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

Daftar Gambar

Pendahuluan

0.1 Mengenal *Internet of Things*

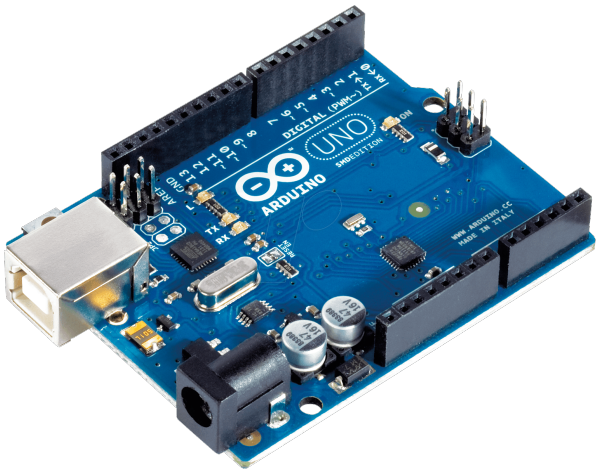
Internet of Things merupakan sebuah teknologi yang di mana mengizinkan setiap perangkat-perangkat yang memiliki kekuatan komputasi untuk berkomunikasi satu dengan yang lainnya tanpa campur tangan manusia untuk menyelesaikan suatu tugas atau fungsi.
Teknologi ini dapat diimplementasikan ke berbagai macam hal tergantung dari tugas atau fungsi yang ingin dicapai. Sebagai contoh untuk mendesain sebuat rumah pintar yang dapat mendeteksi lingkungan sekitar dan melakukan otomatisasi berdasarkan data tersebut.

Gambar 1: Internet of Things

0.2 Perangkat Board IoT

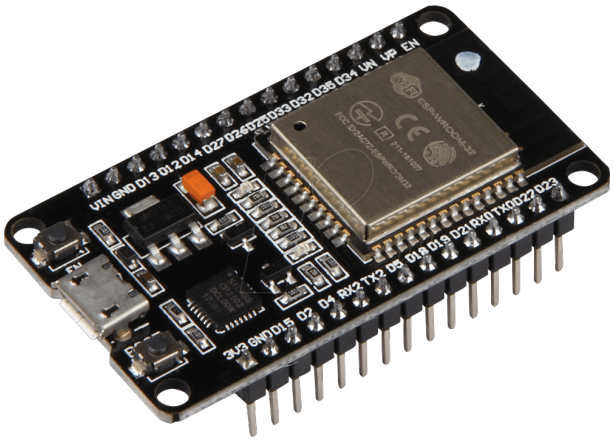
Untuk membangun sebuah perangkat berbasis IoT, komponen dasar seperti **Board** sangatlah vital untuk dipunyai. Terdapat berbagai macam board yang dapat dibeli secara luring maupun daring, dengan variasi harga yang juga berbeda mulai dari paling murah hingga mewah. Semakin kompleks masalah yang dapat 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



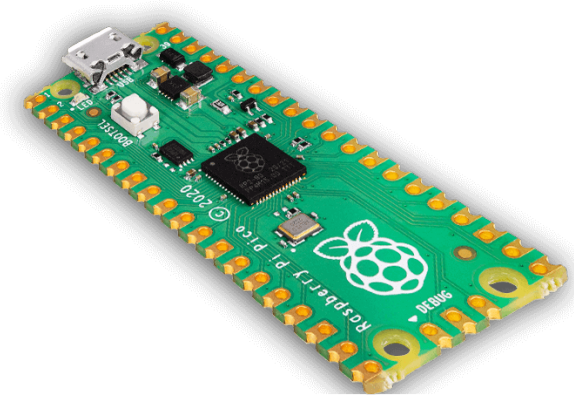
Gambar 2: Board Arduino

2. NodeMCU



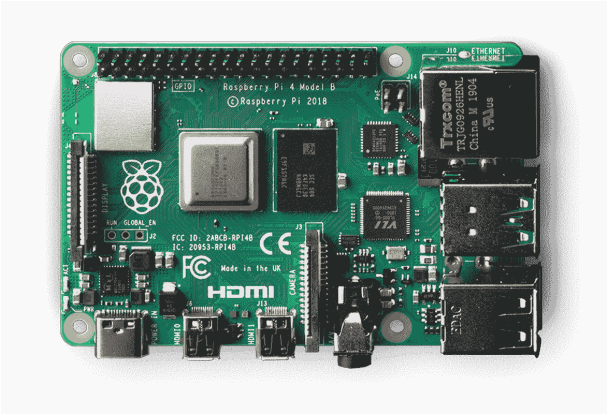
Gambar 3: Board NodeMCU

3. Raspberry Pi Pico



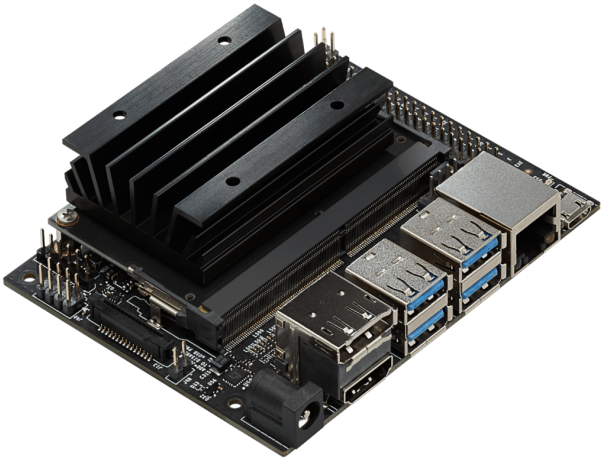
Gambar 4: Board Pico

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



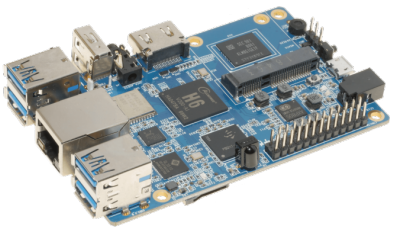
Gambar 5: Board Pi 4B

5. NVidia Jetson



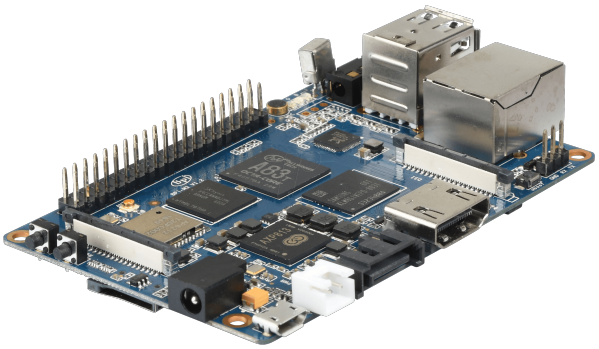
Gambar 6: Board NVidia Jetson

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

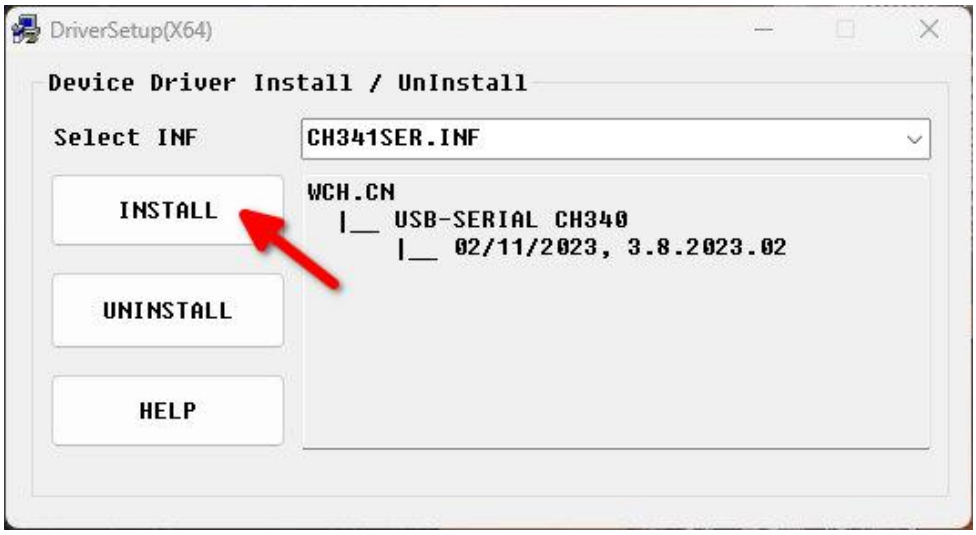
Praktikum 1

1.1 Konfigurasi Arduino IDE dan ESP8266

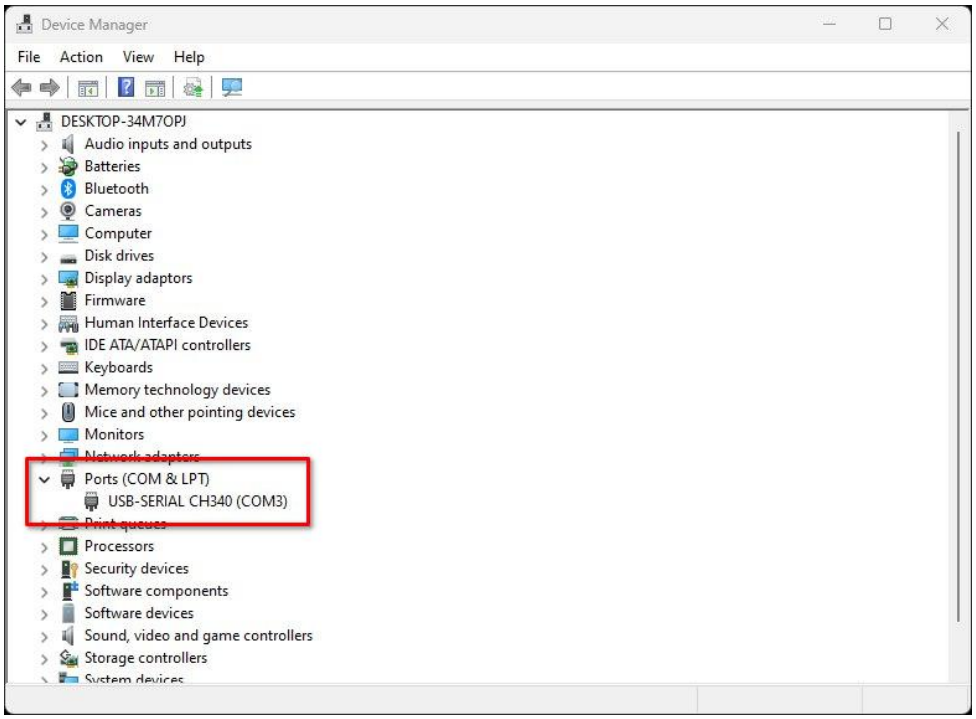
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke komputer beserta konfigurasinya hingga dapat dikenali oleh Arduino IDE. Mahasiswa diharapkan untuk membaca, dan memahami **Persiapan Praktikum** yang ada di halaman sebelumnya.

1.2 Tutorial

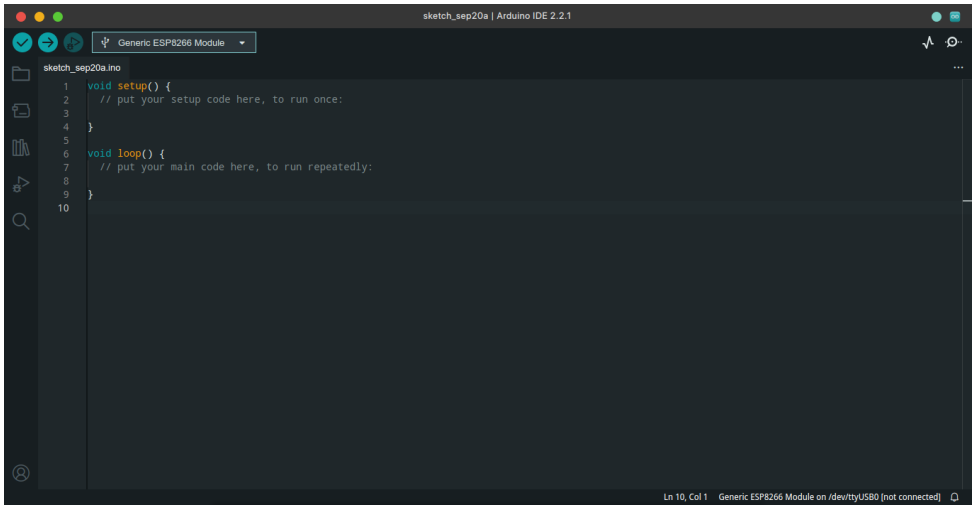
1. Setelah mahasiswa menyiapkan perangkat-perangkat yang diperlukan, maka langkah berikutnya adalah melakukan instalasi driver terlebih dahulu.
2. File driver **CH341SER** yang sudah diunduh, dibuka untuk diinstall. Cukup klik **Install** untuk memasang driver (Windows 10 ke bawah)



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

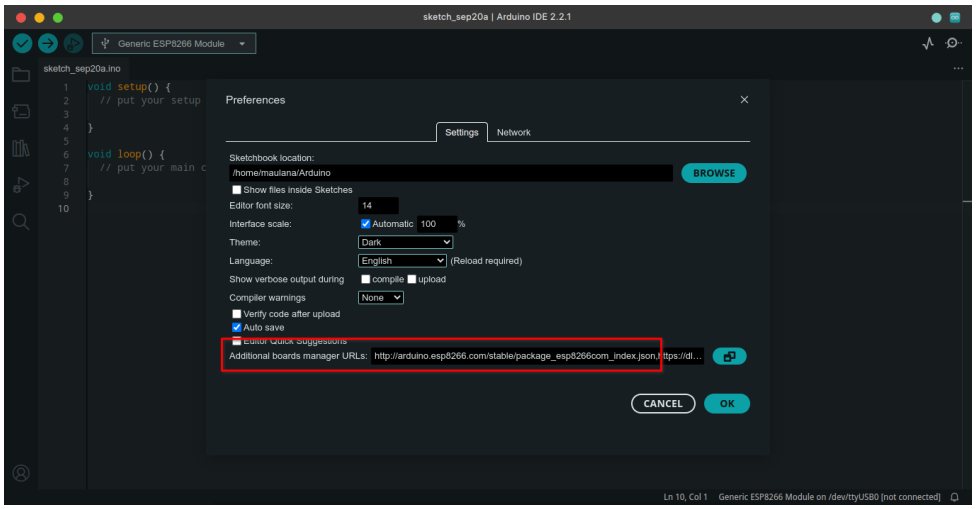


4. Langkah berikutnya adalah mengunduh **Arduino IDE**, usahakan untuk mendapatkan versi terbaru. Setelah unduh, buka aplikasi tersebut

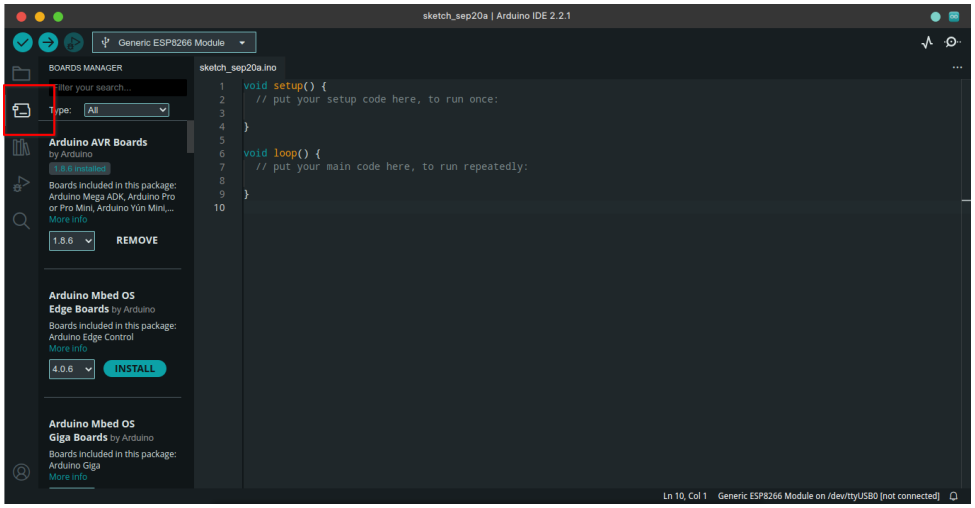


5. Namun **Arduino IDE** ini belum mendukung perangkat yang kita gunakan. Langkah berikutnya buka **File** → **Preferences** →. Tambahkan baris **Alamat URL** berikut ke **Additional board manager URLs**. Klik **OK** untuk mengupdate otomatis.

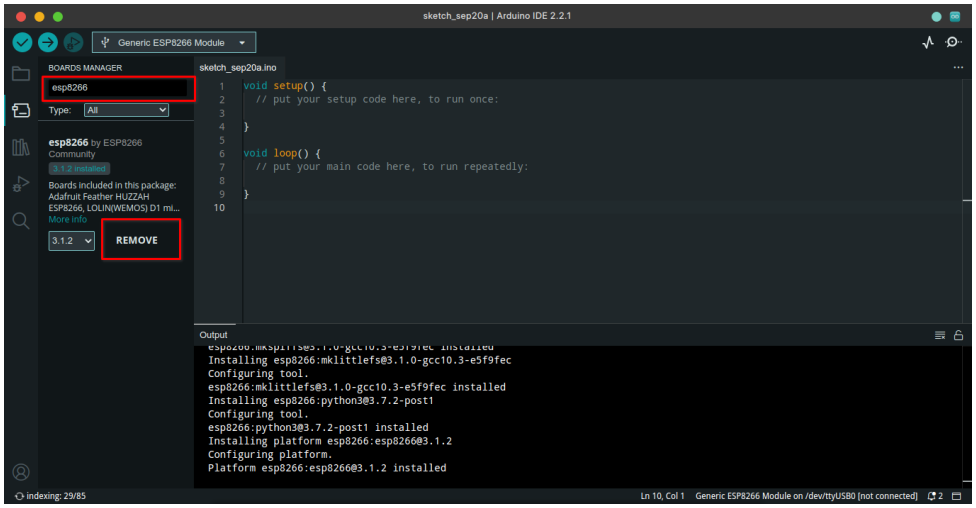
   - http://arduino.esp8266.com/stable/package_esp8266com_index.json



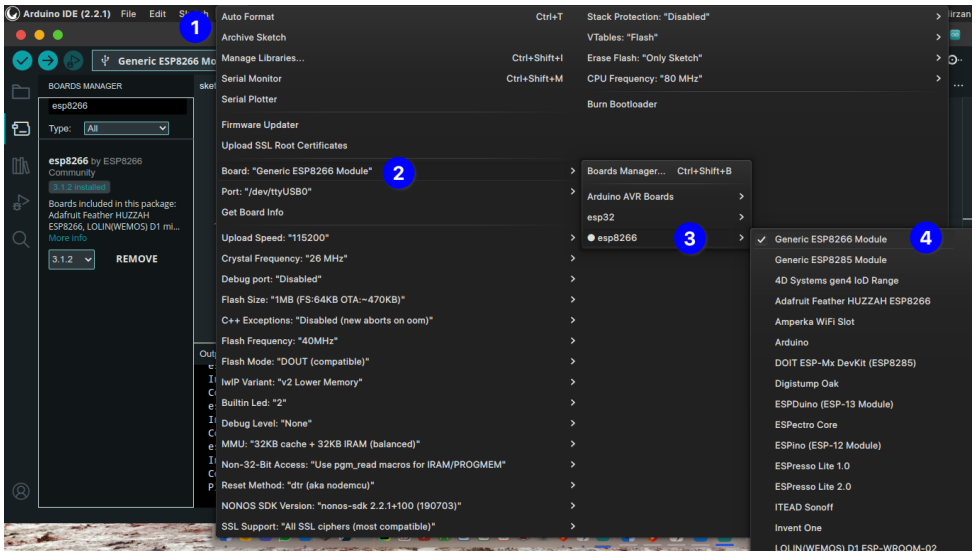
6. Jika sudah, install driver **ESP8266** dengan klik **Boards Manager** di **Sidebar Kiri** atau **Tools** → **Board:** → **Boards Manager**



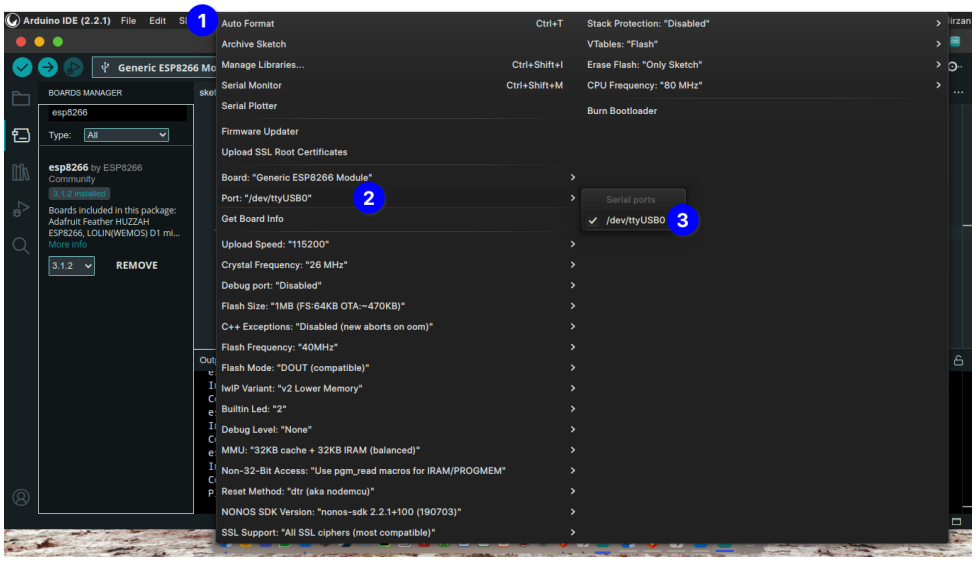
7. Di kolom **Pencarian**, ketik **ESP8266** dan klik **Install**



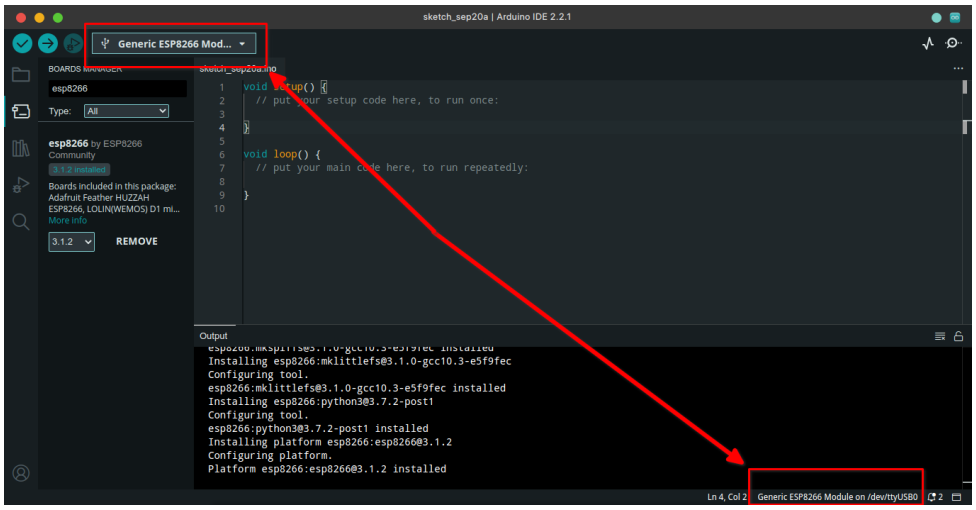
8. **Arduino IDE** sudah siap, namun belum terhubung ke perangkat. Untuk menghubungkan antara **IDE** dengan **ESP8266**, pilih **Tools** → **Board:** → **esp8266** → **Generic ESP8266 Module**



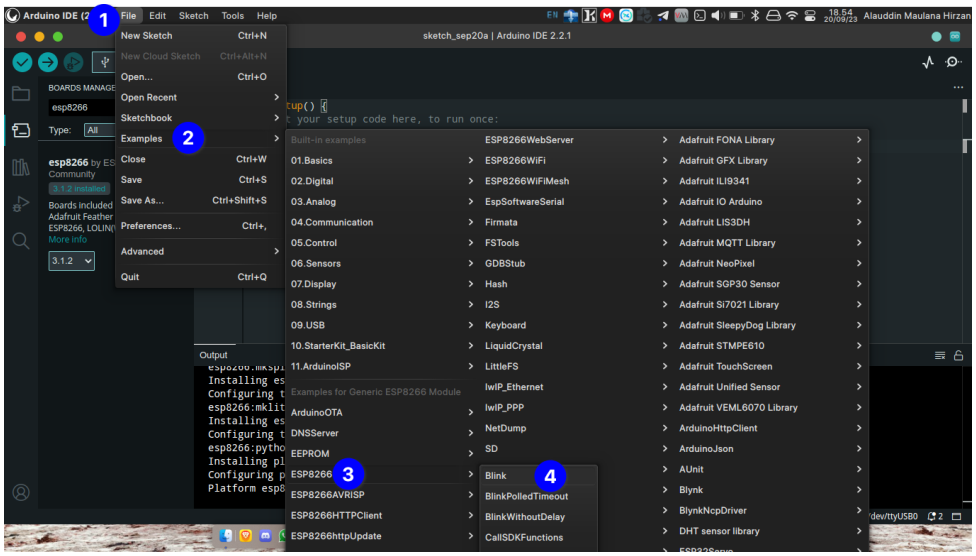
9. Kemudian pastikan **Port Serial** yang digunakan, sama dengan yang ada di **Device Manager**. Cek dengan menu **Tools** → **Port:** → **Pilih COM Sesuai Device Manager**



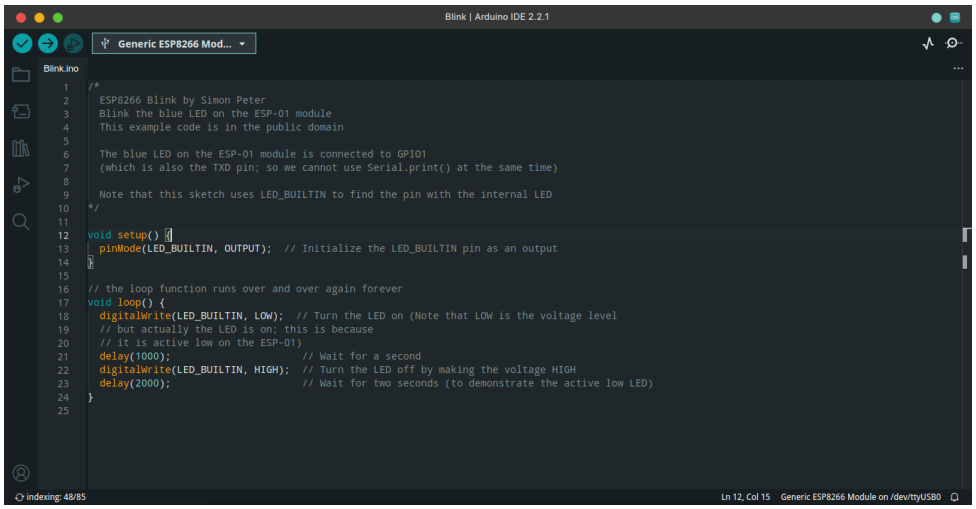
10. Jika sudah terhubung, akan ada tanda tulisan **Generic ESP8266 Module on COMXXX** di bawah kanan maupun simbol USB di atas kiri



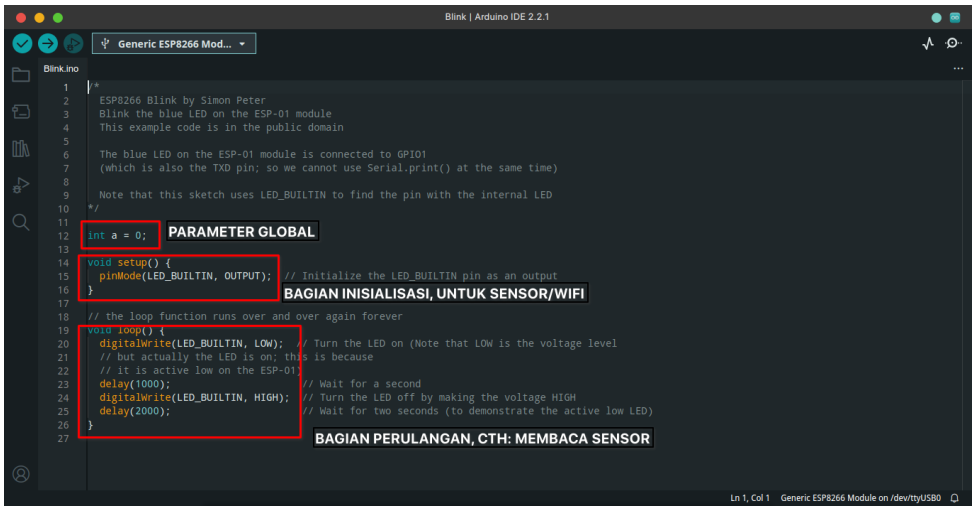
11. **NodeMCU ESP8266** siap diujikan. Untuk menguji alat, **Arduino IDE** sudah menyiapkan template dasar seperti **LED Blinking**. Untuk mengakses kode ini buka menu **File** → **Examples** → **ESP8266** → **Blink**



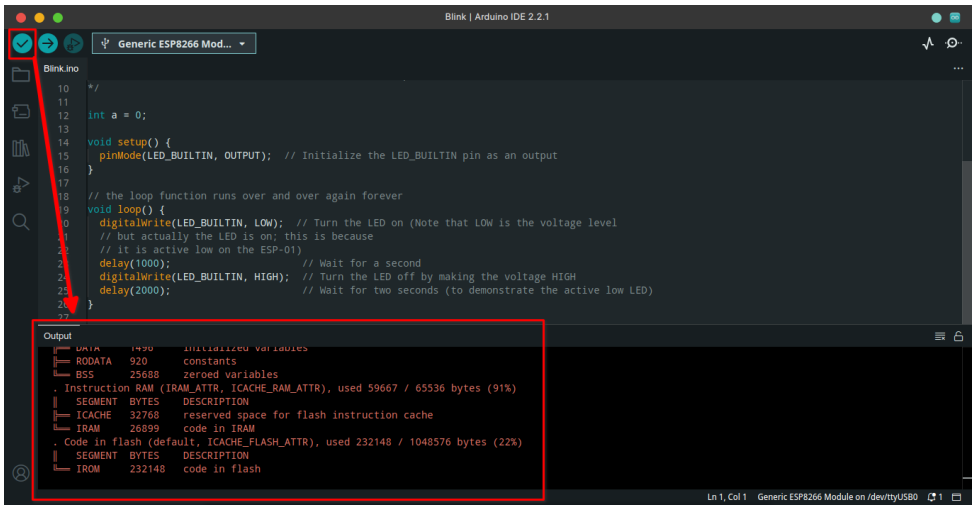
12. **Arduino IDE** akan membuka **Window Baru**. Tutup **Window** sebelumnya agar tidak terganggu.



