Universitas Semarang
Fakultas Teknologi Informasi dan Komunikasi
Teknik Informatika

TIS18755P
Internet of Thing

Modul Praktikum Mahasiswa

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

Daftar Isi

Pendahuluan **2**

0.1 Mengenal *Internet of Things* 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, DHT11, dan AdafruitIO 14

2.2 Tutorial 14

3 Praktikum 3 **27**

3.1 ESP8266, DHT11, dan Thingspeak 27

3.2 Tutorial 27

4 Praktikum 4 **40**

4.1 ESP8266, DHT11, dan Firebase Realtime 40

4.2 Tutorial 40

5 Praktikum 5 **53**

5.1 NodeMCU, DHT11, dan Web App 53

5.2 Tutorial 53

6 Praktikum 6 **64**

6.1 ESP8266, DHT11, dan Telegram Bot 64

6.2 Tutorial 64

7 Praktikum 7 **70**

7.1 Observasi dengan Internet of Things 70

7.2 Tutorial 70

8 Praktikum 8 **71**

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*

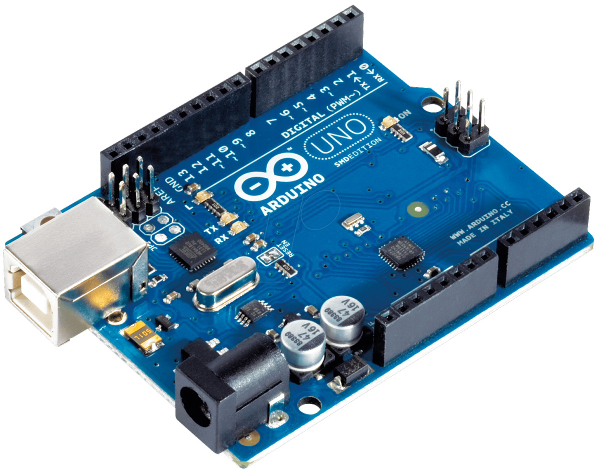
Internet of Things merupakan sebuah teknologi yang di mana mengizinkan setiap perangkat-perangkat yang memiliki kekuatan komputasi untuk berkomunikasi satu dengan yang lainnya tanpa campur tangan manusia untuk menyelesaikan suatu tugas atau fungsi.
Teknologi ini dapat diimplementasikan ke berbagai macam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

Gambar 1: Internet of Things

0.2 Perangkat Board IoT

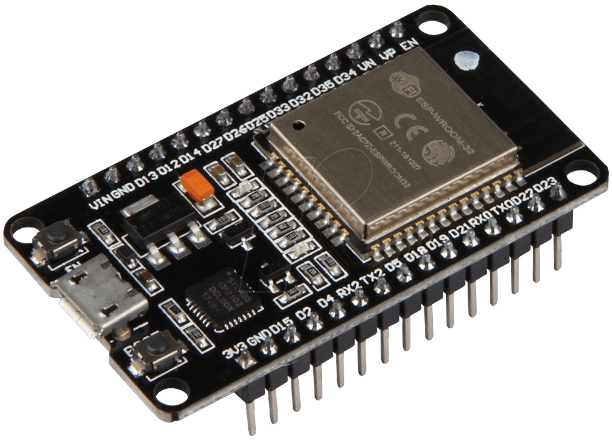
Untuk membangun sebuah perangkat berbasis IoT, komponen dasar seperti **Board** sangatlah vital untuk dipunyai. Terdapat berbagai macam board yang dapat dibeli secara luring maupun daring, dengan variasi harga yang juga berbeda mulai dari paling murah hingga mewah. Semakin kompleks masalah yang dapat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



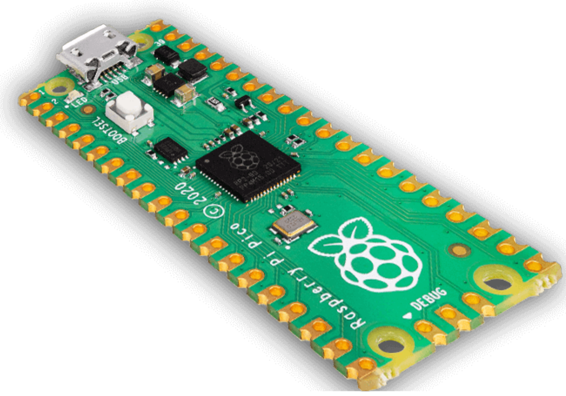
Gambar 2: Board Arduino

2. NodeMCU



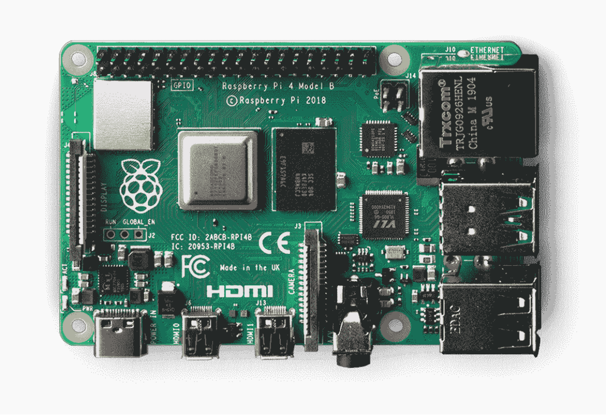
Gambar 3: Board NodeMCU

3. Raspberry Pi Pico



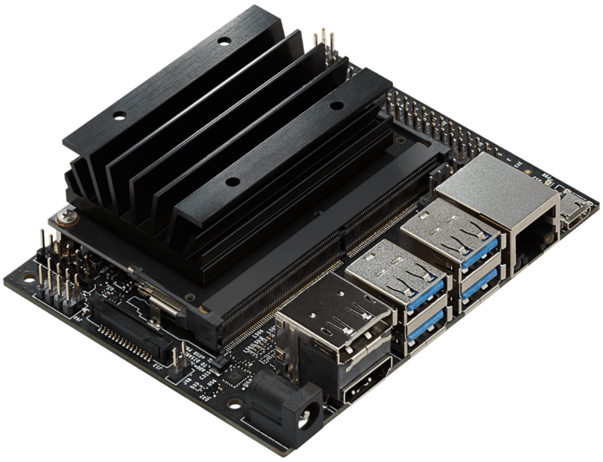
Gambar 4: Board Pico

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



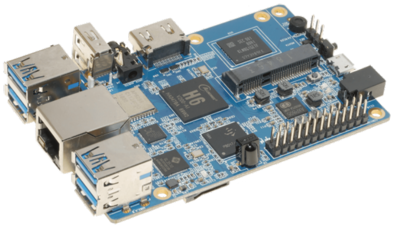
Gambar 5: Board Pi 4B

5. NVidia Jetson



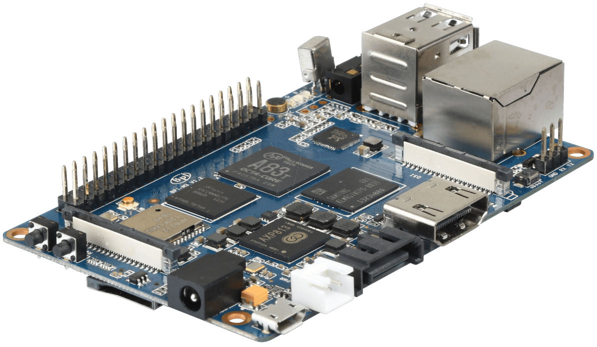
Gambar 6: Board NVidia Jetson

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-11

0.4 Perangkat Lunak

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

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

Bab 1

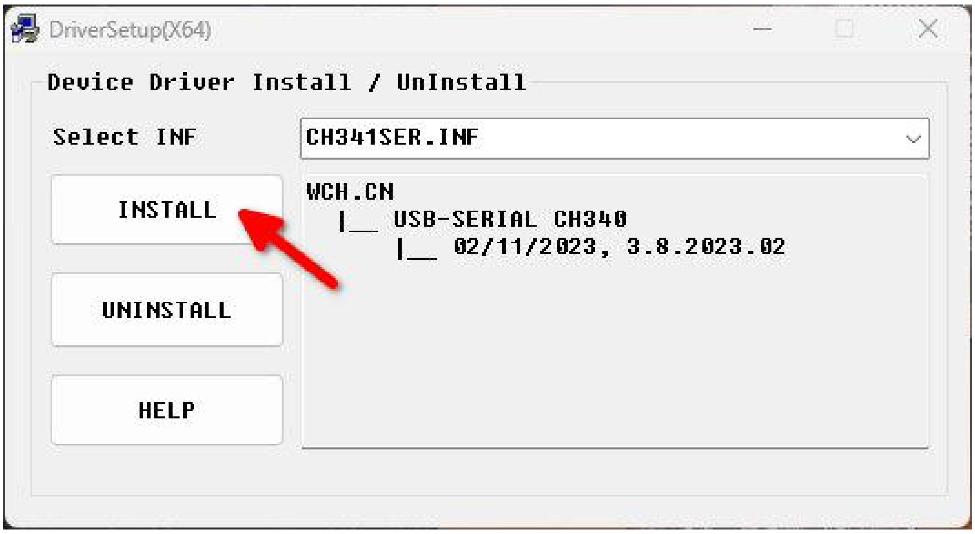
Praktikum 1

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)



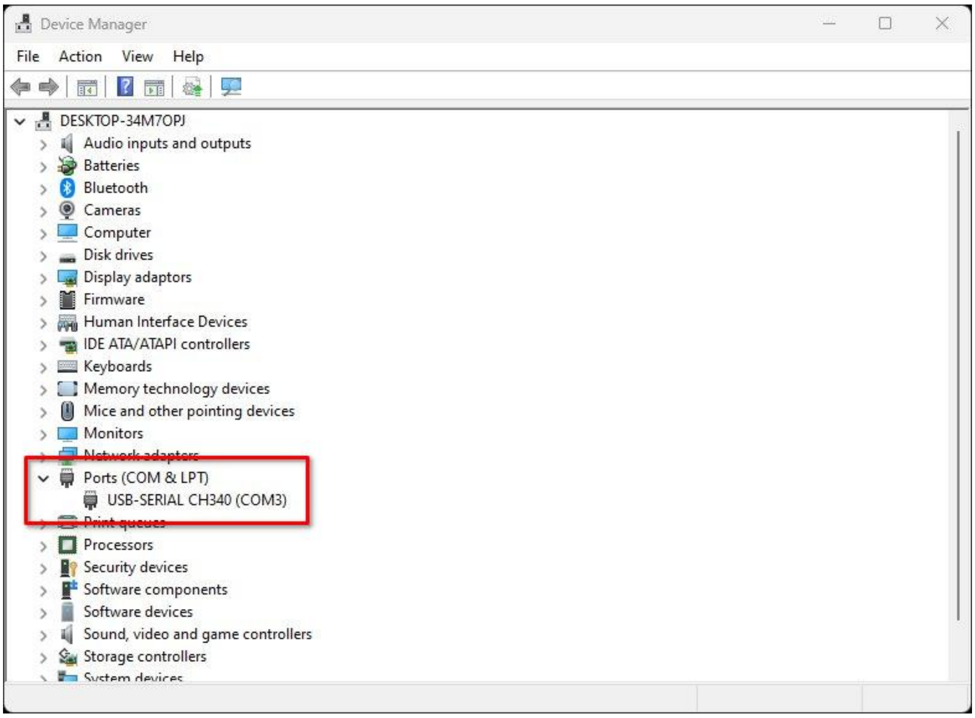
DriverSetup(X64)

Device Driver Install / UnInstall

Select INF: CH341SER.INF

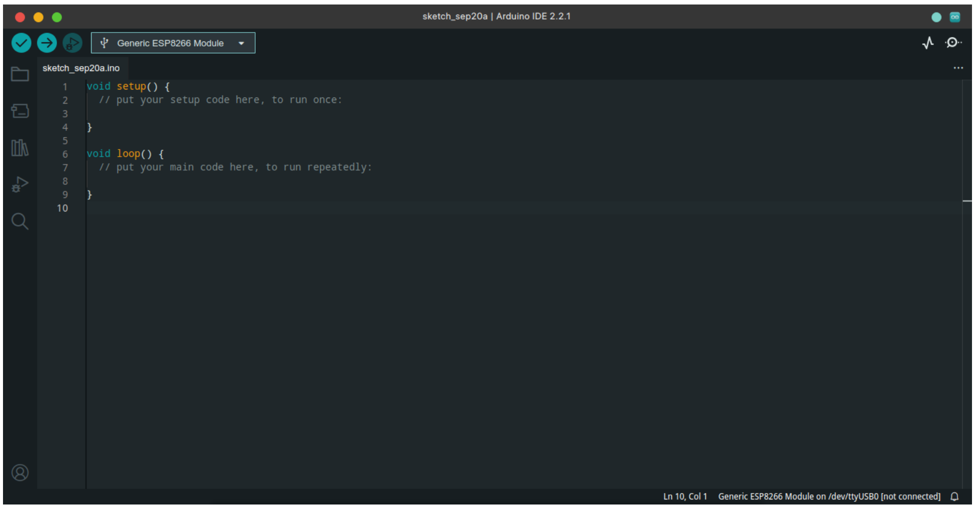
```
WCH.CN
  |__ USB-SERIAL CH340
      |__ 02/11/2023, 3.8.2023.02
```

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



- DESKTOP-34M7OPJ
  - Audio inputs and outputs
  - Batteries
  - Bluetooth
  - Cameras
  - Computer
  - Disk drives
  - Display adaptors
  - Firmware
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Keyboards
  - Memory technology devices
  - Mice and other pointing devices
  - Monitors
  - Network adaptors
  - Ports (COM & LPT)
    - USB-SERIAL CH340 (COM3)
  - Print queues
  - Processors
  - Security devices
  - Software components
  - Software devices
  - Sound, video and game controllers
  - Storage controllers
  - System devices

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

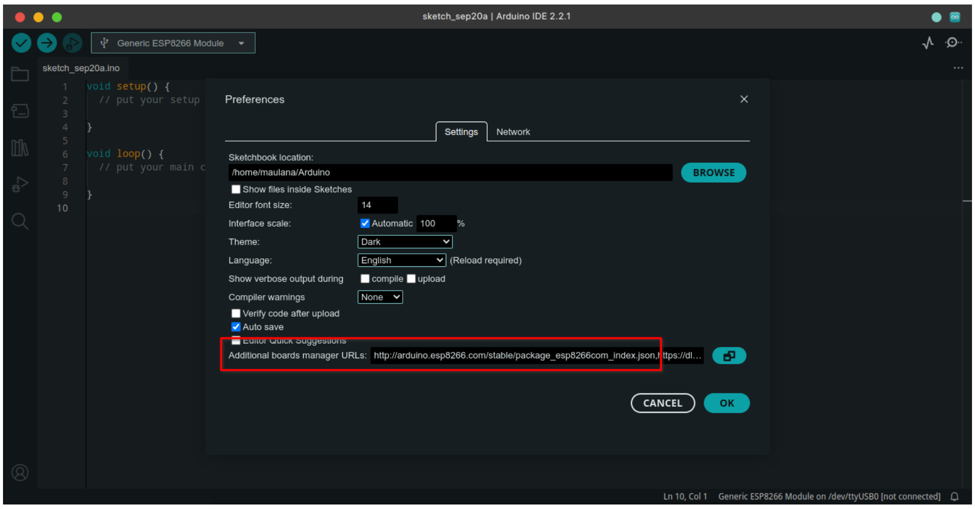


```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}

```

Ln 10, Col 1   Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

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



sketch_sep20a | Arduino IDE 2.2.1

Generic ESP8266 Module

sketch_sep20a.ino

```
void setup() {
  // put your setup

}

void loop() {
  // put your main c

}
```

Preferences

Settings | Network

Sketchbook location: /home/maulana/Arduino BROWSE

□ Show files inside Sketches

Editor font size: 14

Interface scale: ☑ Automatic 100 %

Theme: Dark

Language: English (Reload required)

Show verbose output during □ compile □ upload

Compiler warnings None

□ Verify code after upload

☑ Auto save

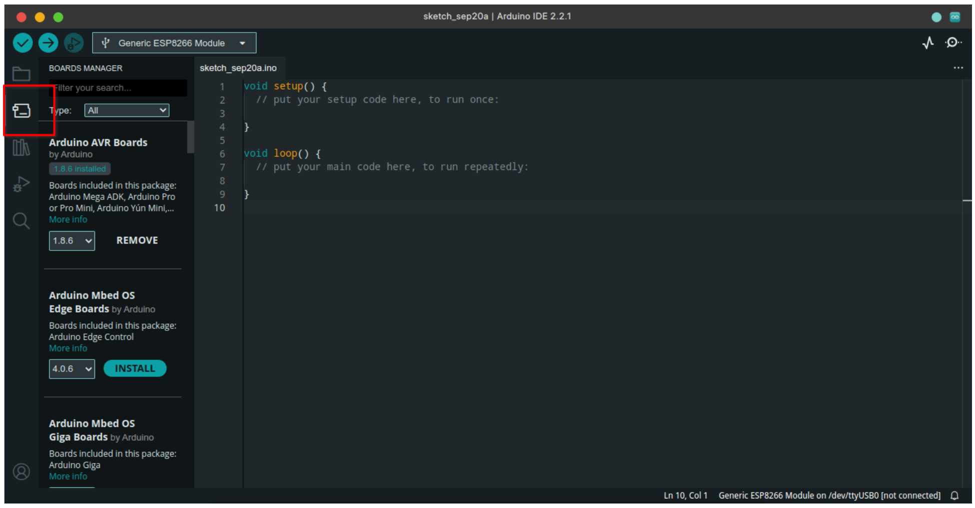
□ Editor Quick Suggestions

Additional boards manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json,https://dl…

CANCEL OK

Ln 10, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

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



sketch_sep20a | Arduino IDE 2.2.1

Generic ESP8266 Module

BOARDS MANAGER

Filter your search...

