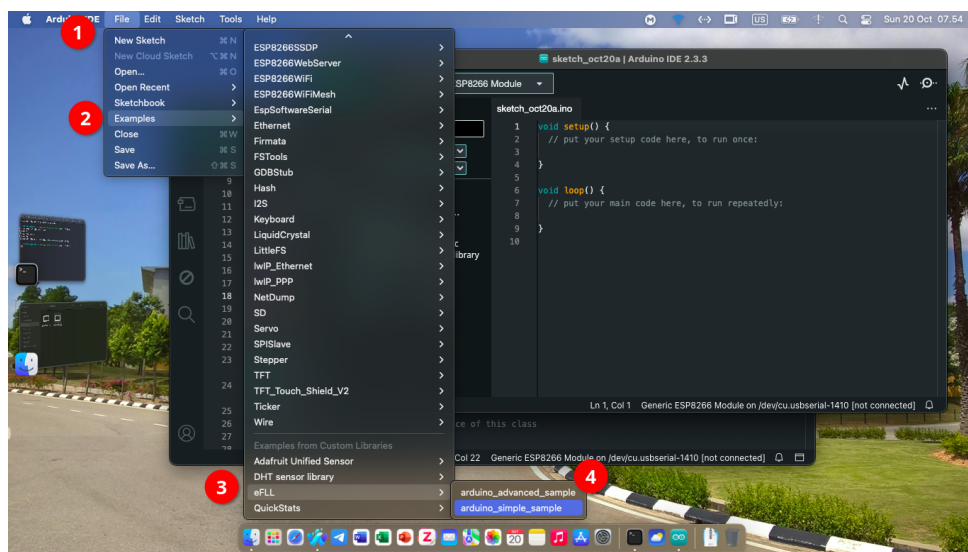
1. Buka Arduino IDE kembali



2. Install Library **eFLL**



3. Buka kode example dari **eFLL** dari menu **File** → **Examples** → **eFLL** → **arduino_simple_sample**.





4. Di bagian bawah kode `#include <Fuzzy.h>`, tambahkan kode library DHT-22

Kode Program

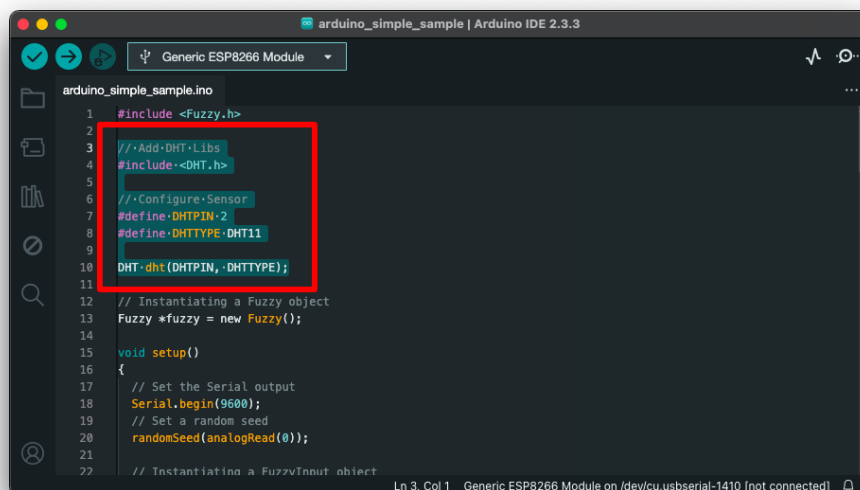
```

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

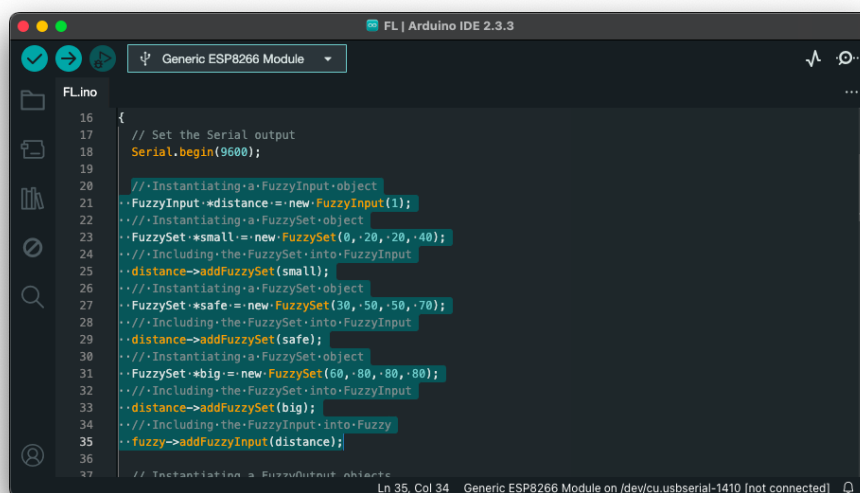
// Configure Sensor
#define DHTPIN 2
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