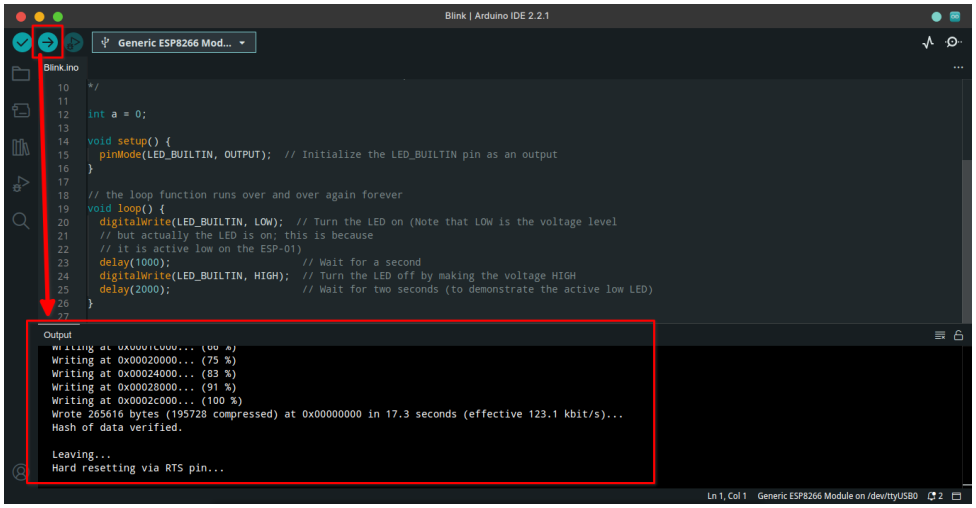
13. Mahasiswa **WAJIB MEMAHAMI ALUR KODE**. Kode dieksekusi dari atas ke bawah. **Fungsi SETUP** digunakan untuk mengatur inisialisasi yang dilakukan **SATU KALI**. Sedangkan **Fungsi LOOP** digunakan untuk proses yang diulang-ulang oleh alat. Kode-kode di atas kedua fungsi tersebut dianggap sebagai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



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

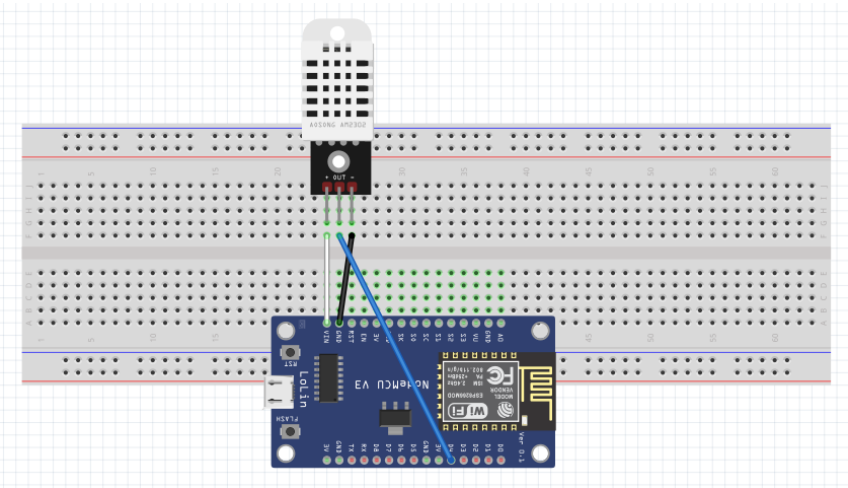
Praktikum 2

2.1 ESP8266, DHT-22, dan AdafruitIO

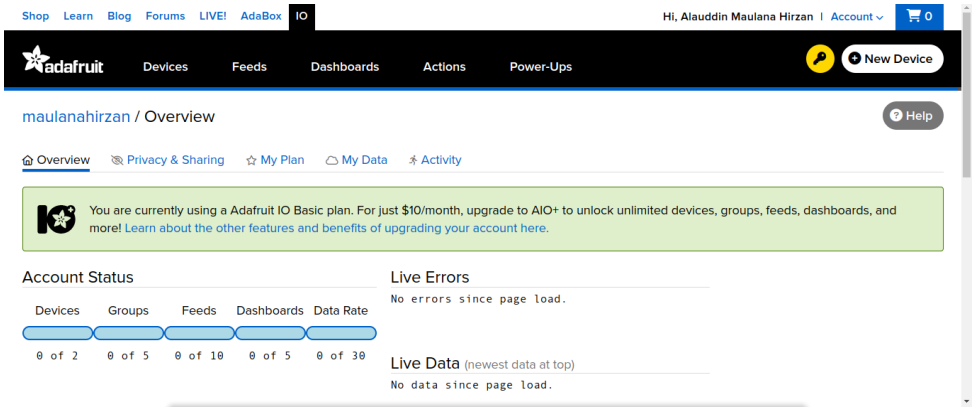
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke sensor DHT-22 dan bagaimana menyimpan data secara daring di layanan AdafruitIO. Mahasiswa diwajibkan memahami **Praktikum 1** yang ada di halaman sebelumnya.

2.2 Tutorial

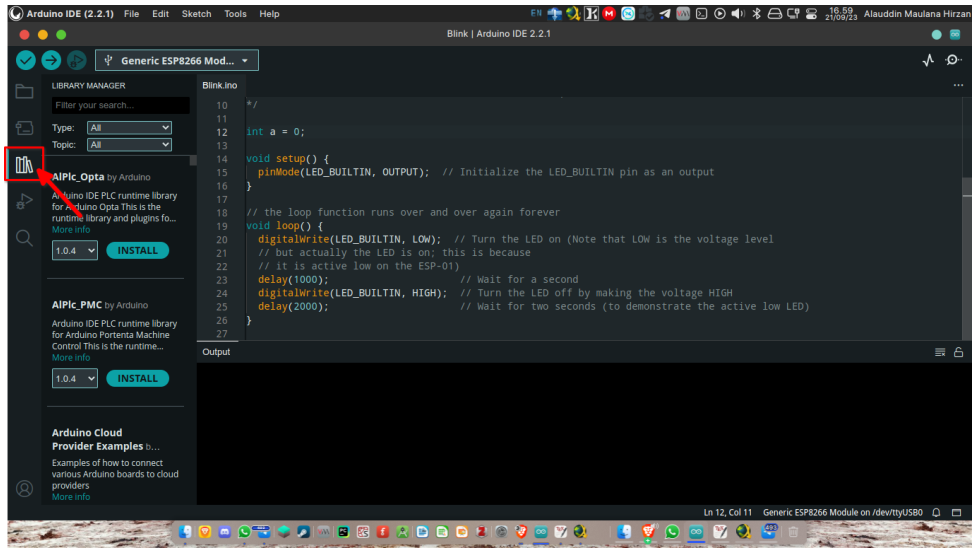
1. Langkah pertama yang perlu dilakukan adalah memasang sensor ke perangkat. Perlu diketahui bahwa dalam memasang sensor harus dalam keadaan **MATI/TIDAK TERTANCAP** untuk menghindari KORSLETING
2. Perhatikan sensor **DHT-22**, di bagian kakinya ada tanda **Plus** +, **Minus -**, dan **Out**. Sambungkan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



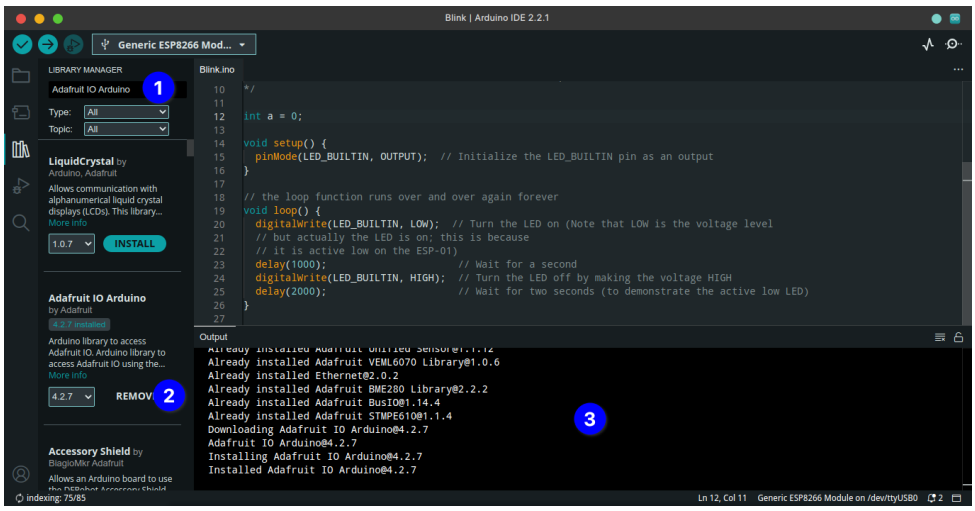
3. Setelah selesai menancapkan sensor, berikutnya adalah melakukan registrasi ke website AdafruitIO dengan link : `https://io.adafruit.com/`. Setelah teregistrasi akan terlihat dasbor seperti berikut:



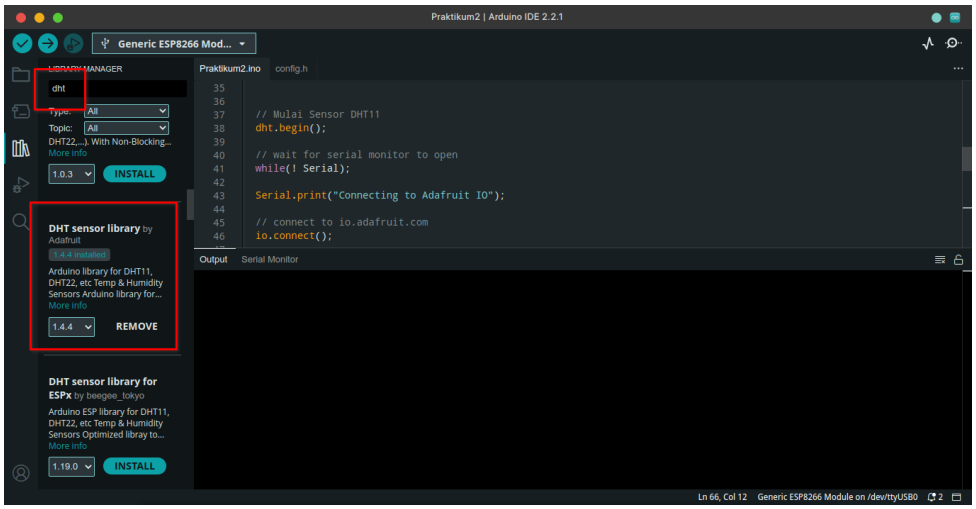
4. Kembali ke **Arduino IDE**, dan install **Library** dengan mengakses menu samping atau **Sketch** → **Include Library** → **Manage Libraries**



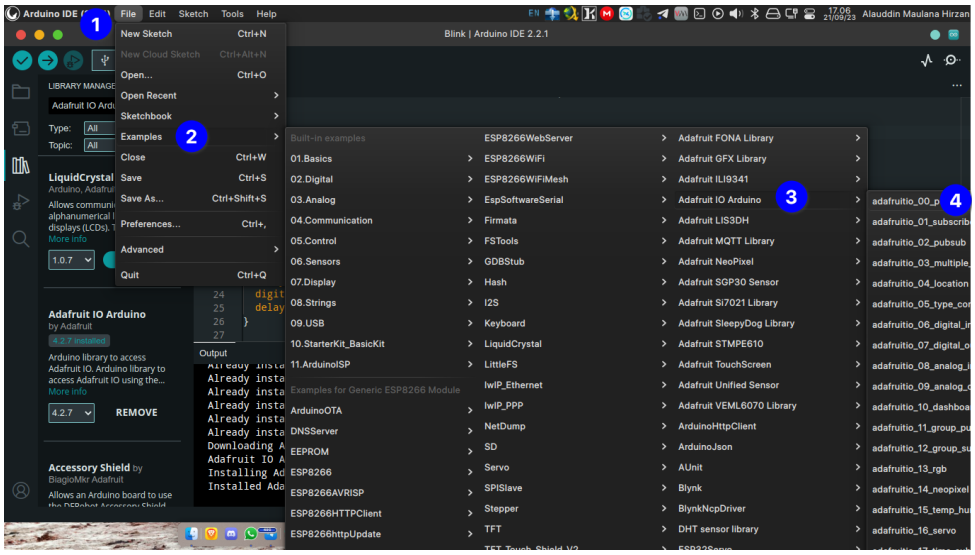
5. Cari **Adafruit IO Arduino**, klik **INSTALL**, lalu **INSTALL ALL**



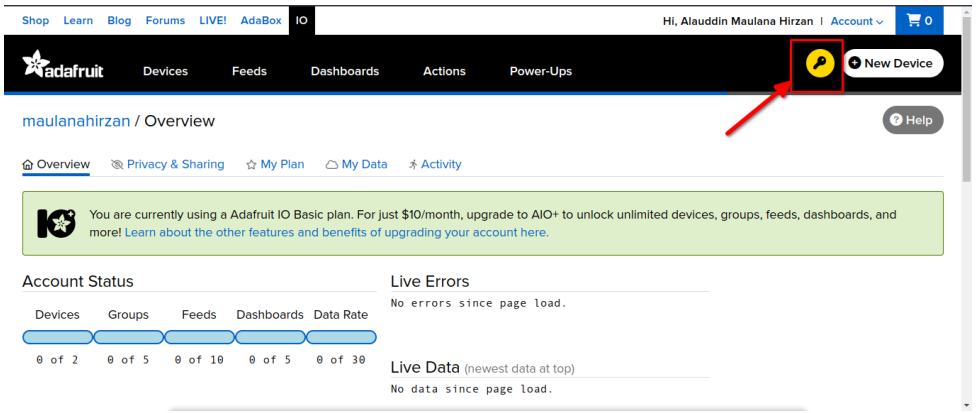
6. Cari **DHT sensor Library**, klik **INSTALL**, lalu **INSTALL ALL**



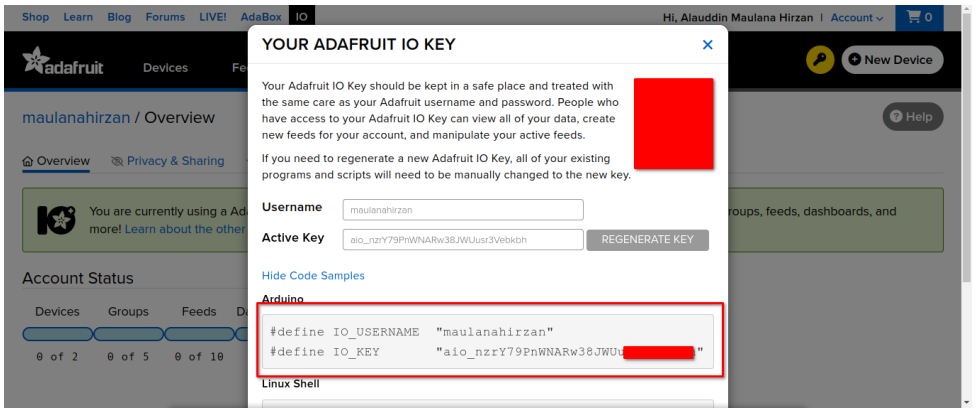
7. Sesudah install, berikutnya adalah membuka **Template Adafruit IO**. Klik menu **File** → **Examples** → **Adafruit IO Arduino** → **adafruit_00_publish**. Tutup **Arduino IDE** lain agar fokus

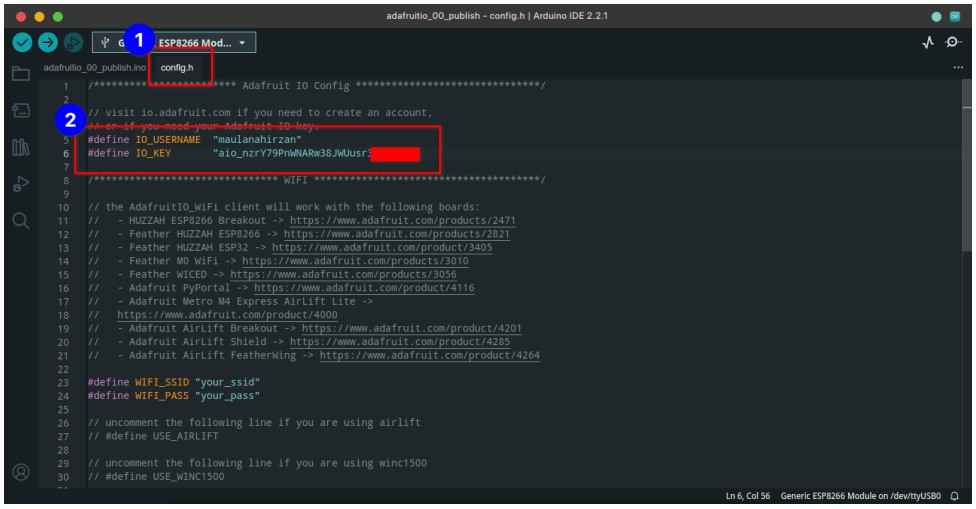


8. Jika sudah terbuka, kembali lagi ke website **Adafruit IO**. Klik **Icon Kunci Kuning** untuk menambahkan perangkat.

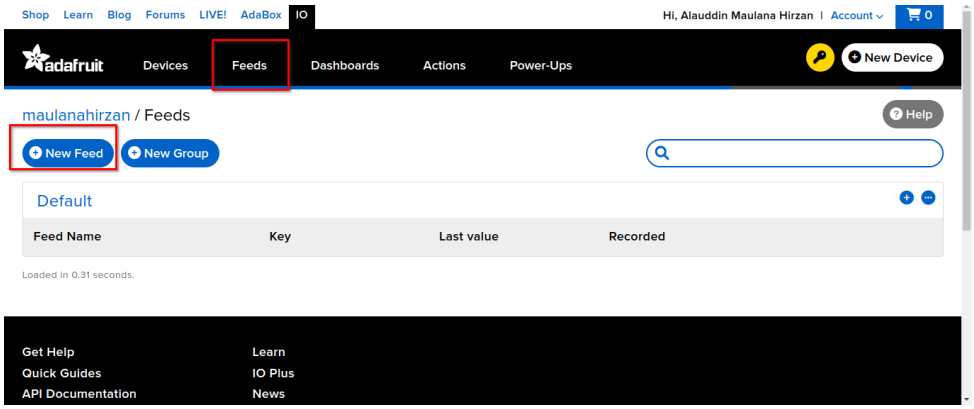


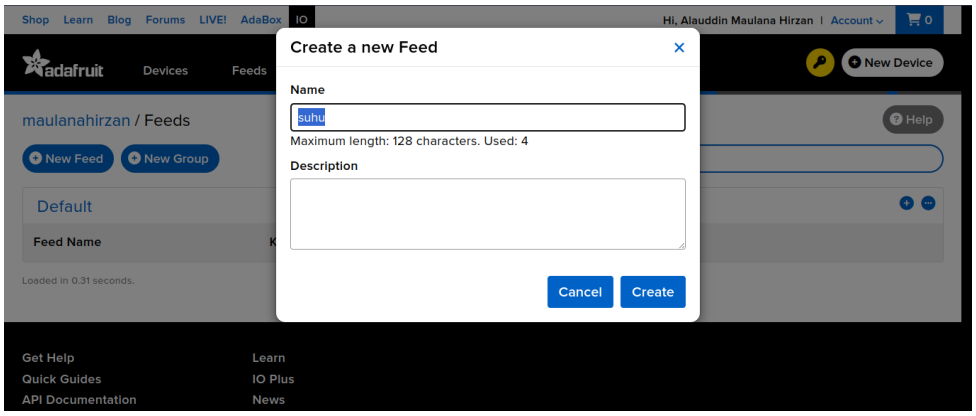
9. **Adafruit IO** akan membuat kunci yang akan dimasukkan ke **Sketch Arduino IDE**. Lihat bagian yang ditandai dan tempelkan ke file **config.h** di **Tab Arduino IDE**

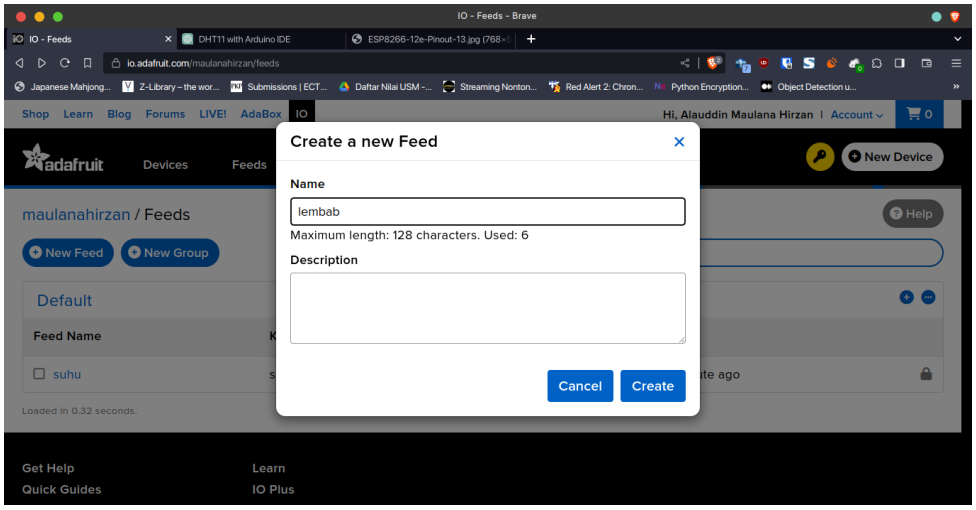




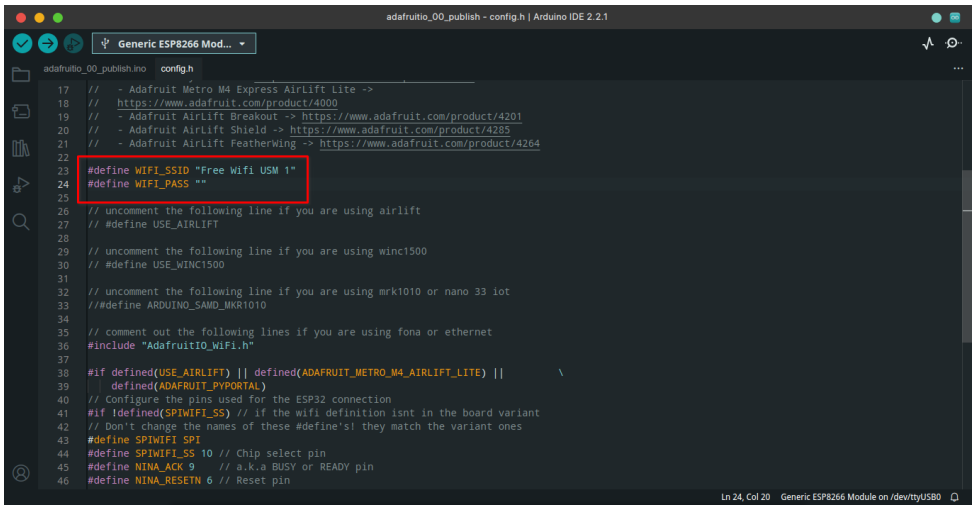
10. Jika sudah, buatlah **Feed** terlebih dahulu dengan meng klik **Menu Feeds**. Lalu buat 2 **Feed** baru dengan nama **suhu** dan **lembab**





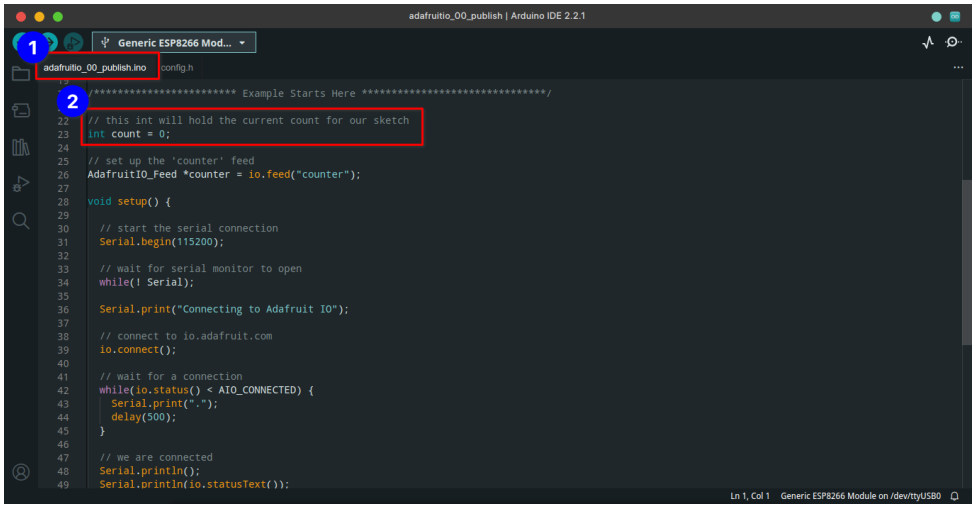


11. Lalu kembali ke **config.h** dan ubah SSID Wifi dan Passwordnya di bagian bawahnya



12. Konfigurasi **Adafruit IO** sudah selesai, berikutnya adalah memasukkan kode untuk mengambil data sensor. Kembali ke tab **arduino_00_publish.ino**

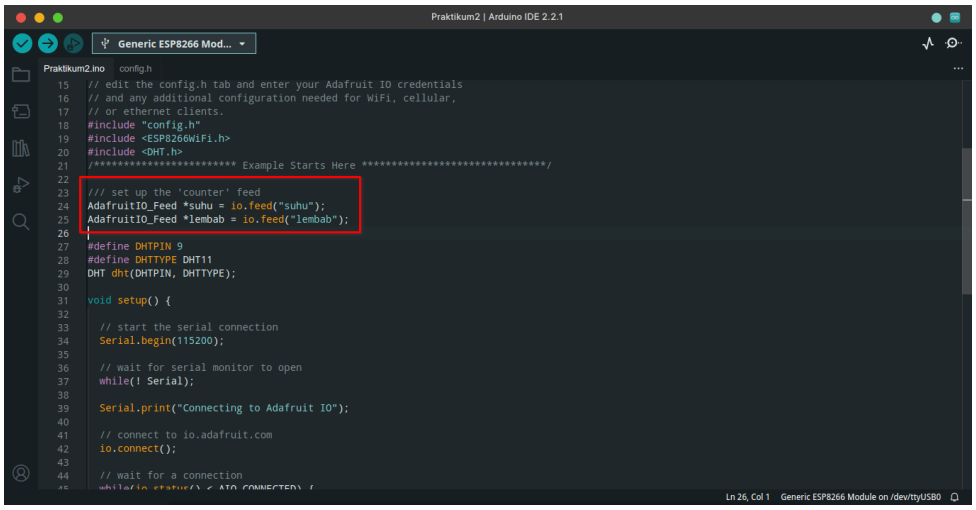
13. Lalu hapus kode yang ditandai



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

Potongan Kode

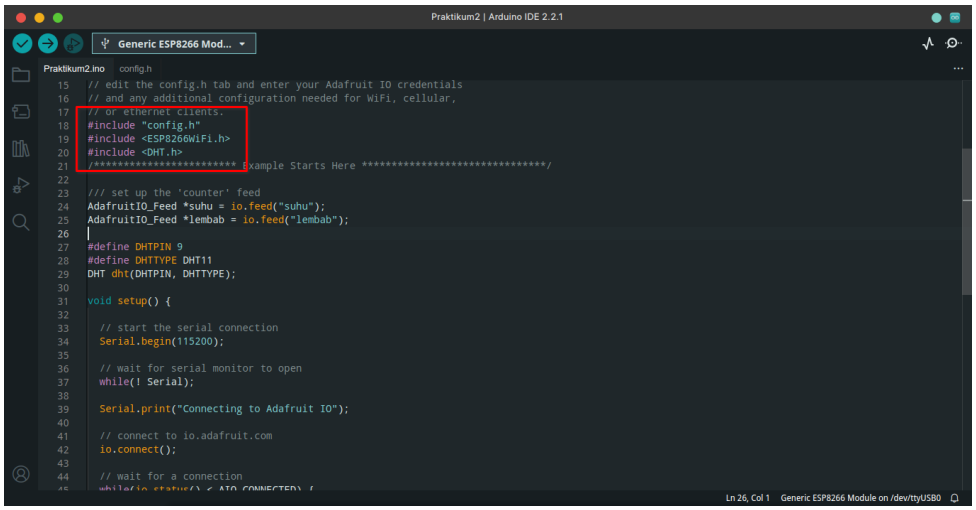
```
AdafruitIO_Feed *suhu = io.feed("suhu");
AdafruitIO_Feed *lembab = io.feed("lembab");
```



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

Potongan Kode

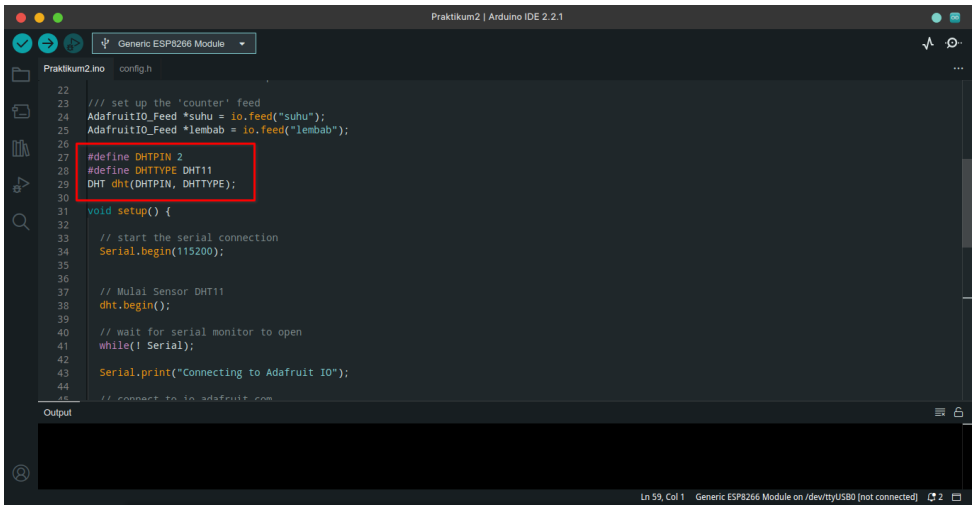
```
#include <ESP8266WiFi.h>
#include <DHT.h>
```



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

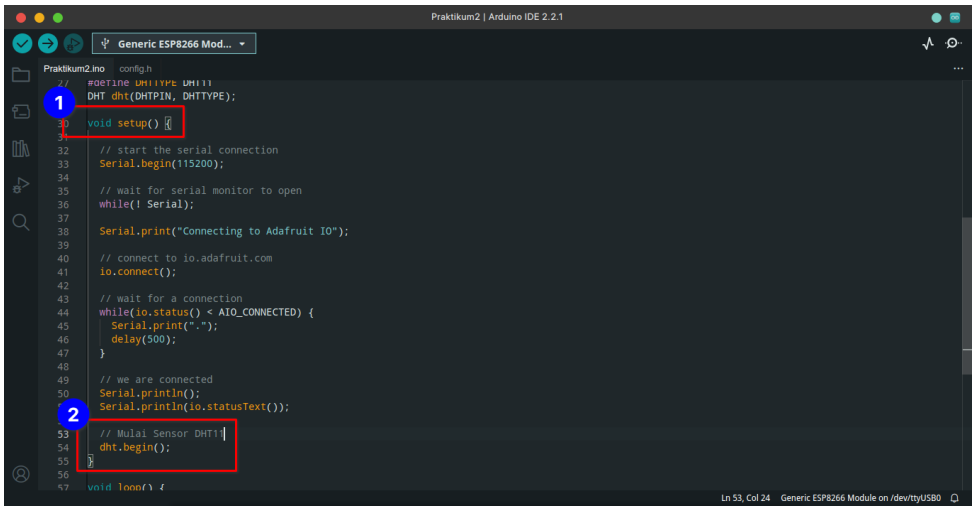
```
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
```



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

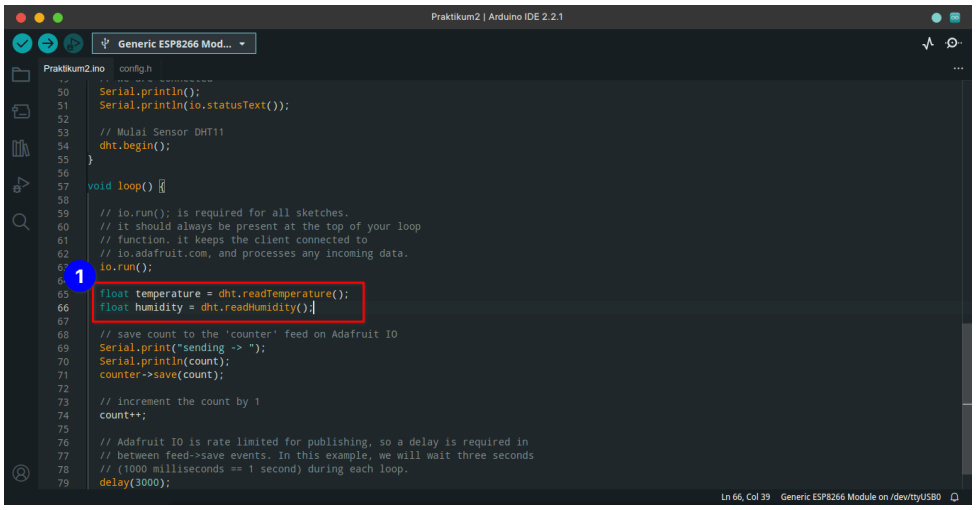
```
// Mulai Sensor DHT-22
dht.begin();
```



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

Potongan Kode

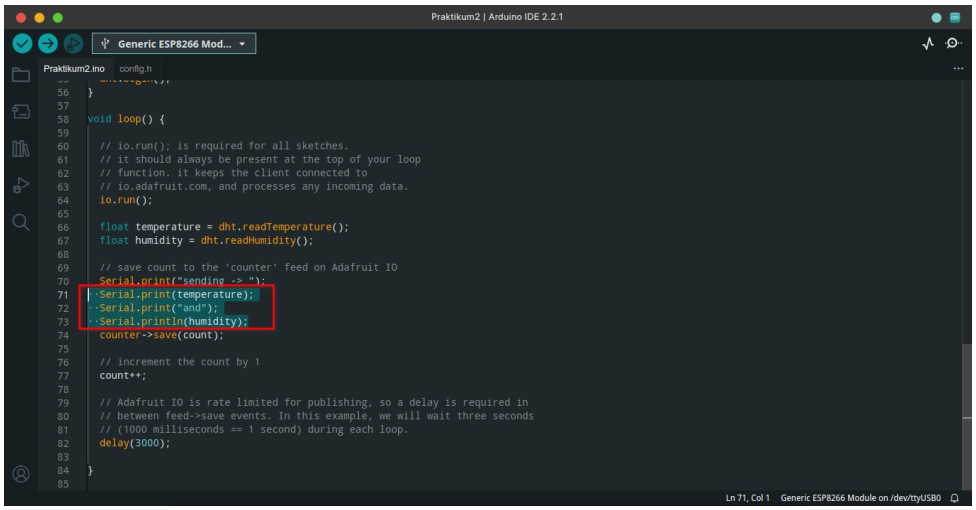
```
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
```