Type: All

Arduino AVR Boards by Arduino
1.8.6 installed
Boards included in this package: Arduino Mega ADK, Arduino Pro or Pro Mini, Arduino Yún Mini,...
More info
1.8.6 REMOVE

Arduino Mbed OS Edge Boards by Arduino
Boards included in this package: Arduino Edge Control
More info
4.0.6 INSTALL

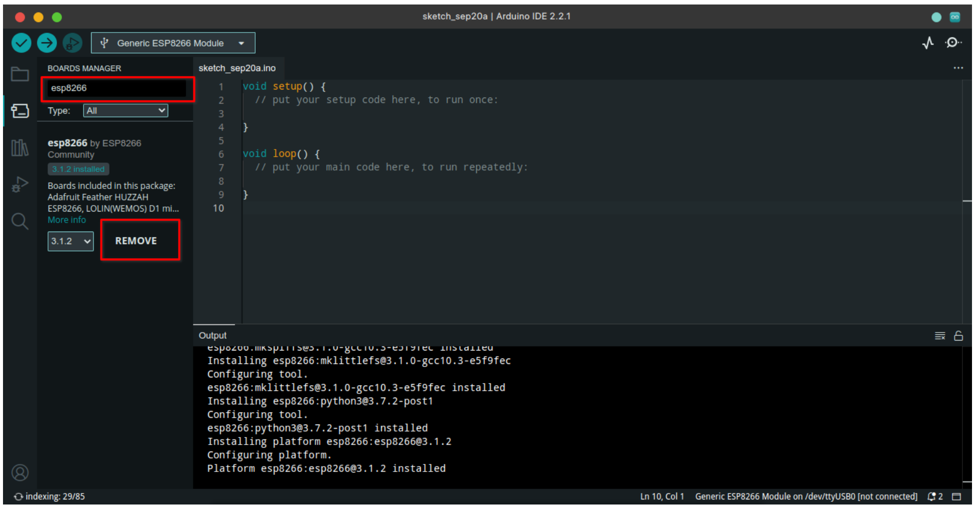
Arduino Mbed OS Giga Boards by Arduino
Boards included in this package: Arduino Giga
More info

sketch_sep20a.ino

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Ln 10, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

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



sketch_sep20a | Arduino IDE 2.2.1

Generic ESP8266 Module

BOARDS MANAGER

esp8266

Type: All

esp8266 by ESP8266 Community
3.1.2 installed
Boards included in this package: Adafruit Feather HUZZAH ESP8266, LOLIN(WEMOS) D1 mi...
More info
3.1.2 REMOVE

sketch_sep20a.ino

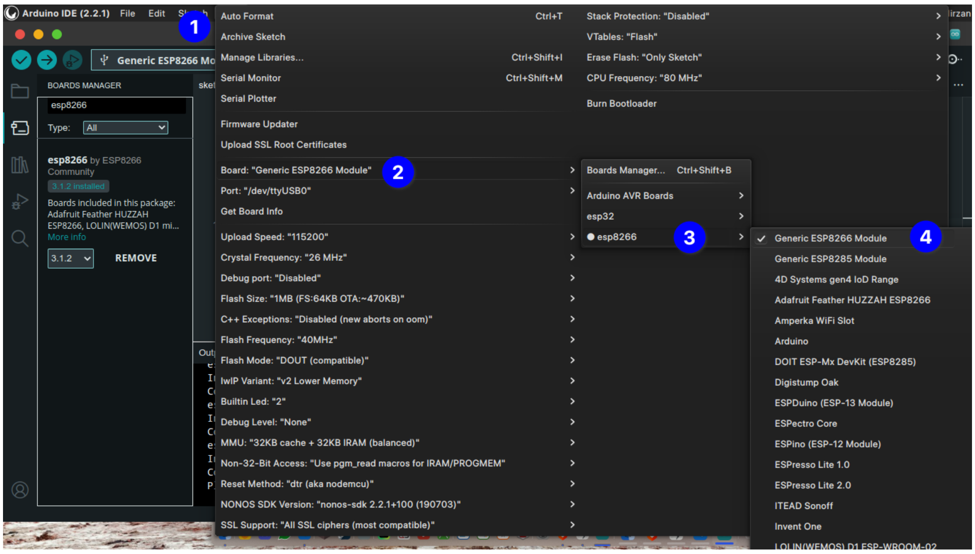
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Output

```
esp8266:mkspiffs@3.1.0-gcc10.3-e5f9fec installed
Installing esp8266:mklittlefs@3.1.0-gcc10.3-e5f9fec
Configuring tool.
esp8266:mklittlefs@3.1.0-gcc10.3-e5f9fec installed
Installing esp8266:python3@3.7.2-post1
Configuring tool.
esp8266:python3@3.7.2-post1 installed
Installing platform esp8266:esp8266@3.1.2
Configuring platform.
Platform esp8266:esp8266@3.1.2 installed
```

indexing: 29/85 Ln 10, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 [not connected] 2

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

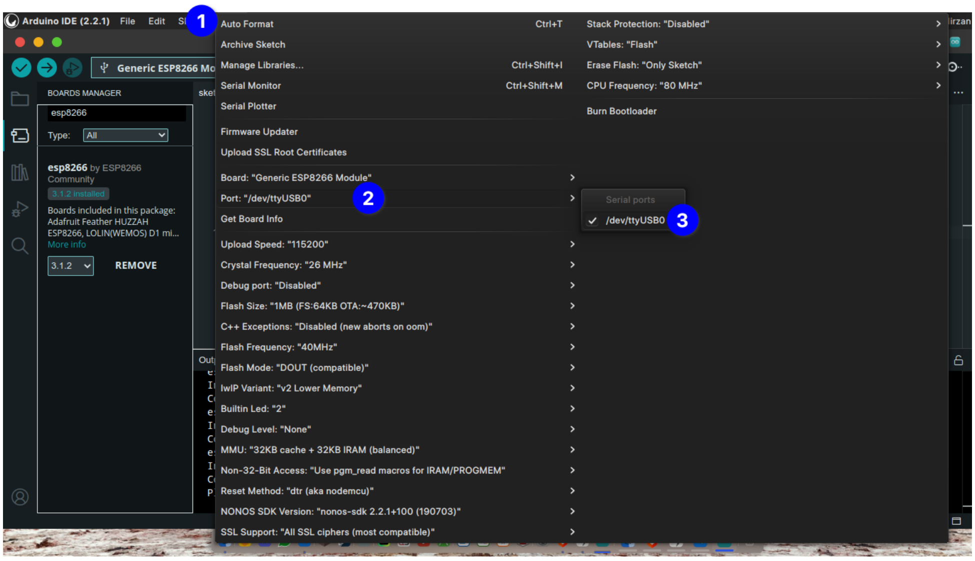


Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Board: "Generic ESP8266 Module"
Port: "/dev/ttyUSB0"
Get Board Info
Upload Speed: "115200"
Crystal Frequency: "26 MHz"
Debug port: "Disabled"
Flash Size: "1MB (FS:64KB OTA:~470KB)"
C++ Exceptions: "Disabled (new aborts on oom)"
Flash Frequency: "40MHz"
Flash Mode: "DOUT (compatible)"
lwIP Variant: "v2 Lower Memory"
Builtin Led: "2"
Debug Level: "None"
MMU: "32KB cache + 32KB IRAM (balanced)"
Non-32-Bit Access: "Use pgm_read macros for IRAM/PROGMEM"
Reset Method: "dtr (aka nodemcu)"
NONOS SDK Version: "nonos-sdk 2.2.1+100 (190703)"
SSL Support: "All SSL ciphers (most compatible)"
Stack Protection: "Disabled"
VTables: "Flash"
Erase Flash: "Only Sketch"
CPU Frequency: "80 MHz"
Burn Bootloader

Boards Manager... Ctrl+Shift+B
Arduino AVR Boards
esp32
● esp8266

✓ Generic ESP8266 Module
Generic ESP8285 Module
4D Systems gen4 IoD Range
Adafruit Feather HUZZAH ESP8266
Amperka WiFi Slot
Arduino
DOIT ESP-Mx DevKit (ESP8285)
Digistump Oak
ESPDuino (ESP-13 Module)
ESPectro Core
ESPino (ESP-12 Module)
ESPresso Lite 1.0
ESPresso Lite 2.0
ITEAD Sonoff
Invent One
LOLIN(WEMOS) D1 ESP-WROOM-02

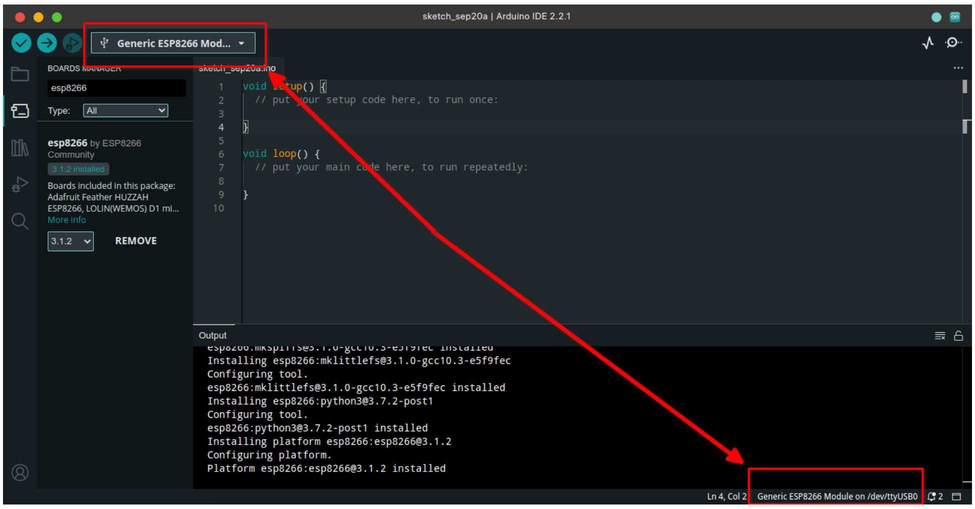
9. Kemudian pastikan **Port Serial** yang digunakan, sama dengan yang ada di **Device Manager**. Cek dengan menu **Tools** → **Port:** → **Pilih COM Sesuai Device Manager**



Board: "Generic ESP8266 Module"
Port: "/dev/ttyUSB0"

Serial ports
✓ /dev/ttyUSB0

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

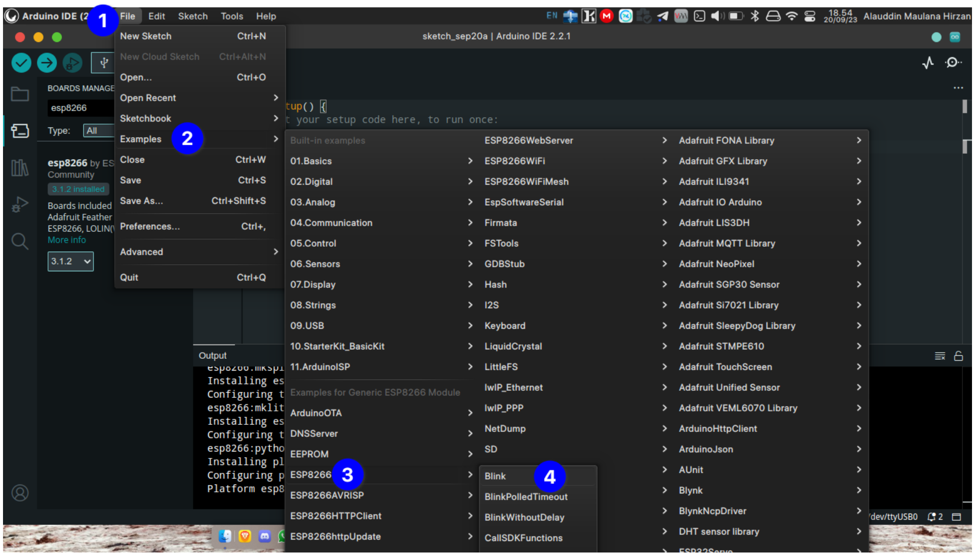


sketch_sep20a | Arduino IDE 2.2.1
Generic ESP8266 Mod...
BOARDS MANAGER
esp8266
Type: All
esp8266 by ESP8266 Community
3.1.2 installed
Boards included in this package: Adafruit Feather HUZZAH ESP8266, LOLIN(WEMOS) D1 mi...
More info
3.1.2
REMOVE
sketch_sep20a.ino
Output
Ln 4, Col 2
Generic ESP8266 Module on /dev/ttyUSB0

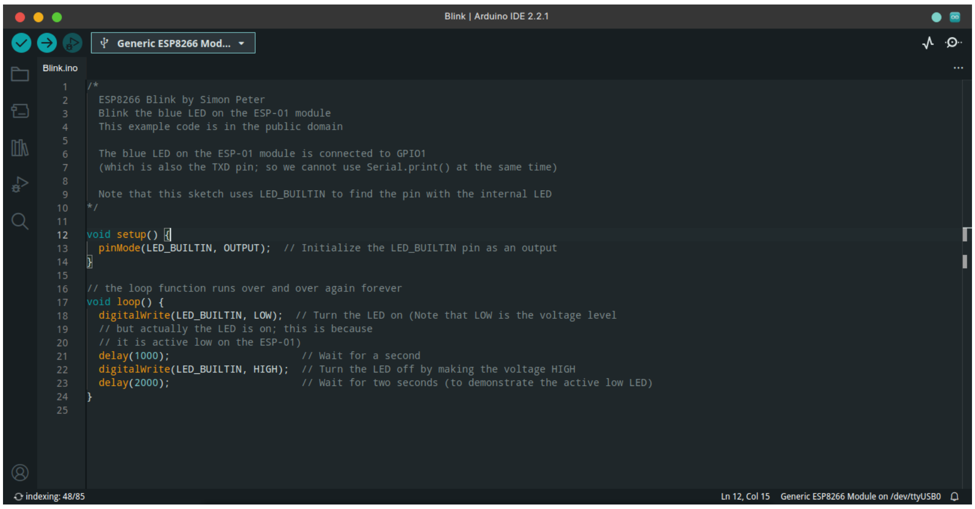
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

```
esp8266:mkspiffs@3.1.0-gcc10.3-e5f9fec installed
Installing esp8266:mklittlefs@3.1.0-gcc10.3-e5f9fec
Configuring tool.
esp8266:mklittlefs@3.1.0-gcc10.3-e5f9fec installed
Installing esp8266:python3@3.7.2-post1
Configuring tool.
esp8266:python3@3.7.2-post1 installed
Installing platform esp8266:esp8266@3.1.2
Configuring platform.
Platform esp8266:esp8266@3.1.2 installed
```

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**

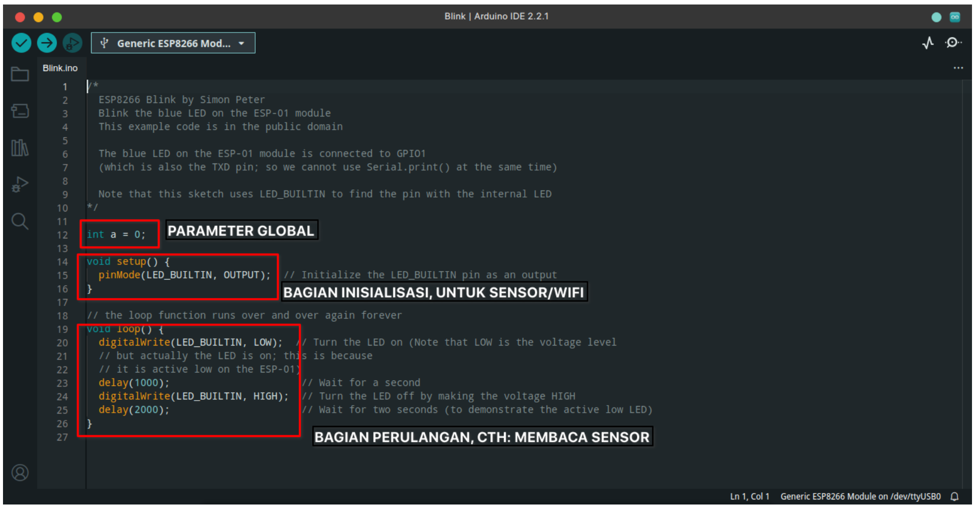


12. **Arduino IDE** akan membuka **Window Baru**. Tutup **Window** sebelumnya agar tidak terganggu.



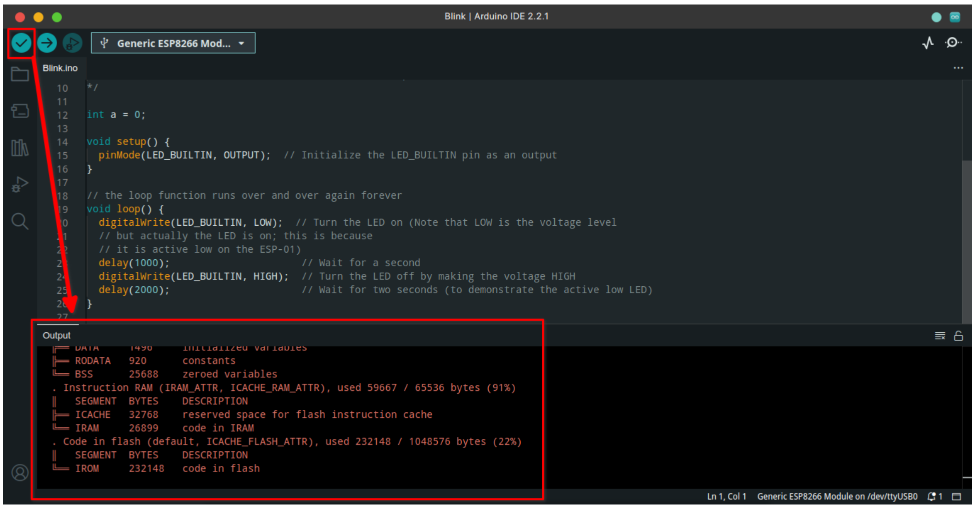
```
/*
  ESP8266 Blink by Simon Peter
  Blink the blue LED on the ESP-01 module
  This example code is in the public domain

  The blue LED on the ESP-01 module is connected to GPIO1
  (which is also the TXD pin; so we cannot use Serial.print() at the same time)

  Note that this sketch uses LED_BUILTIN to find the pin with the internal LED
*/

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);  // Initialize the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, LOW);  // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000);                      // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);                      // Wait for two seconds (to demonstrate the active low LED)
}
```

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**


Blink | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Blink.ino
PARAMETER GLOBAL
BAGIAN INISIALISASI, UNTUK SENSOR/WIFI
BAGIAN PERULANGAN, CTH: MEMBACA SENSOR
Ln 1, Col 1
Generic ESP8266 Module on /dev/ttyUSB0

```
/*
  ESP8266 Blink by Simon Peter
  Blink the blue LED on the ESP-01 module
  This example code is in the public domain

  The blue LED on the ESP-01 module is connected to GPIO1
  (which is also the TXD pin; so we cannot use Serial.print() at the same time)

  Note that this sketch uses LED_BUILTIN to find the pin with the internal LED
*/

int a = 0;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);  // Initialize the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, LOW);  // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000);                      // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);                      // Wait for two seconds (to demonstrate the active low LED)
}
```

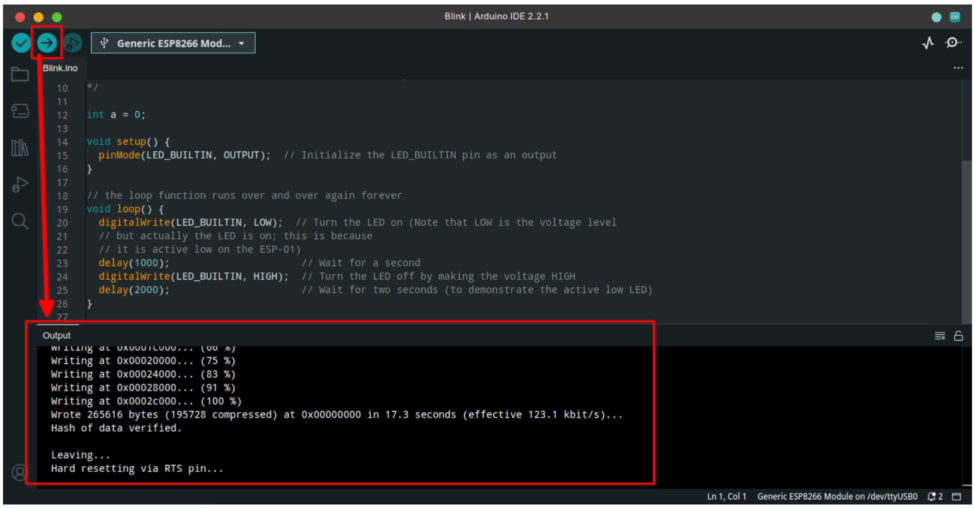
14. Tahap berikutnya adalah verifikasi dan upload kode. Verifikasi memastikan kode sudah benar tanpa typo, sedangkan Upload digunakan mengunggah kode ke alat. Sekarang klik **Verify** untuk memastikan kode sudah benar


Blink | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Blink.ino
Output
Ln 1, Col 1
Generic ESP8266 Module on /dev/ttyUSB0
1

```
*/

int a = 0;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);  // Initialize the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, LOW);  // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000);                      // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);                      // Wait for two seconds (to demonstrate the active low LED)
}
```

```
  ╠══ DATA     1496     initialized variables
  ╠══ RODATA   920      constants
  ╚══ BSS      25688    zeroed variables
. Instruction RAM (IRAM_ATTR, ICACHE_RAM_ATTR), used 59667 / 65536 bytes (91%)
  ║   SEGMENT  BYTES    DESCRIPTION
  ╠══ ICACHE   32768    reserved space for flash instruction cache
  ╚══ IRAM     26899    code in IRAM
. Code in flash (default, ICACHE_FLASH_ATTR), used 232148 / 1048576 bytes (22%)
  ║   SEGMENT  BYTES    DESCRIPTION
  ╚══ IROM     232148   code in flash
```