```



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



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

```

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

```

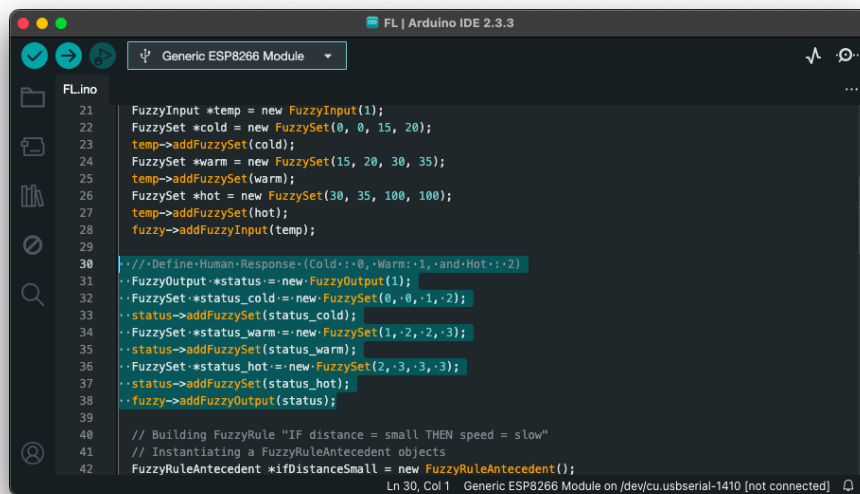
26 FuzzySet *hot = new FuzzySet(30, 35, 100, 100);
27 temp->addFuzzySet(hot);
28 fuzzy->addFuzzyInput(temp);
29
30 // Instantiating a FuzzyOutput objects
31 FuzzyOutput *speed = new FuzzyOutput(1);
32 // Instantiating a FuzzySet object
33 FuzzySet *slow = new FuzzySet(0, 10, 10, 20);
34 // Including 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

```

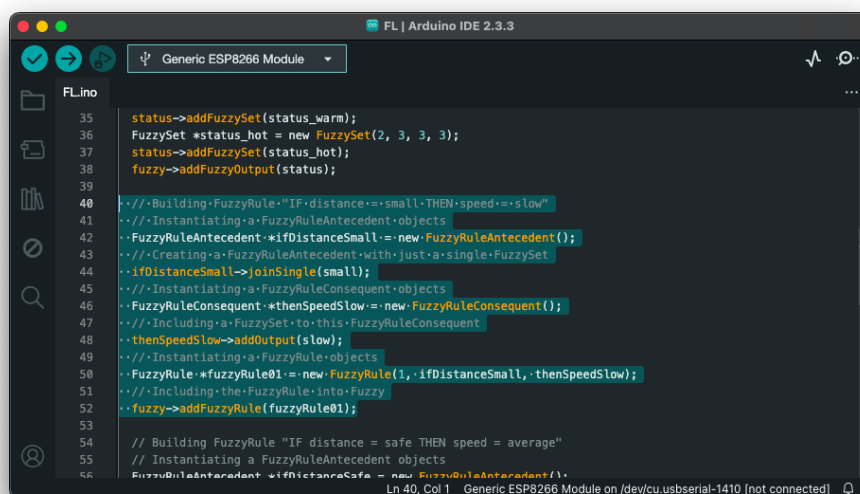
- Lalu tambahkan Kode Parameter Output Fuzzy yang dibuat agar mudah dipahami oleh manusia (Dingin = 1, Hangat = 2, dan Panas = 3) tepat di bawah Kode Parameter Input tadi.

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);
```



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



```

50 FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifDistanceSmall, thenSpeedSlow);
51 // Including the FuzzyRule into Fuzzy
52 fuzzy->addFuzzyRule(fuzzyRule01);
53
54 //Building FuzzyRule "IF distance is safe THEN speed is 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 is big THEN speed is 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 is big THEN speed is 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);