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

Potongan Kode

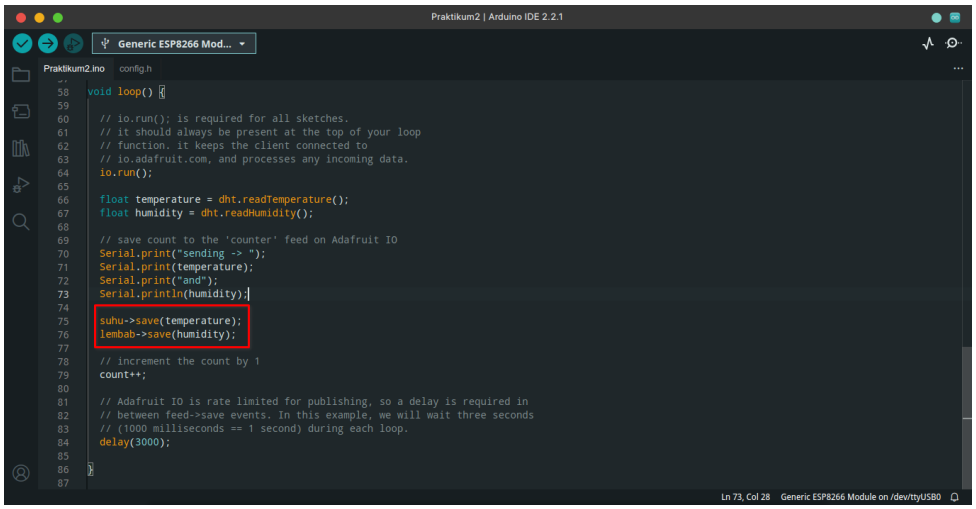
```
Serial.print(temperature);
Serial.print(" and ");
Serial.println(humidity);
```



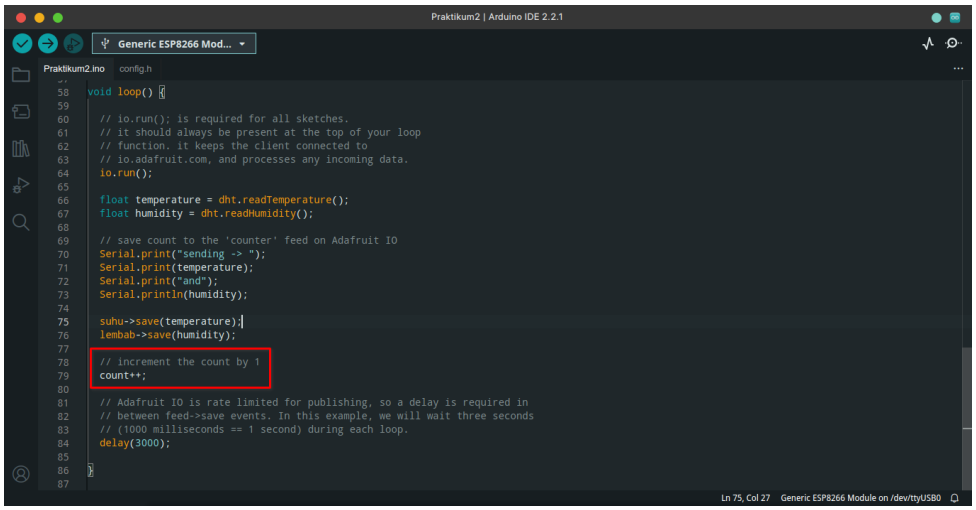
20. Bagian terakhir yang perlu diubah adalah proses unggahnya. Ganti kode **counter->save(count);** menjadi

Potongan Kode

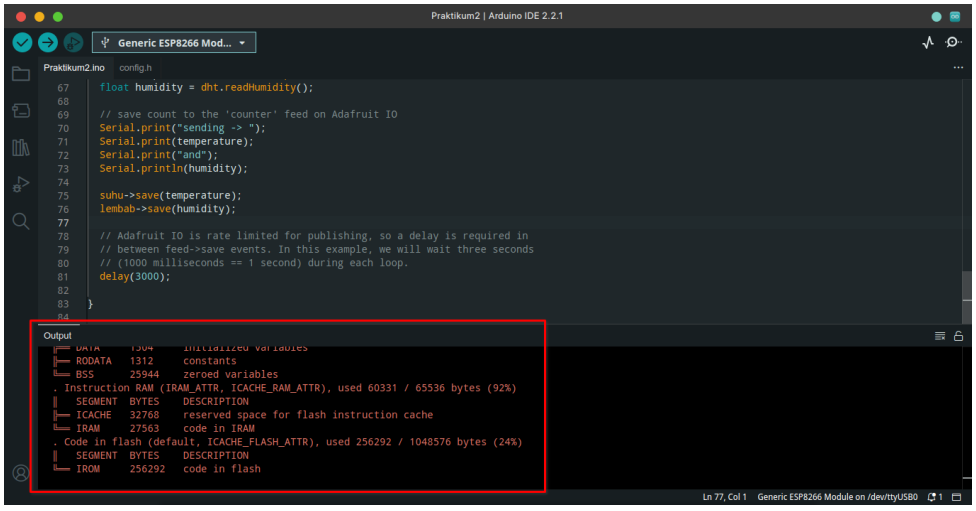
```
suhu->save(temperature);
lembab->save(humidity);
```



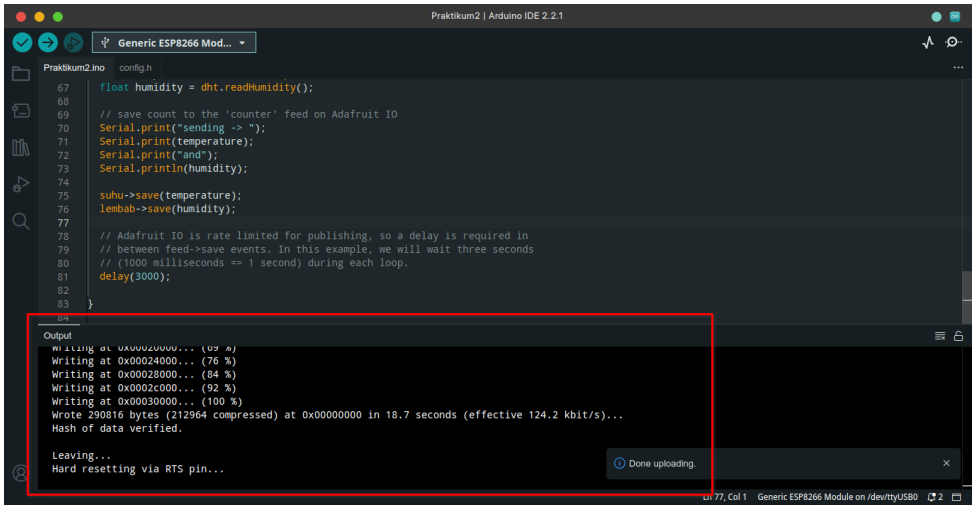
21. Terakhir, hapus kode increment **count++;**



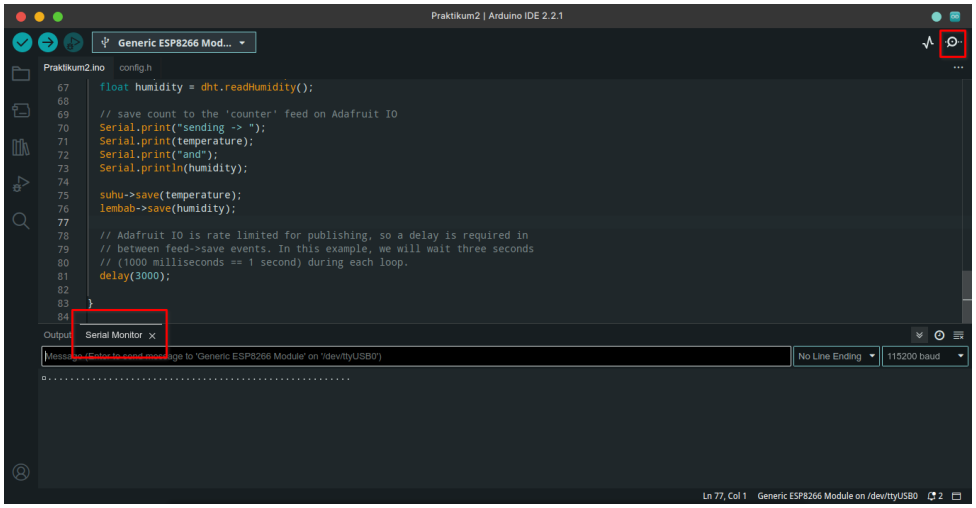
22. Verifikasi kode. Jika tidak ada **Error** seperti digambar. Lanjutkan dengan **Upload**. Pastikan **NodeMCU** tertancap



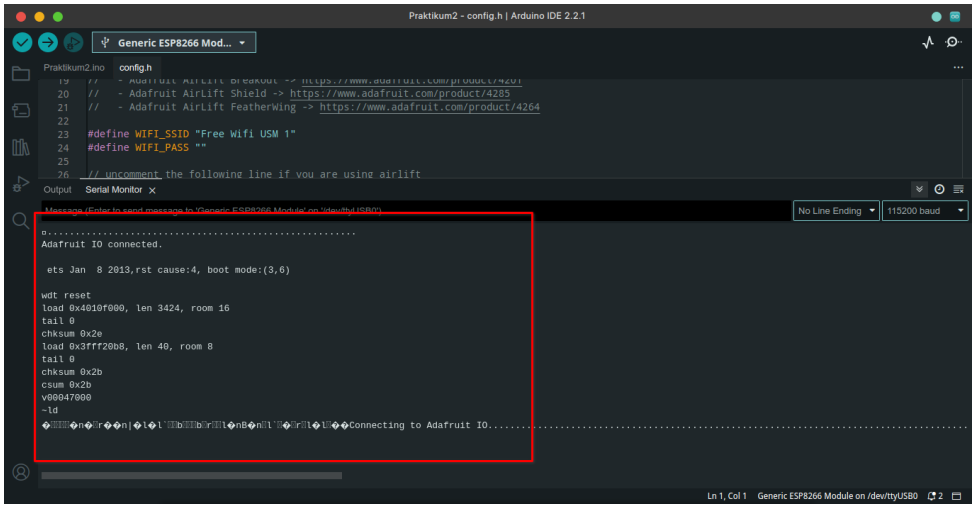
23. Unggah sudah sukses



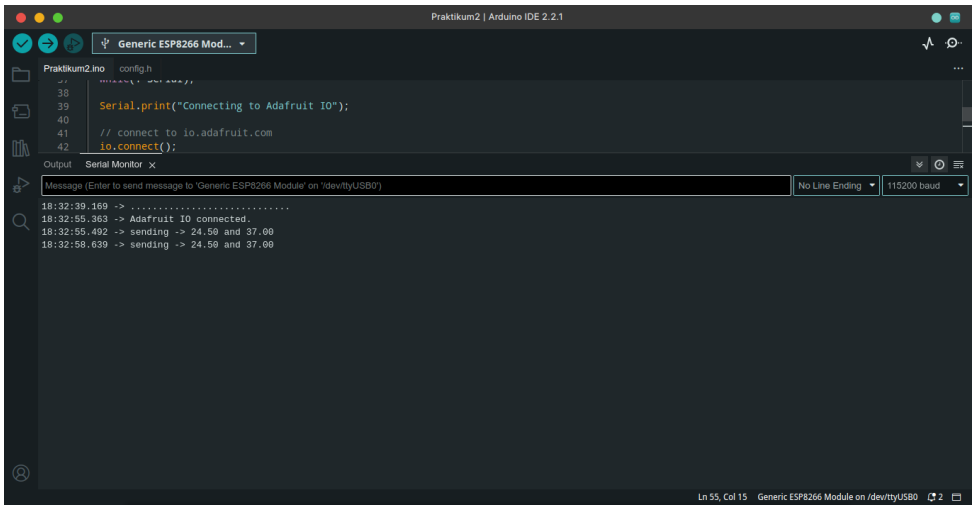
24. Berikutnya adalah mengecek alat. Klik **Tools** → **Serial Monitor**. Pastikan **BAUD** sudah sesuai dengan kode (biasanya 115200 baud)



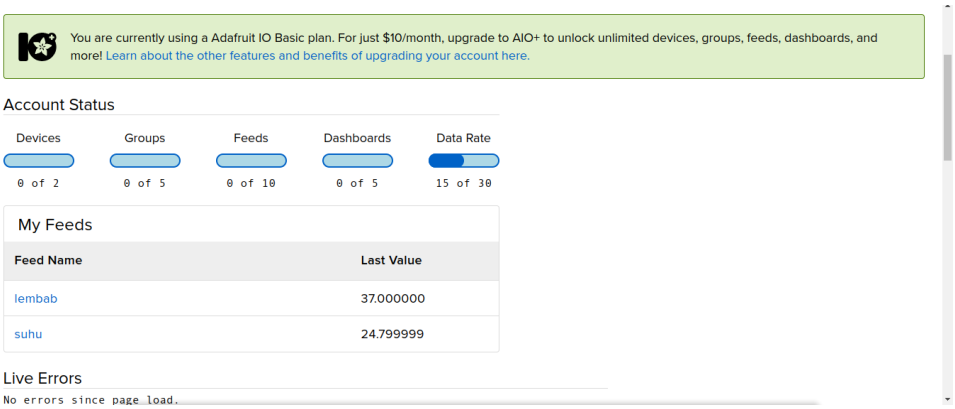
25. Jika proses koneksi lama, cek WiFi SSID apakah sudah benar atau lemot



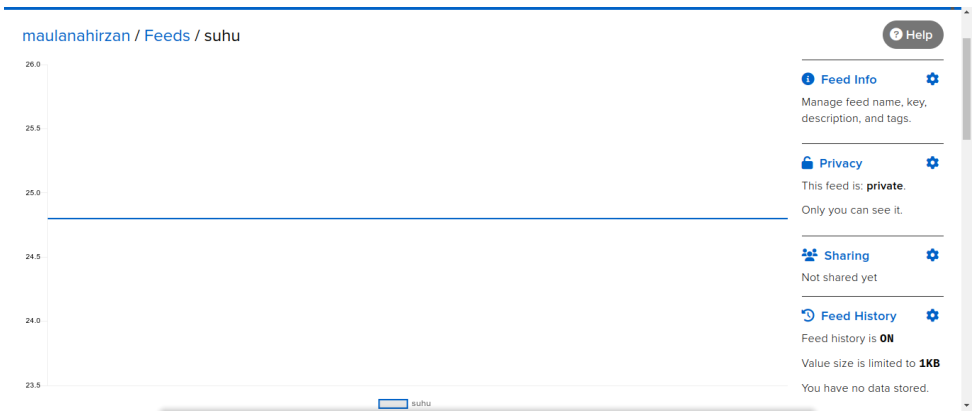
26. Alat terhubung dan berhasil mengirimkan data

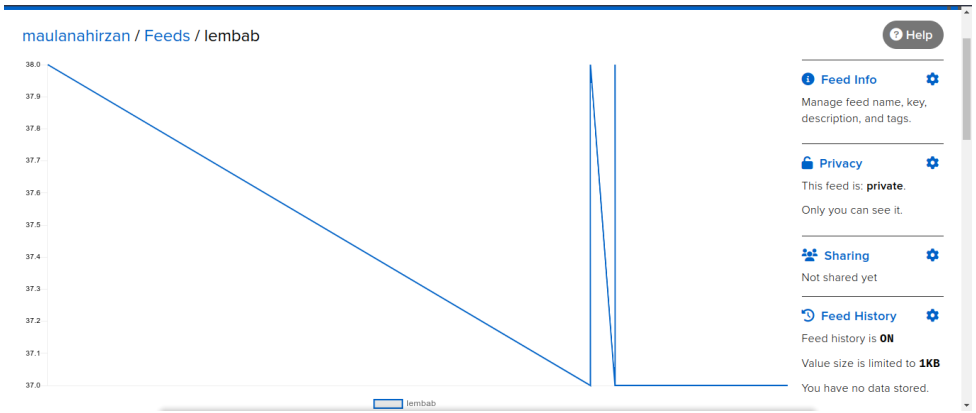


27. Hasil di website **Adafruit IO**



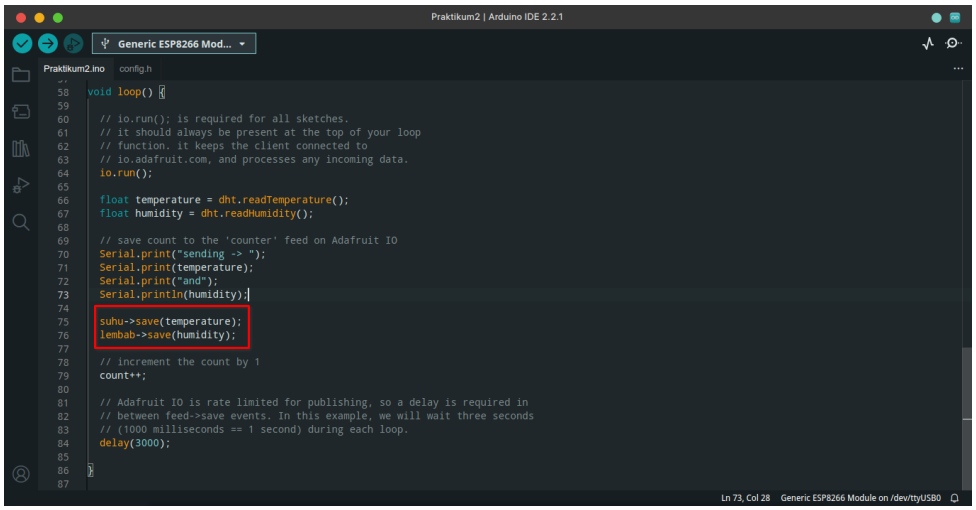
28. Klik salah satu **feed** untuk melihat data



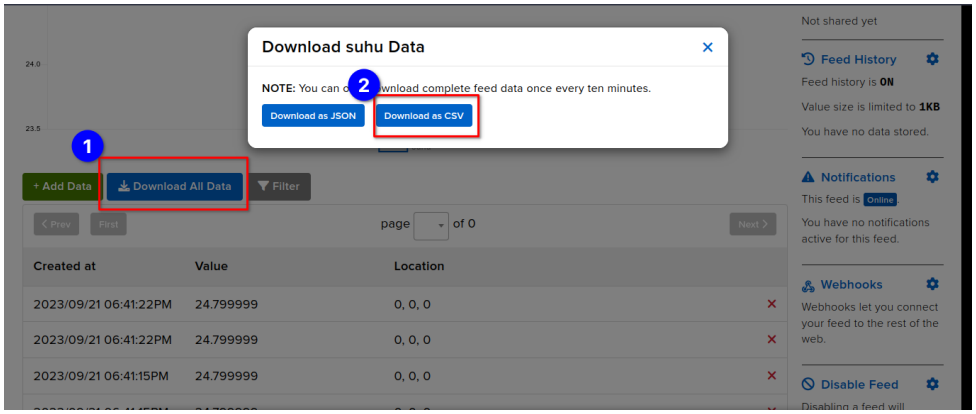


Potongan Kode

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



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



Bab 3

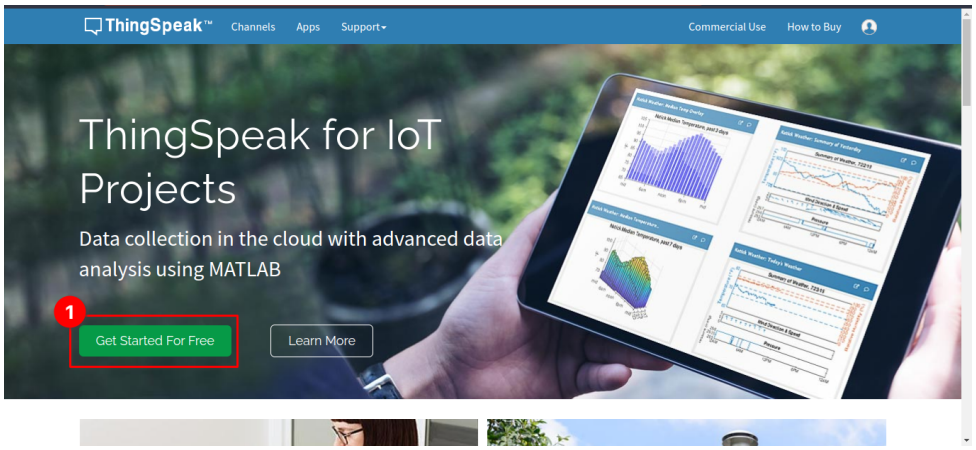
Praktikum 3

3.1 ESP8266, DHT-22, dan Thingspeak

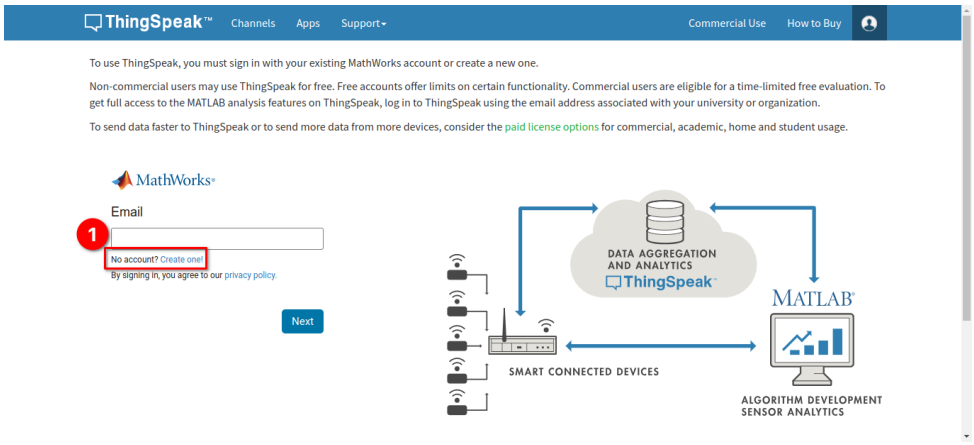
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Thingspeak. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 2** yang ada di halaman sebelumnya.

3.2 Tutorial

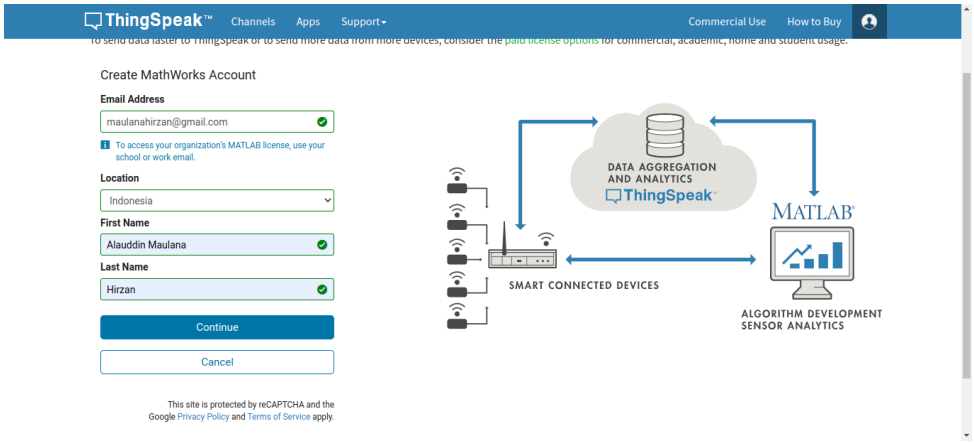
1. Untuk memulai praktikum ini, mahasiswa diwajibkan untuk membuat akun di https://thingspeak.com/ secara gratis. Klik **Get started for free**



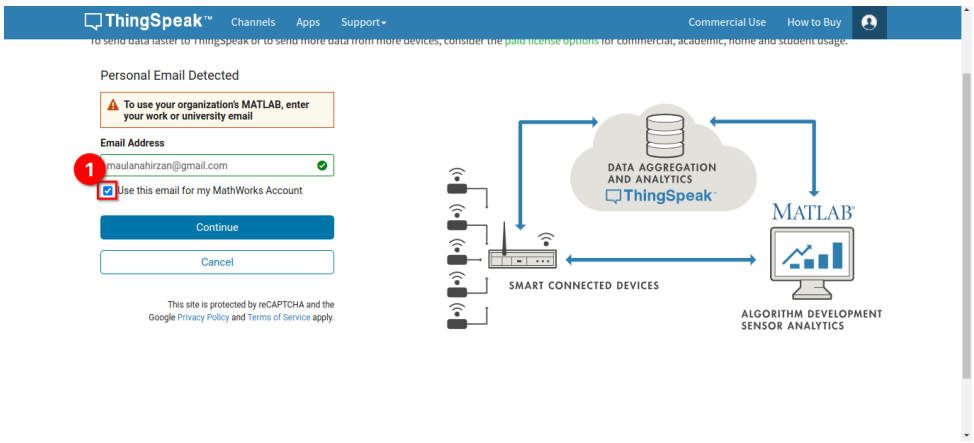
2. Klik **Create one!**



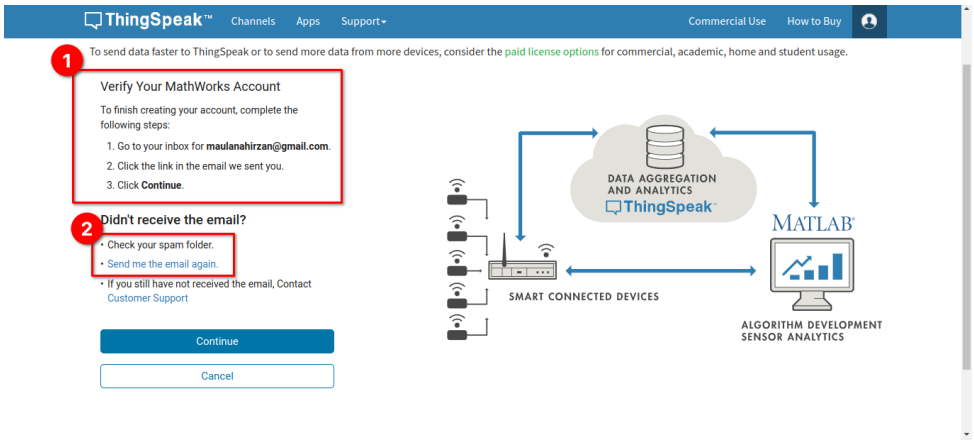
3. Isi informasi identitas

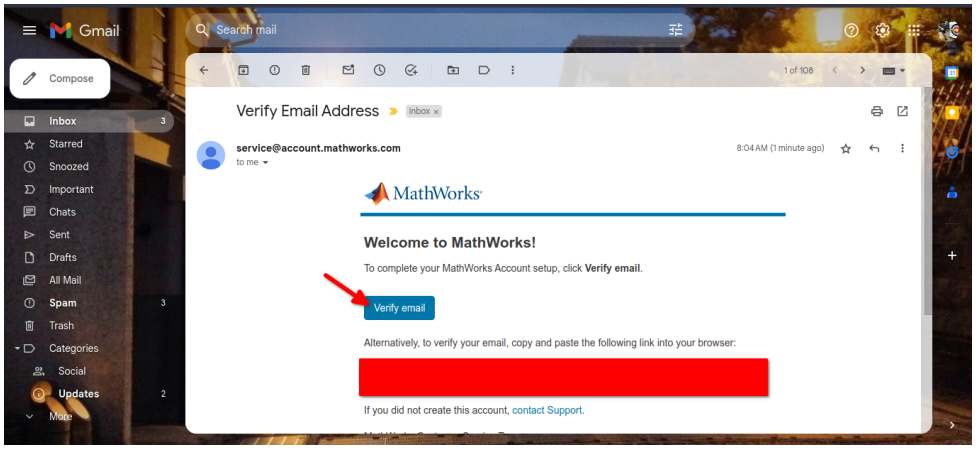


4. Centang untuk menggunakan email pribadi

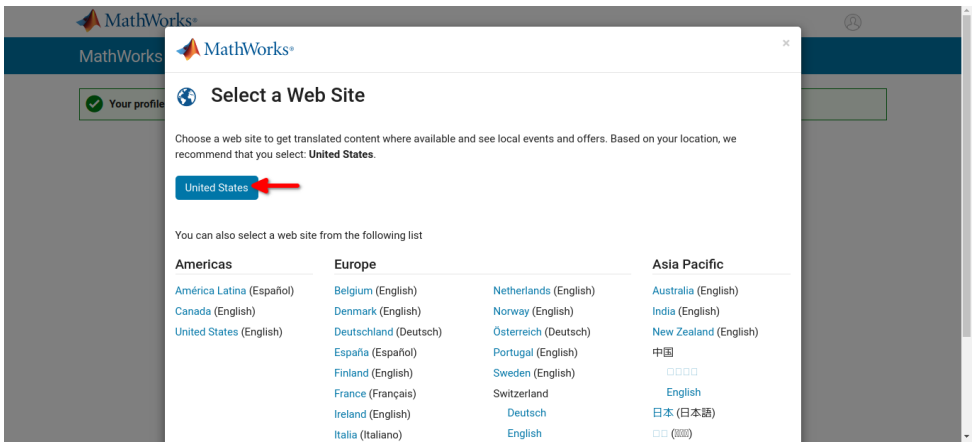


5. Cek email anda (termasuk **SPAM**) untuk verifikasi email. **JANGAN TUTUP WINDOW INI!!!**

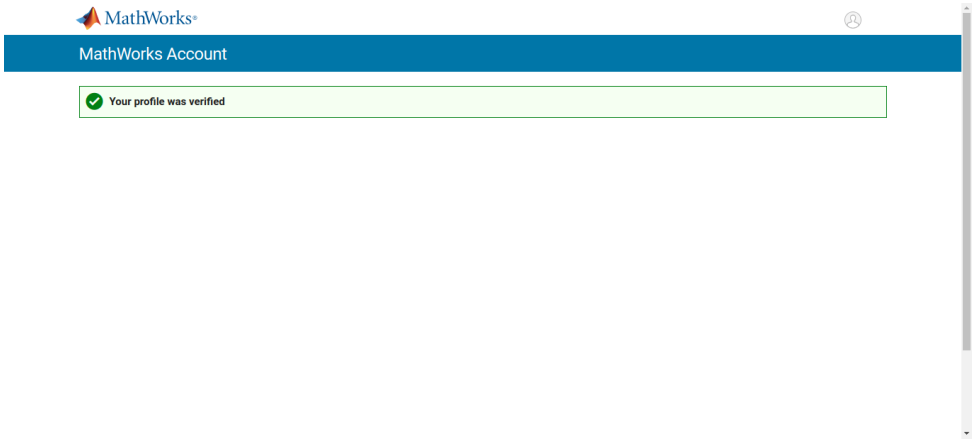




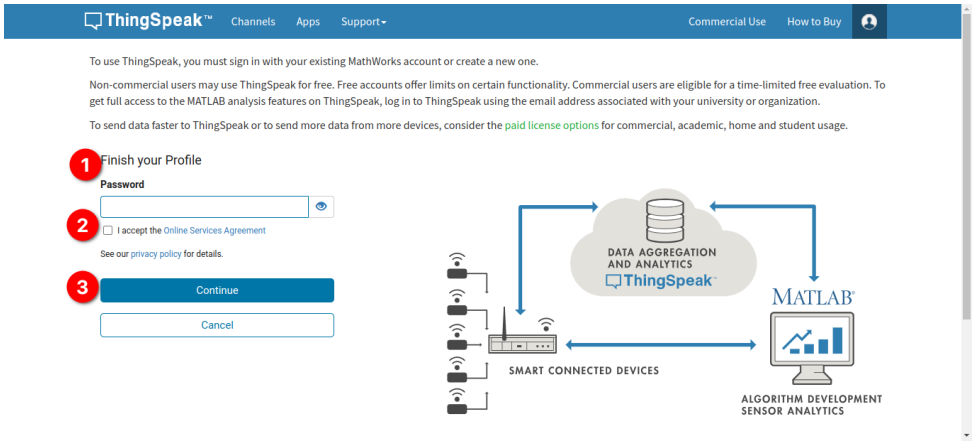
6. Pilih negara untuk website



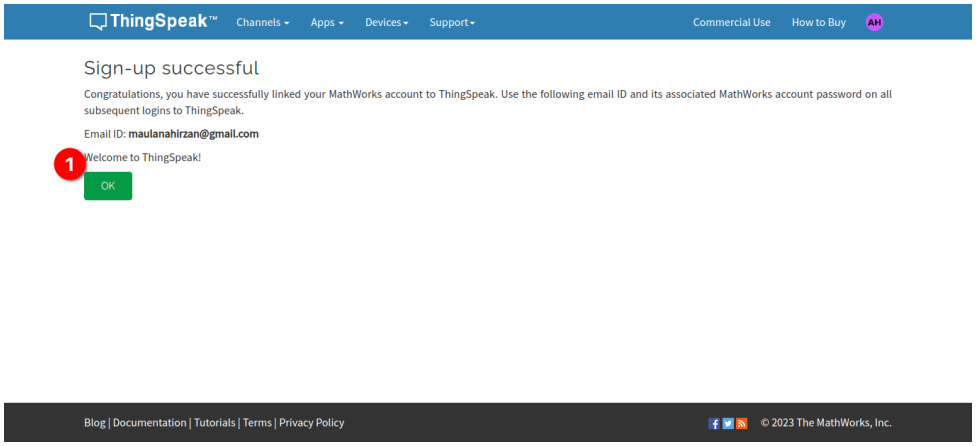
7. Akun sudah terverifikasi



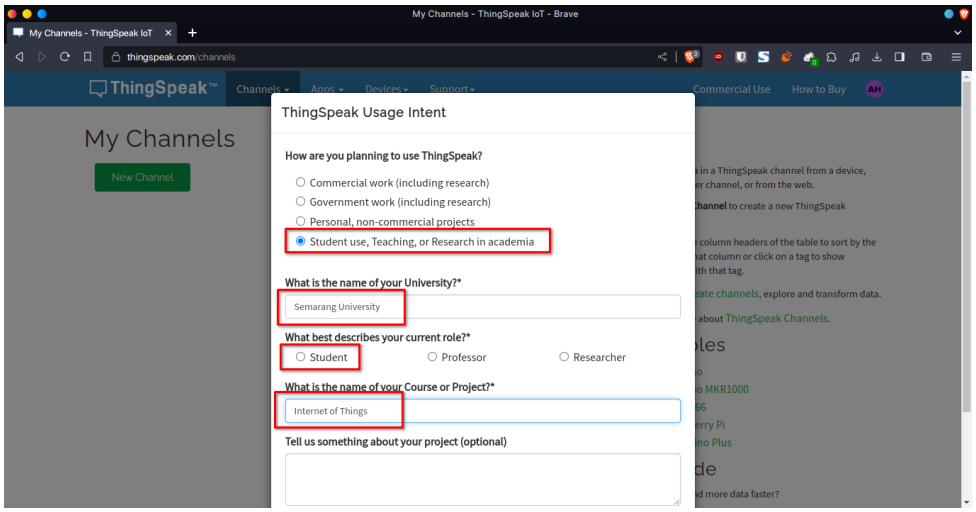
8. Ketika sudah selesai, kembali ke **WINDOW** di **Langkah 5**. dan klik **Continue**