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


Blink | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Blink.ino
Output
Ln 1, Col 1
Generic ESP8266 Module on /dev/ttyUSB0
2

```
*/

int a = 0;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);  // Initialize the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, LOW);  // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000);                      // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);                      // Wait for two seconds (to demonstrate the active low LED)
}
```

```
Writing at 0x0001c000... (66 %)
Writing at 0x00020000... (75 %)
Writing at 0x00024000... (83 %)
Writing at 0x00028000... (91 %)
Writing at 0x0002c000... (100 %)
Wrote 265616 bytes (195728 compressed) at 0x00000000 in 17.3 seconds (effective 123.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Bab 2

Praktikum 2

2.1 ESP8266, DHT11, dan AdafruitIO

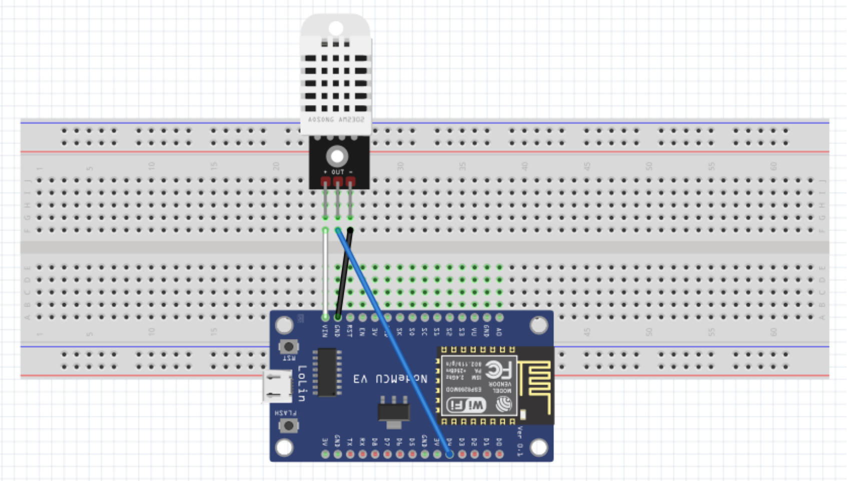
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke sensor DHT11 dan bagaimana menyimpan data secara daring di layanan AdafruitIO. Mahasiswa diwajibkan memahami **Praktikum 1** yang ada di halaman sebelumnya.

2.2 Tutorial

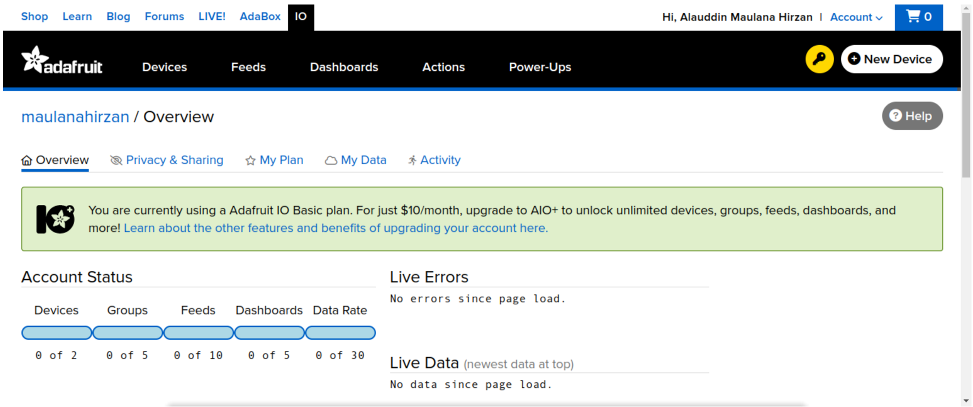
1. Langkah pertama yang perlu dilakukan adalah memasang sensor ke perangkat. Perlu diketahui bahwa dalam memasang sensor harus dalam keadaan **MATI/TIDAK TERTANCAP** untuk menghindari KORSLETING

2. Perhatikan sensor **DHT11**, di bagian kakinya ada tanda **Plus** +, **Minus -**, dan **Out**. Sambungkan sesuai dengan indikator **NodeMCU ESP8266** sebagai berikut:

   - **Plus + → Vin**
   - **Minus - → 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:



You are currently using a Adafruit IO Basic plan. For just $10/month, upgrade to AIO+ to unlock unlimited devices, groups, feeds, dashboards, and more! Learn about the other features and benefits of upgrading your account here.

Account Status

Devices	Groups	Feeds	Dashboards	Data Rate
0 of 2	0 of 5	0 of 10	0 of 5	0 of 30

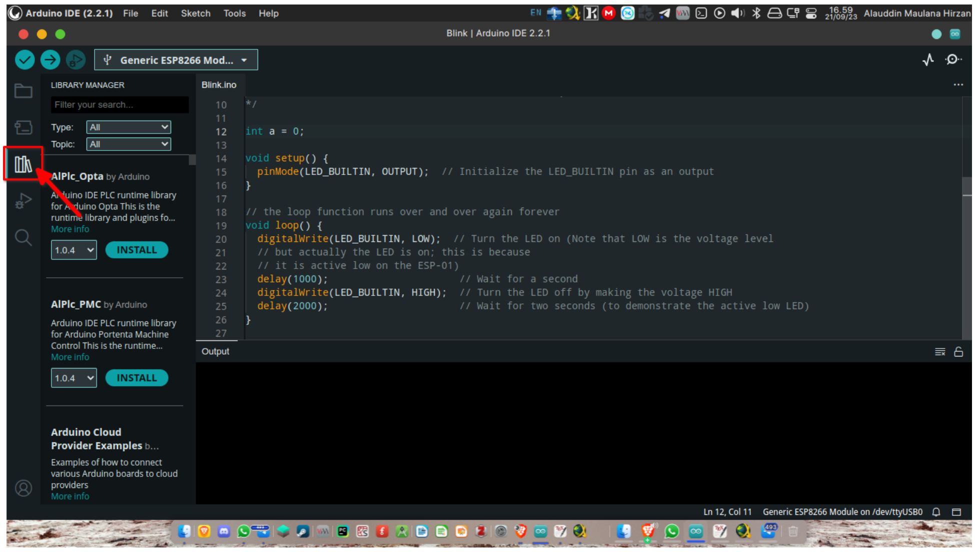
Live Errors

```
No errors since page load.
```

Live Data (newest data at top)

```
No data since page load.
```

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



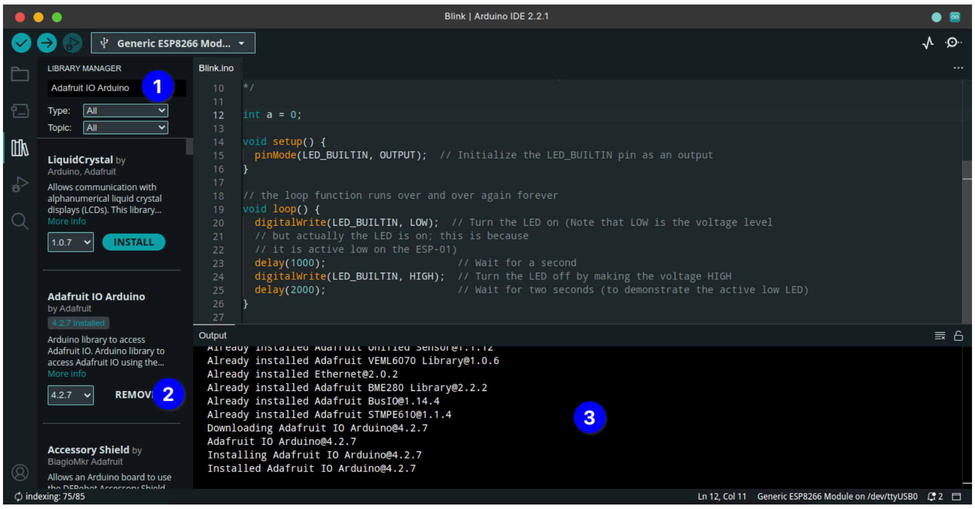
AlPlc_Opta by Arduino
Arduino IDE PLC runtime library for Arduino Opta This is the runtime library and plugins fo...
More info
1.0.4 INSTALL

AlPlc_PMC by Arduino
Arduino IDE PLC runtime library for Arduino Portenta Machine Control This is the runtime...
More info
1.0.4 INSTALL

Arduino Cloud Provider Examples b...
Examples of how to connect various Arduino boards to cloud providers
More info

```
*/

int a = 0;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);  // Initialize the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, LOW);  // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000);                      // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);                      // Wait for two seconds (to demonstrate the active low LED)
}
```

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



LiquidCrystal by Arduino, Adafruit
Allows communication with alphanumerical liquid crystal displays (LCDs). This library...
More info
1.0.7 INSTALL

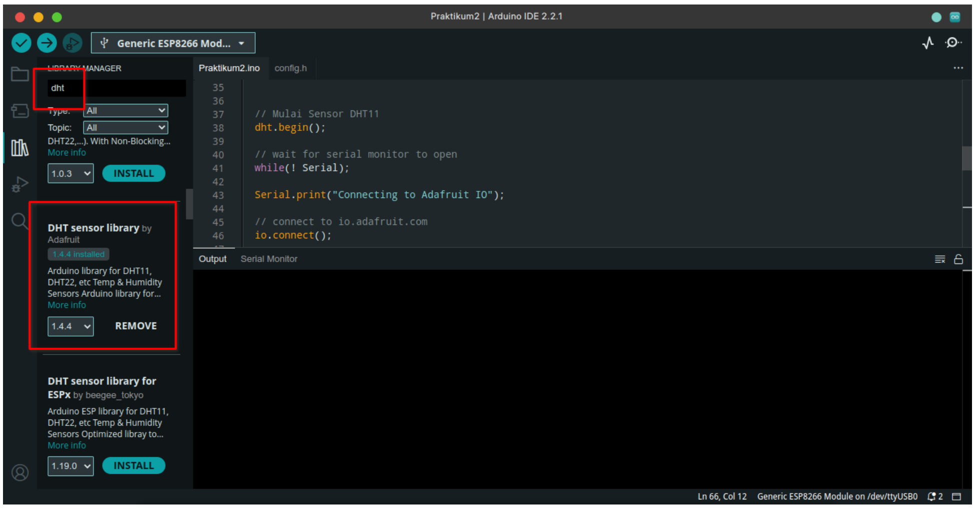
Adafruit IO Arduino by Adafruit
4.2.7 installed
Arduino library to access Adafruit IO. Arduino library to access Adafruit IO using the...
More info
4.2.7 REMOVE

Accessory Shield by BiagioMkr Adafruit
Allows an Arduino board to use the DFRobot Accessory Shield

```
*/

int a = 0;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);  // Initialize the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, LOW);  // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000);                      // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);                      // Wait for two seconds (to demonstrate the active low LED)
}
```

```
Already installed Adafruit Unified Sensor@1.1.12
Already installed Adafruit VEML6070 Library@1.0.6
Already installed Ethernet@2.0.2
Already installed Adafruit BME280 Library@2.2.2
Already installed Adafruit BusIO@1.14.4
Already installed Adafruit STMPE610@1.1.4
Downloading Adafruit IO Arduino@4.2.7
Adafruit IO Arduino@4.2.7
Installing Adafruit IO Arduino@4.2.7
Installed Adafruit IO Arduino@4.2.7
```

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



DHT22,...). With Non-Blocking...

DHT sensor library by Adafruit

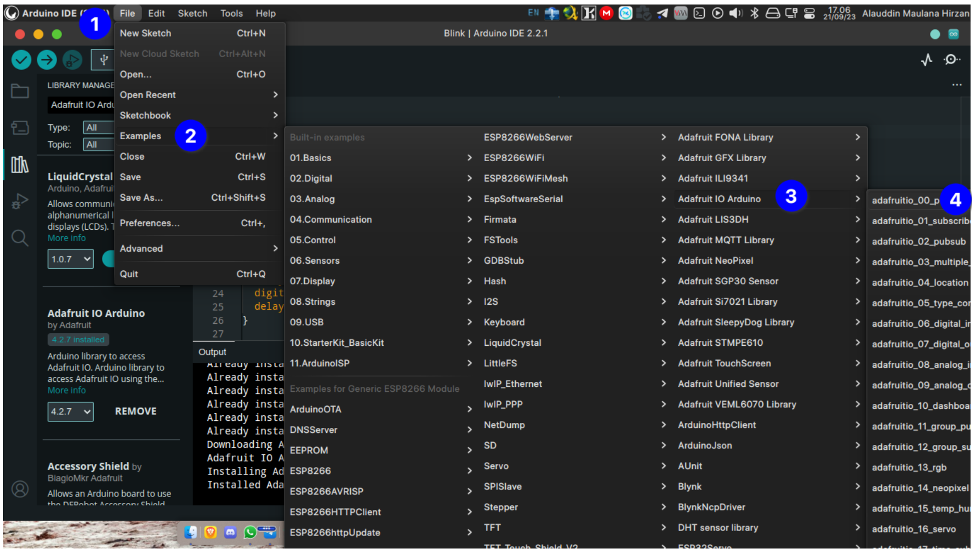
Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for...

DHT sensor library for ESPx by beegee_tokyo

Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to...

```


  // Mulai Sensor DHT11
  dht.begin();

  // wait for serial monitor to open
  while(! Serial);

  Serial.print("Connecting to Adafruit IO");

  // connect to io.adafruit.com
  io.connect();
```

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



Allows communi alphanumerical l displays (LCDs). T

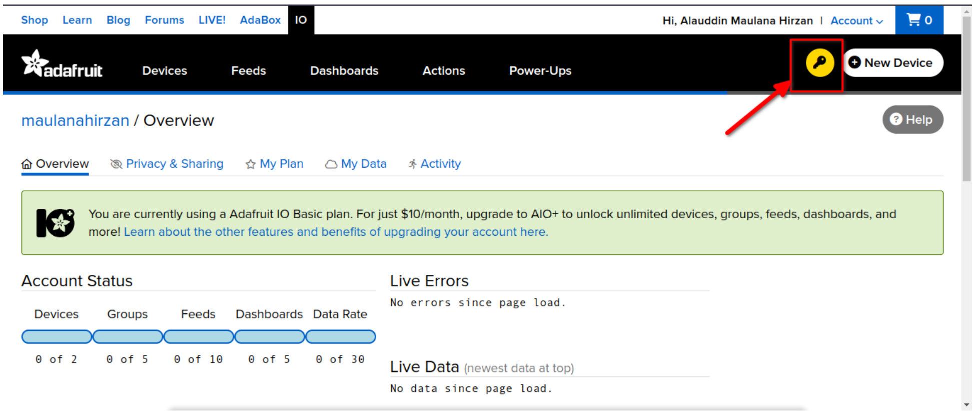
Arduino library to access Adafruit IO. Arduino library to access Adafruit IO using the...

Allows an Arduino board to use

```
digit
delay
}
```

```
Already insta
Already insta
Already insta
Already insta
Already insta
Downloading A
Adafruit IO A
Installing Ad
Installed Ada
```

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



maulanahirzan / Overview

You are currently using a Adafruit IO Basic plan. For just $10/month, upgrade to AIO+ to unlock unlimited devices, groups, feeds, dashboards, and more! Learn about the other features and benefits of upgrading your account here.

Account Status

Devices	Groups	Feeds	Dashboards	Data Rate
0 of 2	0 of 5	0 of 10	0 of 5	0 of 30

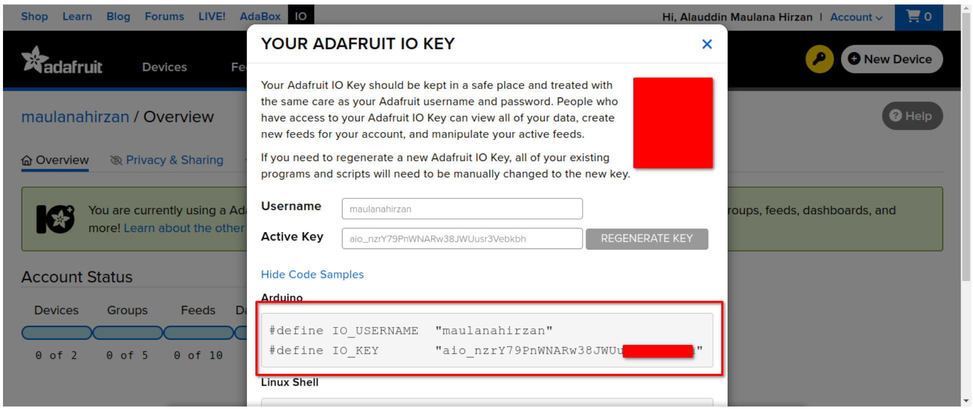
Live Errors

```
No errors since page load.
```

Live Data (newest data at top)

```
No data since page load.
```

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**



YOUR ADAFRUIT IO KEY

Your Adafruit IO Key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your Adafruit IO Key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.

[redacted]

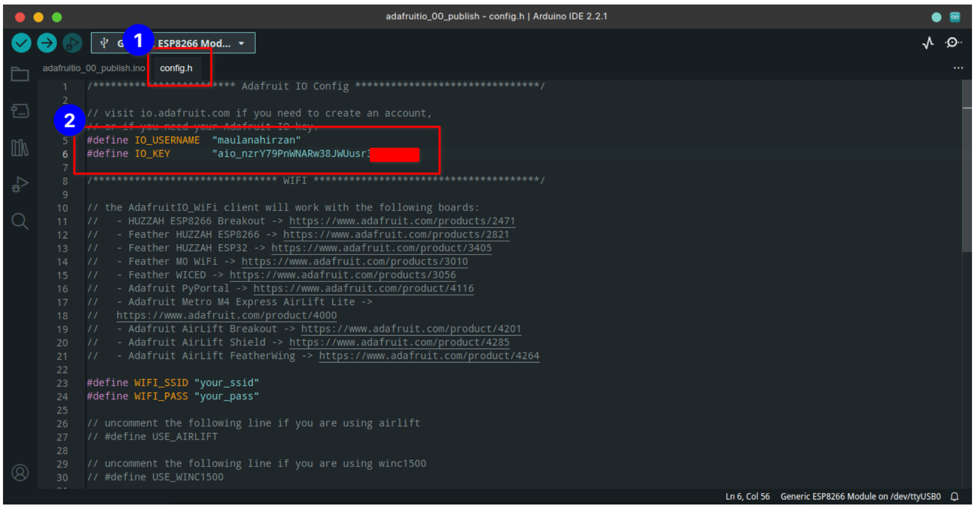
Username: maulanahirzan

Active Key: aio_nzrY79PnWNARw38JWUusr3Vebkbh

Arduino

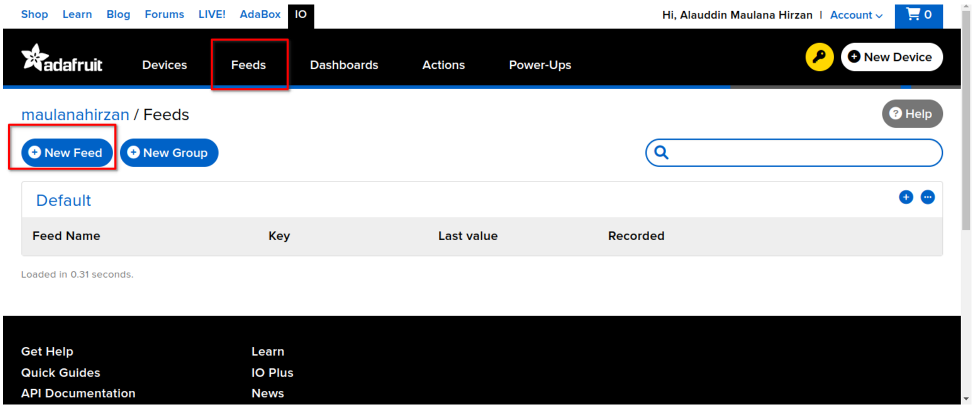
```
#define IO_USERNAME  "maulanahirzan"
#define IO_KEY       "aio_nzrY79PnWNARw38JWUu[redacted]"
```

Linux Shell


adafruitio_00_publish - config.h | Arduino IDE 2.2.1
1
2
ESP8266 Mod...
adafruitio_00_publish.ino
config.h
Ln 6, Col 56
Generic ESP8266 Module on /dev/ttyUSB0

```
/************************** Adafruit IO Config *******************************/