```

```

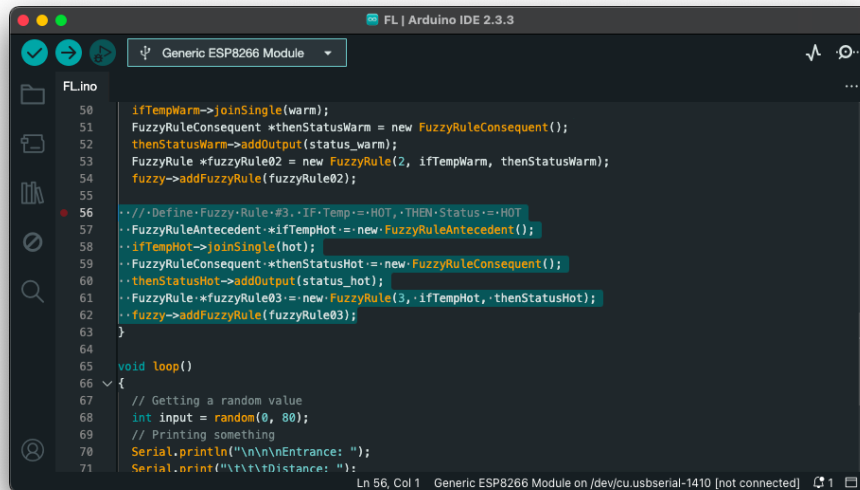
42 ifTempCold->joinSingle(cold);
43 FuzzyRuleConsequent *thenStatusCold = new FuzzyRuleConsequent();
44 thenStatusCold->addOutput(status_cold);
45 FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifTempCold, thenStatusCold);
46 fuzzy->addFuzzyRule(fuzzyRule01);
47
48 // Define Fuzzy Rule #2. IF Temp == WARM, 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);
```

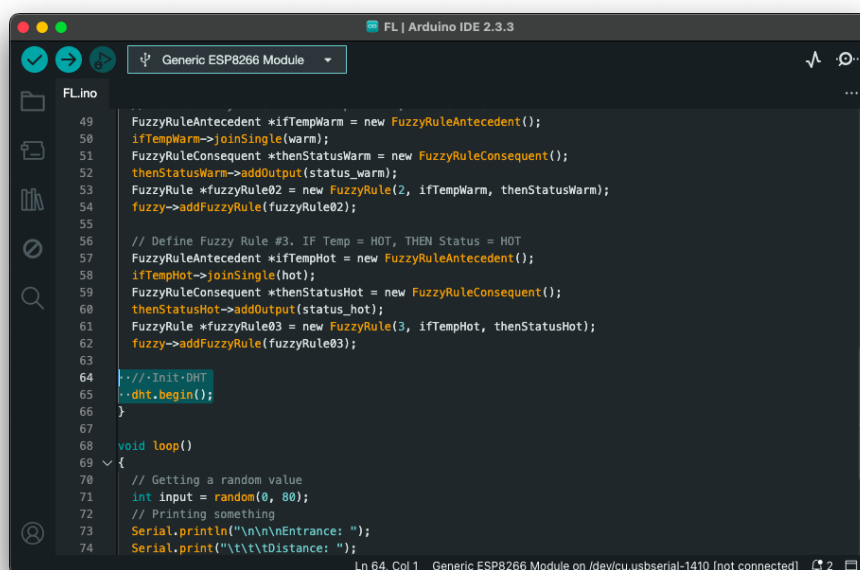


```
FL.ino
50 ifTempWarm->joinSingle(warm);
51 FuzzyRuleConsequent *thenStatusWarm = new FuzzyRuleConsequent();
52 thenStatusWarm->addOutput(status_warm);
53 FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifTempWarm, thenStatusWarm);
54 fuzzy->addFuzzyRule(fuzzyRule02);
55
56 // Define Fuzzy Rule #3. IF Temp = 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
64
65 void loop()
66 {
67 // Getting a random value
68 int input = random(0, 80);
69 // Printing something
70 Serial.println("\n\nEntrance: ");
71 Serial.print("\t\tDistance: ");
```

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

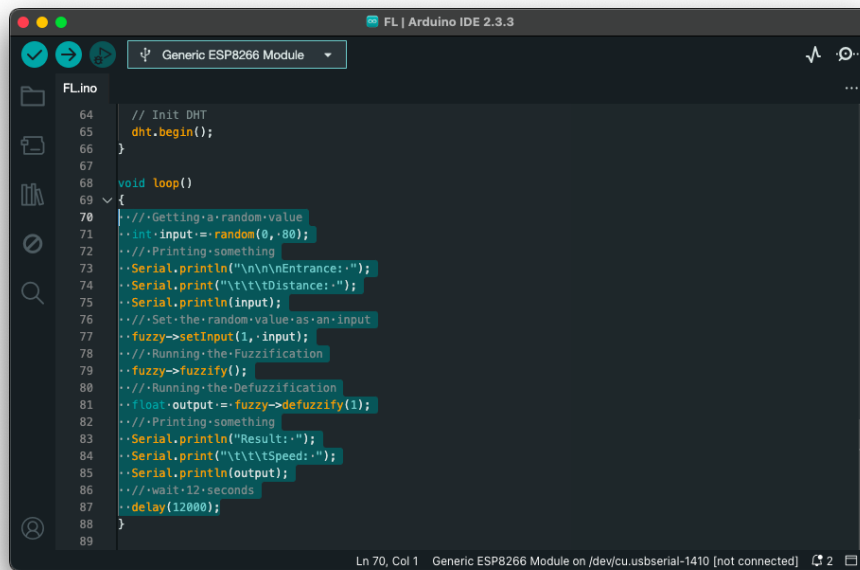
Kode Program

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



```
FL.ino
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
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: ");
```