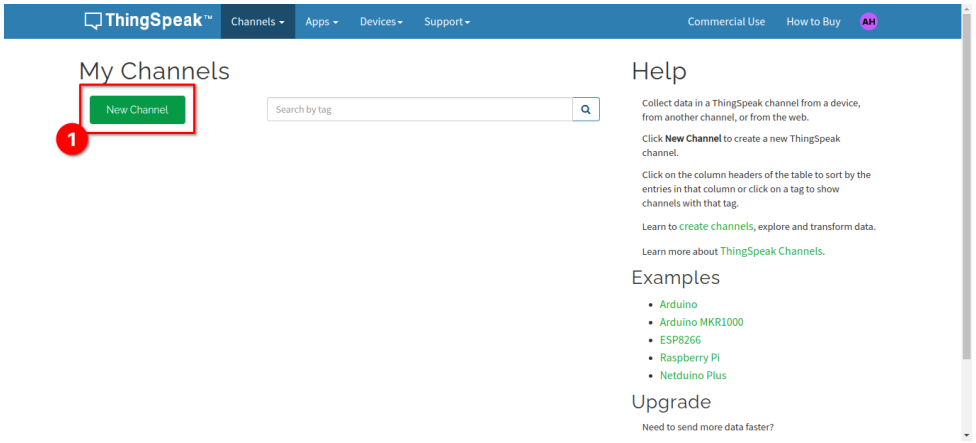
9. Klik **OK** untuk pindah ke **Dasbor**



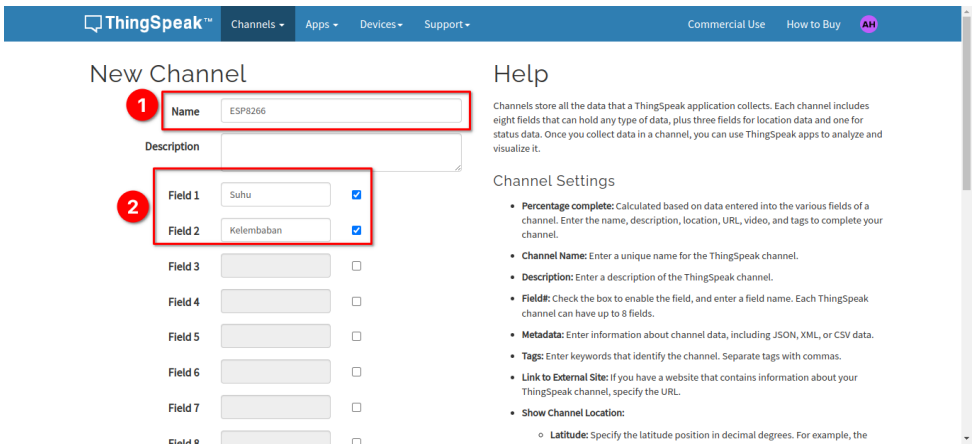
10. Di Dashboar akan ditanya penggunaan **Thingspeak**. Isi sesuai pertanyaan. Jangan lupa untuk klik **OK** atau **Continue**



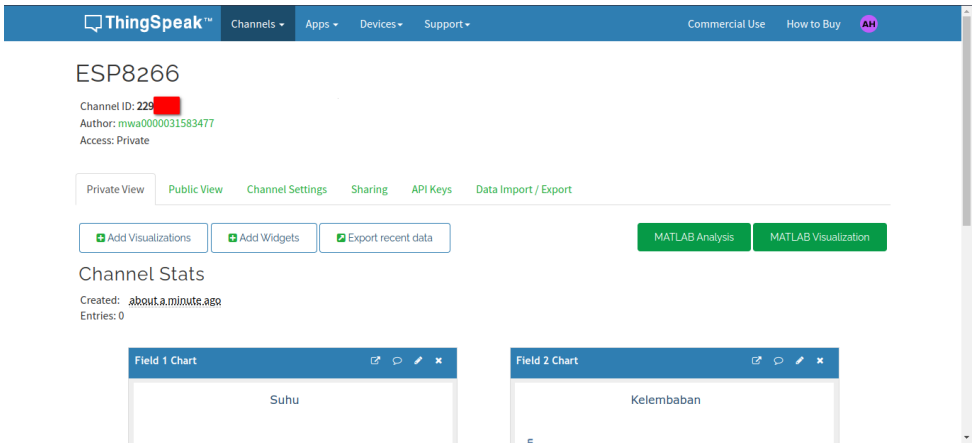
11. Jika sudah, buat **KANAL BARU** dengan klik **New Channel**



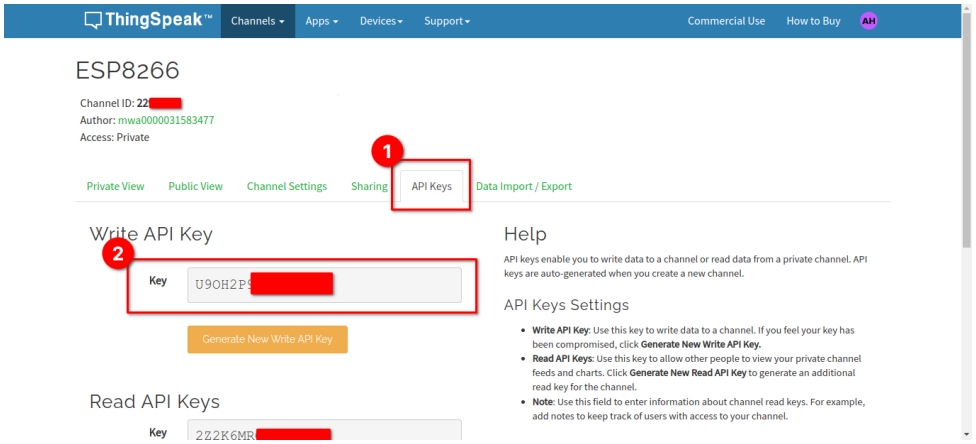
12. Beri nama **KANAL**, dan isi **2 Field** dengan nama **Suhu** dan **Kelembaban**. Klik **Save Channel** di bagian bawah



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

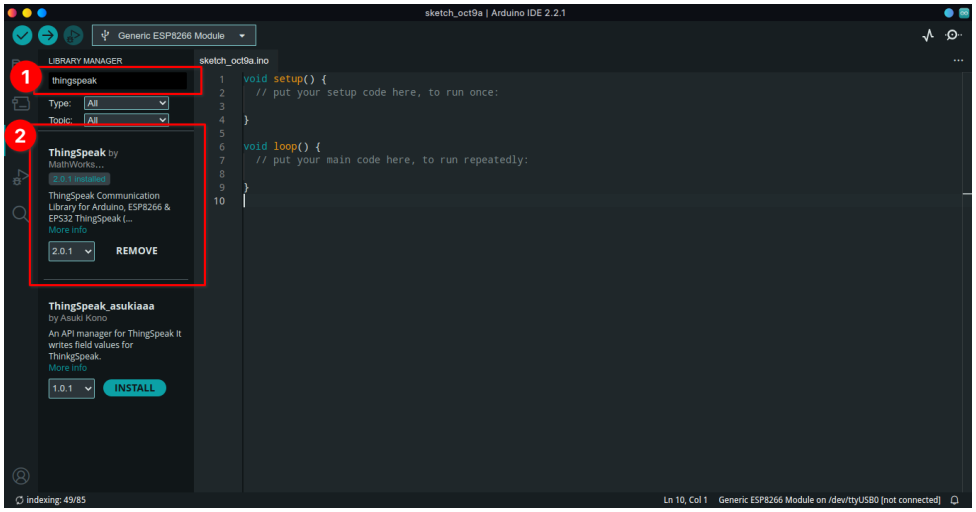


14. Pindah ke tab **API Keys**, dan kopi **Write API Key** untuk **Arduino IDE**



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

16. **Install** Library **Thingspeak**

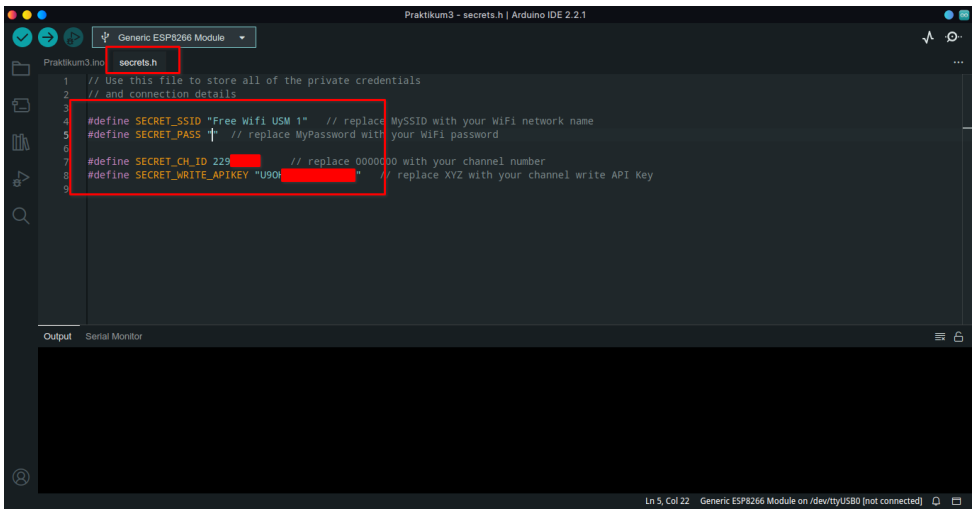


17. Untuk membuat program pengunggah data ke **Thingspeak**, gunakan **Example**

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

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

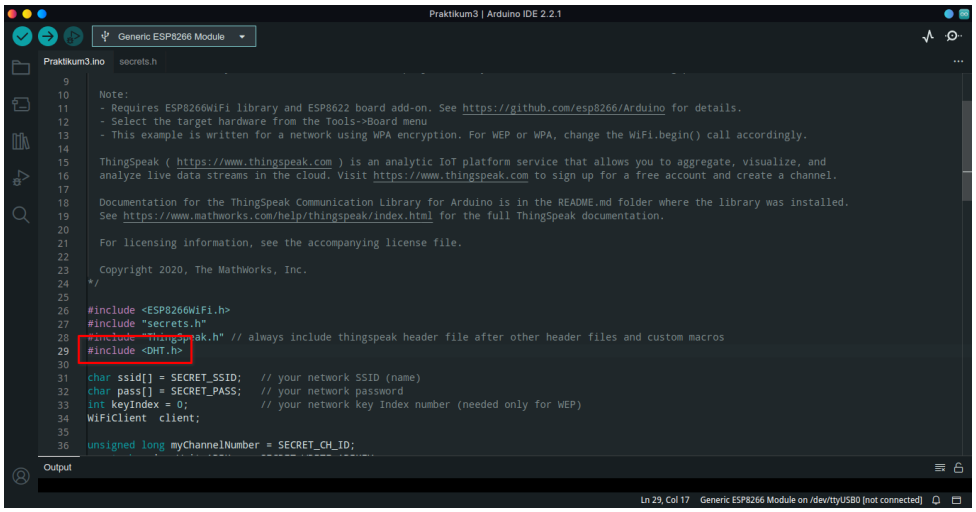
19. Ketika sudah siap, cukup edit file **secrets.h** melalui tab. Isi sesuai konfigurasi sebelumnya.



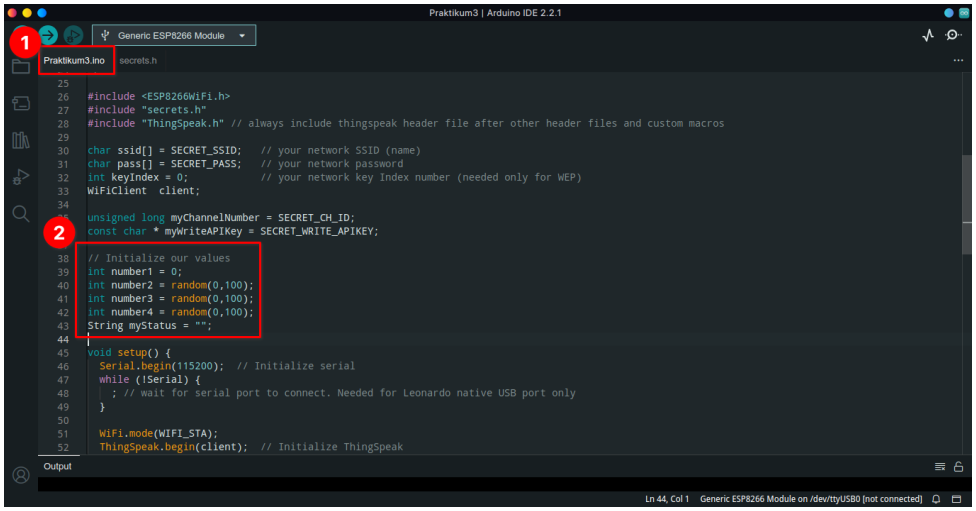
20. Kembali ke file **Praktikum3.ino**. Tambahkan **Library DHT** di bawah **ThingSpeak.h**. Lihat gambar

Potongan Kode

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



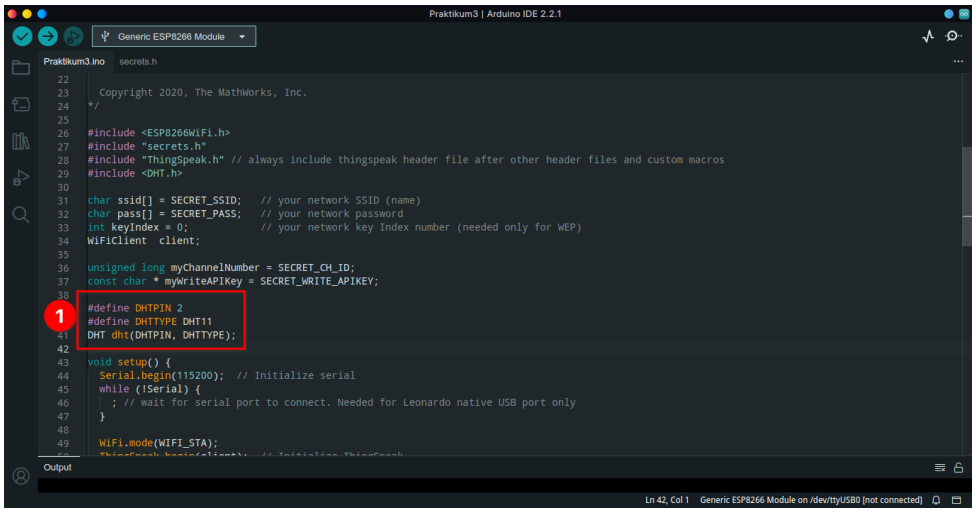
21. Hapus kode berikut



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

Potongan Kode

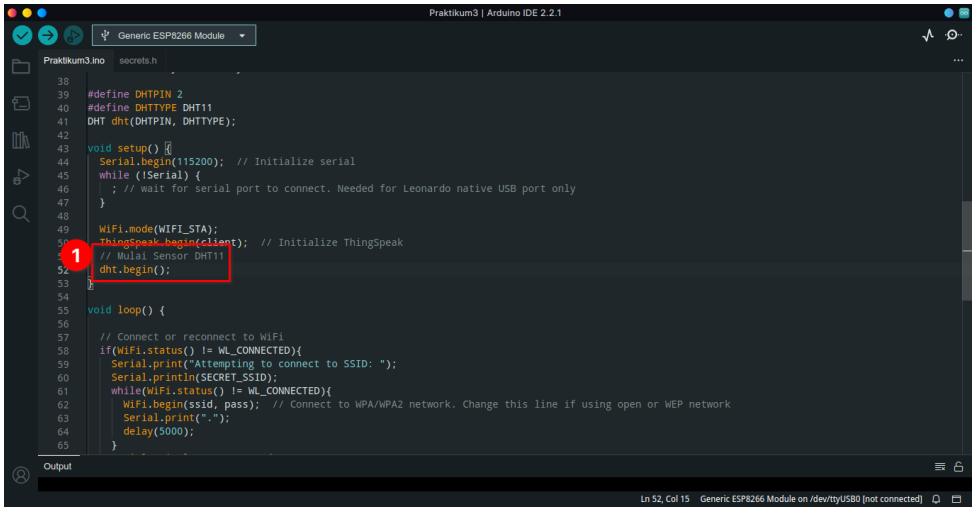
```
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
```



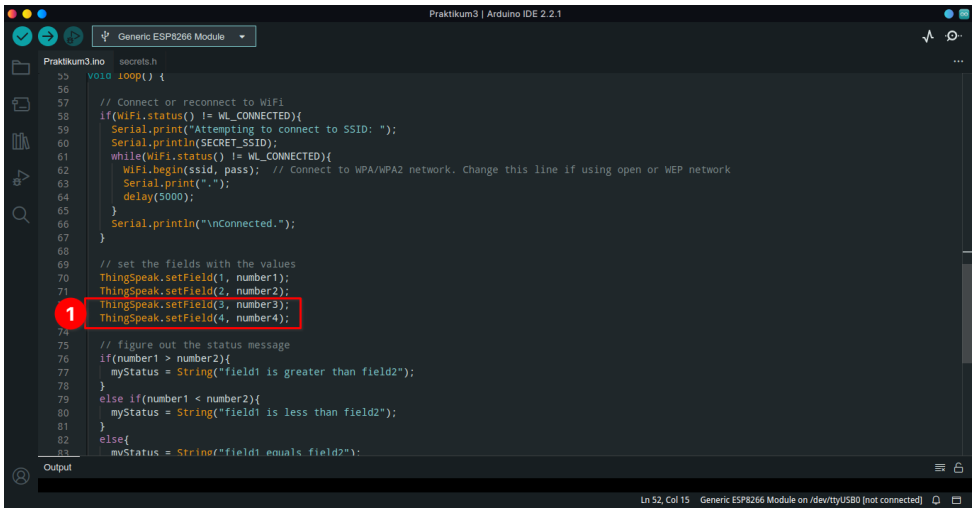
23. Lalu di dalam **FUNGSI SETUP**, tambahkan kode berikut setelah baris **ThingSpeak.begin()**:

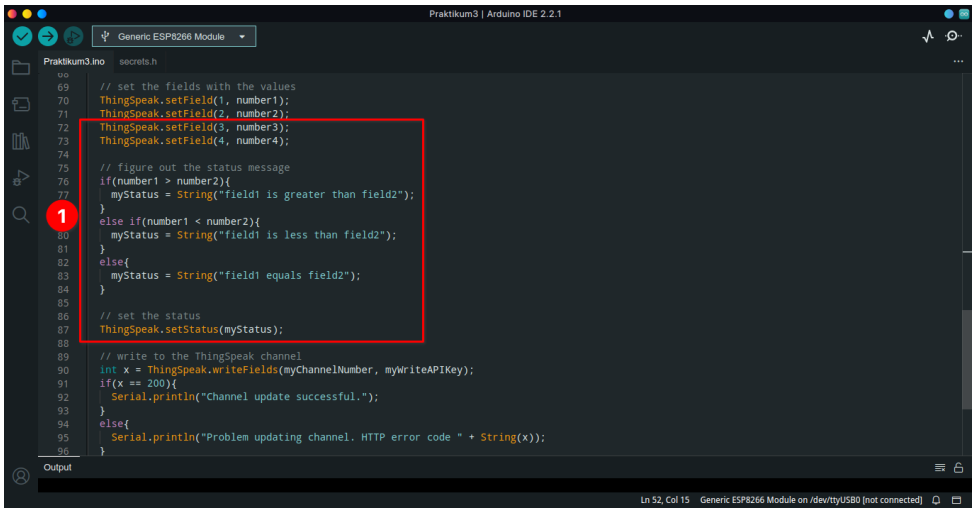
Potongan Kode

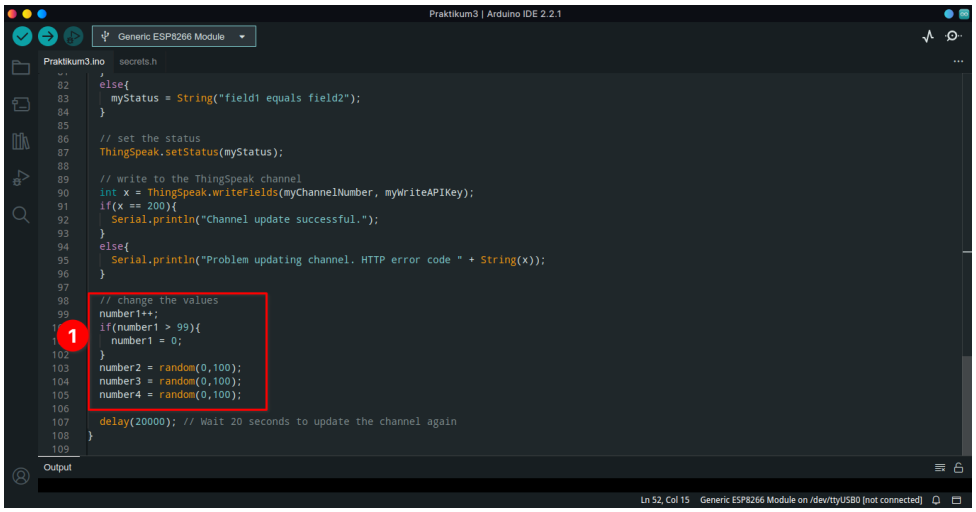
```
// Mulai Sensor DHT-22
dht.begin();
```



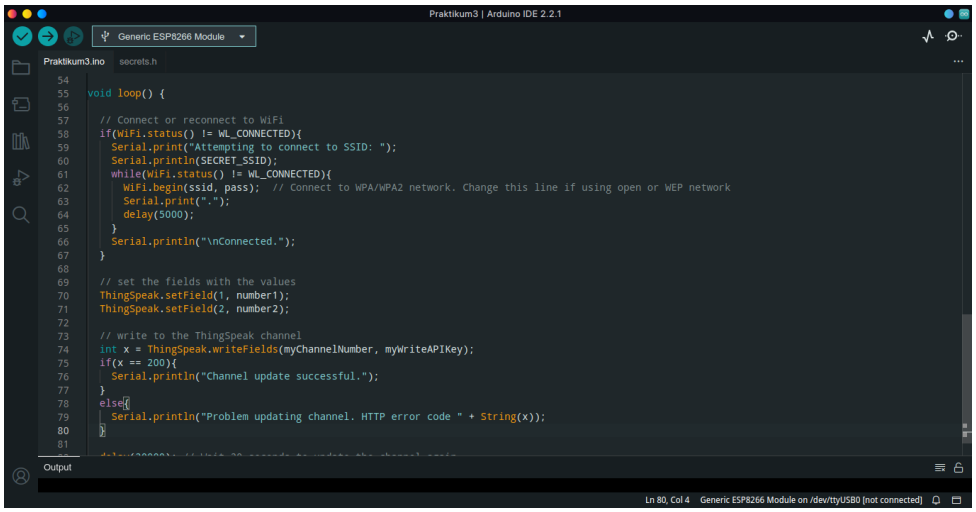
24. Di dalam **FUNGSI LOOP** Hapus kode berikut:







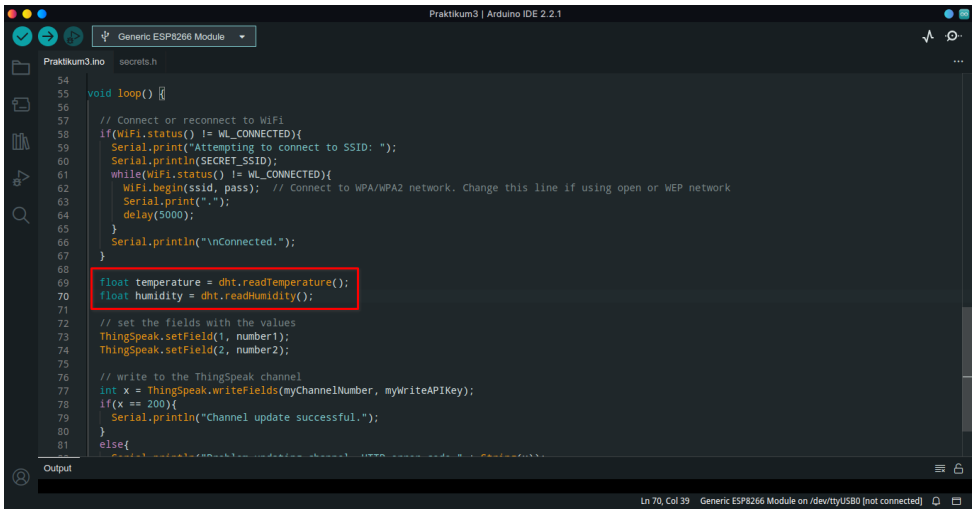
25. Hasil AKHIR SEHARUSNYA:



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

Potongan Kode

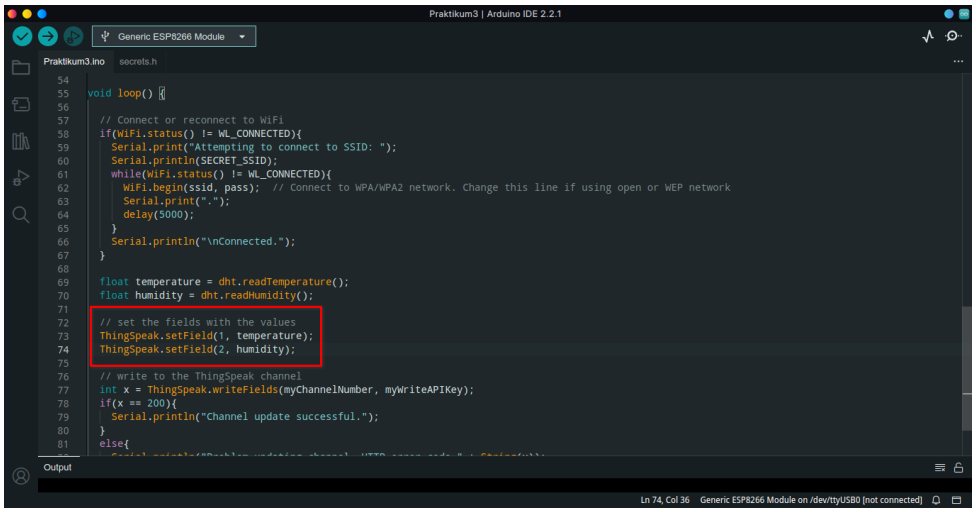
```
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
```



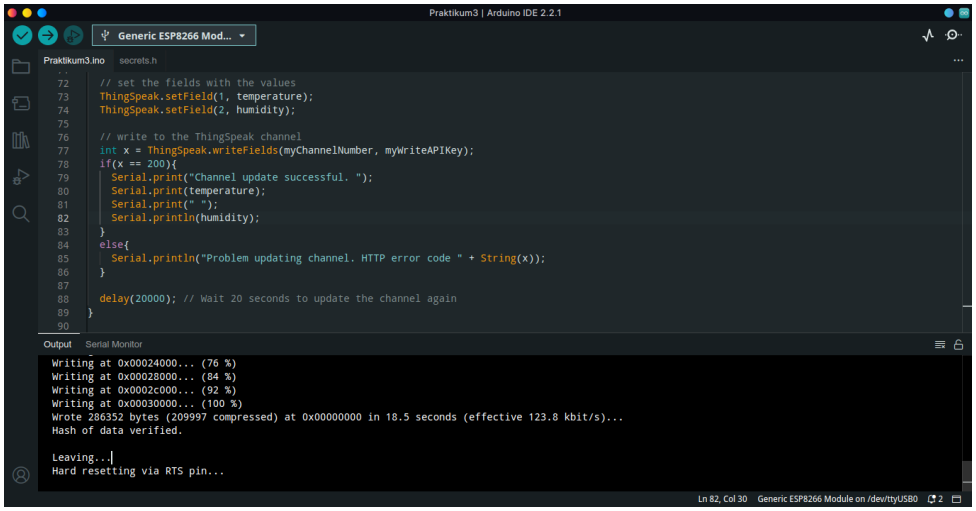
27. Lalu ubah kode di dalam **ThingSpeak.setField** sesuai kode berikut:

Potongan Kode

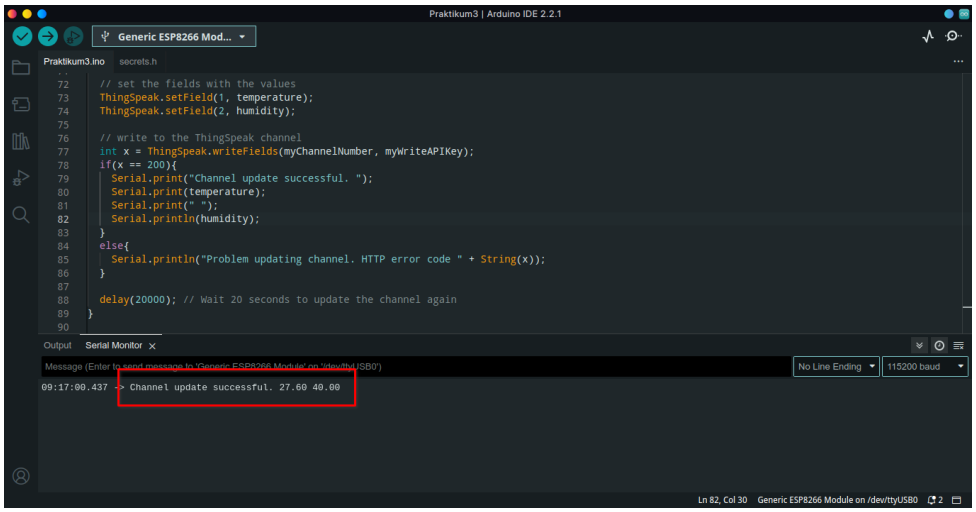
```
ThingSpeak.setField(1, temperature);
ThingSpeak.setField(2, humidity);
```

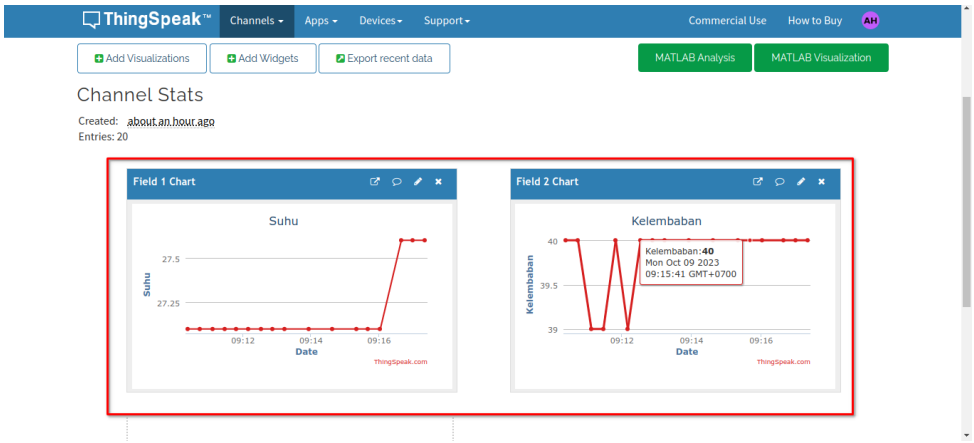


28. **Verifikasi** untuk memastikan kode sudah benar. Lalu klik **Upload**

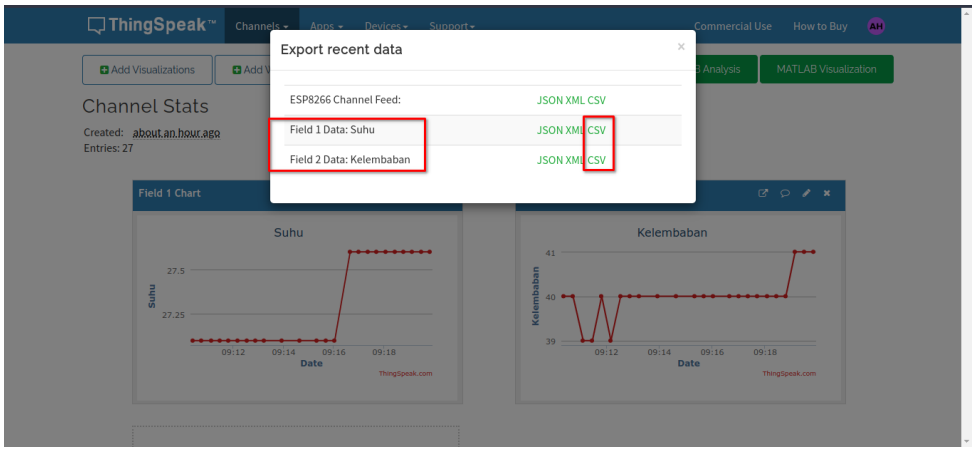


29. Data terkirim dan terunggah





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



Bab 4

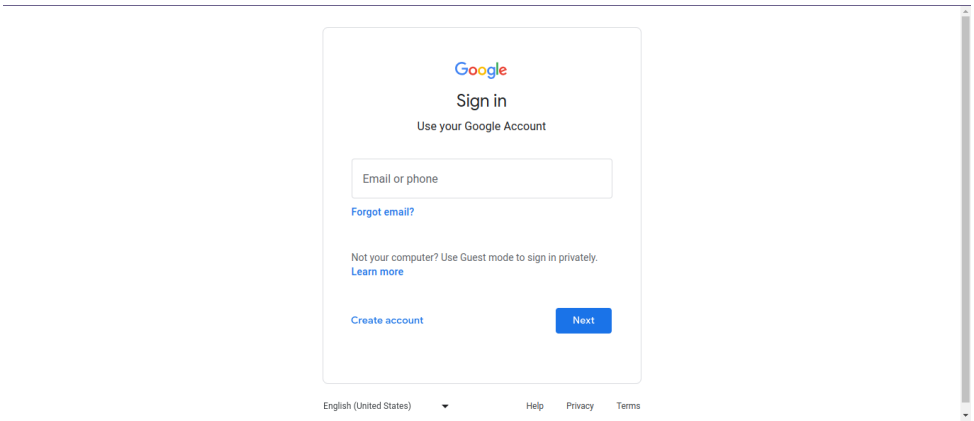
Praktikum 4

4.1 ESP8266, DHT-22, dan Firebase Realtime

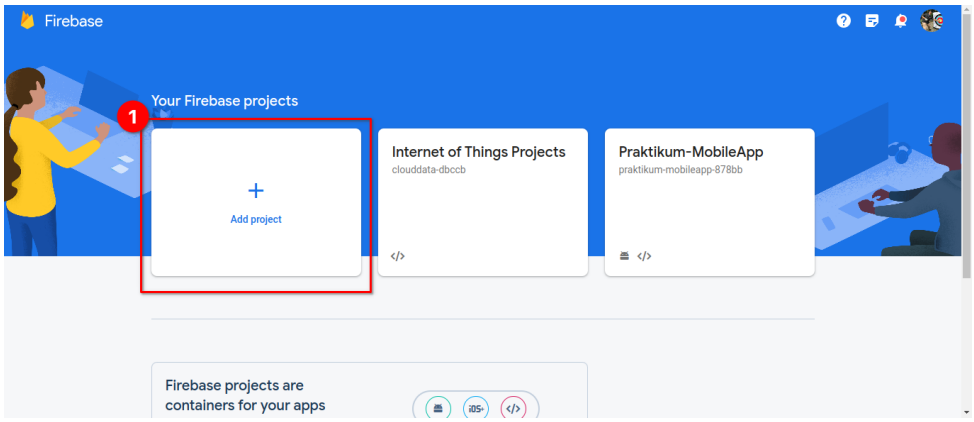
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Firebase Realtime. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 3** yang ada di halaman sebelumnya.