// visit io.adafruit.com if you need to create an account,
// or if you need your Adafruit IO key.
#define IO_USERNAME  "maulanahirzan"
#define IO_KEY       "aio_nzrY79PnWNARw38JWUusr3[redacted]

/******************************* WIFI **************************************/

// the AdafruitIO_WiFi client will work with the following boards:
//   - HUZZAH ESP8266 Breakout -> https://www.adafruit.com/products/2471
//   - Feather HUZZAH ESP8266 -> https://www.adafruit.com/products/2821
//   - Feather HUZZAH ESP32 -> https://www.adafruit.com/product/3405
//   - Feather M0 WiFi -> https://www.adafruit.com/products/3010
//   - Feather WICED -> https://www.adafruit.com/products/3056
//   - Adafruit PyPortal -> https://www.adafruit.com/product/4116
//   - Adafruit Metro M4 Express AirLift Lite ->
//   https://www.adafruit.com/product/4000
//   - Adafruit AirLift Breakout -> https://www.adafruit.com/product/4201
//   - Adafruit AirLift Shield -> https://www.adafruit.com/product/4285
//   - Adafruit AirLift FeatherWing -> https://www.adafruit.com/product/4264

#define WIFI_SSID "your_ssid"
#define WIFI_PASS "your_pass"

// uncomment the following line if you are using airlift
// #define USE_AIRLIFT

// uncomment the following line if you are using winc1500
// #define USE_WINC1500
```

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

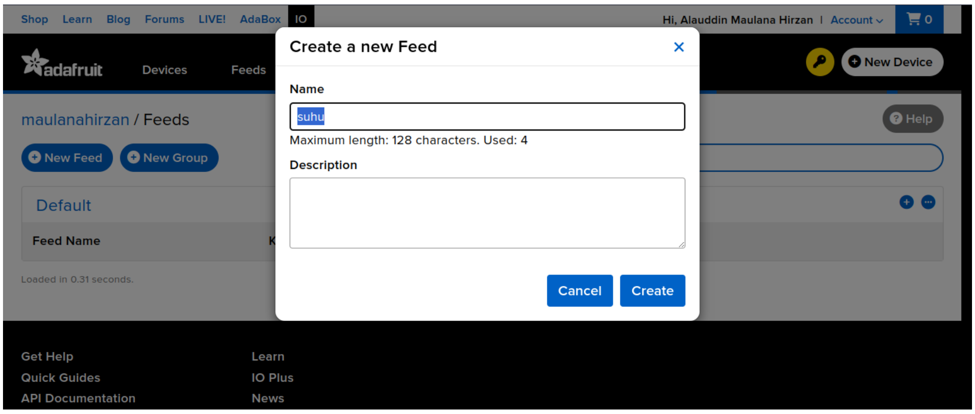


maulanahirzan / Feeds

Default

Feed Name	Key	Last value	Recorded

Loaded in 0.31 seconds.



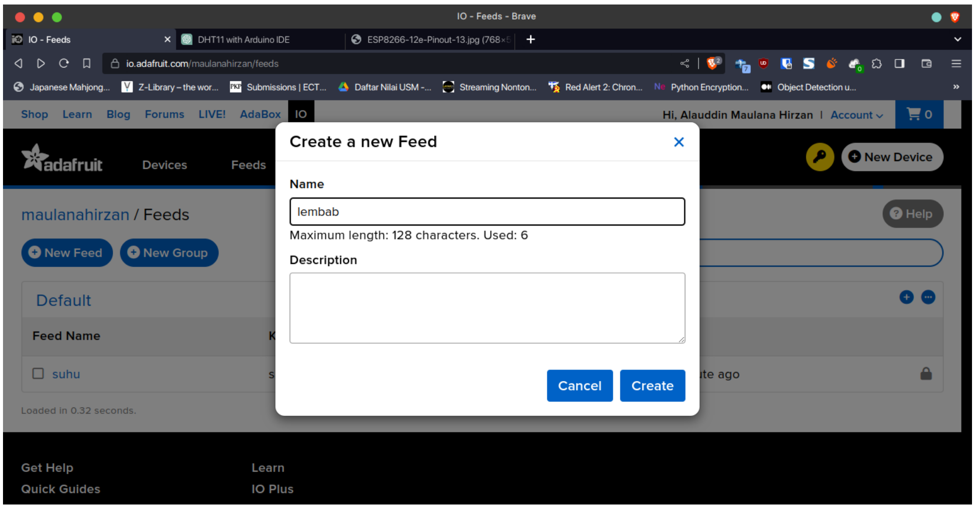
Create a new Feed

Name

suhu

Maximum length: 128 characters. Used: 4

Description



Create a new Feed

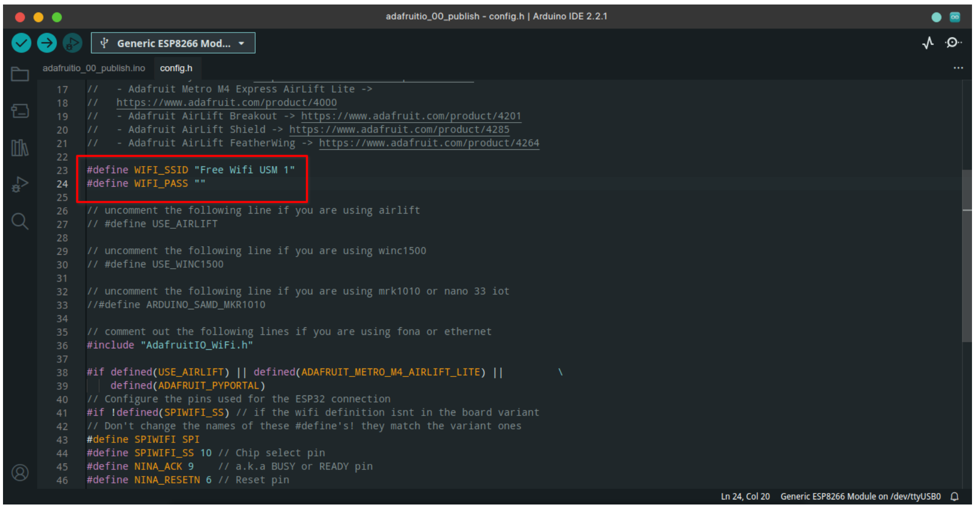
Name

lembab

Maximum length: 128 characters. Used: 6

Description

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

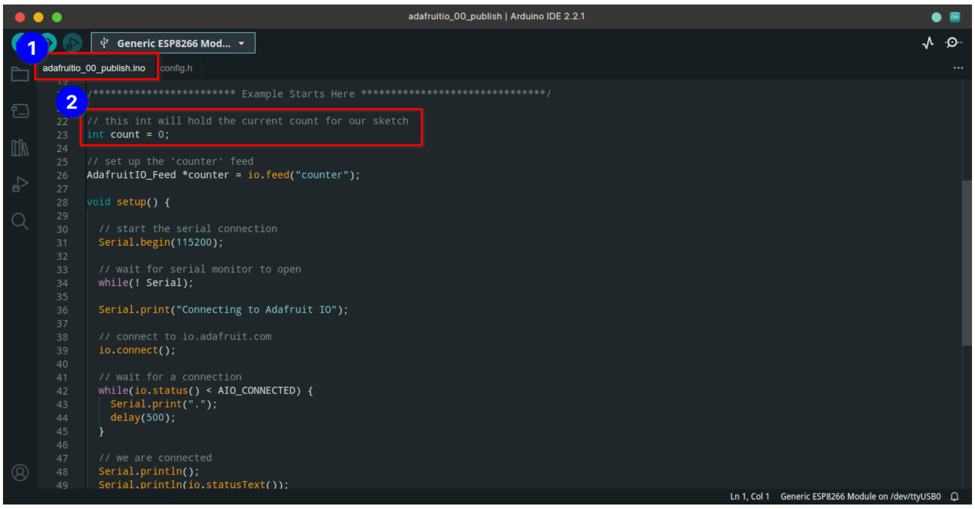


adafruitio_00_publish - config.h | Arduino IDE 2.2.1
Generic ESP8266 Mod...
adafruitio_00_publish.ino
config.h
Ln 24, Col 20 Generic ESP8266 Module on /dev/ttyUSB0

```
//   - Adafruit Metro M4 Express AirLift Lite ->
//   https://www.adafruit.com/product/4000
//   - Adafruit AirLift Breakout -> https://www.adafruit.com/product/4201
//   - Adafruit AirLift Shield -> https://www.adafruit.com/product/4285
//   - Adafruit AirLift FeatherWing -> https://www.adafruit.com/product/4264

#define WIFI_SSID "Free Wifi USM 1"
#define WIFI_PASS ""

// uncomment the following line if you are using airlift
// #define USE_AIRLIFT

// uncomment the following line if you are using winc1500
// #define USE_WINC1500

// uncomment the following line if you are using mrk1010 or nano 33 iot
//#define ARDUINO_SAMD_MKR1010

// comment out the following lines if you are using fona or ethernet
#include "AdafruitIO_WiFi.h"

#if defined(USE_AIRLIFT) || defined(ADAFRUIT_METRO_M4_AIRLIFT_LITE) ||         \
    defined(ADAFRUIT_PYPORTAL)
// Configure the pins used for the ESP32 connection
#if !defined(SPIWIFI_SS) // if the wifi definition isnt in the board variant
// Don't change the names of these #define's! they match the variant ones
#define SPIWIFI SPI
#define SPIWIFI_SS 10 // Chip select pin
#define NINA_ACK 9    // a.k.a BUSY or READY pin
#define NINA_RESETN 6 // Reset pin
```

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

13. Lalu hapus kode yang ditandai



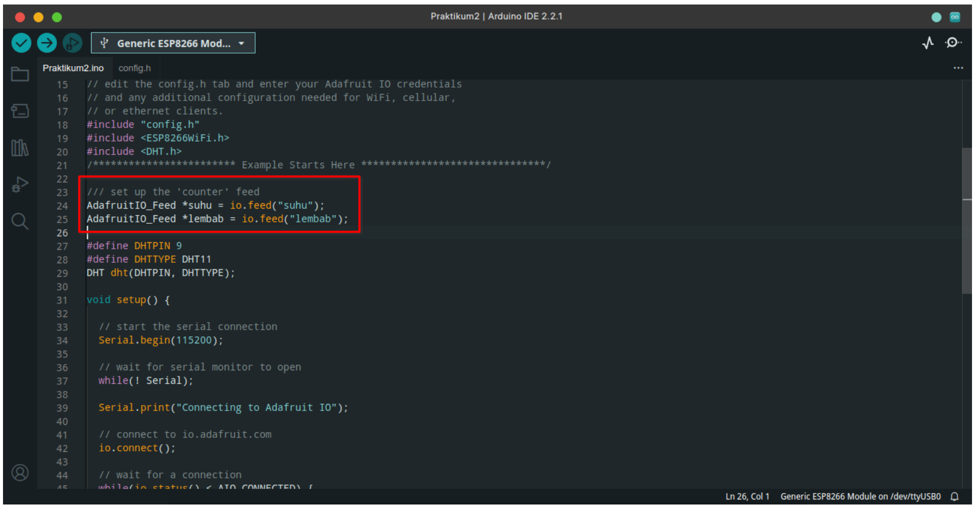
adafruitio_00_publish | Arduino IDE 2.2.1
1
2
Generic ESP8266 Mod...
adafruitio_00_publish.ino
config.h
Ln 1, Col 1 Generic ESP8266 Module on /dev/ttyUSB0

```
/*********************** Example Starts Here *******************************/

// this int will hold the current count for our sketch
int count = 0;

// set up the 'counter' feed
AdafruitIO_Feed *counter = io.feed("counter");

void setup() {

  // start the serial connection
  Serial.begin(115200);

  // wait for serial monitor to open
  while(! Serial);

  Serial.print("Connecting to Adafruit IO");

  // connect to io.adafruit.com
  io.connect();

  // wait for a connection
  while(io.status() < AIO_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  // we are connected
  Serial.println();
  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");
```

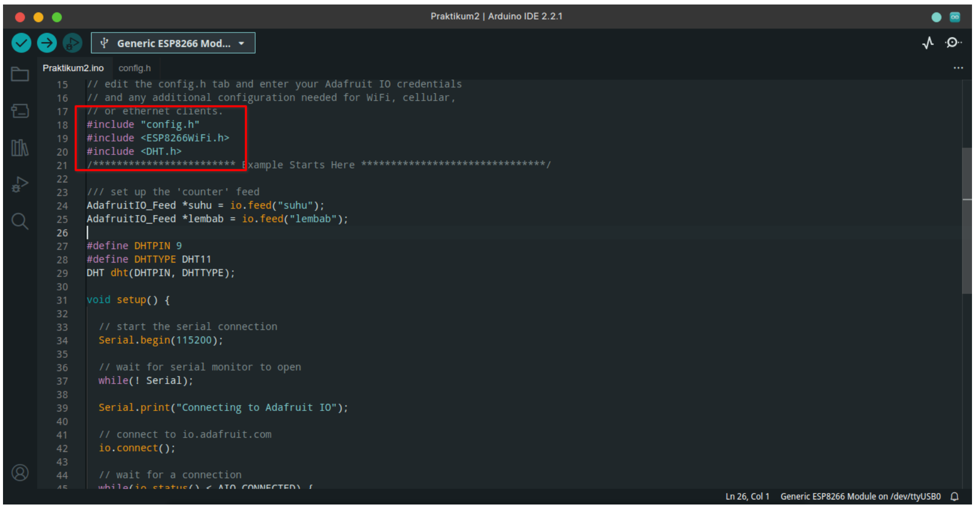

Praktikum2 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum2.ino
config.h
Ln 26, Col 1
Generic ESP8266 Module on /dev/ttyUSB0

```
// edit the config.h tab and enter your Adafruit IO credentials
// and any additional configuration needed for WiFi, cellular,
// or ethernet clients.
#include "config.h"
#include <ESP8266WiFi.h>
#include <DHT.h>
/*********************** Example Starts Here *******************************/

/// set up the 'counter' feed
AdafruitIO_Feed *suhu = io.feed("suhu");
AdafruitIO_Feed *lembab = io.feed("lembab");

#define DHTPIN 9
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {

  // start the serial connection
  Serial.begin(115200);

  // wait for serial monitor to open
  while(! Serial);

  Serial.print("Connecting to Adafruit IO");

  // connect to io.adafruit.com
  io.connect();

  // wait for a connection
  while(io.status() < AIO_CONNECTED) {
```

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

    Potongan Kode

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

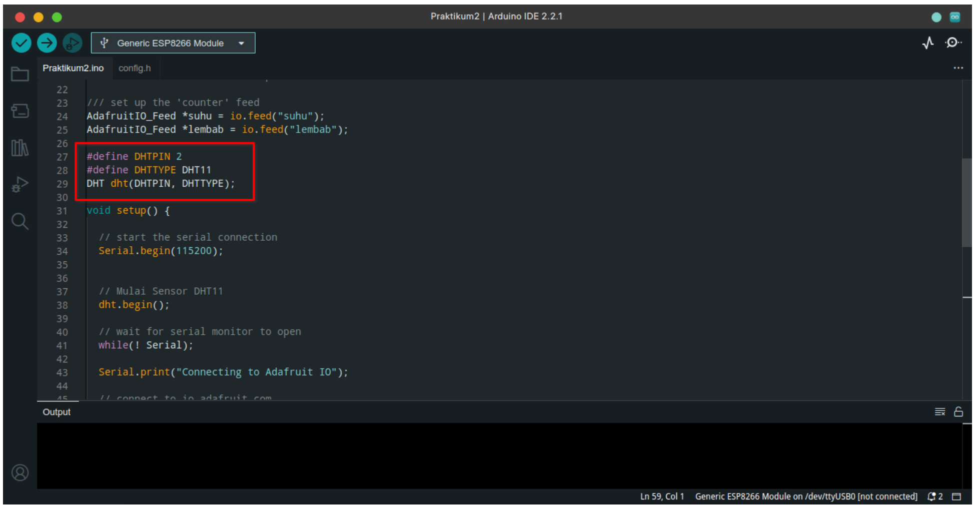

Praktikum2 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum2.ino
config.h
Ln 26, Col 1
Generic ESP8266 Module on /dev/ttyUSB0

```
// edit the config.h tab and enter your Adafruit IO credentials
// and any additional configuration needed for WiFi, cellular,
// or ethernet clients.
#include "config.h"
#include <ESP8266WiFi.h>
#include <DHT.h>
/*********************** Example Starts Here *******************************/

/// set up the 'counter' feed
AdafruitIO_Feed *suhu = io.feed("suhu");
AdafruitIO_Feed *lembab = io.feed("lembab");

#define DHTPIN 9
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {

  // start the serial connection
  Serial.begin(115200);

  // wait for serial monitor to open
  while(! Serial);

  Serial.print("Connecting to Adafruit IO");

  // connect to io.adafruit.com
  io.connect();

  // wait for a connection
  while(io.status() < AIO_CONNECTED) {
```

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

    Potongan Kode

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

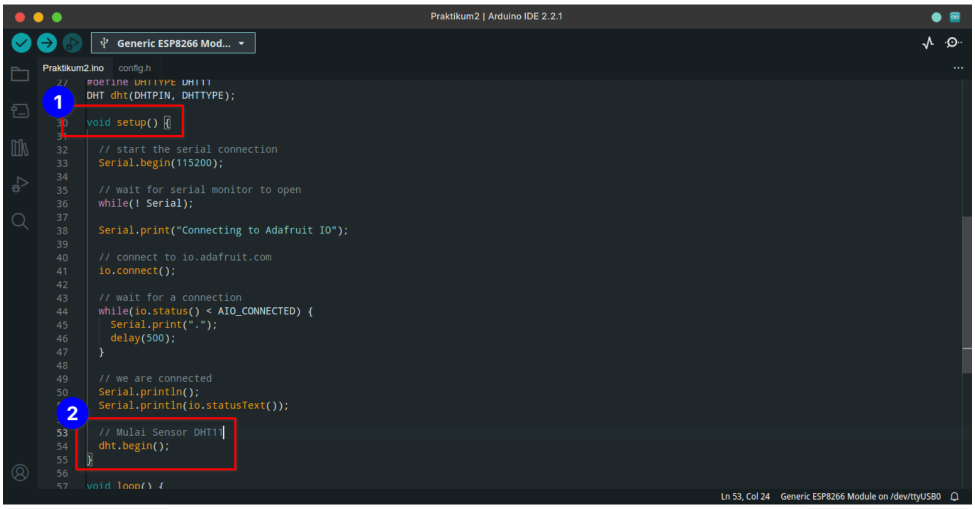

Praktikum2 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum2.ino
config.h
Output
Ln 59, Col 1
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]
2

```
/// set up the 'counter' feed
AdafruitIO_Feed *suhu = io.feed("suhu");
AdafruitIO_Feed *lembab = io.feed("lembab");

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

void setup() {

  // start the serial connection
  Serial.begin(115200);


  // Mulai Sensor DHT11
  dht.begin();

  // wait for serial monitor to open
  while(! Serial);

  Serial.print("Connecting to Adafruit IO");

  // connect to io.adafruit.com
```

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

Potongan Kode

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

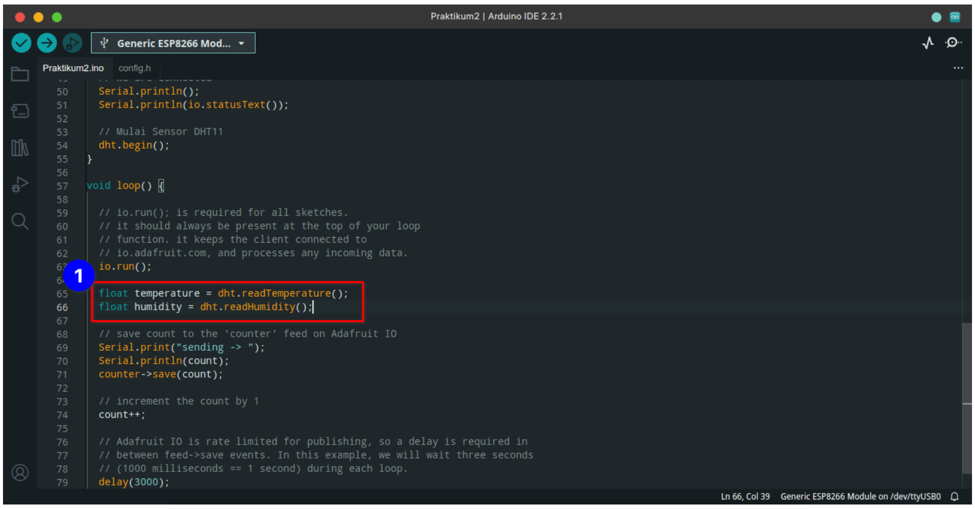

Praktikum2 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum2.ino
config.h
1
2
Ln 53, Col 24
Generic ESP8266 Module on /dev/ttyUSB0

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

void setup() {

  // start the serial connection
  Serial.begin(115200);

  // wait for serial monitor to open
  while(! Serial);

  Serial.print("Connecting to Adafruit IO");

  // connect to io.adafruit.com
  io.connect();

  // wait for a connection
  while(io.status() < AIO_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  // we are connected
  Serial.println();
  Serial.println(io.statusText());

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

void loop() {
```