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



The screenshot shows the Arduino IDE interface with the 'Generic ESP8266 Module' selected. The 'FL.ino' file is open, and the 'loop()' function is being edited. The code inside the loop function is as follows:

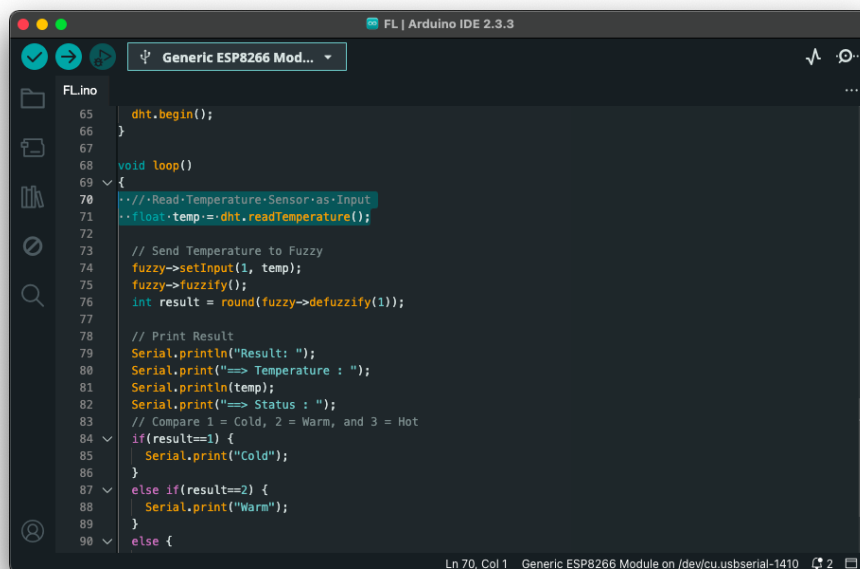
```
64 // Init DHT
65 dht.begin();
66 }
67
68 void loop()
69 {
70   //Getting a random value
71   int input = random(0, 80);
72   //Printing something
73   Serial.println("\n\nEntrance:");
74   Serial.print("\t\tDistance:");
75   Serial.println(input);
76   //Set the random value as an input
77   fuzzy->setInput(1, 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\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();
```



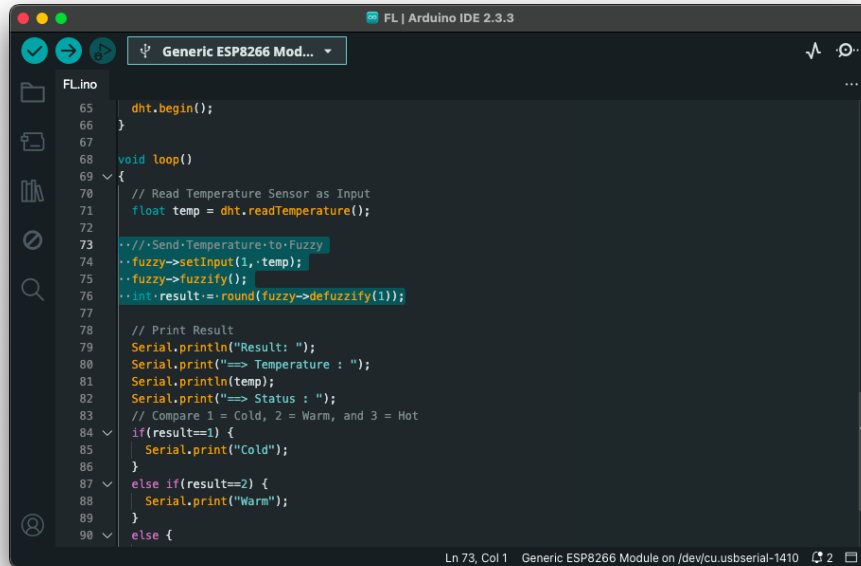
The screenshot shows the Arduino IDE interface with the 'Generic ESP8266 Module' selected. The 'FL.ino' file is open, and the 'loop()' function is being edited. The code inside the loop function is as follows:

```
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 {
91
92   }
93 }
```

(b) Berikutnya kode Fuzzifikasi dan Defuzzifikasi

Kode Program

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



(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 DHT-22, dan unggah kode. Lihat hasil seperti berikut

```

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** → DHT-22 (DHT22 jika paham)
 3. **Opsional** → Breakout Board ESP8266
 - Wajib menggunakan **Regresi Liniear**
 - 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)