4.2 Tutorial

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

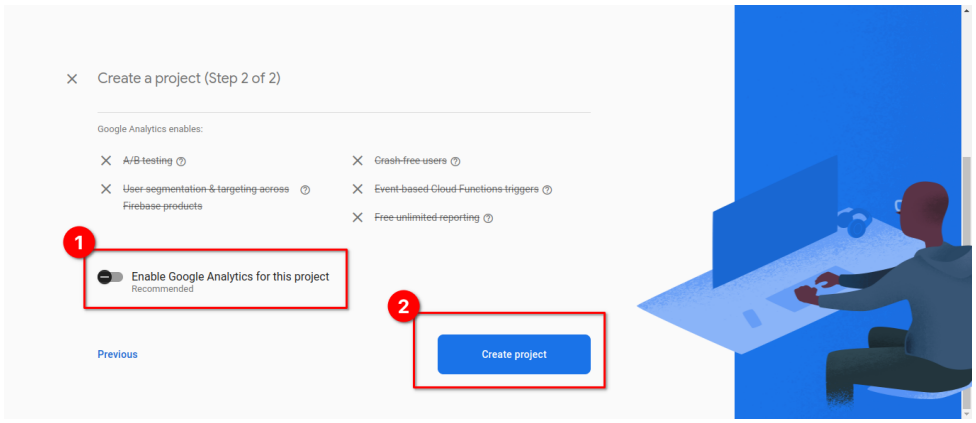


2. Buat projek baru dengan melakukan klik tanda +

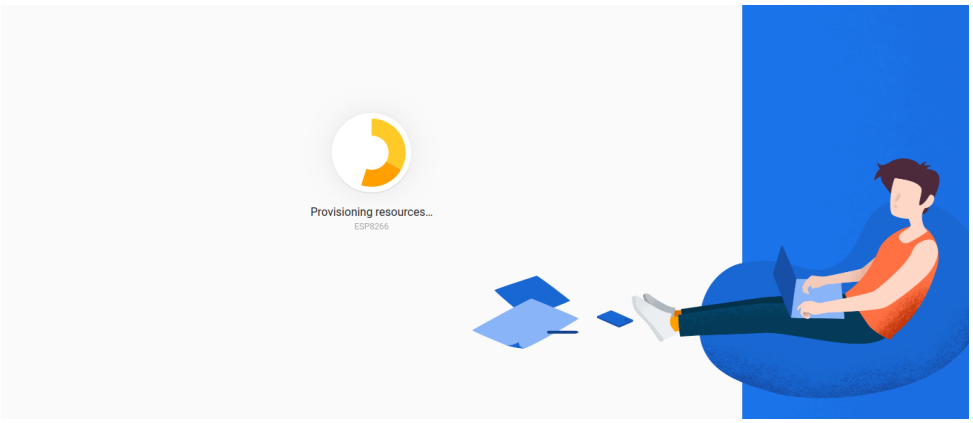


3. Isi nama projek

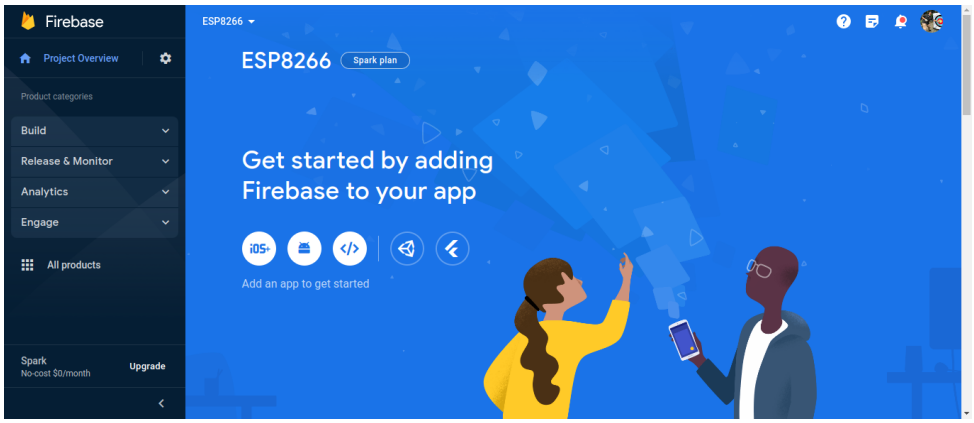
4. Matikan **Google Analytic** dan klik **Create Project**



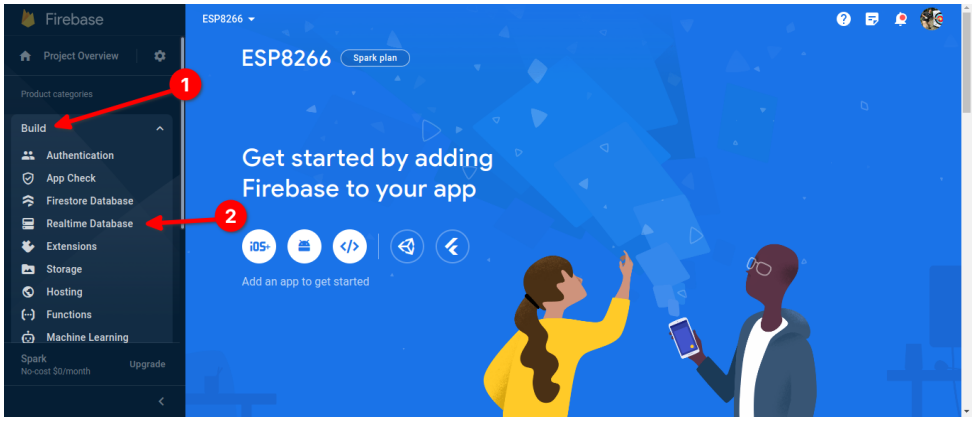
5. Tunggu proses berlangsung dan klik tombol apabila sudah muncul



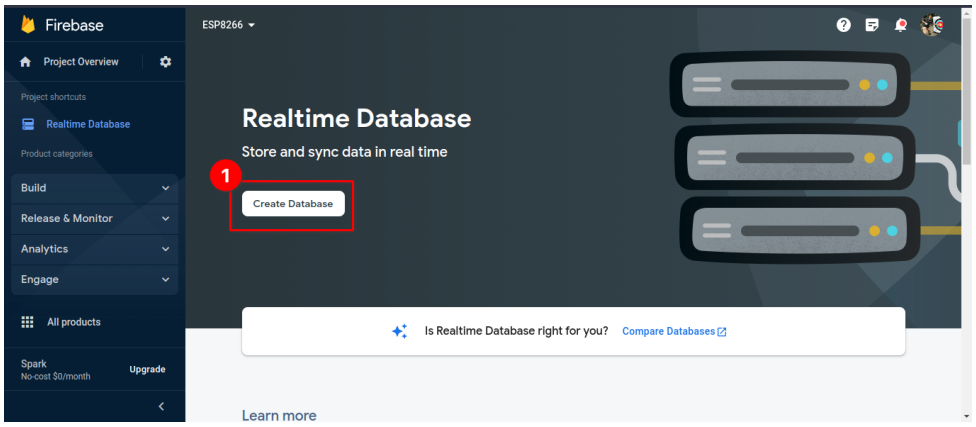
6. Firebase akan menampilkan dasbor sistem



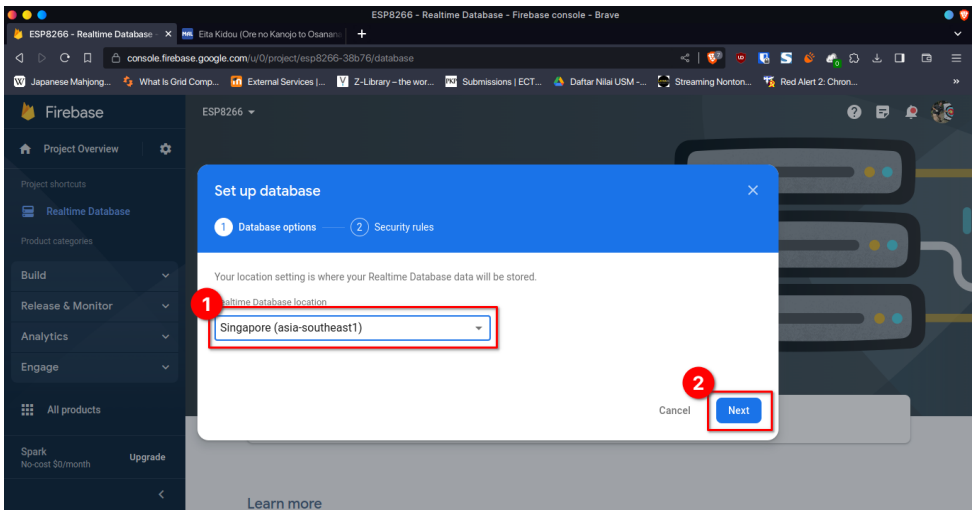
7. Klik **Build** dan pilih **Realtime Database**



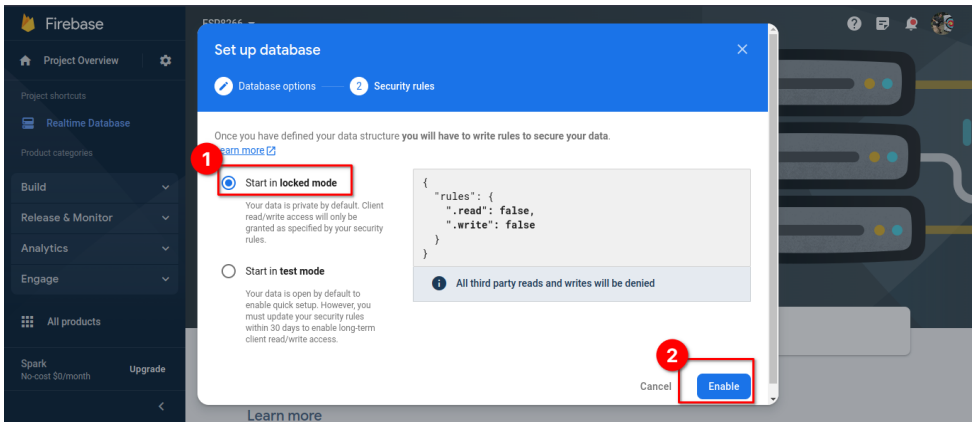
8. Klik **Create Database**



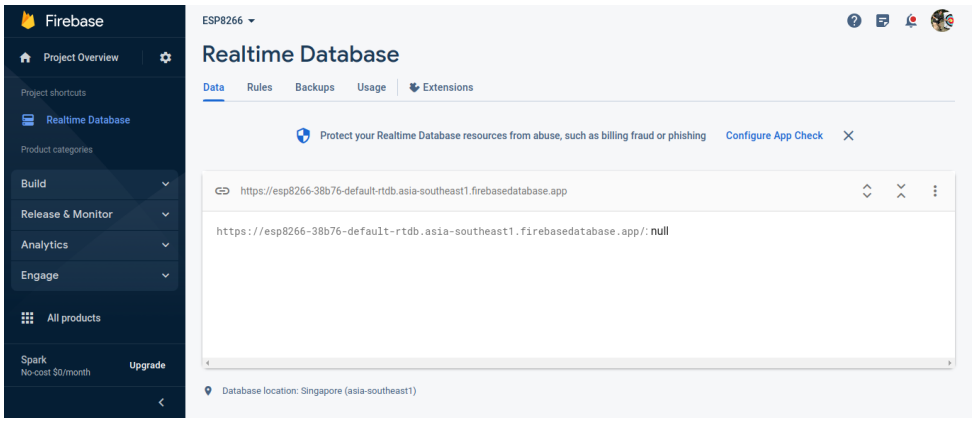
9. Pilih Lokasi dan Klik **Next**



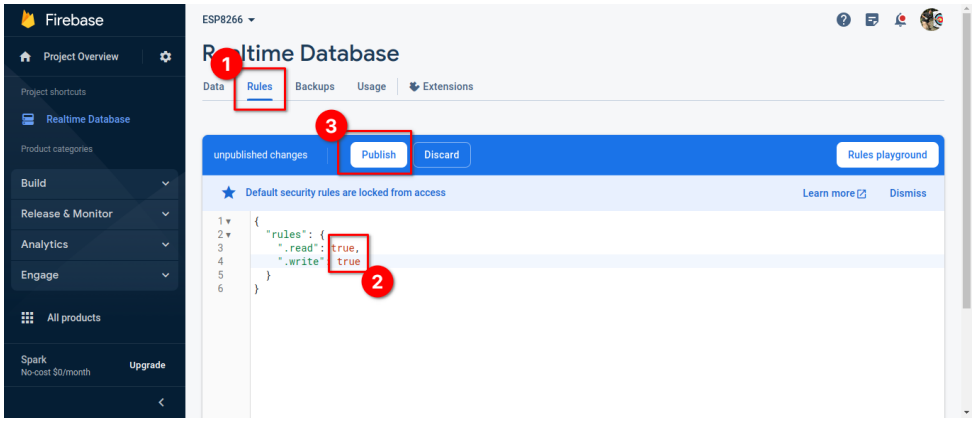
10. Pilih **Locked Mode** dan klik **Enable**



11. Database sudah dibuat

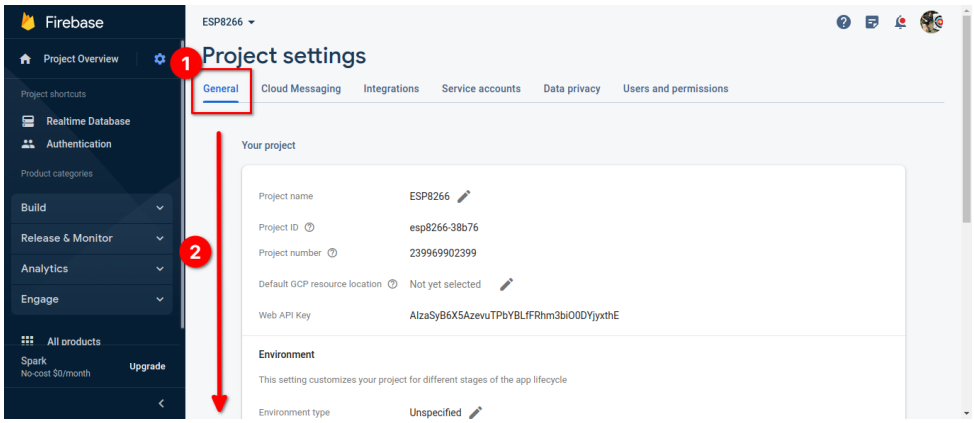


12. Sebelumnya ubah aturan database dengan klik **Rules**, dan ubah kata **false** menjadi **true**. dan klik **Publish**

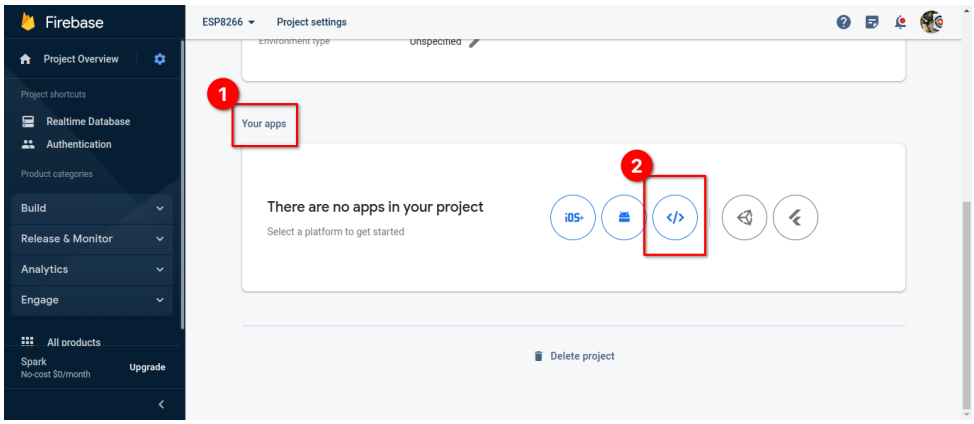


13. Untuk membuat kunci, klik **Roda Gigi Project Overview**, pilih **Project Settings**

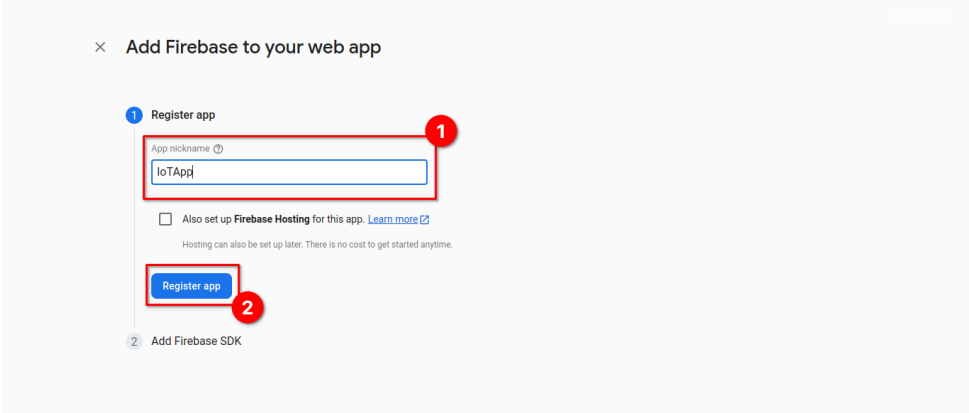
14. Di bagian **General**, scroll turun hingga menemukan **Apps**



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

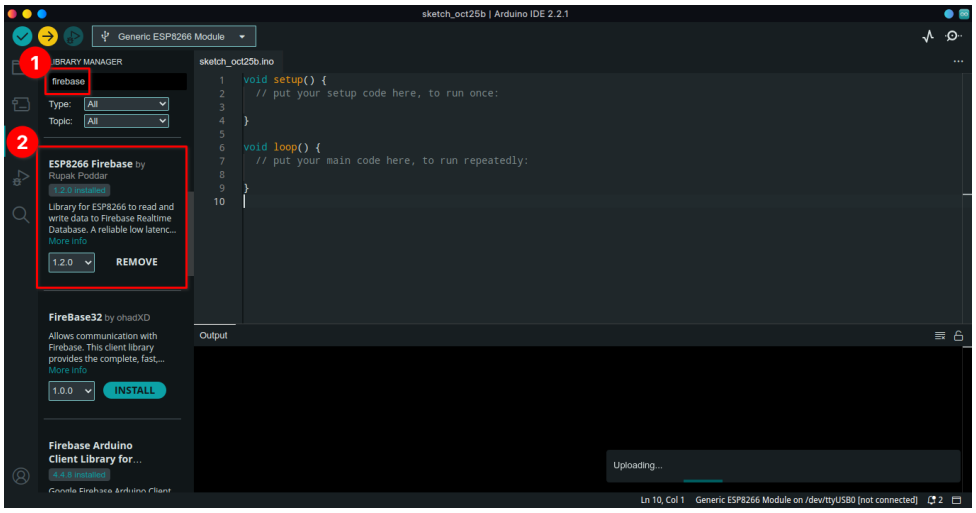


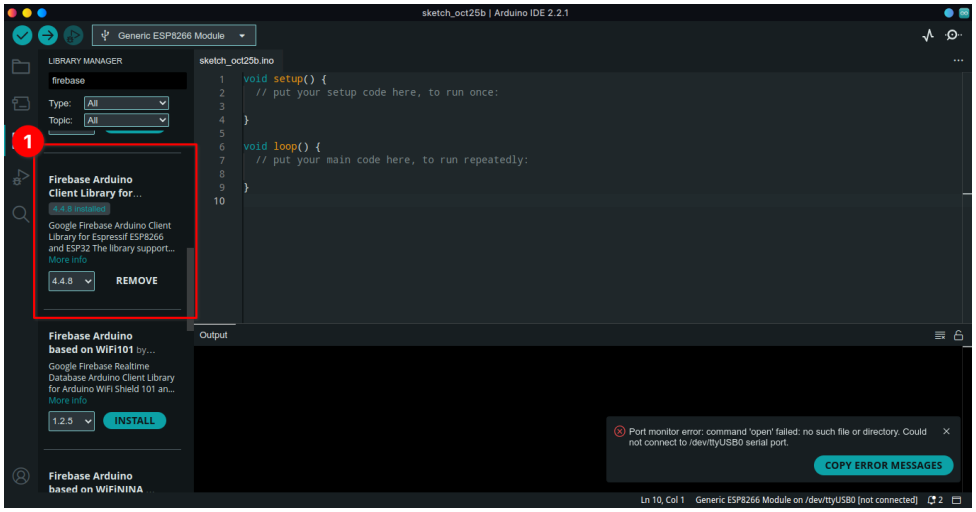
16. Isikan nama app, dan pilih **Register app**



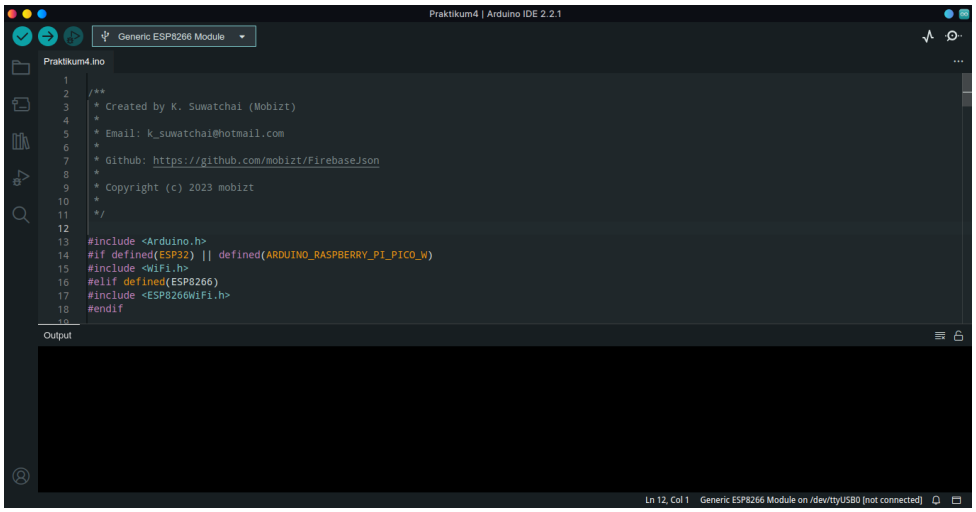
17. Di tahap selanjutnya, sistem akan membuat **API Key** dan **Database URL**. Kopi data ini ke Notepad

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



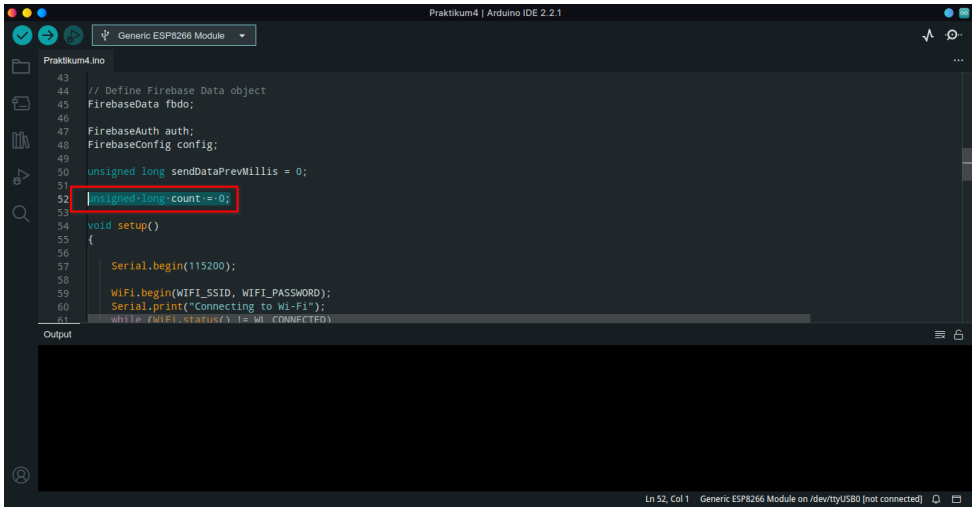


19. Buat projek baru dengan template yang sudah ada. Klik **File** → **Examples** → **Firebase Arduino Client Library for ESP8266 and ESP32** → **FirebaseJson** → **Client** → **Firebase**

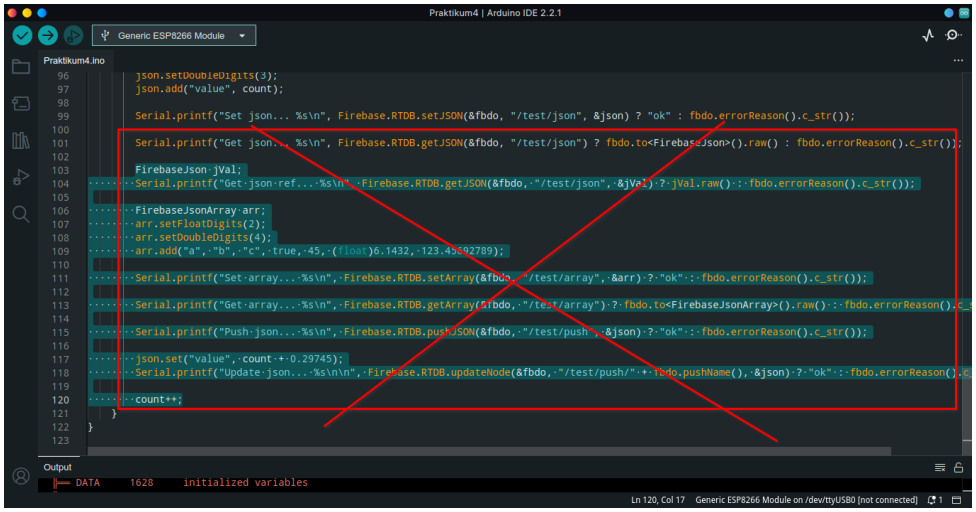


20. Hapus beberapa bagian kode berikut:

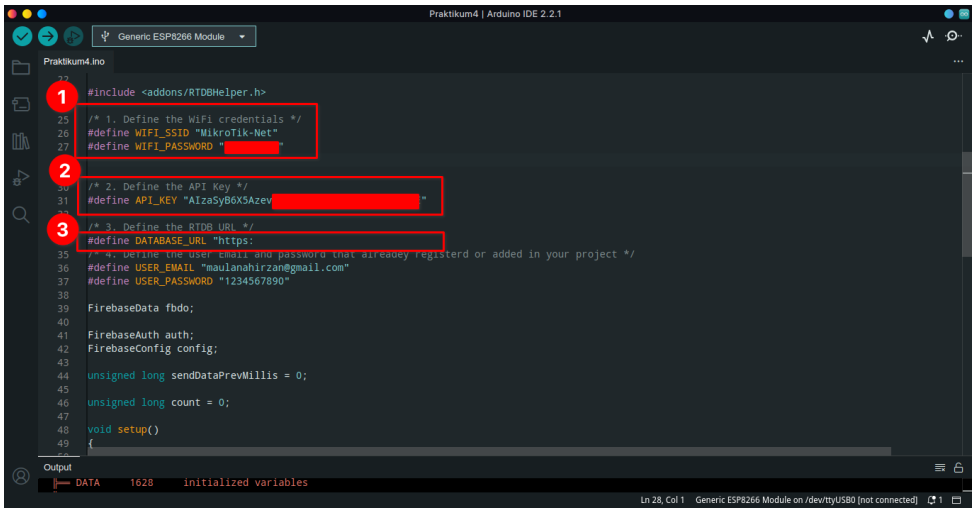
- Bagian 1



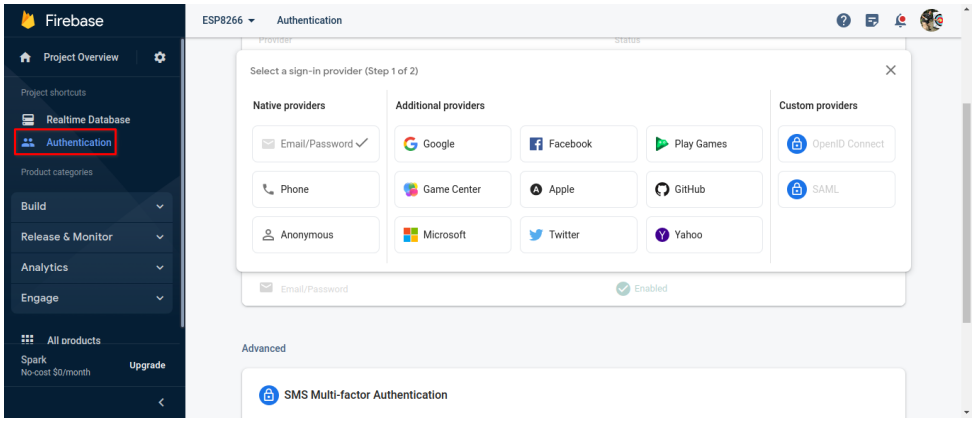
- Bagian 2



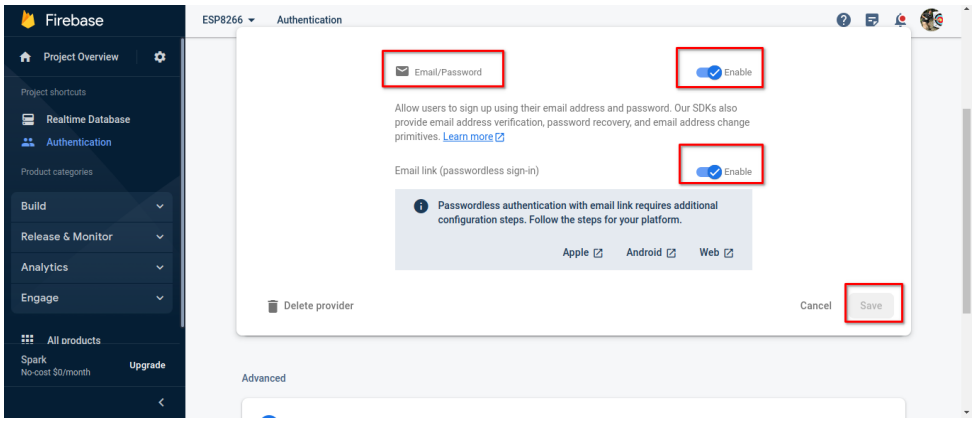
21. Lalu kembali ke bagian atas, dan ubah kode berikut:



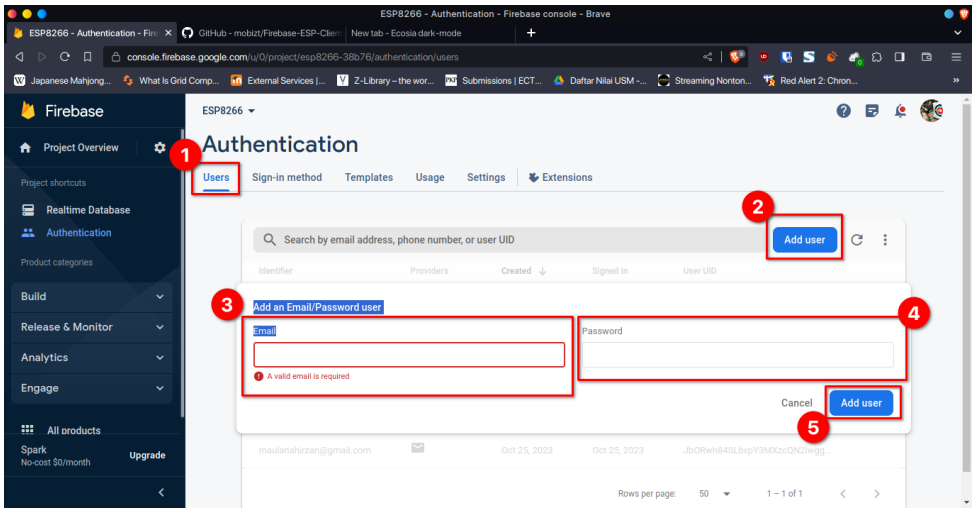
22. Bagian berikutnya adalah akun. Buka kembali **Firebase**, buka menu **Build** lalu **Firebase Authentication**



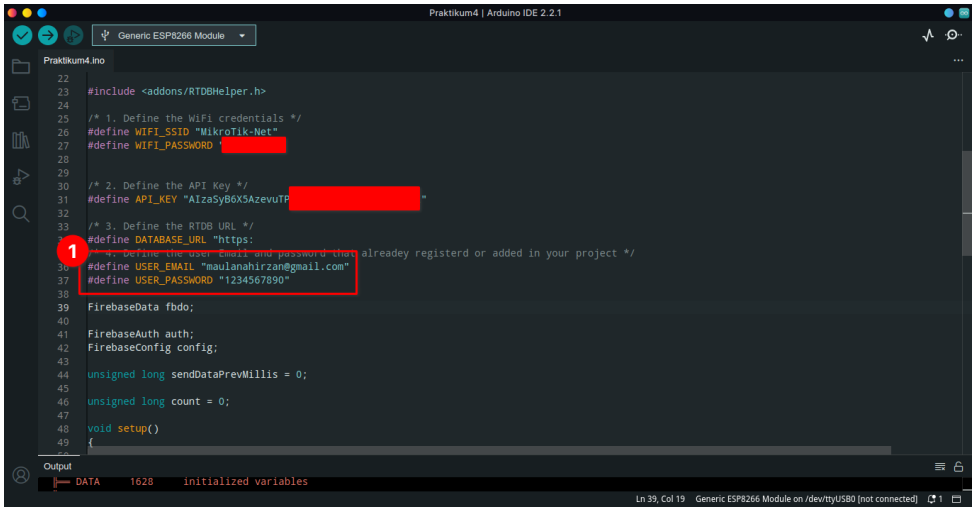
23. Pilih **Email/Password**, klik semua menjadi **Enable**, dan **Save**



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



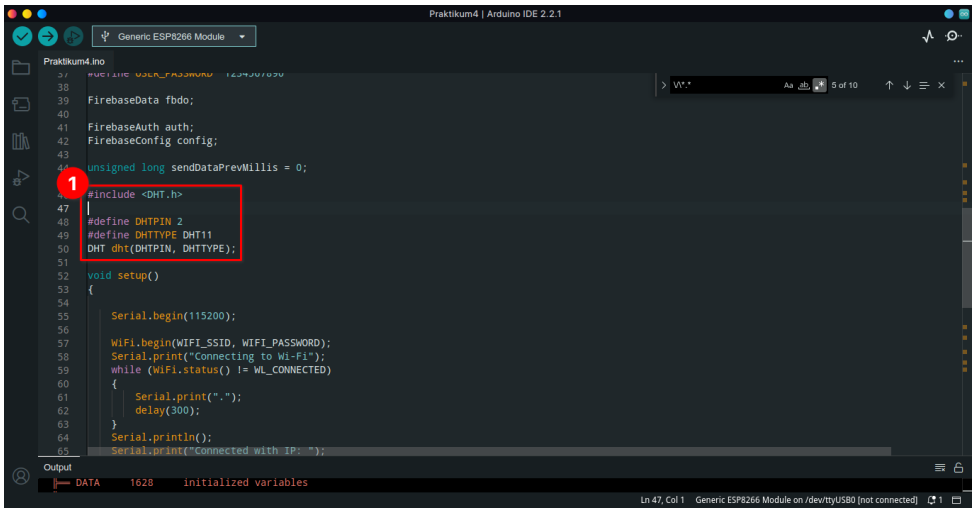
25. Kembali lagi ke **Arduino IDE** dan ubah bagian **Email** dan **Password**



26. Berikutnya adalah menambahkan kode untuk sensor DHT

Potongan Kode

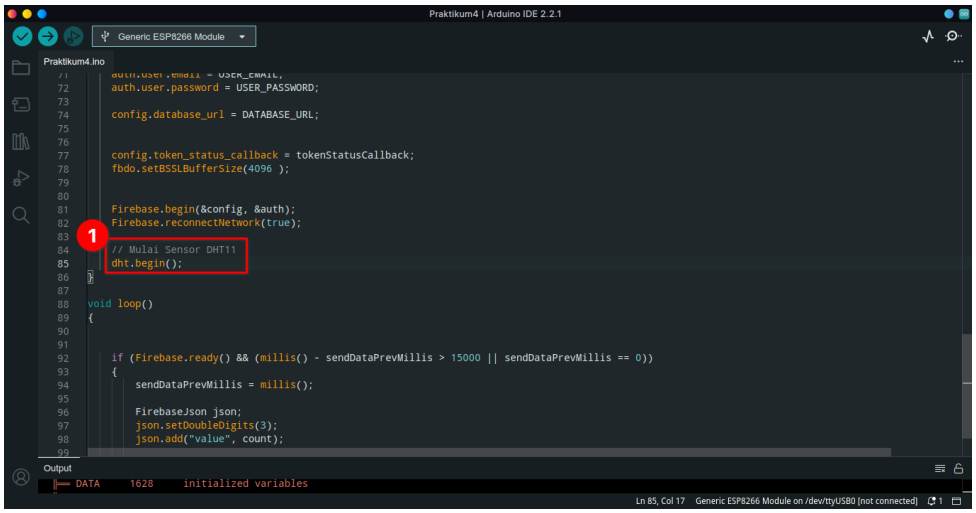
```
#include <DHT.h>

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



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

Potongan Kode

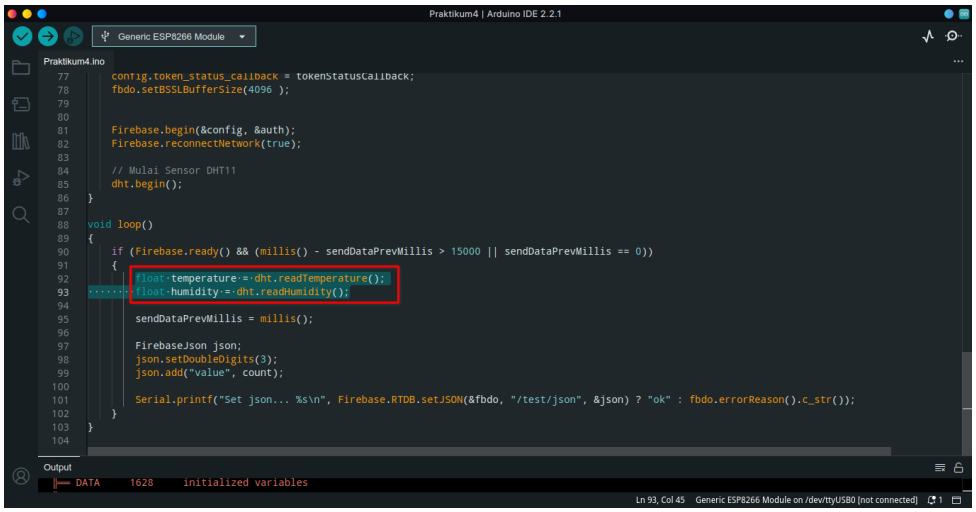
```
// Mulai Sensor DHT-22
dht.begin();
```



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

Potongan Kode

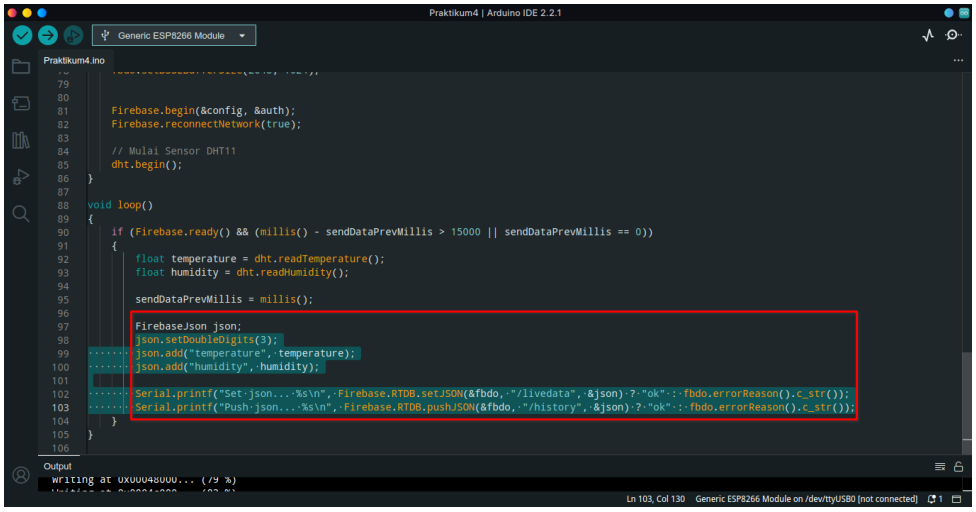
```
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
```



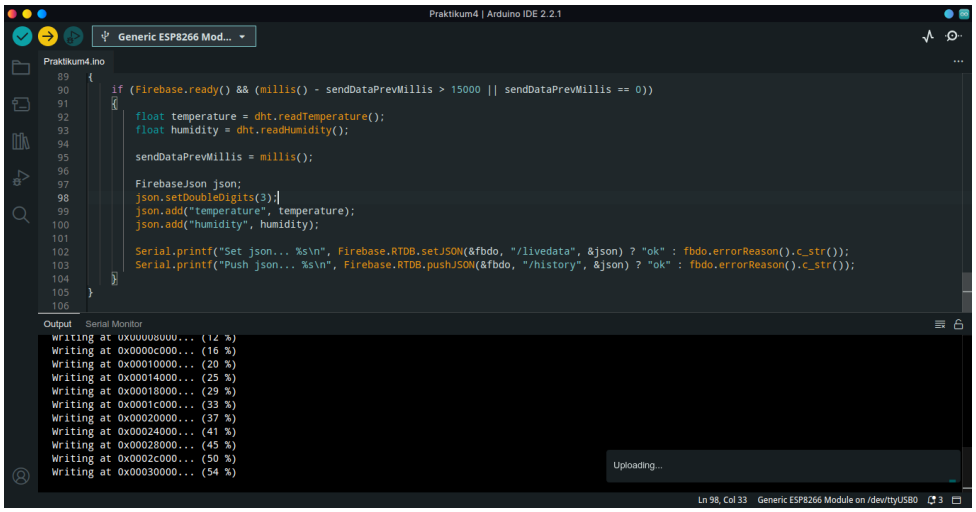
29. Setelah itu untuk menyusun query nya, masukkan kode berikut. GANTI baris yang ditandai sesuai dengan kode berikut

Potongan Kode

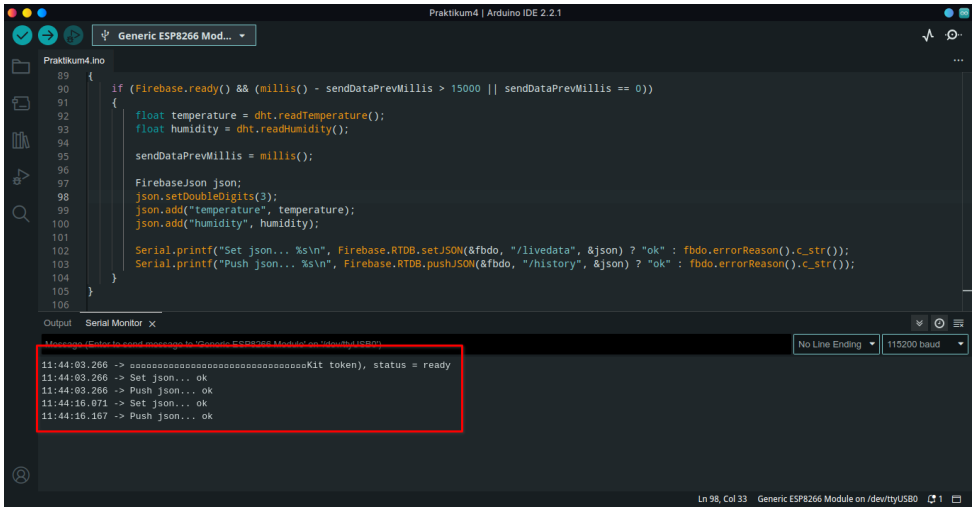
```
FirebaseJson json;
json.setDoubleDigits(3);
json.add("temperature", temperature);
json.add("humidity", humidity);

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());
```



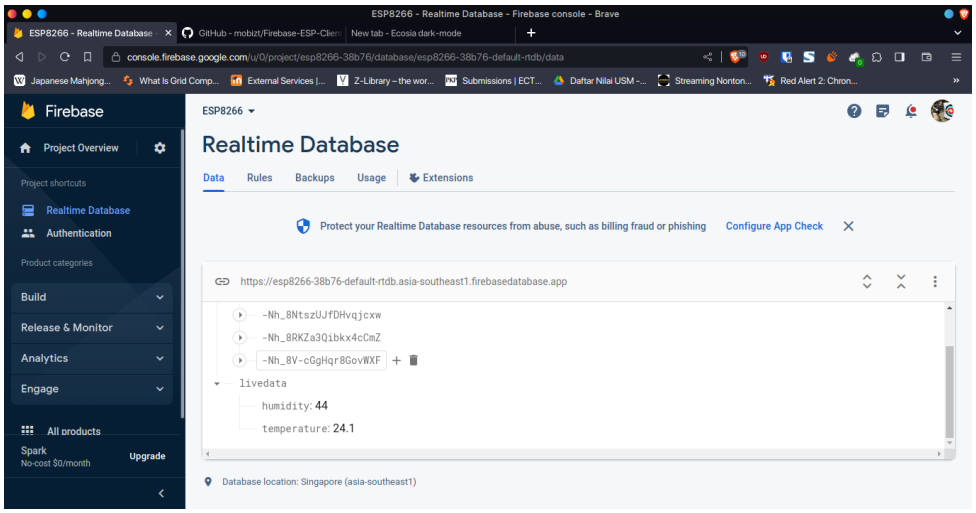
30. Verifikasi dan Upload aplikasi



31. Data sukses diunggah



32. Hasil



Bab 5

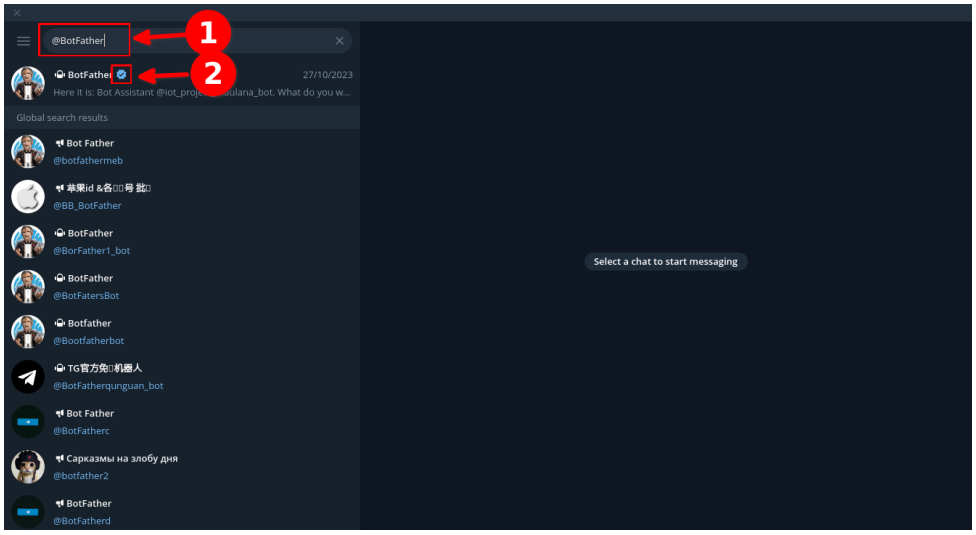
Praktikum 5

5.1 ESP8266, DHT-22, dan Telegram Bot

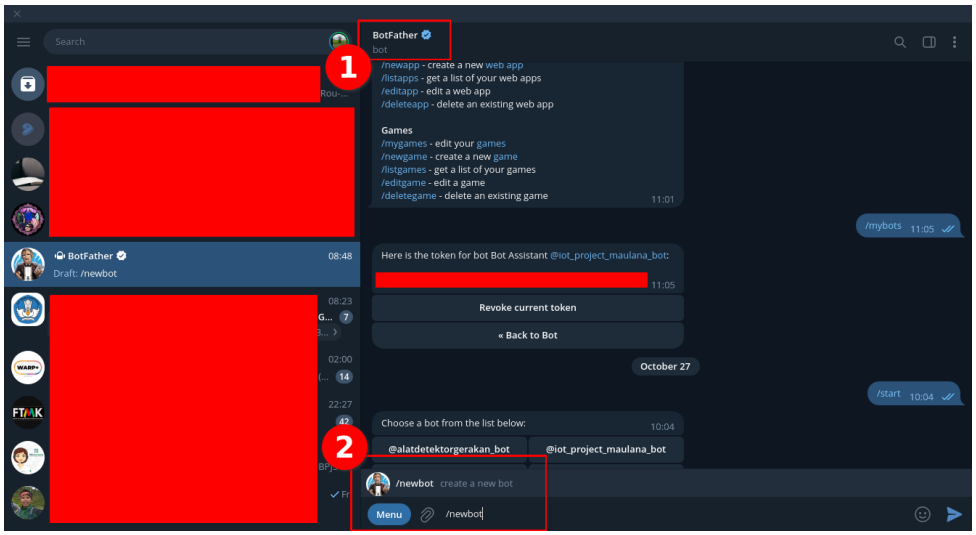
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Telegram Bot. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 4** yang ada di halaman sebelumnya.

5.2 Tutorial

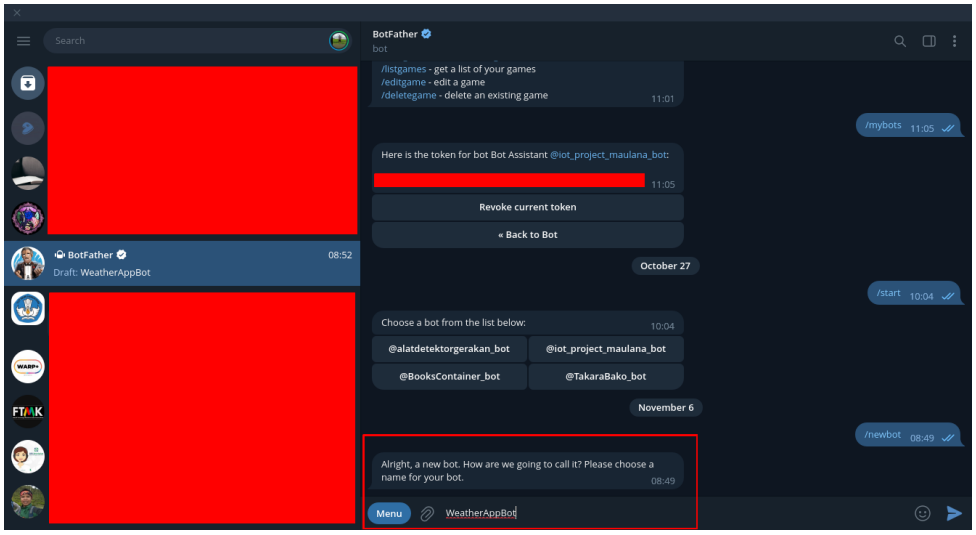
1. Tahap pertama yang dilakukan adalah membuat **Telegram Bot**. Pastikan untuk memiliki Akun Telegram untuk bisa memulai langkah ini
2. Cari **Bog Manager** dengan **@BotFather**



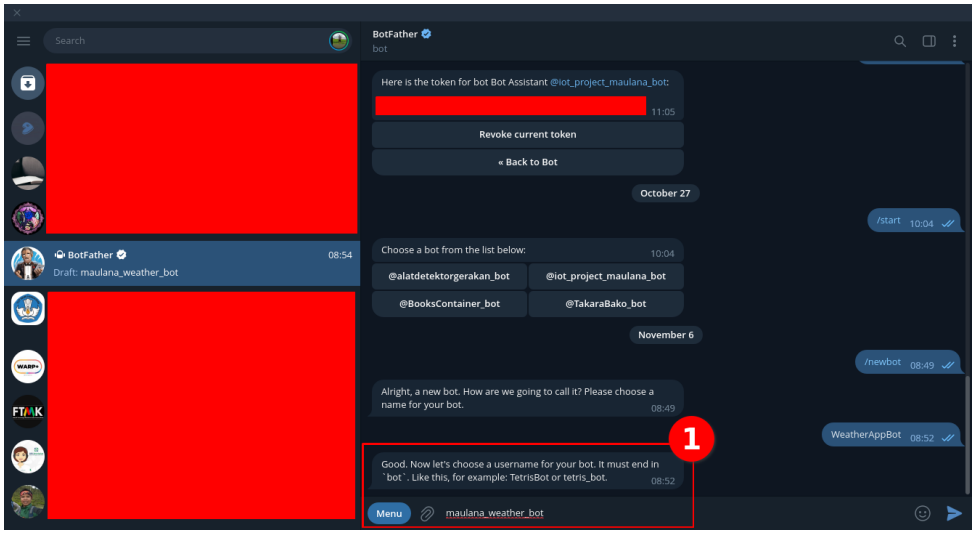
3. Gunakan perintah /**newbot** untuk membuat **Telegram Bot** baru



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

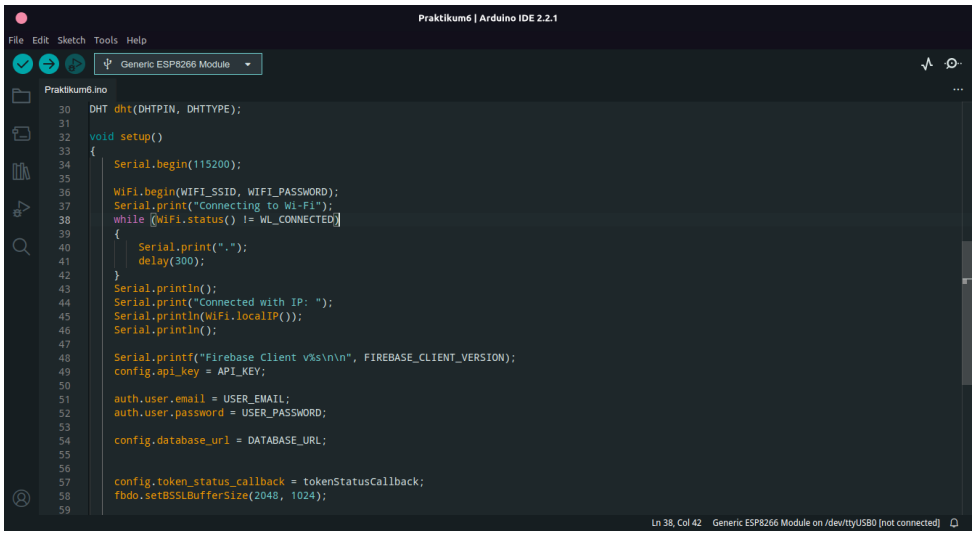


5. Lalu masukkan **username** untuk mempermudah pencarian **Telegram Bot**. Pastikan memiliki akhiran **_bot**

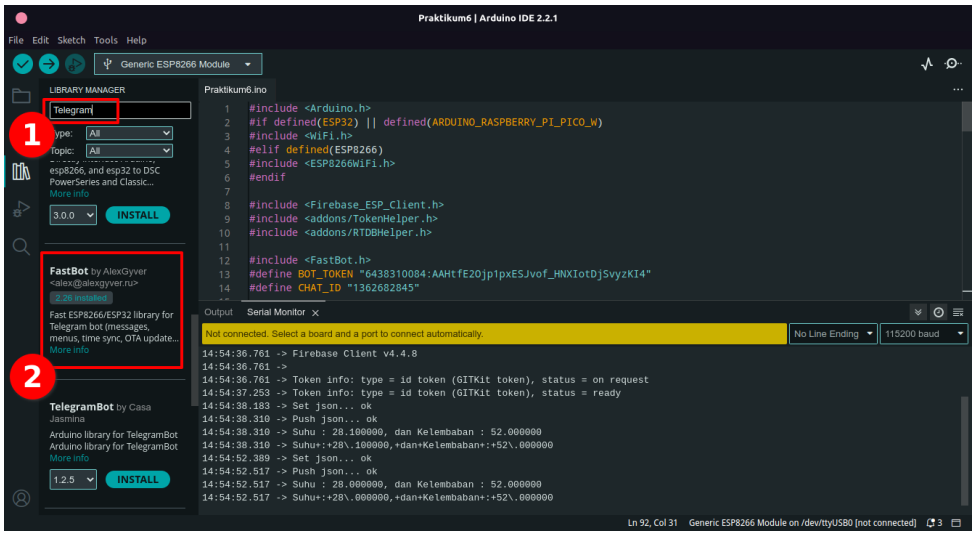


6. **Telegram Bot** sudah jadi dan **Token API** akan ditampilkan. Simpan baik-baik kode tersebut

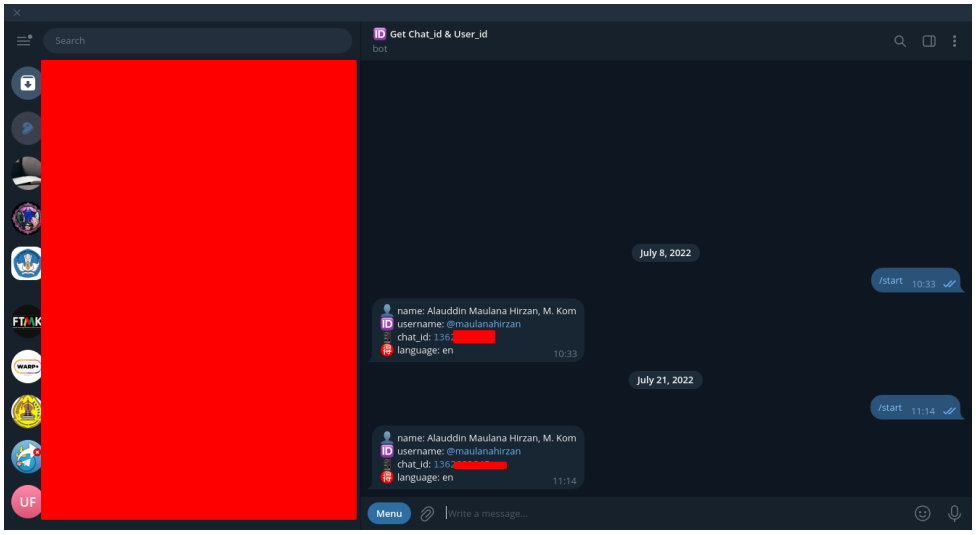
7. Berikutnya adalah membuka kembali **Praktikum 4** dengan menggunakan **Arduino IDE**. Lakukan **Save As** untuk menyimpan sebagai **Praktikum 6**



8. Install Library dengan nama **FastBot**



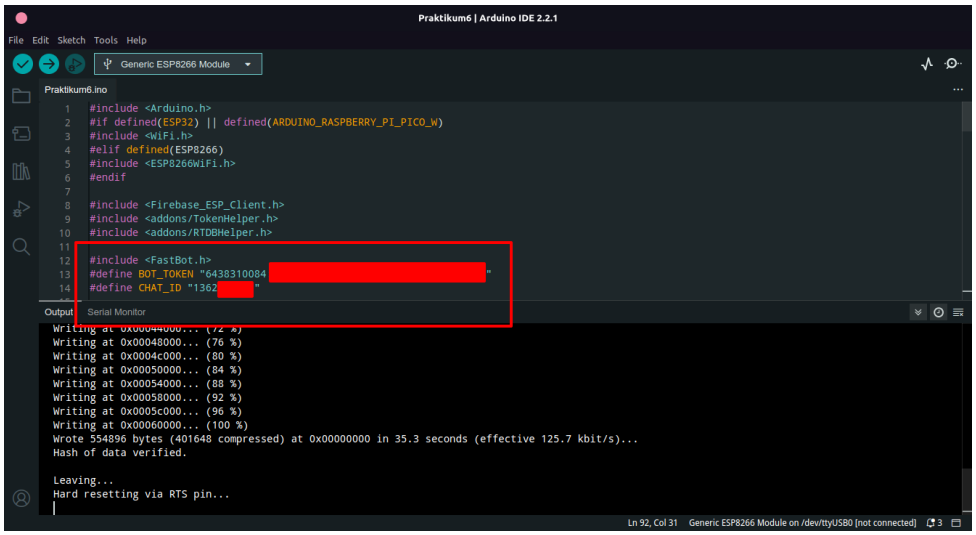
9. Berikutnya adalah mendapatkan **Chat ID** melalui Bot `https://t.me/chatIDrobot`



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

Sesudah

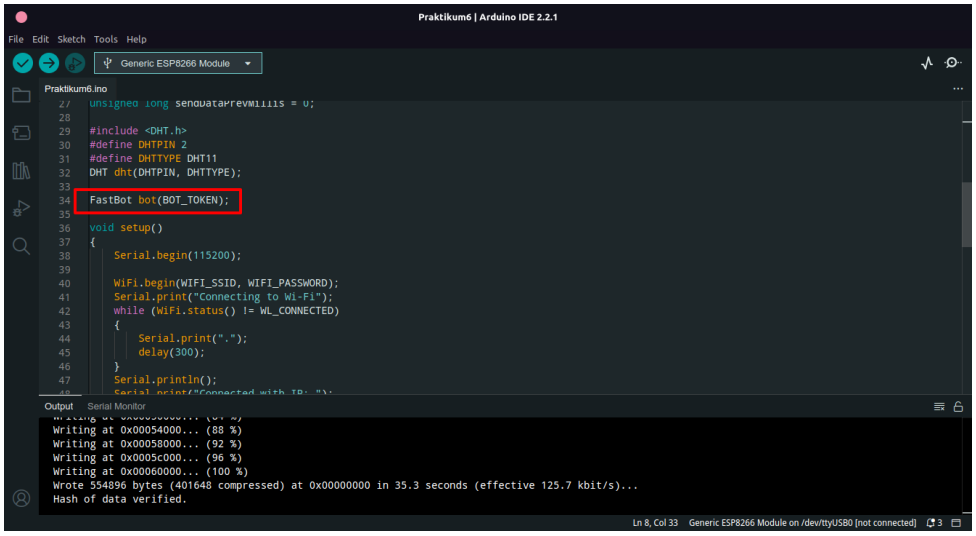
```
#include <FastBot.h>
#define BOT_TOKEN "<TOKEN BOT>"
#define CHAT_ID "<Chat ID>"
```



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

Sesudah

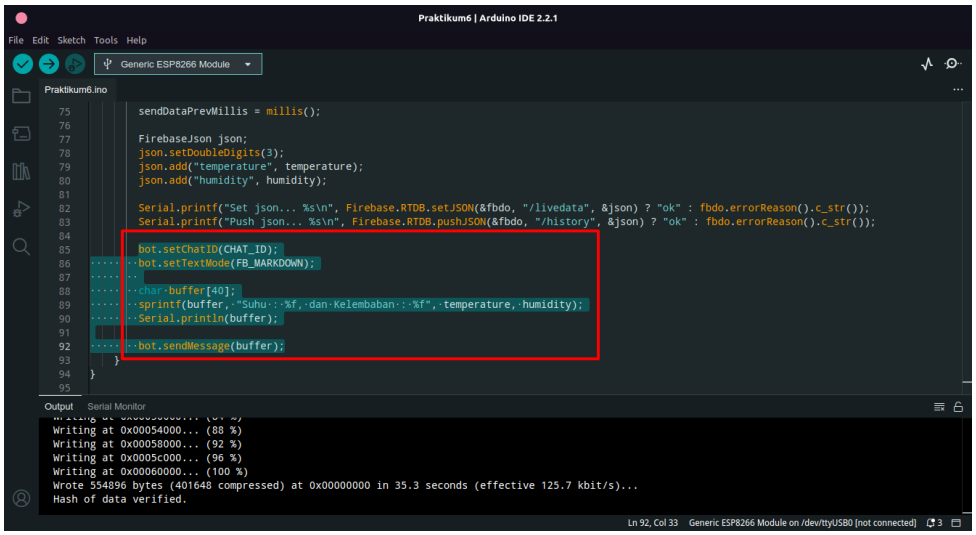
```
FastBot bot(BOT_TOKEN);
```



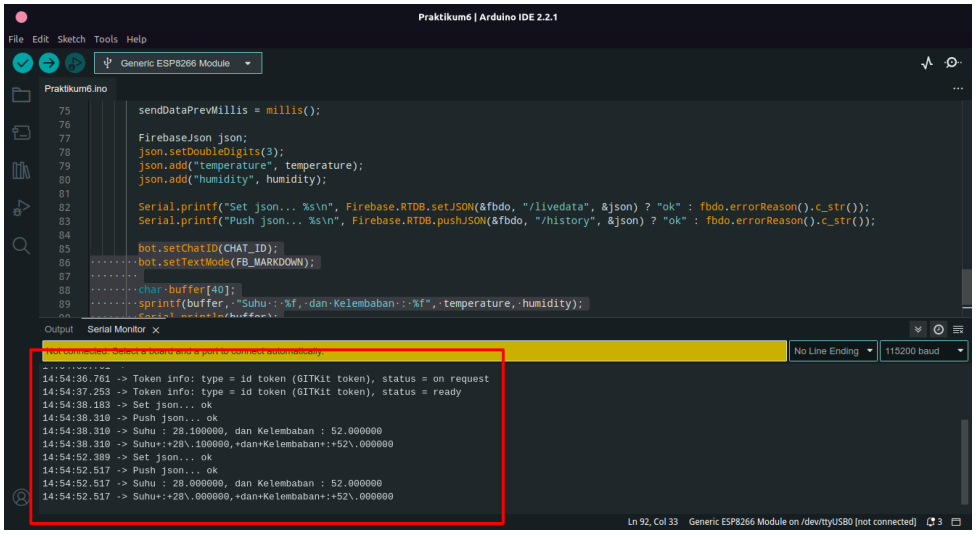
12. Setelah itu, cukup tambahkan kode berikut tepat di akhir fungsi **void loop()**

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);
```