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

Potongan Kode

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

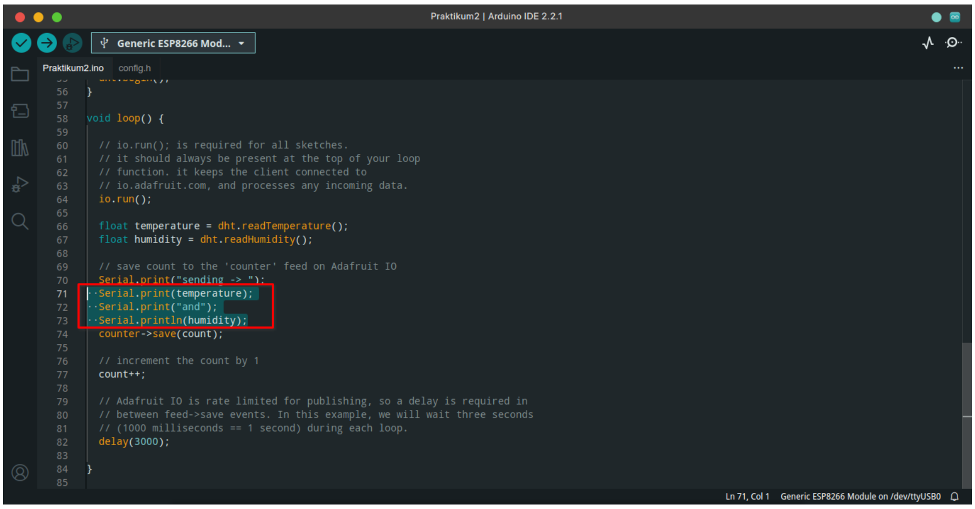

Praktikum2 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum2.ino
config.h
1
Ln 66, Col 39
Generic ESP8266 Module on /dev/ttyUSB0

```
  Serial.println();
  Serial.println(io.statusText());

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

void loop() {

  // io.run(); is required for all sketches.
  // it should always be present at the top of your loop
  // function. it keeps the client connected to
  // io.adafruit.com, and processes any incoming data.
  io.run();

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

  // save count to the 'counter' feed on Adafruit IO
  Serial.print("sending -> ");
  Serial.println(count);
  counter->save(count);

  // increment the count by 1
  count++;

  // Adafruit IO is rate limited for publishing, so a delay is required in
  // between feed->save events. In this example, we will wait three seconds
  // (1000 milliseconds == 1 second) during each loop.
  delay(3000);
```

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

Potongan Kode

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

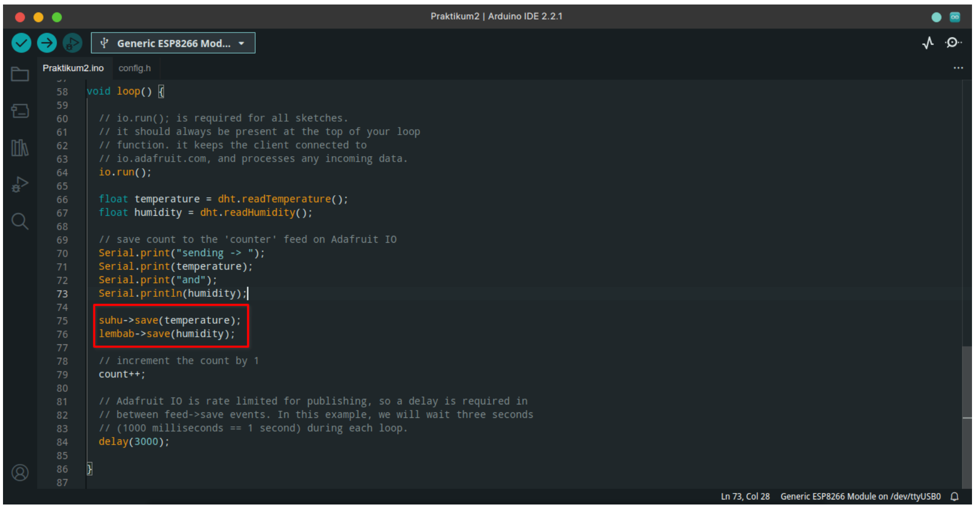

Praktikum2 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum2.ino
config.h
Ln 71, Col 1
Generic ESP8266 Module on /dev/ttyUSB0

```
}

void loop() {

  // io.run(); is required for all sketches.
  // it should always be present at the top of your loop
  // function. it keeps the client connected to
  // io.adafruit.com, and processes any incoming data.
  io.run();

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

  // save count to the 'counter' feed on Adafruit IO
  Serial.print("sending -> ");
  Serial.print(temperature);
  Serial.print("and");
  Serial.println(humidity);
  counter->save(count);

  // increment the count by 1
  count++;

  // Adafruit IO is rate limited for publishing, so a delay is required in
  // between feed->save events. In this example, we will wait three seconds
  // (1000 milliseconds == 1 second) during each loop.
  delay(3000);

}
```

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

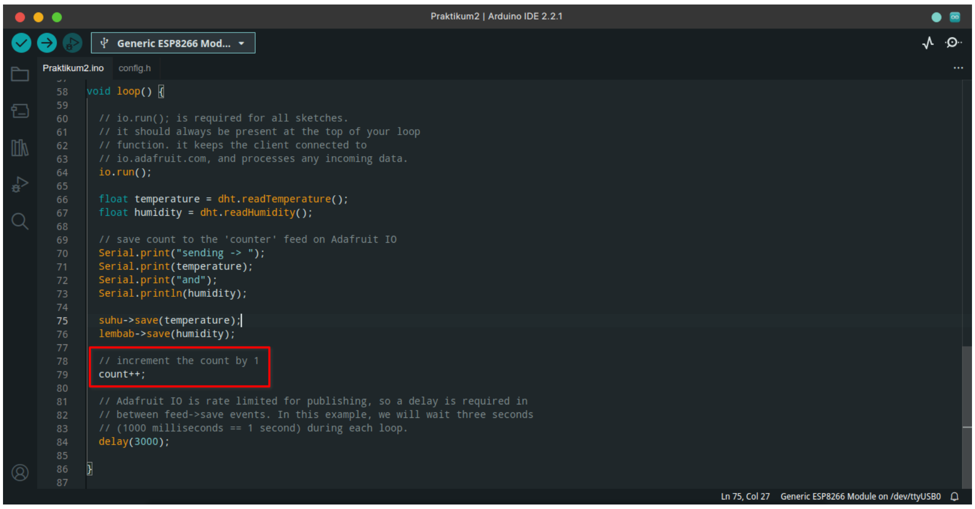
Potongan Kode

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


Praktikum2 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum2.ino
config.h
Ln 73, Col 28 Generic ESP8266 Module on /dev/ttyUSB0

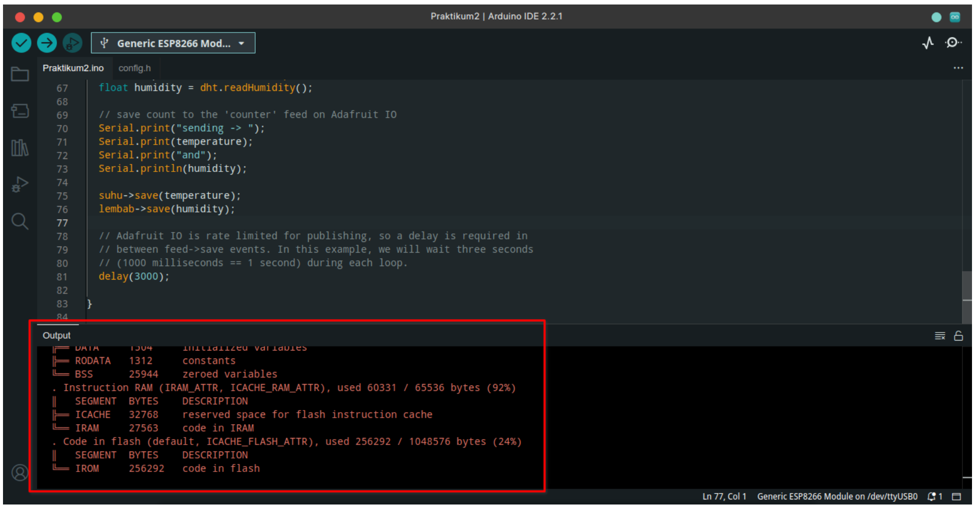
```
void loop() {

  // io.run(); is required for all sketches.
  // it should always be present at the top of your loop
  // function. it keeps the client connected to
  // io.adafruit.com, and processes any incoming data.
  io.run();

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

  // save count to the 'counter' feed on Adafruit IO
  Serial.print("sending -> ");
  Serial.print(temperature);
  Serial.print("and");
  Serial.println(humidity);

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

  // increment the count by 1
  count++;

  // Adafruit IO is rate limited for publishing, so a delay is required in
  // between feed->save events. In this example, we will wait three seconds
  // (1000 milliseconds == 1 second) during each loop.
  delay(3000);

}
```

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


Praktikum2 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum2.ino
config.h
Ln 75, Col 27 Generic ESP8266 Module on /dev/ttyUSB0

```
void loop() {

  // io.run(); is required for all sketches.
  // it should always be present at the top of your loop
  // function. it keeps the client connected to
  // io.adafruit.com, and processes any incoming data.
  io.run();

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

  // save count to the 'counter' feed on Adafruit IO
  Serial.print("sending -> ");
  Serial.print(temperature);
  Serial.print("and");
  Serial.println(humidity);

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

  // increment the count by 1
  count++;

  // Adafruit IO is rate limited for publishing, so a delay is required in
  // between feed->save events. In this example, we will wait three seconds
  // (1000 milliseconds == 1 second) during each loop.
  delay(3000);

}
```

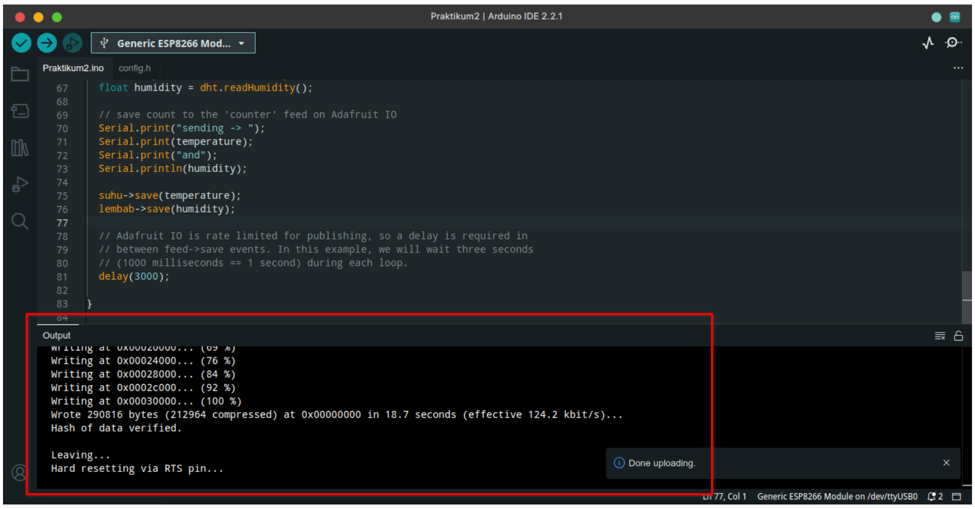
22. Verifikasi kode. Jika tidak ada **Error** seperti digambar. Lanjutkan dengan **Upload**. Pastikan **NodeMCU** tertancap


Praktikum2 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum2.ino
config.h
Output
Ln 77, Col 1 Generic ESP8266 Module on /dev/ttyUSB0

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

  // save count to the 'counter' feed on Adafruit IO
  Serial.print("sending -> ");
  Serial.print(temperature);
  Serial.print("and");
  Serial.println(humidity);

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

  // Adafruit IO is rate limited for publishing, so a delay is required in
  // between feed->save events. In this example, we will wait three seconds
  // (1000 milliseconds == 1 second) during each loop.
  delay(3000);

}
```

```
 ╠══ DATA     1504     initialized variables
 ╠══ RODATA   1312     constants
 ╚══ BSS      25944    zeroed variables
. Instruction RAM (IRAM_ATTR, ICACHE_RAM_ATTR), used 60331 / 65536 bytes (92%)
 ║   SEGMENT  BYTES    DESCRIPTION
 ╠══ ICACHE   32768    reserved space for flash instruction cache
 ╚══ IRAM     27563    code in IRAM
. Code in flash (default, ICACHE_FLASH_ATTR), used 256292 / 1048576 bytes (24%)
 ║   SEGMENT  BYTES    DESCRIPTION
 ╚══ IROM     256292   code in flash
```

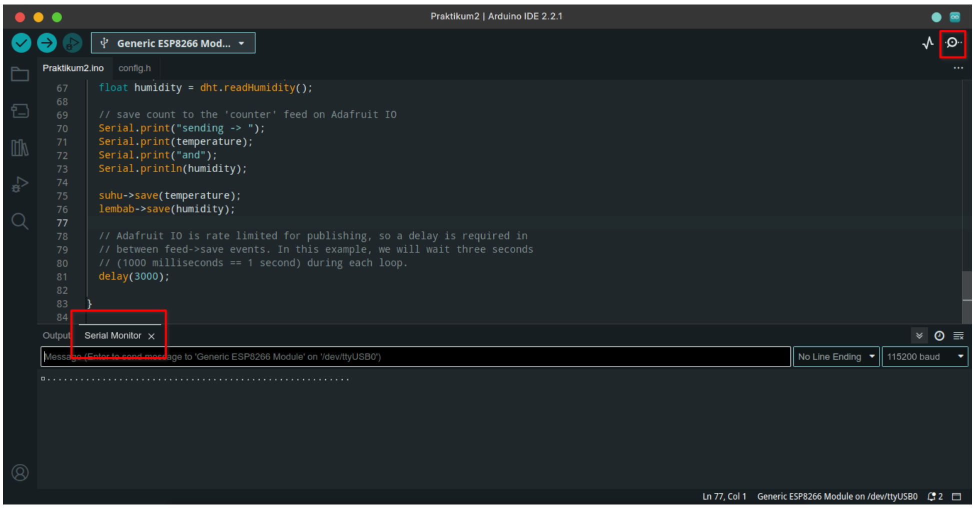
23. Unggah sudah sukses



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

  // save count to the 'counter' feed on Adafruit IO
  Serial.print("sending -> ");
  Serial.print(temperature);
  Serial.print("and");
  Serial.println(humidity);

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

  // Adafruit IO is rate limited for publishing, so a delay is required in
  // between feed->save events. In this example, we will wait three seconds
  // (1000 milliseconds == 1 second) during each loop.
  delay(3000);

}
```

```
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...
```

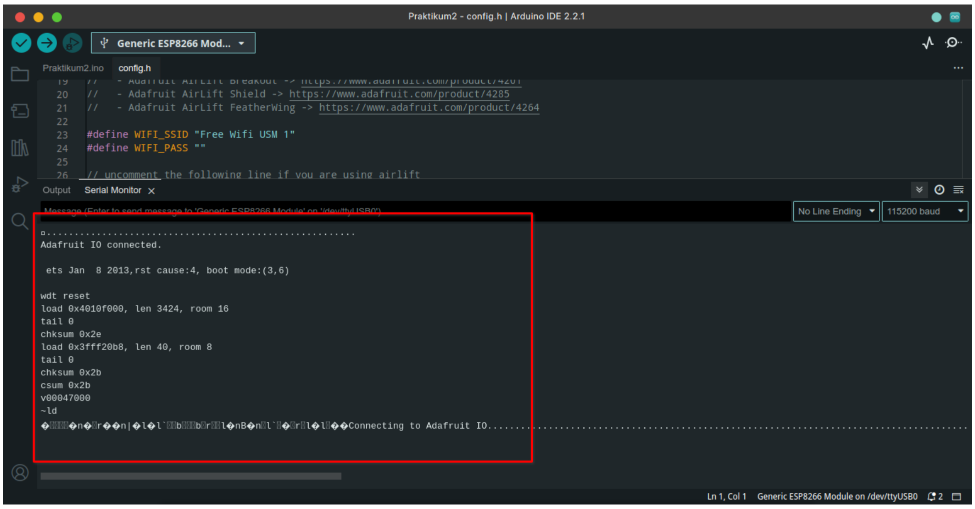
24. Berikutnya adalah mengecek alat. Klik **Tools** → **Serial Monitor**


Praktikum2 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum2.ino
config.h
Output
Serial Monitor
Message (Enter to send message to 'Generic ESP8266 Module' on '/dev/ttyUSB0')
No Line Ending
115200 baud
Ln 77, Col 1
Generic ESP8266 Module on /dev/ttyUSB0

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

  // save count to the 'counter' feed on Adafruit IO
  Serial.print("sending -> ");
  Serial.print(temperature);
  Serial.print("and");
  Serial.println(humidity);

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

  // Adafruit IO is rate limited for publishing, so a delay is required in
  // between feed->save events. In this example, we will wait three seconds
  // (1000 milliseconds == 1 second) during each loop.
  delay(3000);

}
```

```
¤.......................................................
```

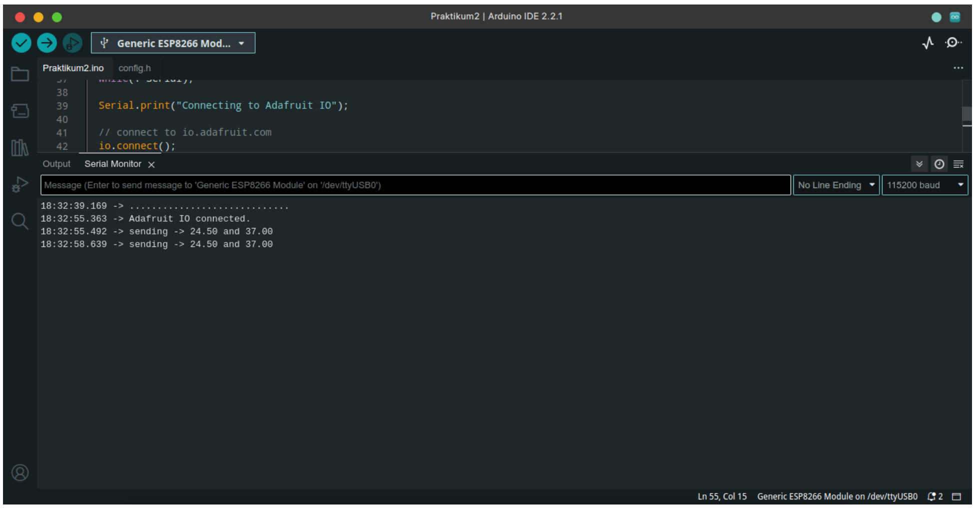
25. Jika proses koneksi lama, cek WiFi SSID apakah sudah benar atau lemot


Praktikum2 - config.h | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum2.ino
config.h
Output
Serial Monitor
Message (Enter to send message to 'Generic ESP8266 Module' on '/dev/ttyUSB0')
No Line Ending
115200 baud
Ln 1, Col 1
Generic ESP8266 Module on /dev/ttyUSB0