13. Verifikasi dan Upload kode ke Perangkat



Bab 6

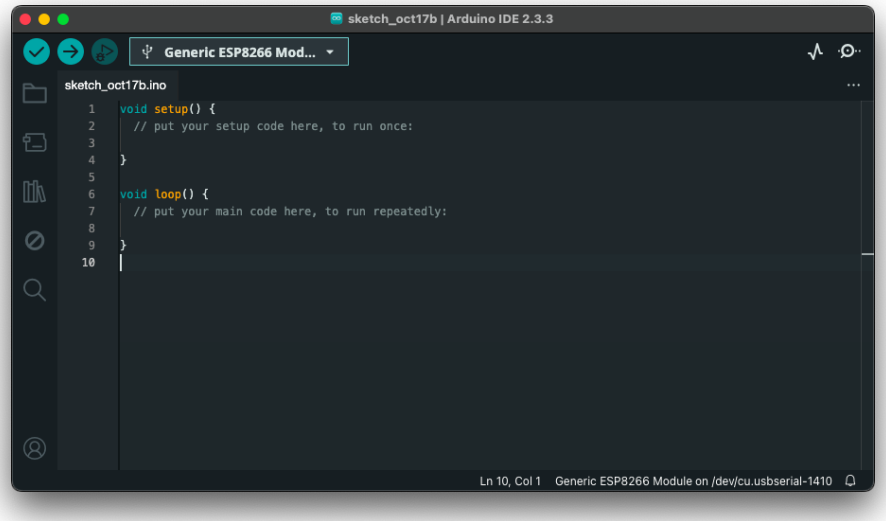
Praktikum 6

6.1 ESP8266, DHT-22, dan Linear Regression

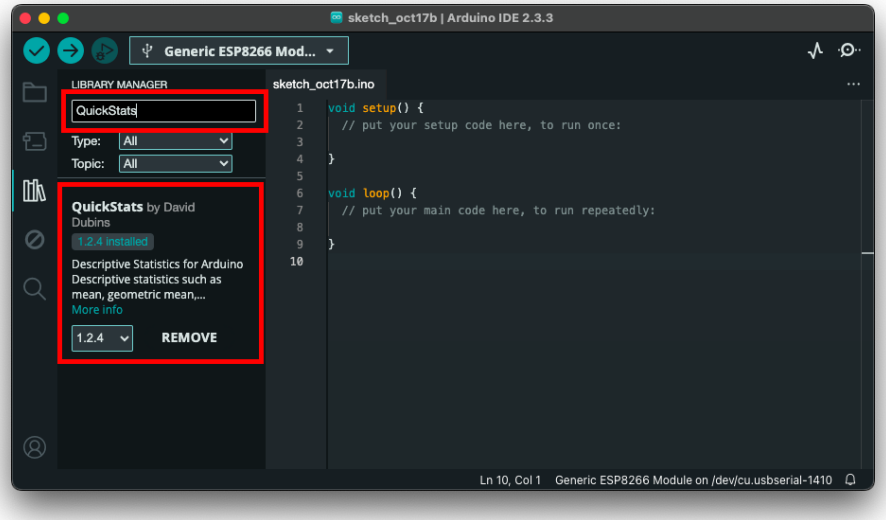
Di bagian ini mahasiswa diajarkan bagaimana mengimplementasikan algoritma *Linear Regression* sederhana dengan perangkat **ESP8266, dan DHT-22**. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 5** yang ada di halaman sebelumnya.

6.2 Tutorial

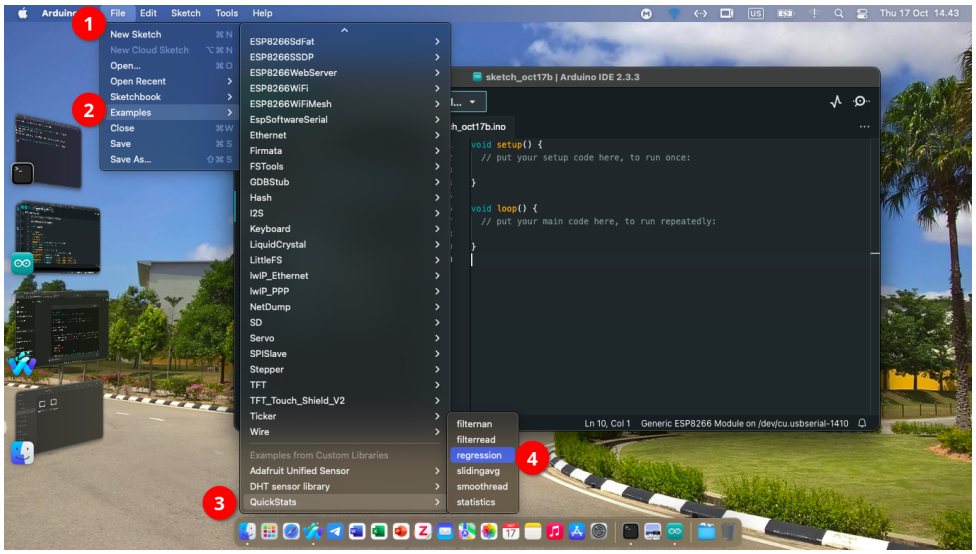
1. Buka kembali Arduino IDE untuk memulai projek baru. ESP8266 tidak harus tercolok.



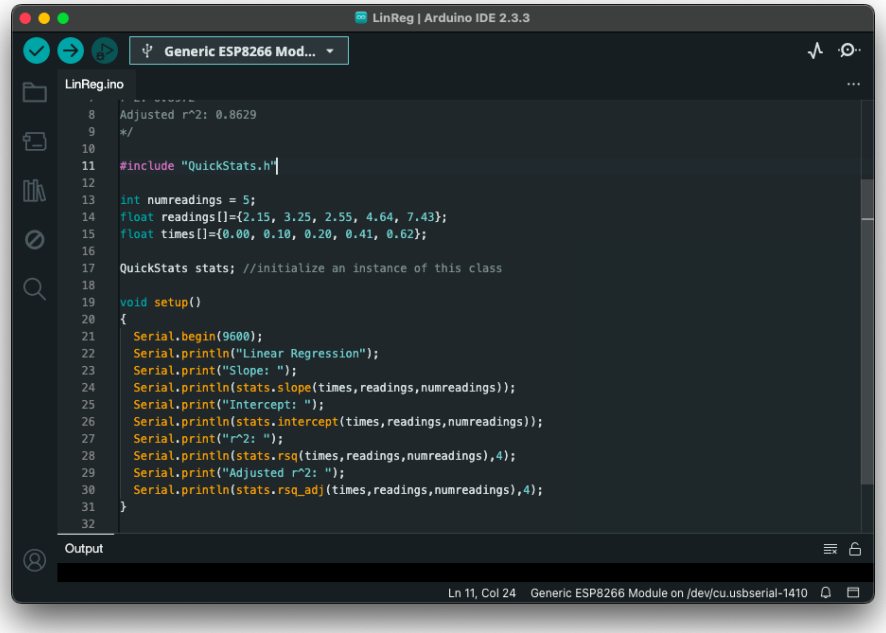
2. Install Library **QuickStats**



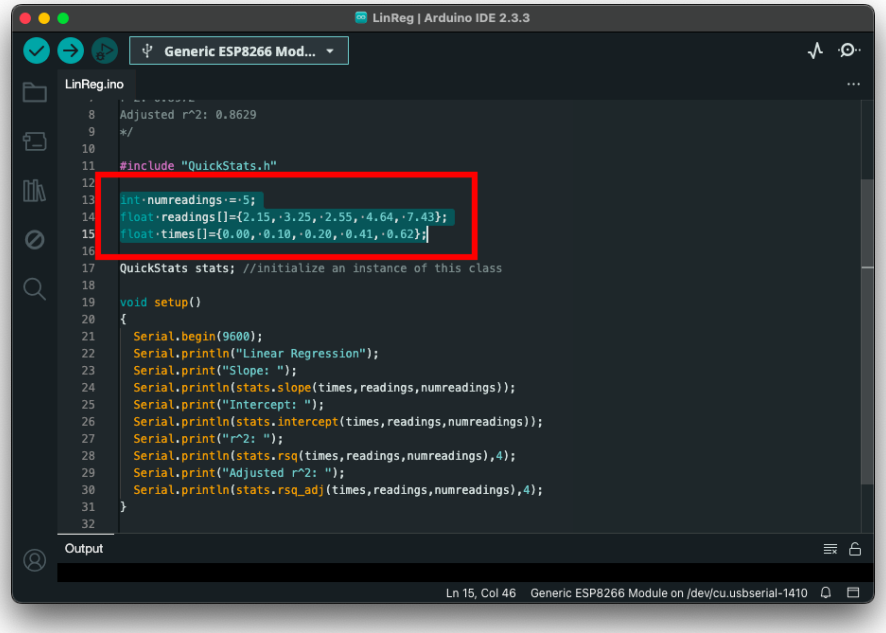
3. Buka contoh kode dari **Linear Regression** melalui menu **File** → **Examples** → **QuickStats** → **regresison**

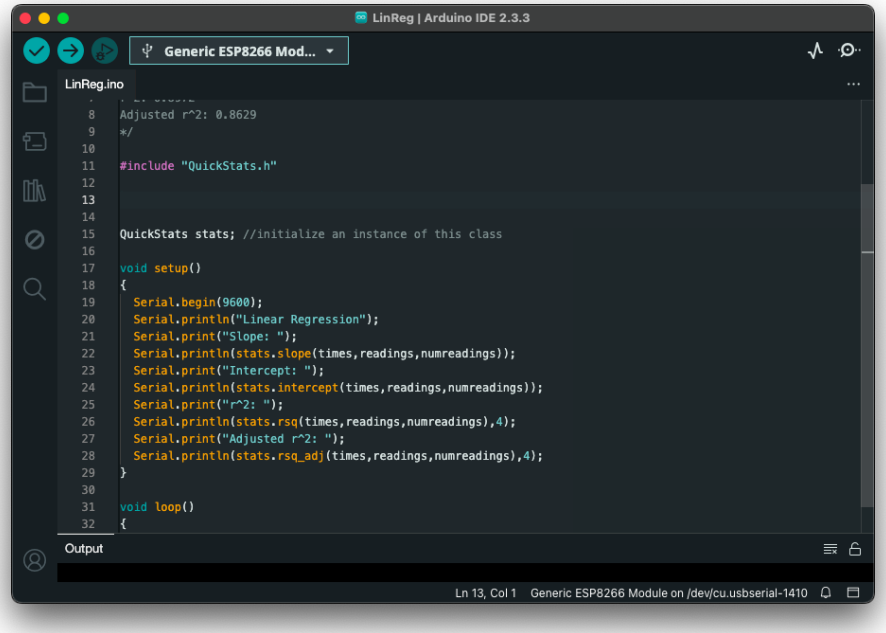


4. Tunggu window contoh kode **QuickStats** muncul



5. Hapus kode bagian yang ditandai

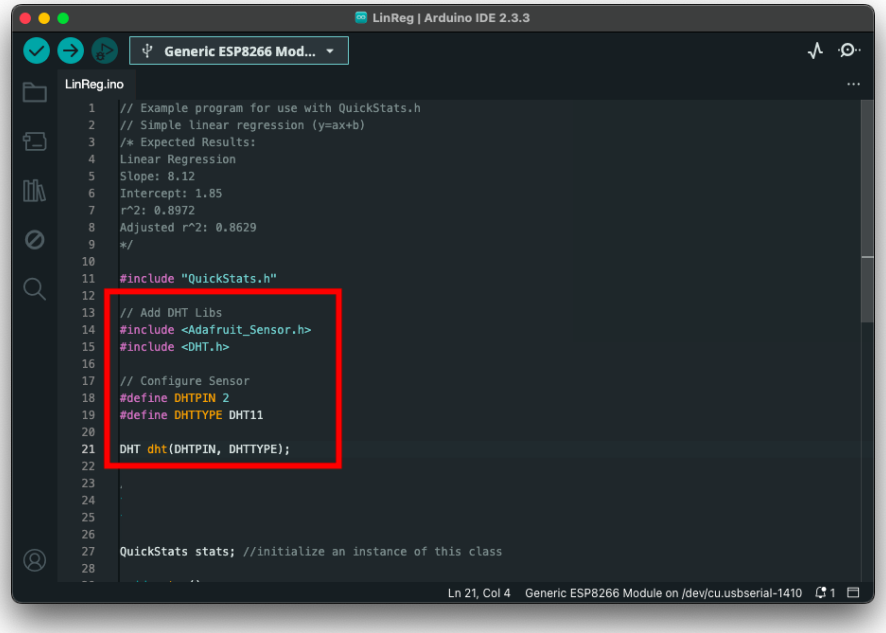




6. Di bagian bawah dari #**include "QuickStats.h"**, tambahkan kode berikut untuk sensor DHT-22:

Kode Program

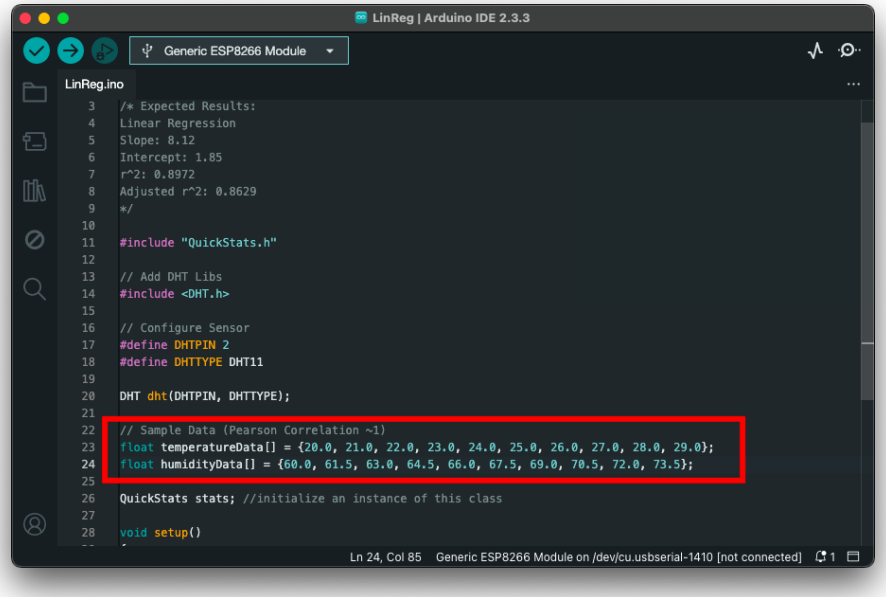
```
// Add DHT Libs
#include <DHT.h>

// 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

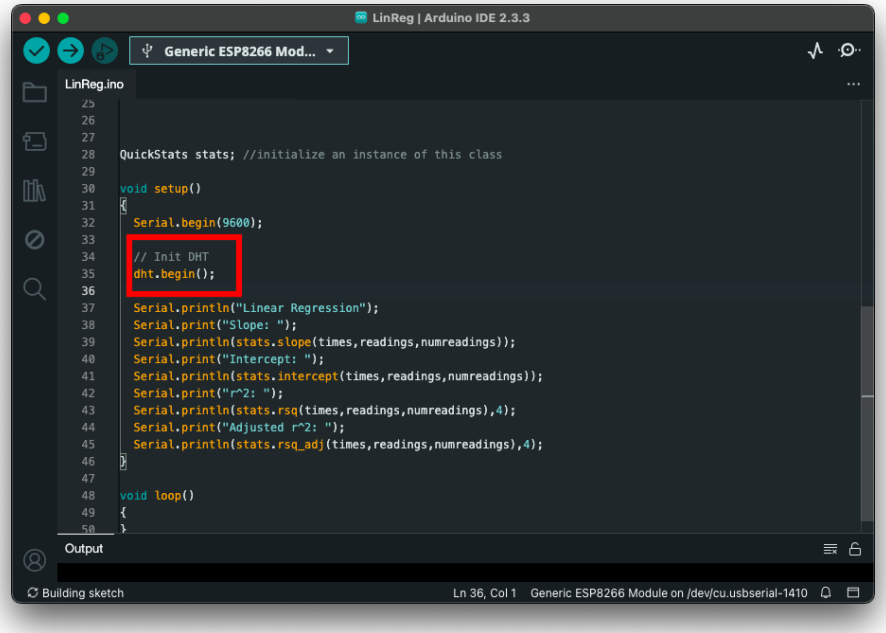
```
// Sample Data (Pearson Correlation ~1)
float temperatureData[] = { 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0,
27.0, 28.0, 29.0};

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



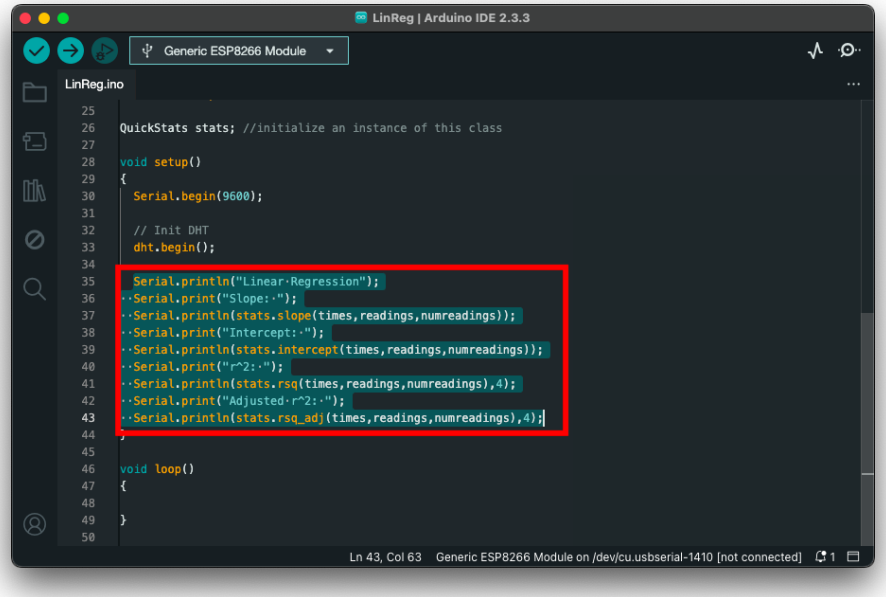
8. Setelah mendapatkan dana *training* untuk **Regresi Linier**, maka langkah berikutnya adalah menambahkan kode inisialisasi untuk sensor **DHT 11** dengan kode berikut setelah kode **Serial.begin(9600);**:

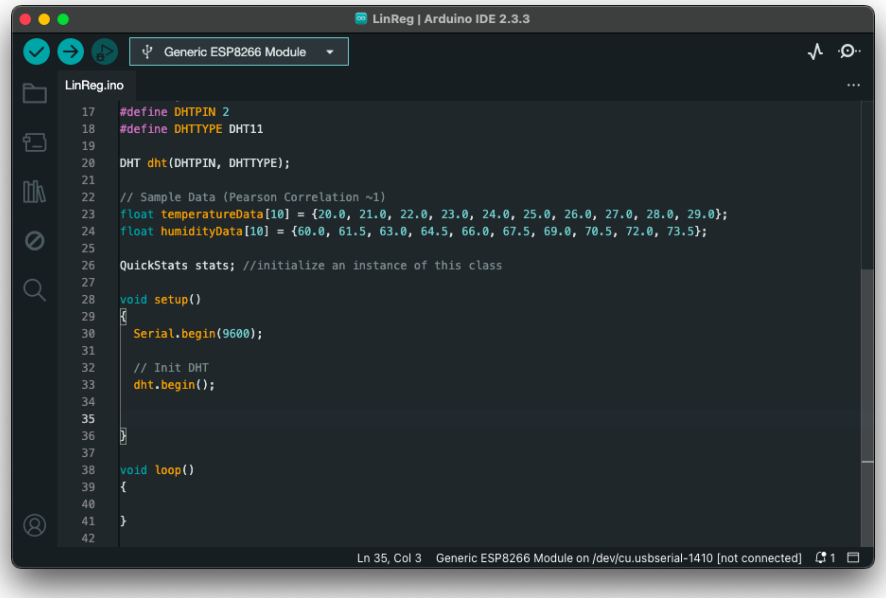
Kode Program

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



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

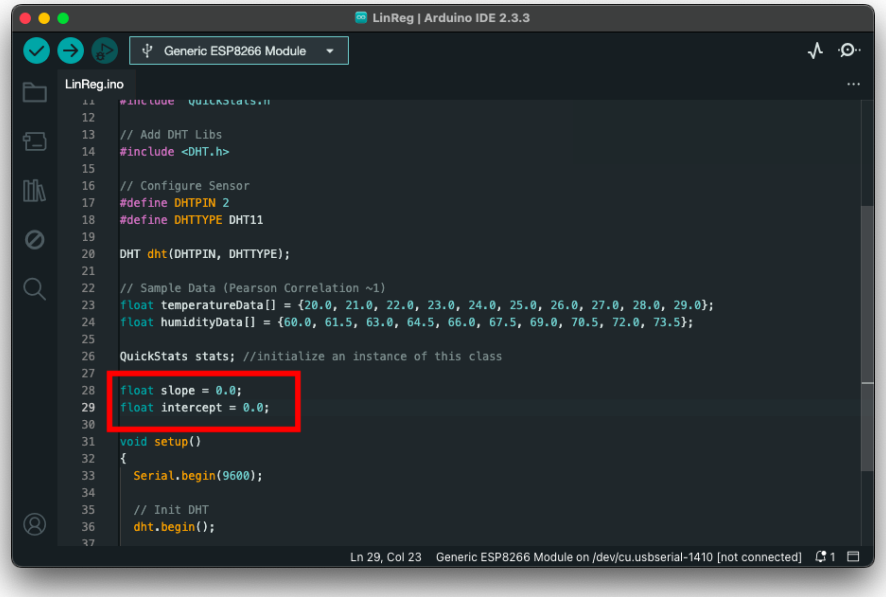




10. Kembali ke bagian atas, tambahkan kode setelah kode **QuickStats stats;**

Kode Program

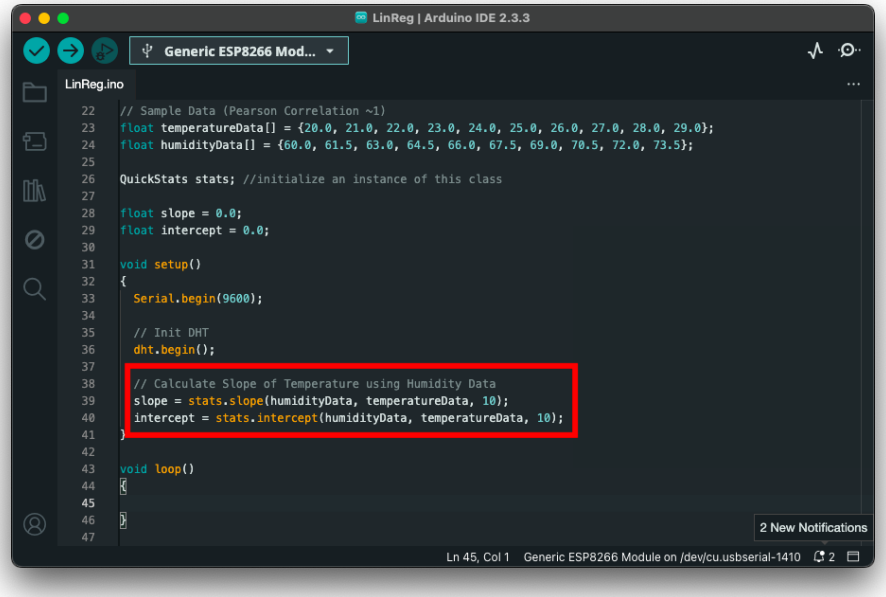
```
float slope = 0.0;
float intercept = 0.0;
```



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);
```

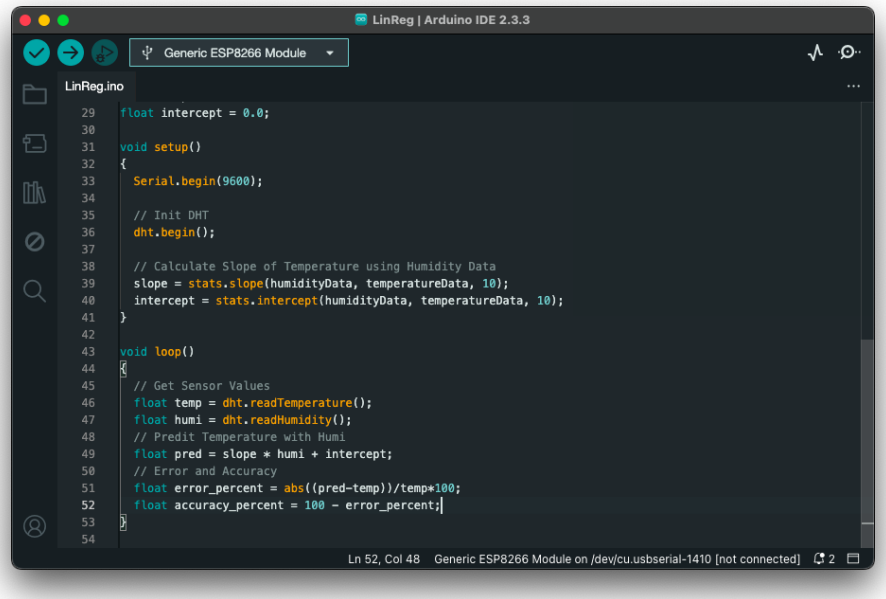


12. Bagian terakhir untuk kode **loop()** ada dua kode yang ditambahkan:

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

Kode Program

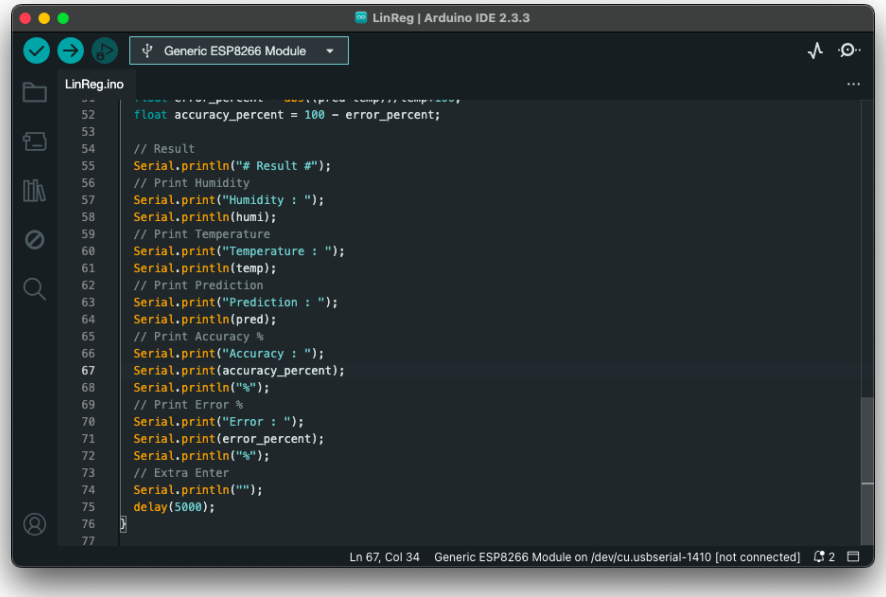
```
// Get Sensor Values
float temp = dht.readTemperature();
float humi = dht.readHumidity();
// Predit Temperature with Humi
float pred = slope * humi + intercept;
// Error and Accuracy
float error_percent = abs((pred-temp))/temp*100;
float accuracy_percent = 100 - error_percent;
```



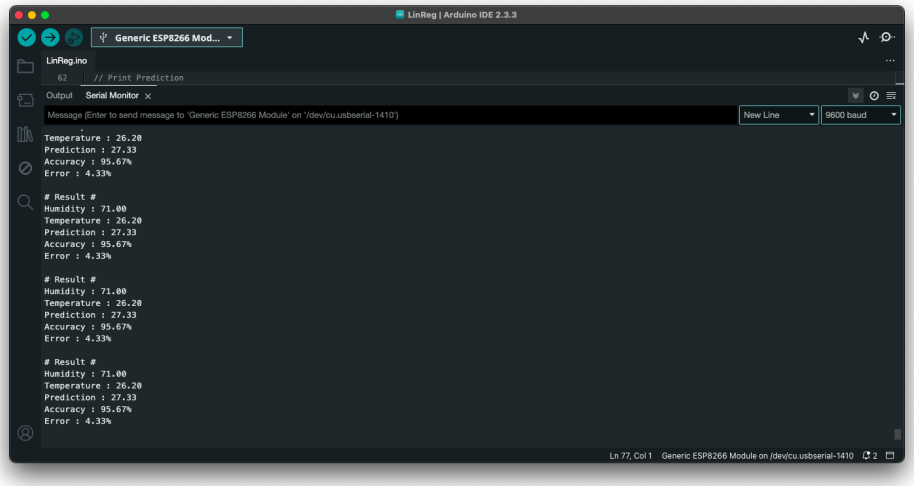
(b) Kode untuk menampilkan hasilnya, beserta delay 5 detik

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**


LinReg | Arduino IDE 2.3.3
Generic ESP8266 Mod...
LinReg.ino
62 // Print Prediction
Output
Serial Monitor
Message (Enter to send message to 'Generic ESP8266 Module' on '/dev/cu.usbserial-1410')
New Line
9600 baud
Temperature : 26.20
Prediction : 27.33
Accuracy : 95.67%
Error : 4.33%
Result
Humidity : 71.00
Temperature : 26.20
Prediction : 27.33
Accuracy : 95.67%
Error : 4.33%
Result
Humidity : 71.00
Temperature : 26.20
Prediction : 27.33
Accuracy : 95.67%
Error : 4.33%
Result
Humidity : 71.00
Temperature : 26.20
Prediction : 27.33
Accuracy : 95.67%
Error : 4.33%
Ln 77, Col 1
Generic ESP8266 Module on /dev/cu.usbserial-1410

Bab 7

Praktikum 7

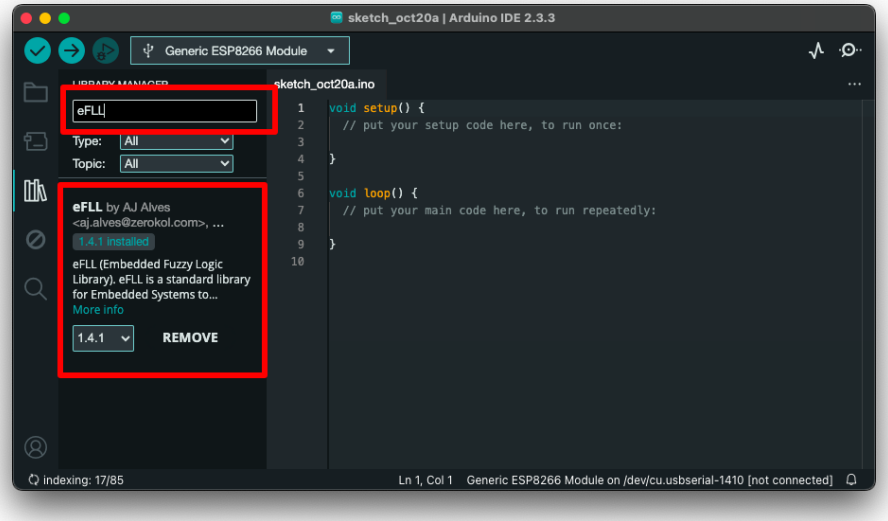
7.1 ESP8266, DHT-22, dan Fuzzy Logic

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.

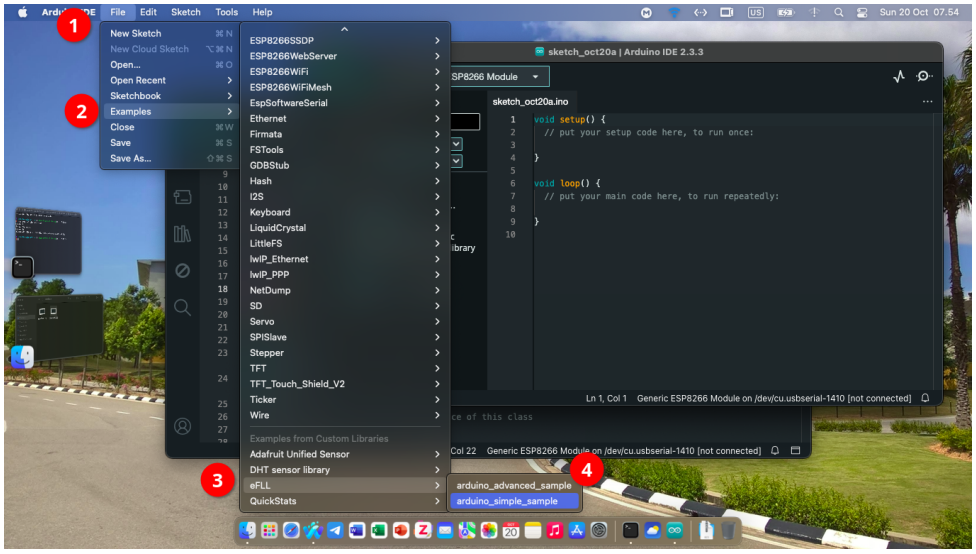
7.2 Tutorial

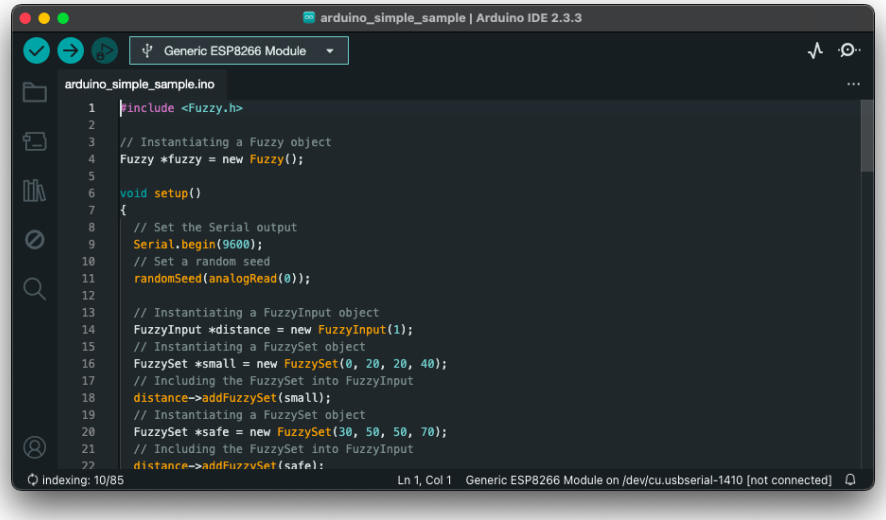
1. Buka Arduino IDE kembali

2. Install Library **eFLL**



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



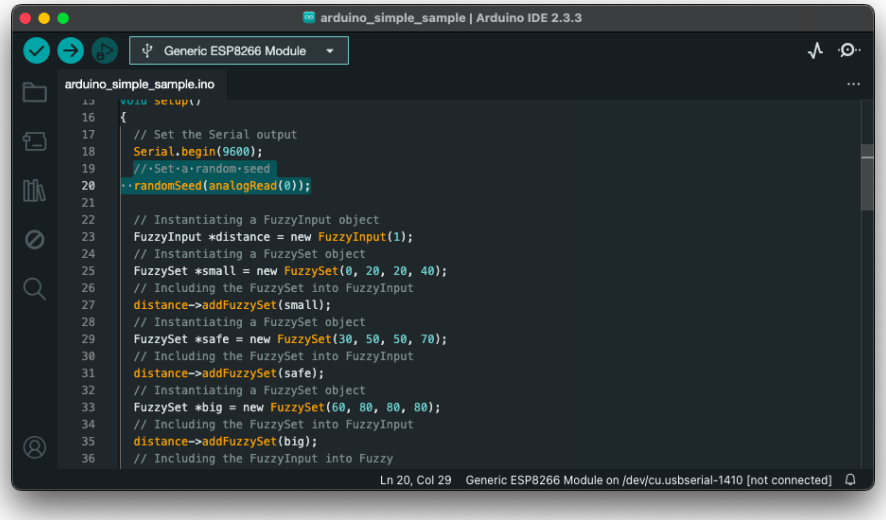


4. Di bagian bawah kode **#include <Fuzzy.h>**, tambahkan kode library DHT-22

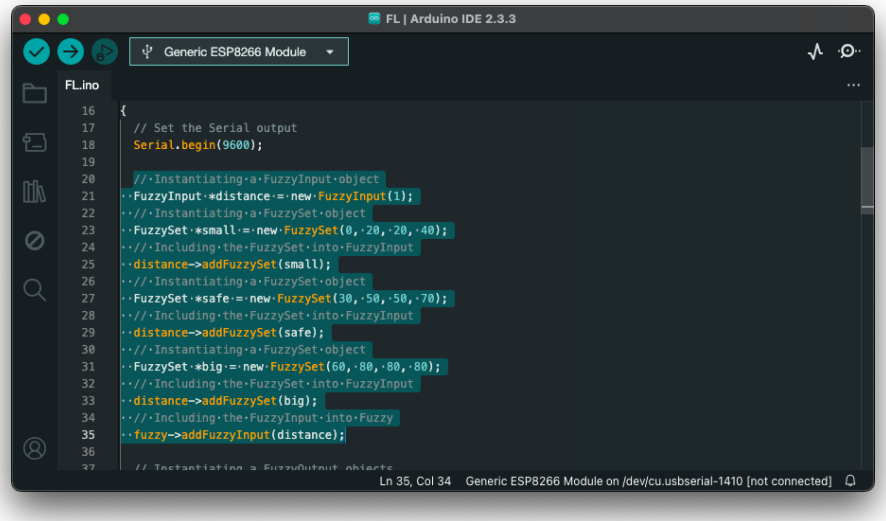
Kode Program