```
//   - Adafruit AirLift Breakout -> https://www.adafruit.com/product/4201
//   - Adafruit AirLift Shield -> https://www.adafruit.com/product/4285
//   - Adafruit AirLift FeatherWing -> https://www.adafruit.com/product/4264

#define WIFI_SSID "Free Wifi USM 1"
#define WIFI_PASS ""

// uncomment the following line if you are using airlift
```

```
¤........................................................
Adafruit IO connected.

 ets Jan  8 2013,rst cause:4, boot mode:(3,6)

wdt reset
load 0x4010f000, len 3424, room 16
tail 0
chksum 0x2e
load 0x3fff20b8, len 40, room 8
tail 0
chksum 0x2b
csum 0x2b
v00047000
~ld
�����n��r��n|�l�l`��b���b�r��l�nB�n�l`��r�l�l���Connecting to Adafruit IO..........................................................................................
```

26. Alat terhubung dan berhasil mengirimkan data



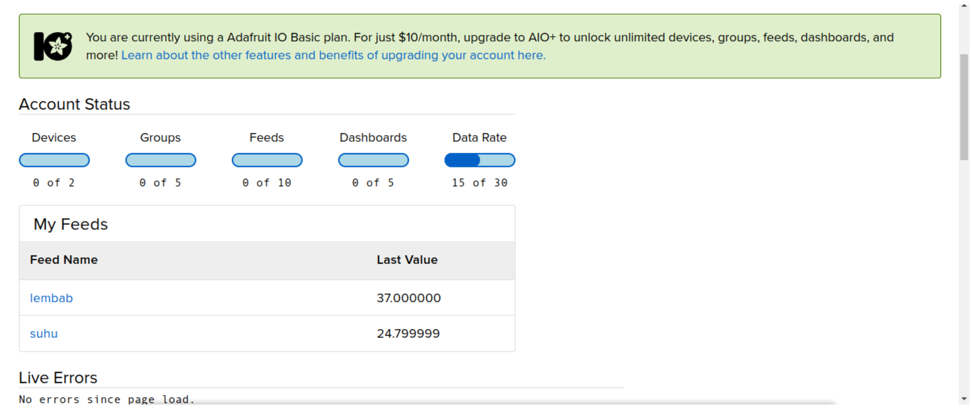
```
Serial.print("Connecting to Adafruit IO");

// connect to io.adafruit.com
io.connect();
```

Message (Enter to send message to 'Generic ESP8266 Module' on '/dev/ttyUSB0')   No Line Ending   115200 baud

```
18:32:39.169 -> ............................
18:32:55.363 -> Adafruit IO connected.
18:32:55.492 -> sending -> 24.50 and 37.00
18:32:58.639 -> sending -> 24.50 and 37.00
```

Ln 55, Col 15   Generic ESP8266 Module on /dev/ttyUSB0

27. Hasil di website **Adafruit IO**



You are currently using a Adafruit IO Basic plan. For just $10/month, upgrade to AIO+ to unlock unlimited devices, groups, feeds, dashboards, and more! Learn about the other features and benefits of upgrading your account here.

Account Status

Devices	Groups	Feeds	Dashboards	Data Rate
0 of 2	0 of 5	0 of 10	0 of 5	15 of 30

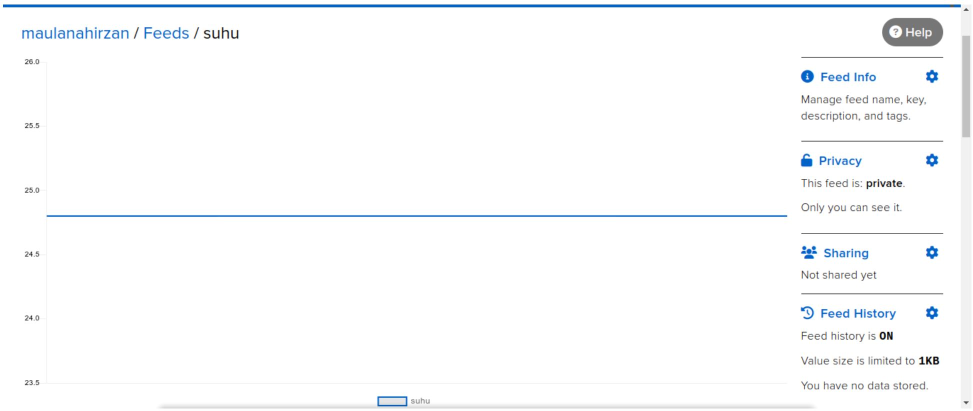
My Feeds

Feed Name	Last Value
lembab	37.000000
suhu	24.799999

Live Errors

No errors since page load.

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



maulanahirzan / Feeds / suhu

Feed Info

Manage feed name, key, description, and tags.

Privacy

This feed is: **private**.

Only you can see it.

Sharing

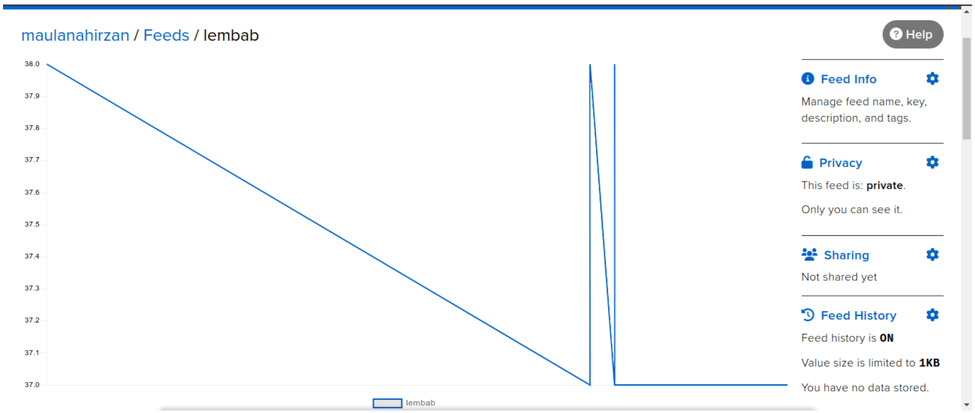
Not shared yet

Feed History

Feed history is **ON**

Value size is limited to **1KB**

You have no data stored.



Feed Info

Manage feed name, key, description, and tags.

Privacy

This feed is: **private**.

Only you can see it.

Sharing

Not shared yet

Feed History

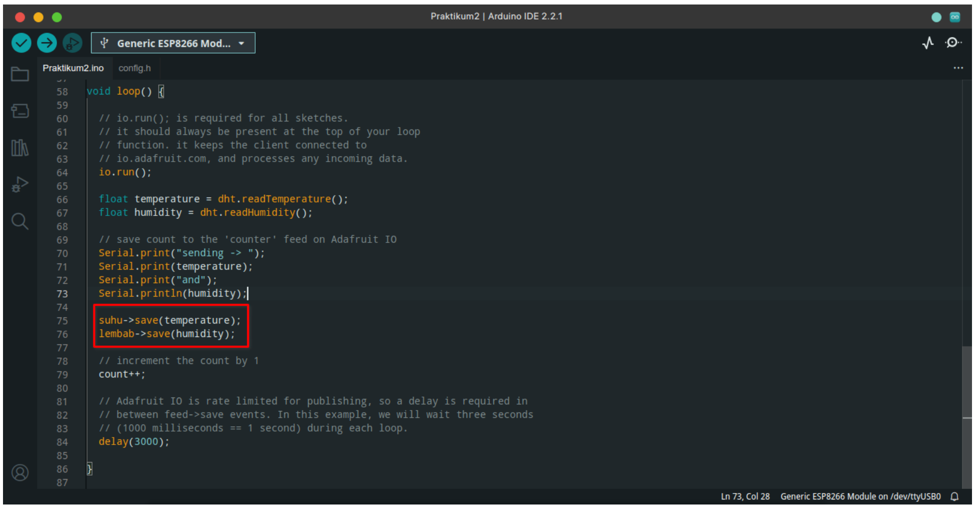
Feed history is **ON**

Value size is limited to **1KB**

You have no data stored.

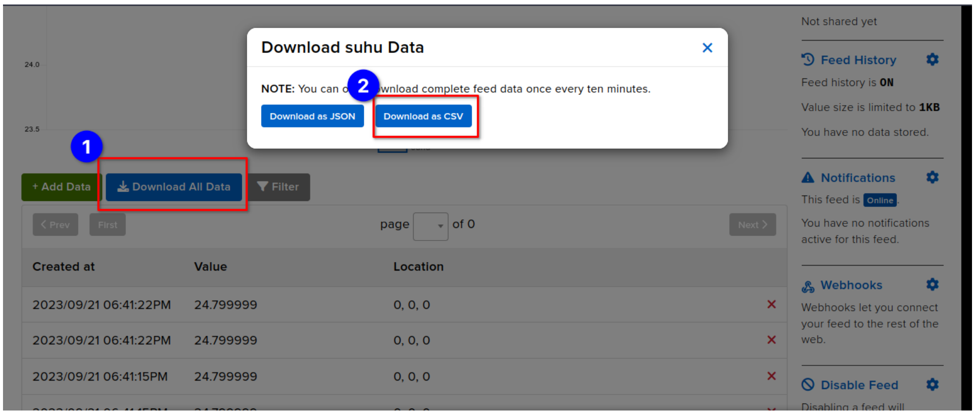
Potongan Kode

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



```
void loop() {

  // io.run(); is required for all sketches.
  // it should always be present at the top of your loop
  // function. it keeps the client connected to
  // io.adafruit.com, and processes any incoming data.
  io.run();

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

  // save count to the 'counter' feed on Adafruit IO
  Serial.print("sending -> ");
  Serial.print(temperature);
  Serial.print("and");
  Serial.println(humidity);

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

  // increment the count by 1
  count++;

  // Adafruit IO is rate limited for publishing, so a delay is required in
  // between feed->save events. In this example, we will wait three seconds
  // (1000 milliseconds == 1 second) during each loop.
  delay(3000);

}
```

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



Download suhu Data

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

Not shared yet

Feed History

Feed history is **ON**

Value size is limited to **1KB**

You have no data stored.

This feed is Online

You have no notifications active for this feed.

Webhooks let you connect your feed to the rest of the web.

Disabling a feed will

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

Bab 3

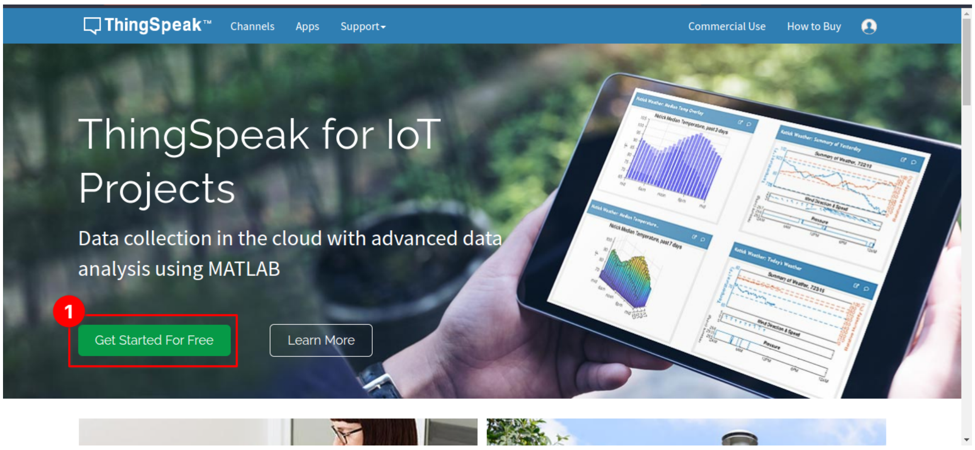
Praktikum 3

3.1 ESP8266, DHT11, dan Thingspeak

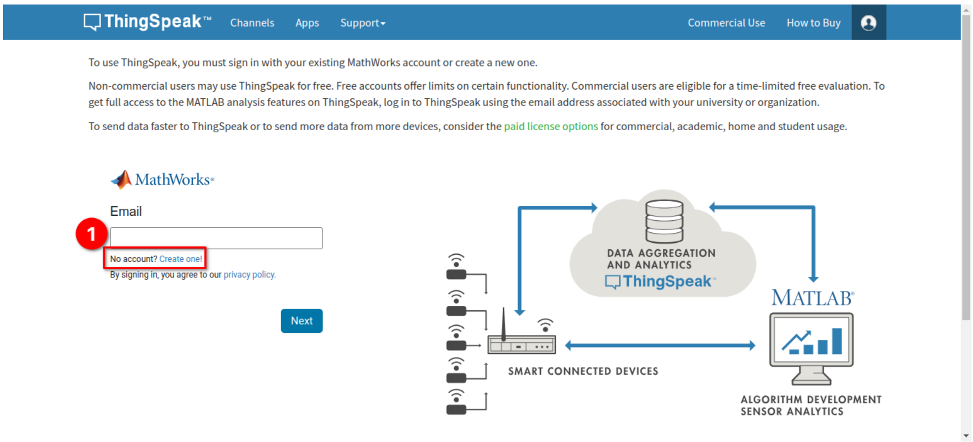
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Thingspeak. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 2** yang ada di halaman sebelumnya.

3.2 Tutorial

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


ThingSpeak™
Channels
Apps
Support▾
Commercial Use
How to Buy
ThingSpeak for IoT Projects
Data collection in the cloud with advanced data analysis using MATLAB
1
Get Started For Free
Learn More

2. Klik **Create one!**



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

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

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

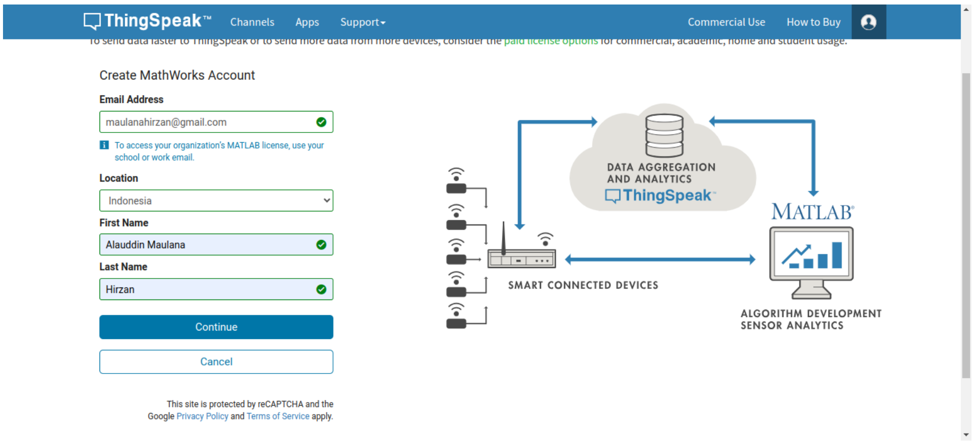
Email

No account? Create one!

By signing in, you agree to our privacy policy.

Next

3. Isi informasi identitas



Create MathWorks Account

Email Address: maulanahirzan@gmail.com

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

Location: Indonesia

First Name: Alauddin Maulana

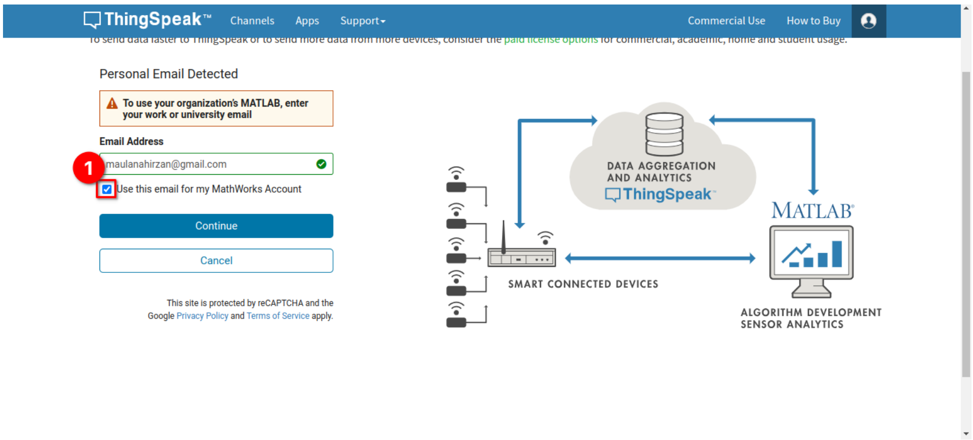
Last Name: Hirzan

Continue

Cancel

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

4. Centang untuk menggunakan email pribadi



Personal Email Detected

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

Email Address: maulanahirzan@gmail.com

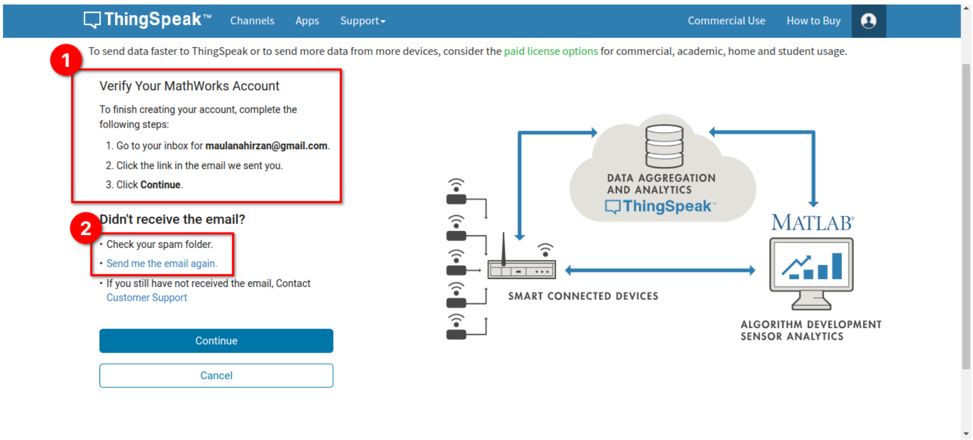
☑ Use this email for my MathWorks Account

Continue

Cancel

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

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



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

Verify Your MathWorks Account

To finish creating your account, complete the following steps:

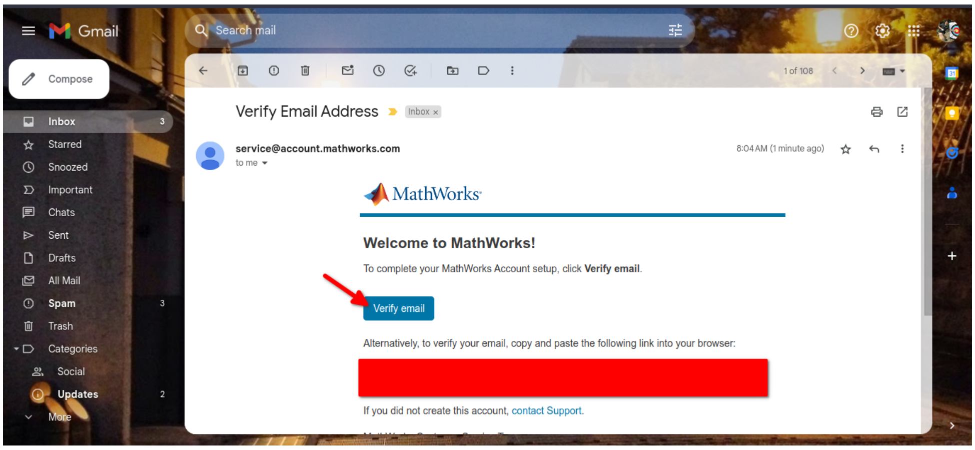
1. Go to your inbox for **maulanahirzan@gmail.com**.
2. Click the link in the email we sent you.
3. Click **Continue**.

Didn't receive the email?

- Check your spam folder.
- Send me the email again.
- If you still have not received the email, Contact Customer Support

Continue

Cancel



Verify Email Address

service@account.mathworks.com
to me

8:04 AM (1 minute ago)

MathWorks

Welcome to MathWorks!

To complete your MathWorks Account setup, click **Verify email**.

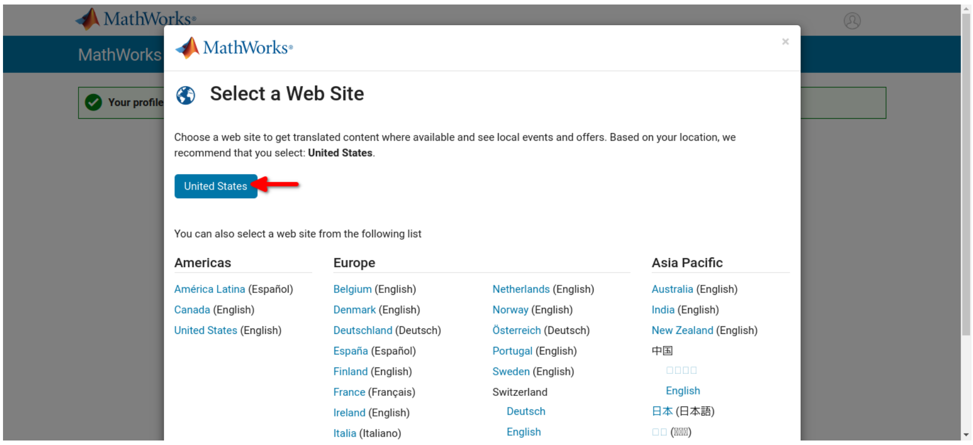
Verify email

Alternatively, to verify your email, copy and paste the following link into your browser:

[redacted]

If you did not create this account, contact Support.

6. Pilih negara untuk website



MathWorks

Select a Web Site

Choose a web site to get translated content where available and see local events and offers. Based on your location, we recommend that you select: **United States**.

United States

You can also select a web site from the following list

Americas

- América Latina (Español)
- Canada (English)
- United States (English)

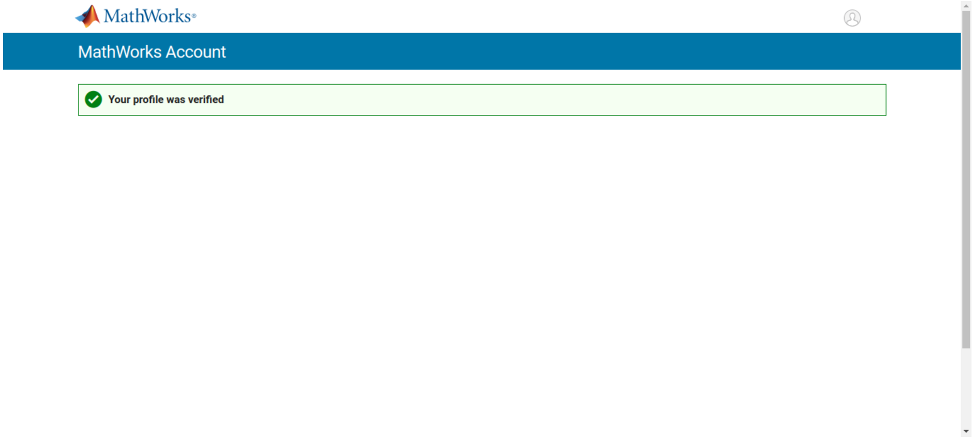
Europe

- Belgium (English)
- Denmark (English)
- Deutschland (Deutsch)
- España (Español)
- Finland (English)
- France (Français)
- Ireland (English)
- Italia (Italiano)
- Netherlands (English)
- Norway (English)
- Österreich (Deutsch)
- Portugal (English)
- Sweden (English)
- Switzerland
  - Deutsch
  - English

Asia Pacific

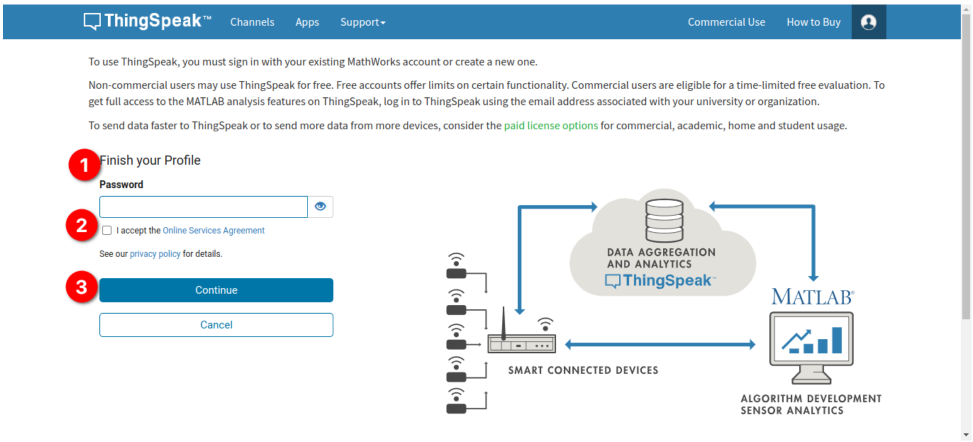
- Australia (English)
- India (English)
- New Zealand (English)
- 中国
  - □□□□
  - English
- 日本 (日本語)
- □□ (한국어)

7. Akun sudah terverifikasi



Your profile was verified

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



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

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

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

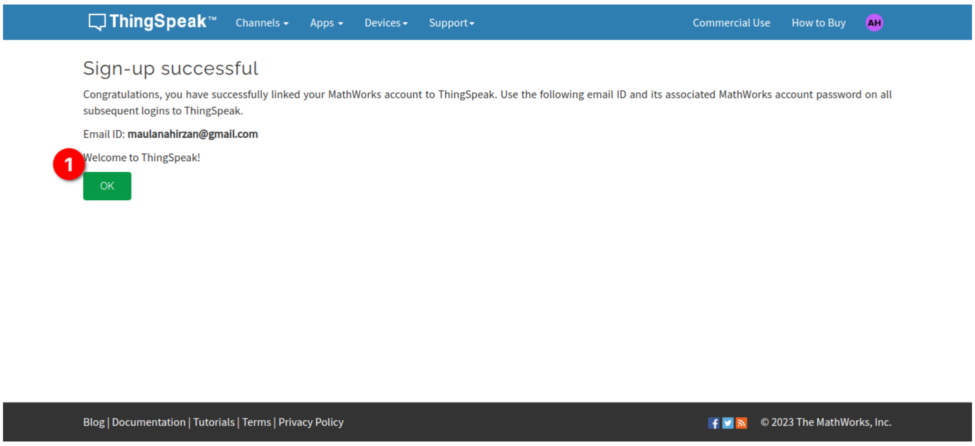
Finish your Profile

Password

□ I accept the Online Services Agreement

See our privacy policy for details.

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


ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
AH
1
OK
Blog | Documentation | Tutorials | Terms | Privacy Policy
© 2023 The MathWorks, Inc.

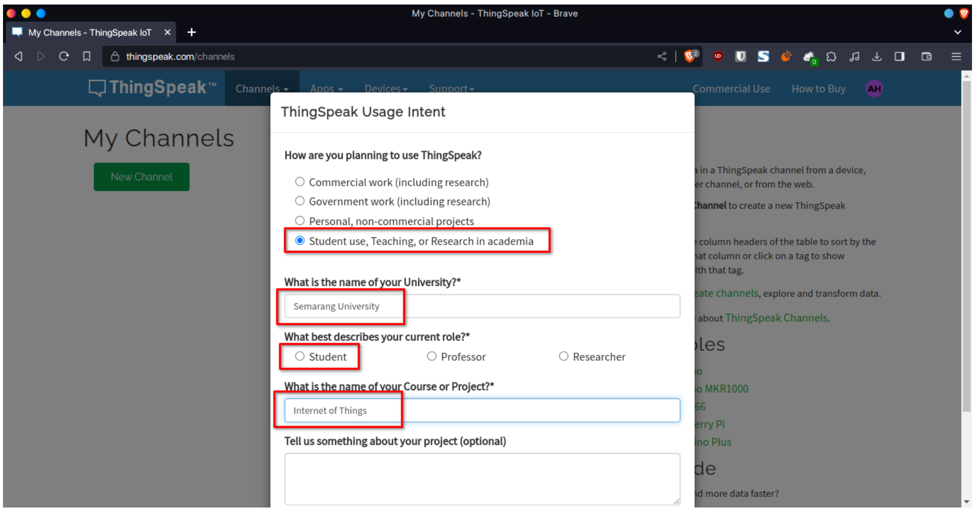
Sign-up successful

Congratulations, you have successfully linked your MathWorks account to ThingSpeak. Use the following email ID and its associated MathWorks account password on all subsequent logins to ThingSpeak.

Email ID: **maulanahirzan@gmail.com**

Welcome to ThingSpeak!

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?*

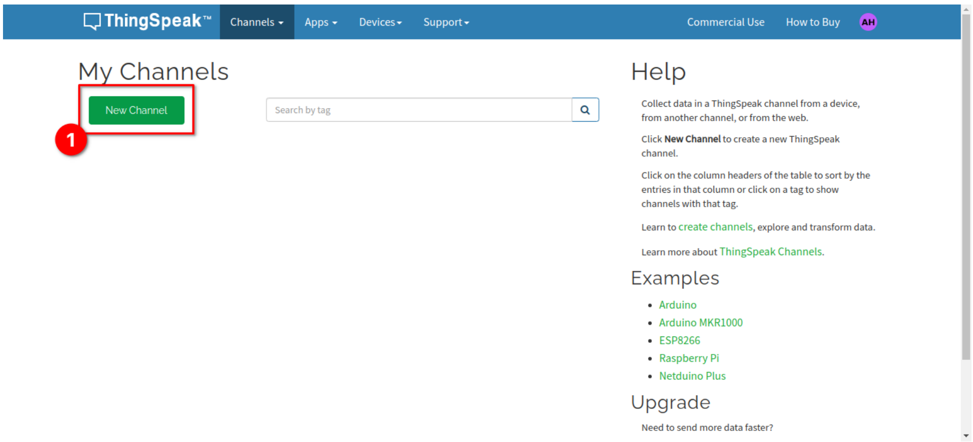
○ Student ○ Professor ○ Researcher

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

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.

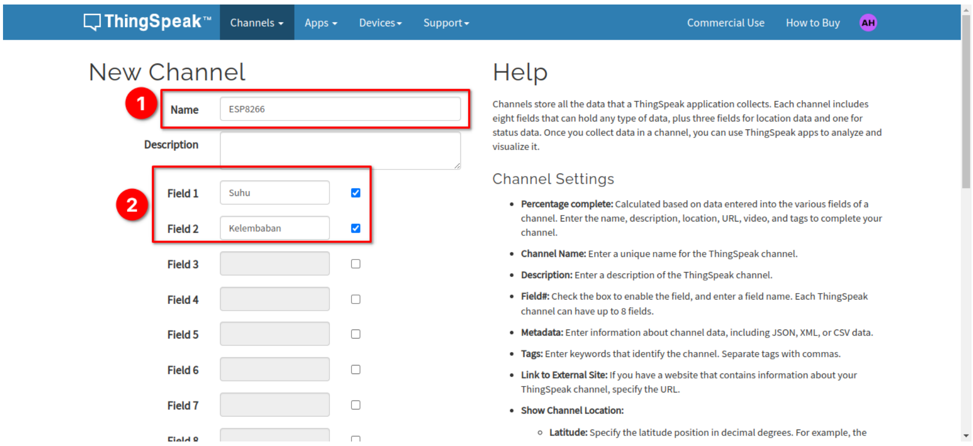
Examples

- 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

Name	ESP8266	
Description		
Field 1	Suhu	☑
Field 2	Kelembaban	☑
Field 3		□
Field 4		□
Field 5		□
Field 6		□
Field 7		□
Field 8		□

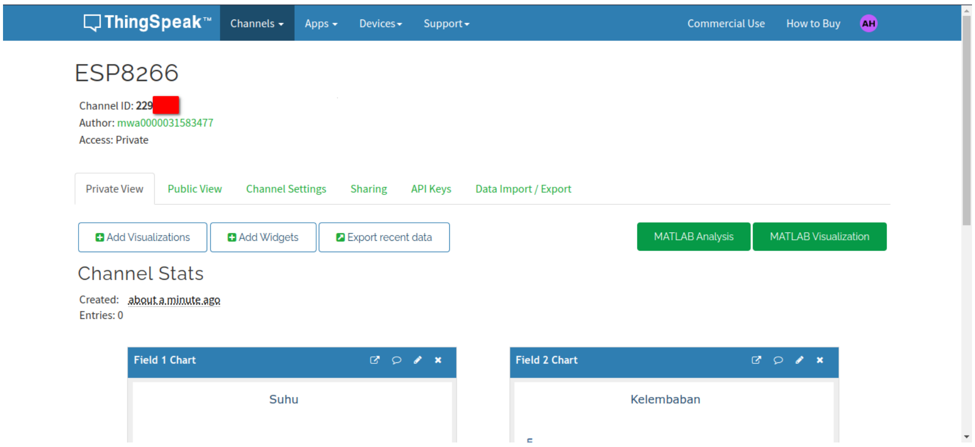
Help

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.
- **Field#:** 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.



ESP8266

Channel ID: 229[redacted]

Author: mwa0000031583477

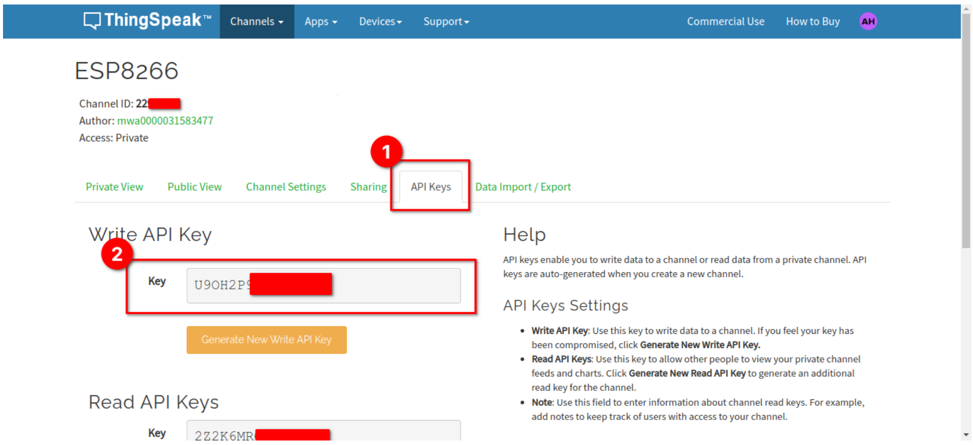
Access: Private

Channel Stats

Created: about a minute ago

Entries: 0

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



ESP8266

Channel ID: 22[redacted]

Author: mwa0000031583477

Access: Private

Write API Key

Key: U9OH2P9[redacted]

Read API Keys

Key: 2Z2K6MR[redacted]

Help

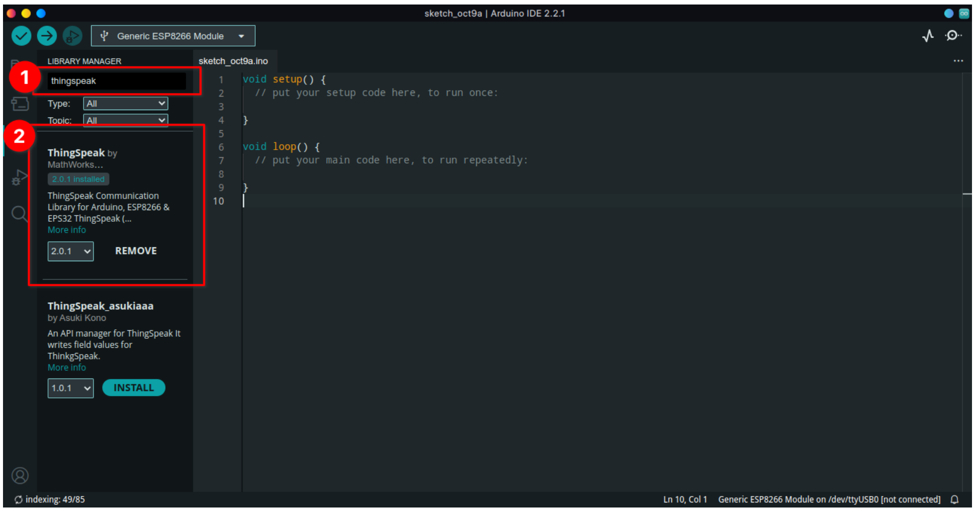
API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key**: Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**
- **Read API Keys**: Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note**: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

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

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



Search: thingspeak

Type: All

Topic: All

ThingSpeak by MathWorks...

2.0.1 installed

ThingSpeak Communication Library for Arduino, ESP8266 & EPS32 ThingSpeak (...

More info

2.0.1 REMOVE

ThingSpeak_asukiaaa

by Asuki Kono

An API manager for ThingSpeak It writes field values for ThinkgSpeak.

More info

1.0.1 INSTALL

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

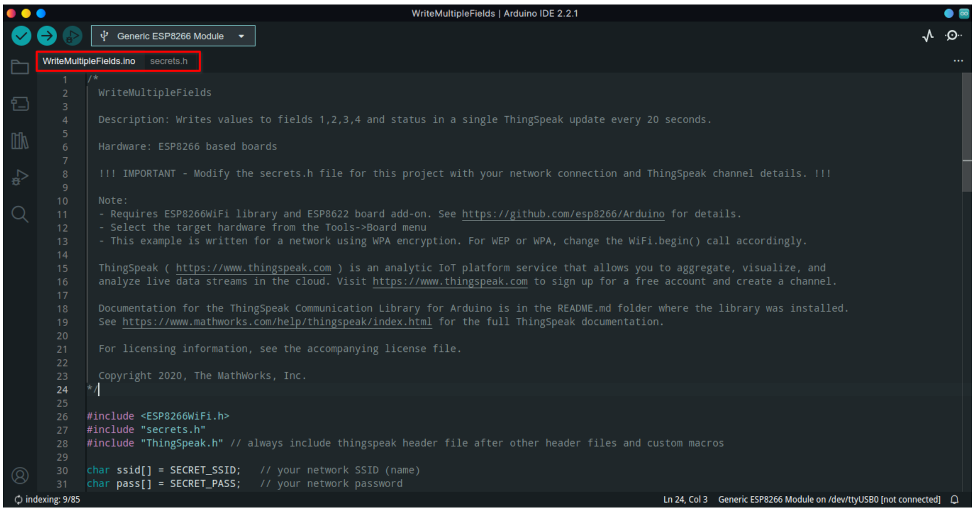
indexing: 49/85

Ln 10, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

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

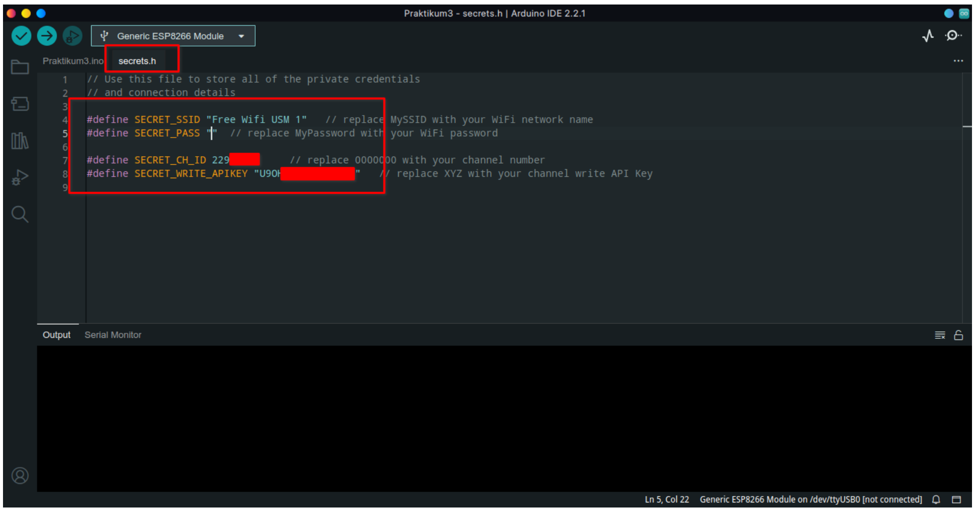
yang sudah disiapkan oleh **Library**. Klik **File** → **Examples** → **ThingSpeak** → **ESP8266** → **Program Board Directly** → **Write Multiple Fields**

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


WriteMultipleFields | Arduino IDE 2.2.1
Generic ESP8266 Module
WriteMultipleFields.ino
secrets.h
indexing: 9/85
Ln 24, Col 3
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```
/*
 WriteMultipleFields

 Description: Writes values to fields 1,2,3,4 and status in a single ThingSpeak update every 20 seconds.

 Hardware: ESP8266 based boards

 !!! IMPORTANT - Modify the secrets.h file for this project with your network connection and ThingSpeak channel details. !!!

 Note:
 - Requires ESP8266WiFi library and ESP8622 board add-on. See https://github.com/esp8266/Arduino for details.
 - Select the target hardware from the Tools->Board menu
 - This example is written for a network using WPA encryption. For WEP or WPA, change the WiFi.begin() call accordingly.

 ThingSpeak ( https://www.thingspeak.com ) is an analytic IoT platform service that allows you to aggregate, visualize, and
 analyze live data streams in the cloud. Visit https://www.thingspeak.com to sign up for a free account and create a channel.

 Documentation for the ThingSpeak Communication Library for Arduino is in the README.md folder where the library was installed.
 See https://www.mathworks.com/help/thingspeak/index.html for the full ThingSpeak documentation.

 For licensing information, see the accompanying license file.

 Copyright 2020, The MathWorks, Inc.
*/

#include <ESP8266WiFi.h>
#include "secrets.h"
#include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros

char ssid[] = SECRET_SSID;   // your network SSID (name)
char pass[] = SECRET_PASS;   // your network password
```