```
// 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.



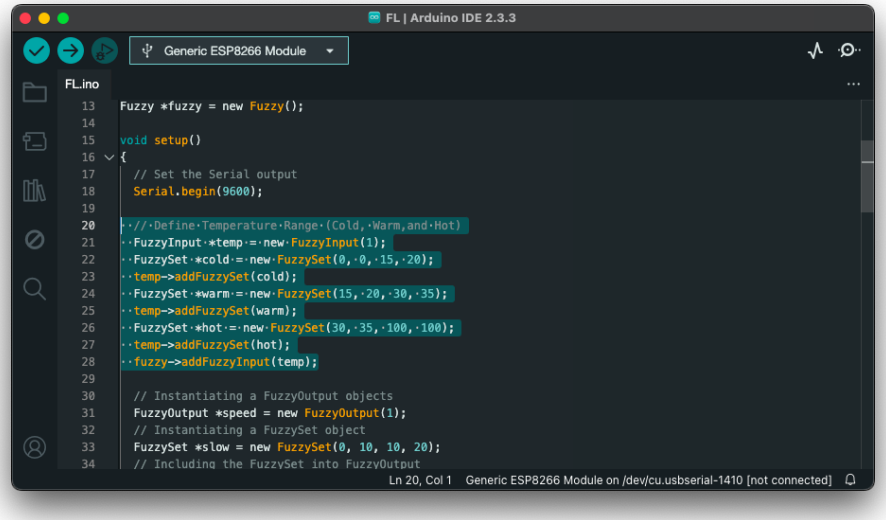
6. Kemudian hapus lagi bagian kode yang ditunjuk oleh gambar



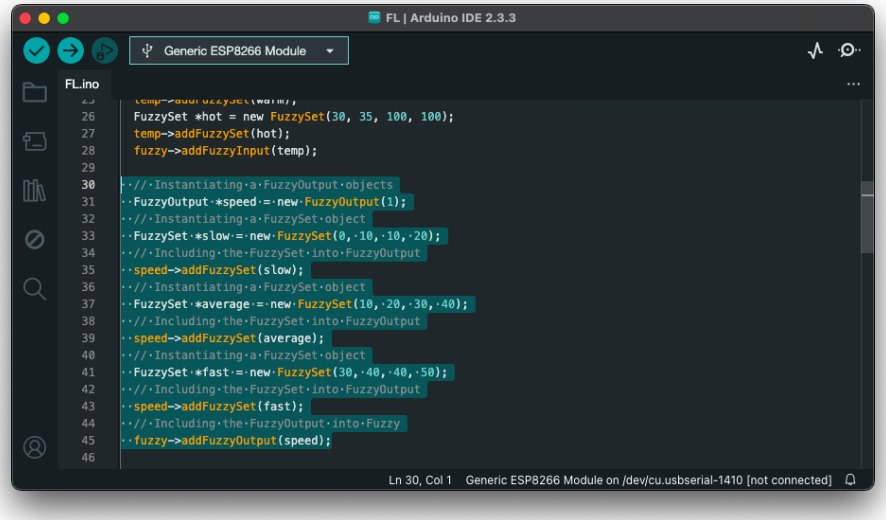
7. Di bawah baris kode **Serial.begin(9600);** yang sudah bersih tadi, tambahkan persis di bawahnya **Kode Parameter Input Fuzzy, dengan contoh Suhu/Temperature (Dingin, Hangat, dan Panas)**:

Kode Program

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



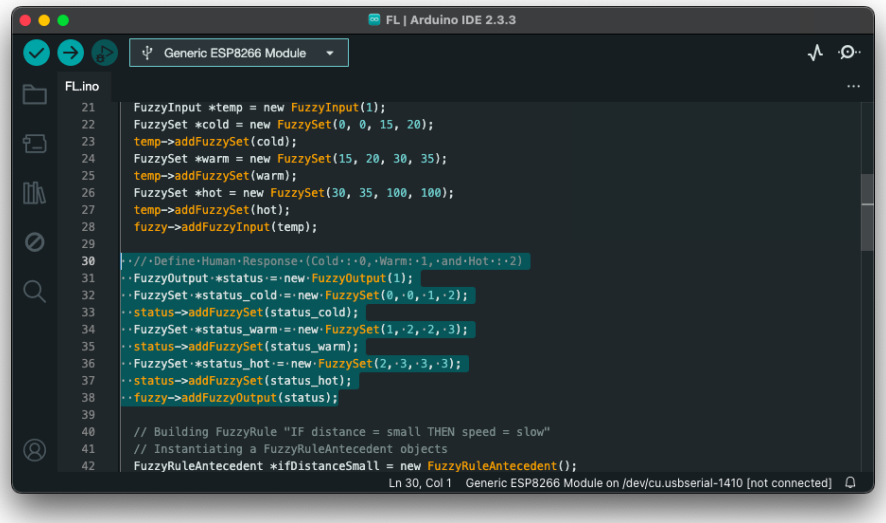
8. Setelah itu, hapus bagian kode yang ditunjuk gambar. Kode yang dihapus tepat berada di bawah kode yang baru dimasukkan.



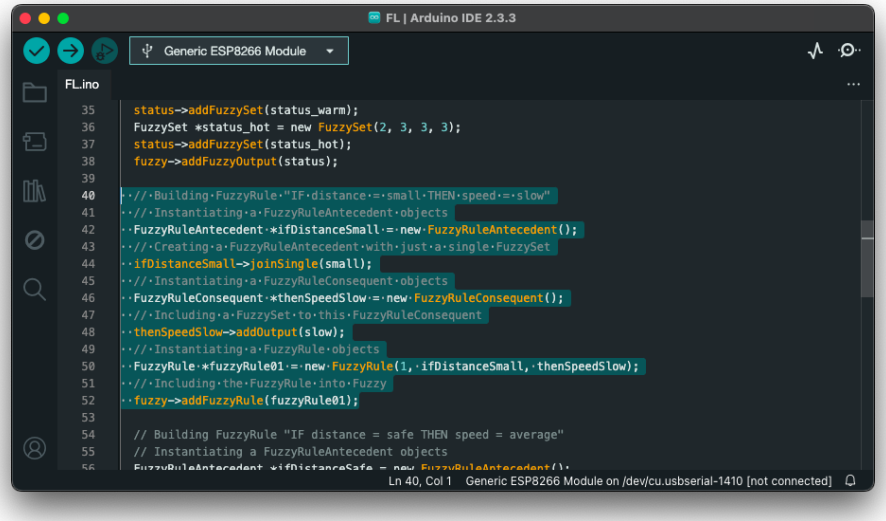
9. Lalu tambahkan **Kode Parameter Output Fuzzy yang dibuat agar mudah dipahami oleh manusia (Dingin = 1, Hangat = 2, dan Panas = 3)** tepat di bawah **Kode Parameter Input** tadi.

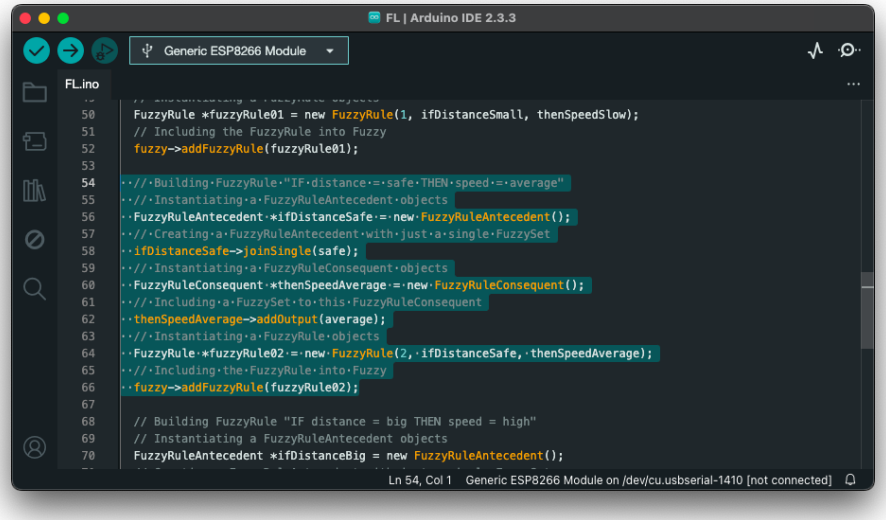
Kode Program

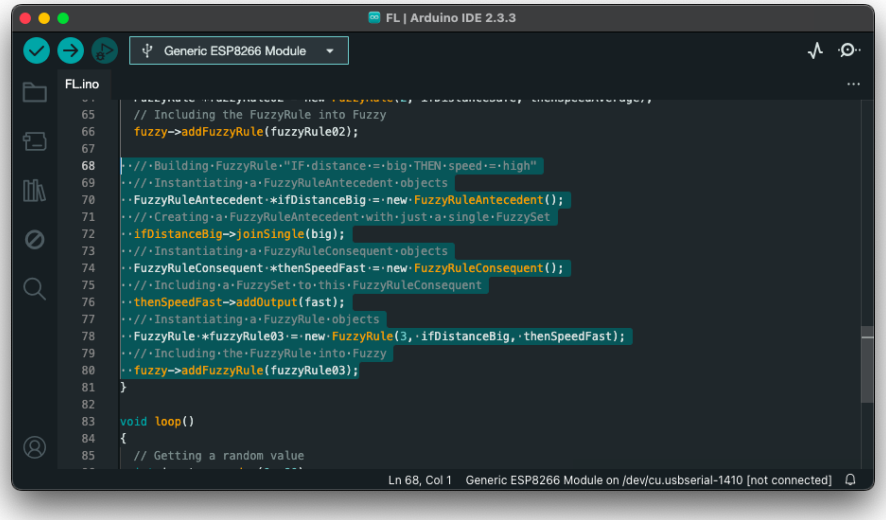
```
// Define Human Response (Cold : 1, Warm: 2, and Hot : 3)
FuzzyOutput *status = new FuzzyOutput(1);
FuzzySet *status_cold = new FuzzySet(0, 1, 1, 2);
status->addFuzzySet(status_cold);
FuzzySet *status_warm = new FuzzySet(1, 2, 2, 3);
status->addFuzzySet(status_warm);
FuzzySet *status_hot = new FuzzySet(2,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



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





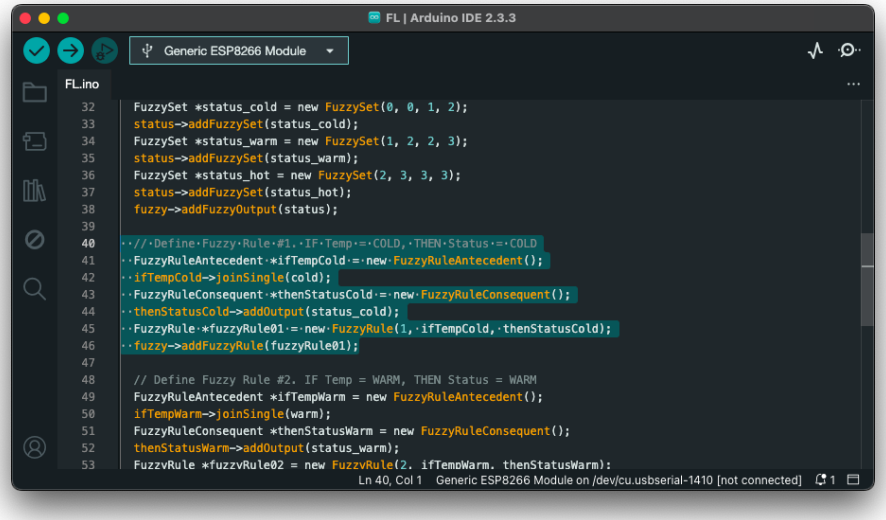


11. Setelah dihapus, masukkan kode **Fuzzy Rules** baru tepat di bawah kode **Fuzzy Output** yang telah dibuat sebelumnya

    (a) **Fuzzy Rule 1**

Kode Program

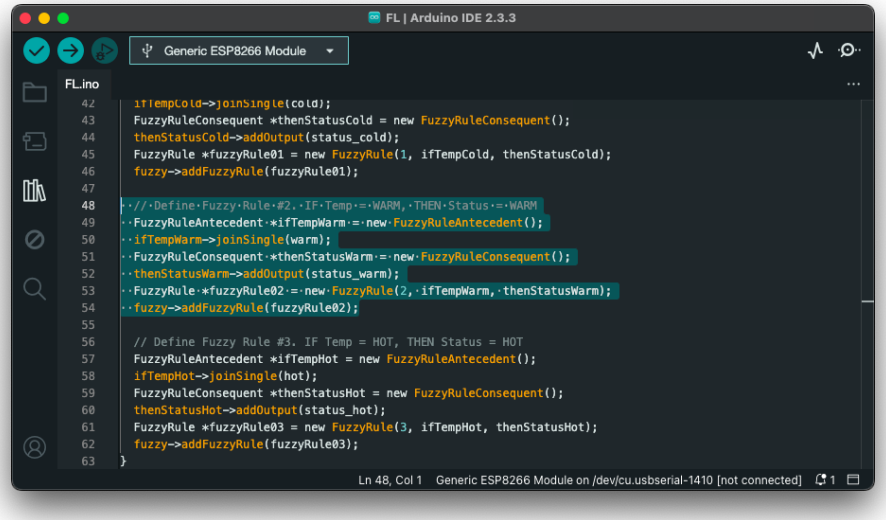
```
// Define Fuzzy Rule #1. IF Temp = COLD, THEN Status = COLD
FuzzyRuleAntecedent *ifTempCold = new FuzzyRuleAntecedent();
ifTempCold->joinSingle(cold);
FuzzyRuleConsequent *thenStatusCold = new FuzzyRuleConsequent();
thenStatusCold->addOutput(status_cold);
FuzzyRule *fuzzyRule01 = new FuzzyRule(1,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



(b) **Fuzzy Rule 2**

Kode Program

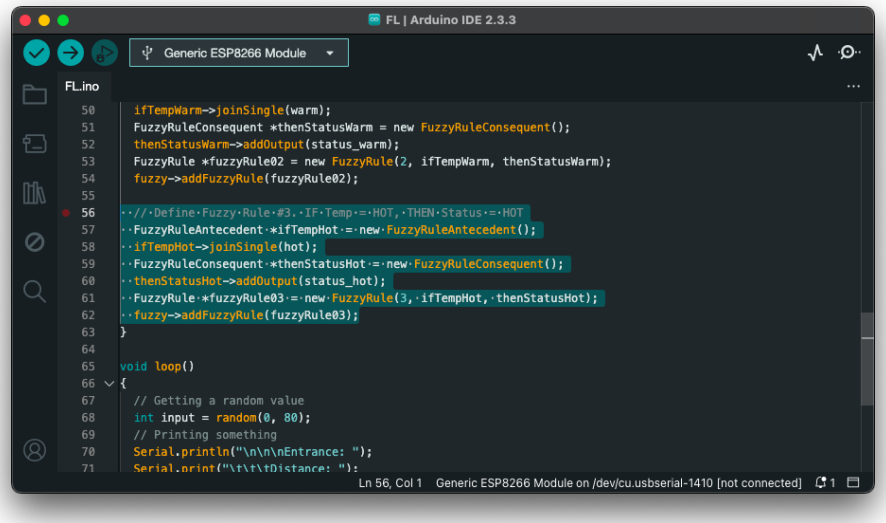
```
// Define Fuzzy Rule #2. IF Temp = WARM, THEN Status = WARM
FuzzyRuleAntecedent *ifTempWarm = new FuzzyRuleAntecedent();
ifTempWarm->joinSingle(warm);
FuzzyRuleConsequent *thenStatusWarm = new FuzzyRuleConsequent();
thenStatusWarm->addOutput(status_warm);
FuzzyRule *fuzzyRule02 = new FuzzyRule(2,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



(c) **Fuzzy Rule 3**

Kode Program

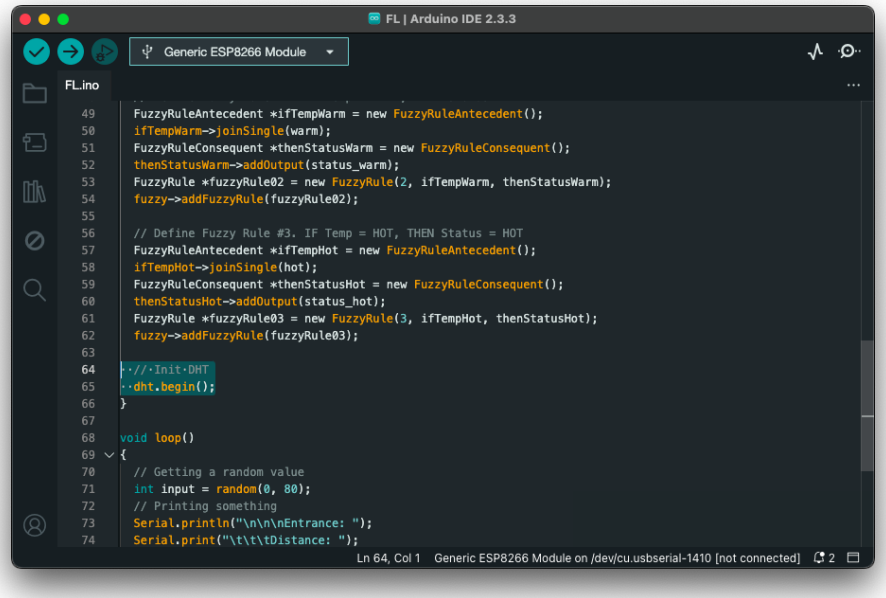
```
// Define Fuzzy Rule #3. IF Temp = HOT, THEN Status = HOT
FuzzyRuleAntecedent *ifTempHot = new FuzzyRuleAntecedent();
ifTempHot->joinSingle(hot);
FuzzyRuleConsequent *thenStatusHot = new FuzzyRuleConsequent();
thenStatusHot->addOutput(status_hot);
FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifTempHot,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



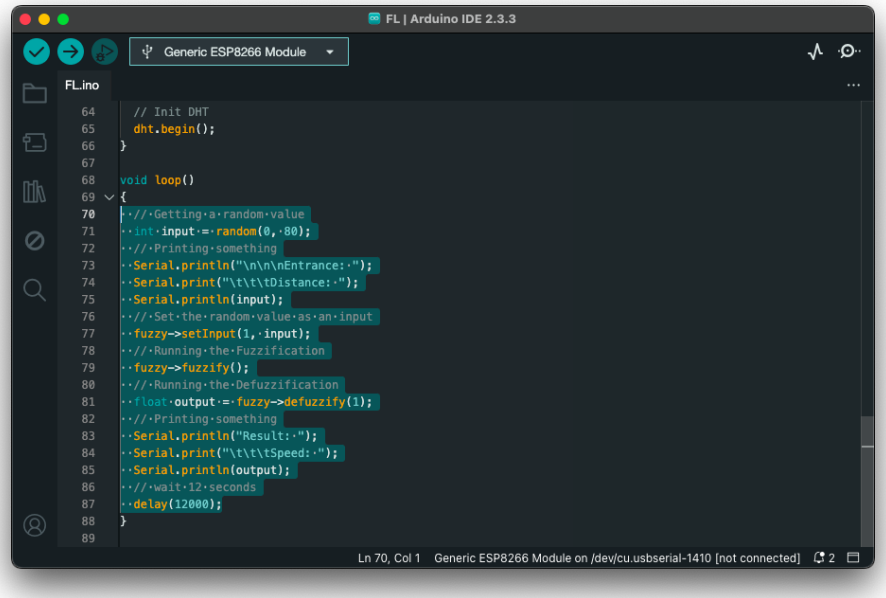
12. Bagian terakhir dari fungsi **setup()** adalah kode inisialisasi dht

Kode Program

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



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

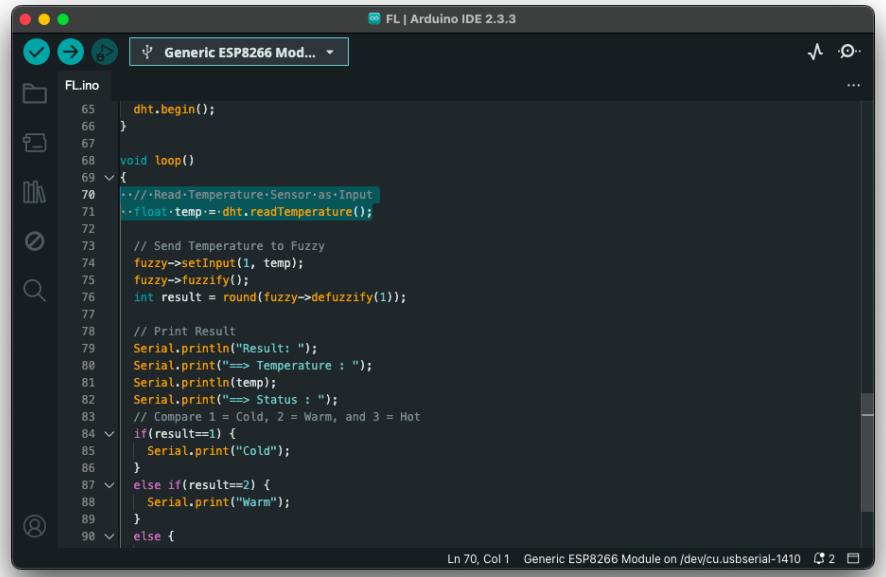


14. Lalu masukkan satu per satu kode berikut:

    (a) Baris kode pembaca sensor suhu

Kode Program

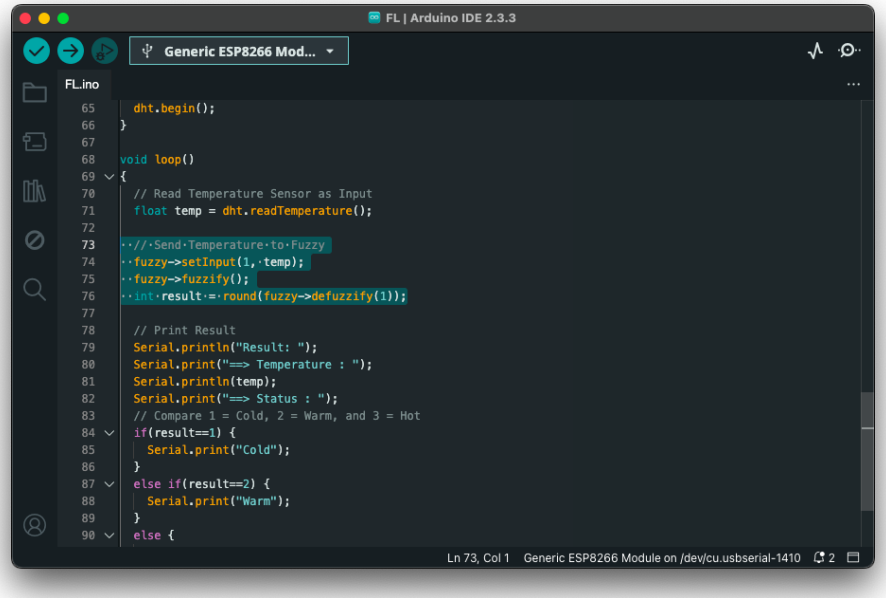
```
// Read Temperature Sensor as Input
float temp = dht.readTemperature();
```



    (b) Berikutnya kode Fuzzifikasi dan Defuzzifikasi

Kode Program

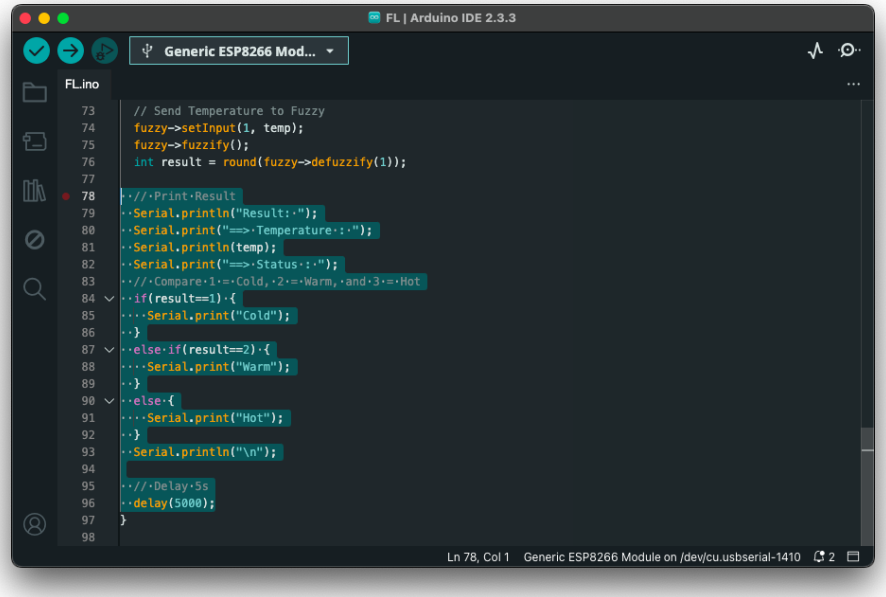
```
// Send Temperature to Fuzzy
fuzzy->setInput(1, temp);
fuzzy->fuzzify();
int result = round(fuzzy->defuzzify(1));
```



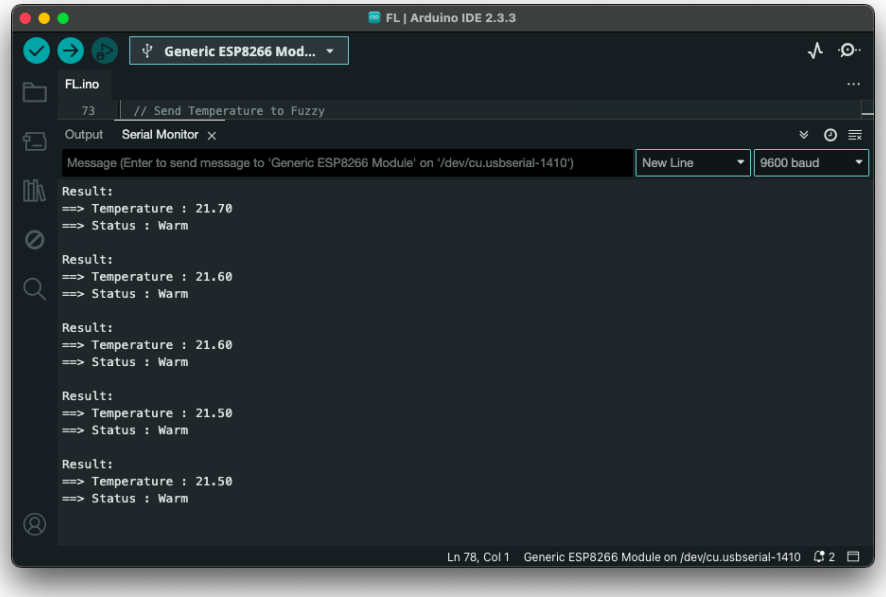
(c) Terakhir, kode cetak hasil dan delay

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);
```



15. Colokkan ESP8266 dan DHT-22, dan unggah kode. Lihat hasil seperti berikut



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)