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

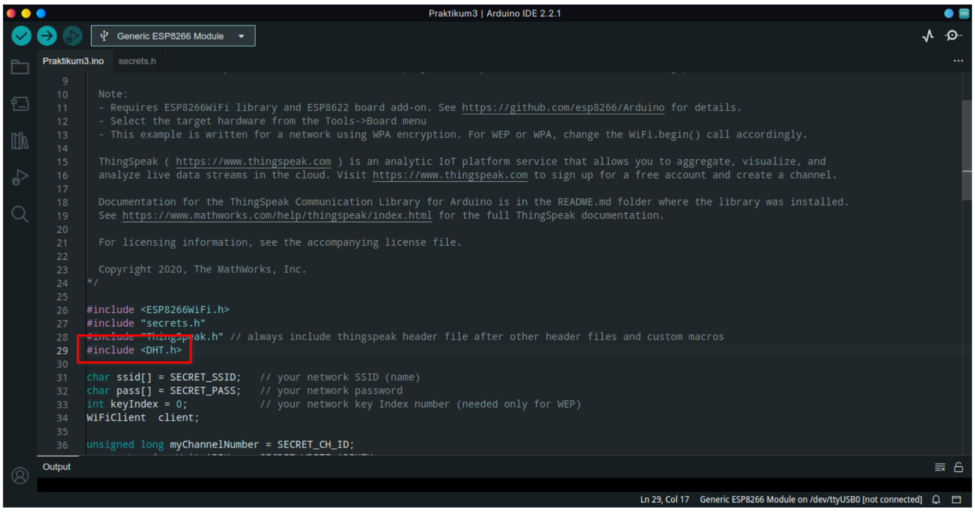

Praktikum3 - secrets.h | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino
secrets.h
Output
Serial Monitor
Ln 5, Col 22
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```
// Use this file to store all of the private credentials
// and connection details

#define SECRET_SSID "Free Wifi USM 1"   // replace MySSID with your WiFi network name
#define SECRET_PASS ""  // replace MyPassword with your WiFi password

#define SECRET_CH_ID 229[redacted]      // replace 0000000 with your channel number
#define SECRET_WRITE_APIKEY "U9O[redacted]"   // replace XYZ with your channel write API Key

```

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

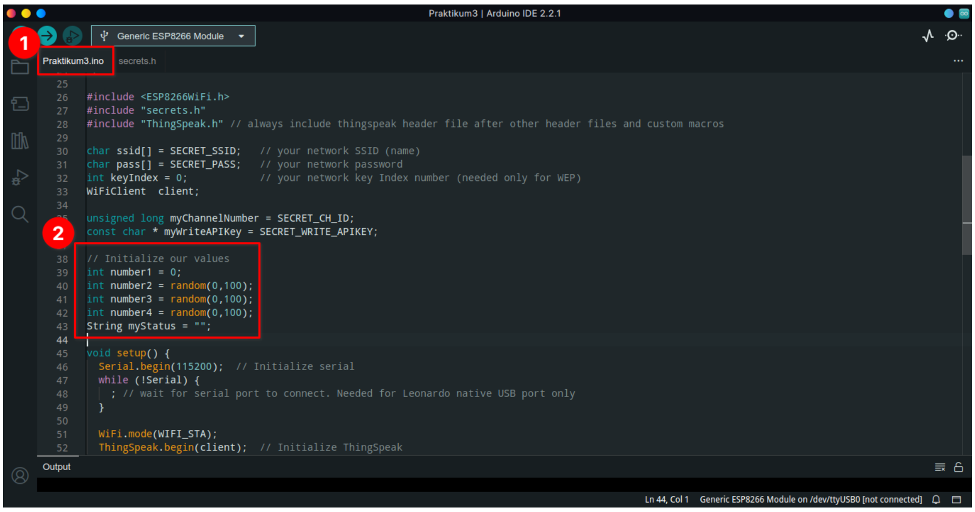
Potongan Kode

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


Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino
secrets.h
Output
Ln 29, Col 17
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```
  Note:
  - Requires ESP8266WiFi library and ESP8622 board add-on. See https://github.com/esp8266/Arduino for details.
  - Select the target hardware from the Tools->Board menu
  - This example is written for a network using WPA encryption. For WEP or WPA, change the WiFi.begin() call accordingly.

  ThingSpeak ( https://www.thingspeak.com ) is an analytic IoT platform service that allows you to aggregate, visualize, and
  analyze live data streams in the cloud. Visit https://www.thingspeak.com to sign up for a free account and create a channel.

  Documentation for the ThingSpeak Communication Library for Arduino is in the README.md folder where the library was installed.
  See https://www.mathworks.com/help/thingspeak/index.html for the full ThingSpeak documentation.

  For licensing information, see the accompanying license file.

  Copyright 2020, The MathWorks, Inc.
*/

#include <ESP8266WiFi.h>
#include "secrets.h"
#include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros
#include <DHT.h>

char ssid[] = SECRET_SSID;   // your network SSID (name)
char pass[] = SECRET_PASS;   // your network password
int keyIndex = 0;            // your network key Index number (needed only for WEP)
WiFiClient  client;

unsigned long myChannelNumber = SECRET_CH_ID;
```

21. Hapus kode berikut

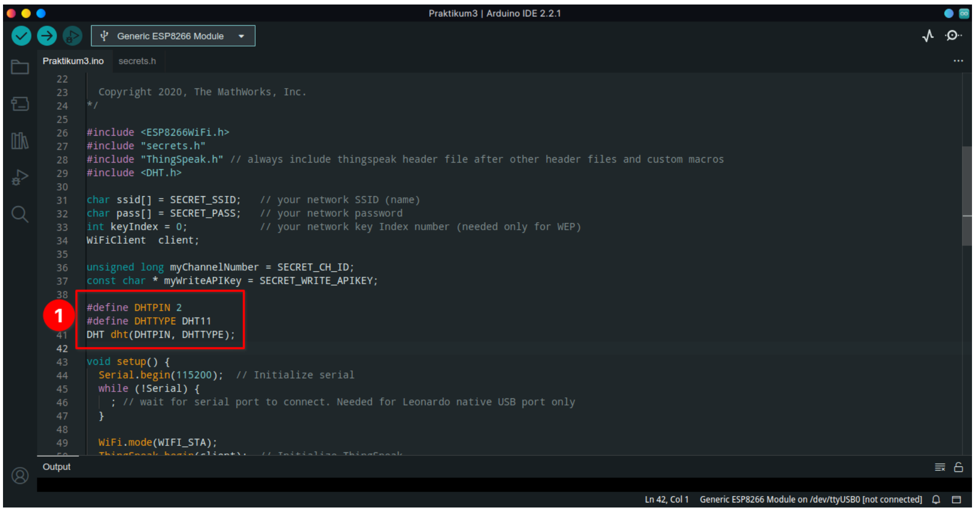

Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
1
Praktikum3.ino
secrets.h
2
Output
Ln 44, Col 1
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```

#include <ESP8266WiFi.h>
#include "secrets.h"
#include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros

char ssid[] = SECRET_SSID;   // your network SSID (name)
char pass[] = SECRET_PASS;   // your network password
int keyIndex = 0;            // your network key Index number (needed only for WEP)
WiFiClient  client;

unsigned long myChannelNumber = SECRET_CH_ID;
const char * myWriteAPIKey = SECRET_WRITE_APIKEY;

// Initialize our values
int number1 = 0;
int number2 = random(0,100);
int number3 = random(0,100);
int number4 = random(0,100);
String myStatus = "";

void setup() {
  Serial.begin(115200);  // Initialize serial
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo native USB port only
  }

  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);  // Initialize ThingSpeak
```

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

Potongan Kode

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

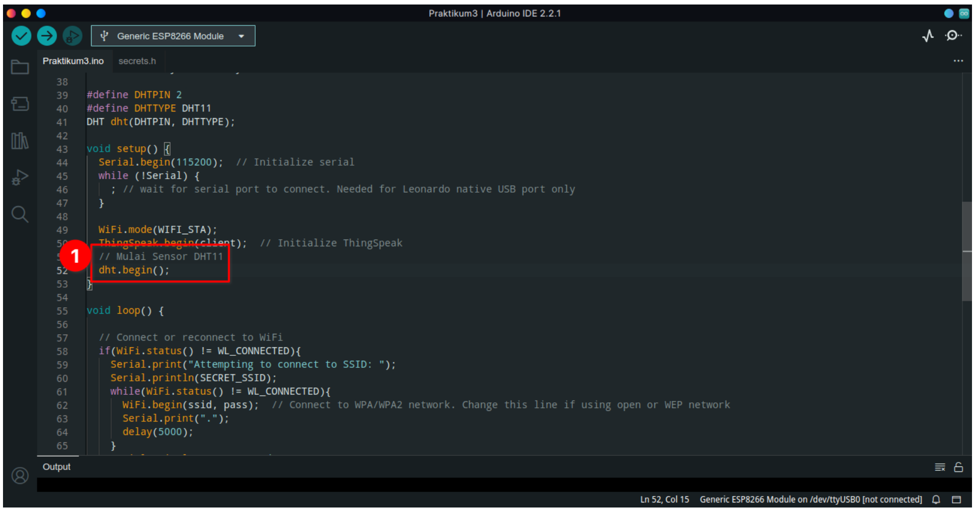

Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino
secrets.h
1
Output
Ln 42, Col 1
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```
  Copyright 2020, The MathWorks, Inc.
*/

#include <ESP8266WiFi.h>
#include "secrets.h"
#include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros
#include <DHT.h>

char ssid[] = SECRET_SSID;   // your network SSID (name)
char pass[] = SECRET_PASS;   // your network password
int keyIndex = 0;            // your network key Index number (needed only for WEP)
WiFiClient  client;

unsigned long myChannelNumber = SECRET_CH_ID;
const char * myWriteAPIKey = SECRET_WRITE_APIKEY;

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

void setup() {
  Serial.begin(115200);  // Initialize serial
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo native USB port only
  }

  WiFi.mode(WIFI_STA);
```

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

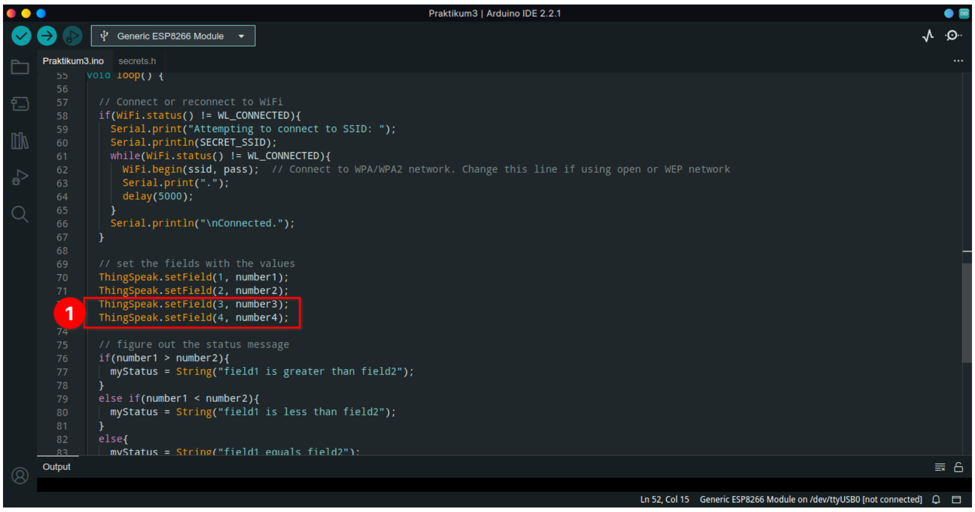
Potongan Kode

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


Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino
secrets.h
1
Output
Ln 52, Col 15
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

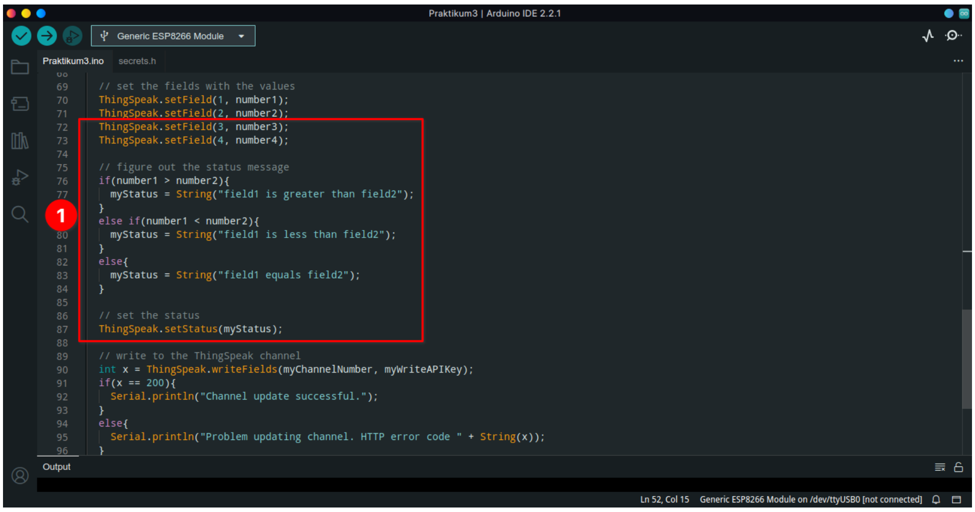
```

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

void setup() {
  Serial.begin(115200);  // Initialize serial
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo native USB port only
  }

  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);  // Initialize ThingSpeak
  // Mulai Sensor DHT11
  dht.begin();
}

void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open or WEP network
      Serial.print(".");
      delay(5000);
    }
```

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



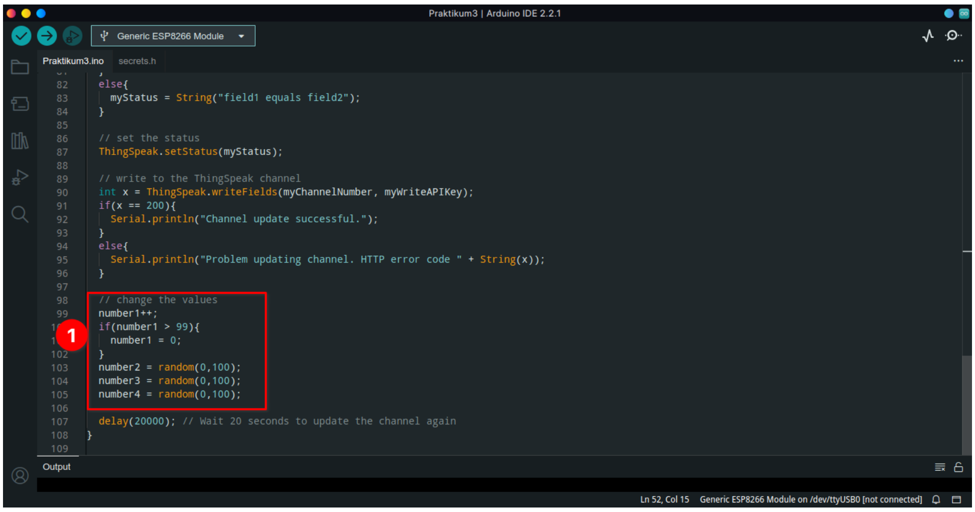
Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino
secrets.h
1
Output
Ln 52, Col 15
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```
void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open or WEP network
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

  // set the fields with the values
  ThingSpeak.setField(1, number1);
  ThingSpeak.setField(2, number2);
  ThingSpeak.setField(3, number3);
  ThingSpeak.setField(4, number4);

  // figure out the status message
  if(number1 > number2){
    myStatus = String("field1 is greater than field2");
  }
  else if(number1 < number2){
    myStatus = String("field1 is less than field2");
  }
  else{
    myStatus = String("field1 equals field2");
```



Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino
secrets.h
1
Output
Ln 52, Col 15
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

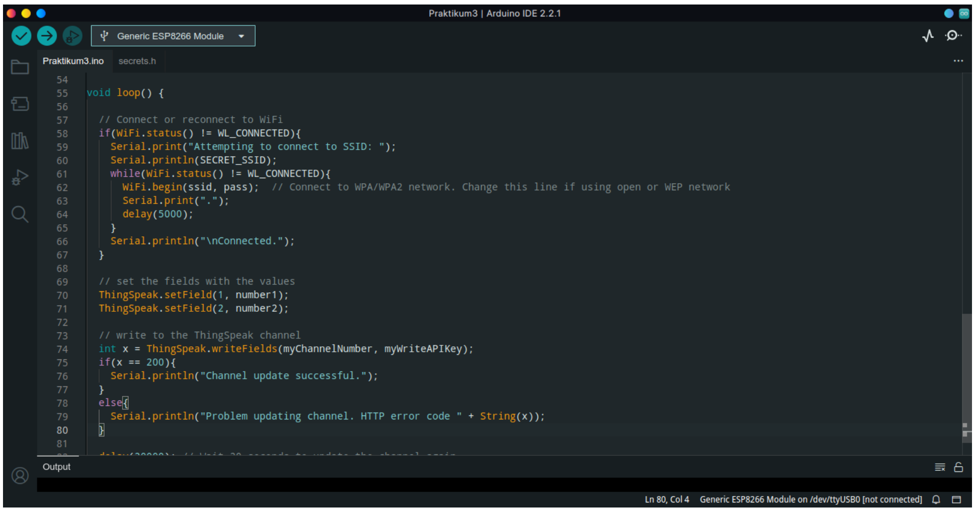
```
  // set the fields with the values
  ThingSpeak.setField(1, number1);
  ThingSpeak.setField(2, number2);
  ThingSpeak.setField(3, number3);
  ThingSpeak.setField(4, number4);

  // figure out the status message
  if(number1 > number2){
    myStatus = String("field1 is greater than field2");
  }
  else if(number1 < number2){
    myStatus = String("field1 is less than field2");
  }
  else{
    myStatus = String("field1 equals field2");
  }

  // set the status
  ThingSpeak.setStatus(myStatus);

  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }
```



Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino
secrets.h
1
Output
Ln 52, Col 15
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```
  else{
    myStatus = String("field1 equals field2");
  }

  // set the status
  ThingSpeak.setStatus(myStatus);

  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }

  // change the values
  number1++;
  if(number1 > 99){
    number1 = 0;
  }
  number2 = random(0,100);
  number3 = random(0,100);
  number4 = random(0,100);

  delay(20000); // Wait 20 seconds to update the channel again
}
```

25. Hasil AKHIR SEHARUSNYA:



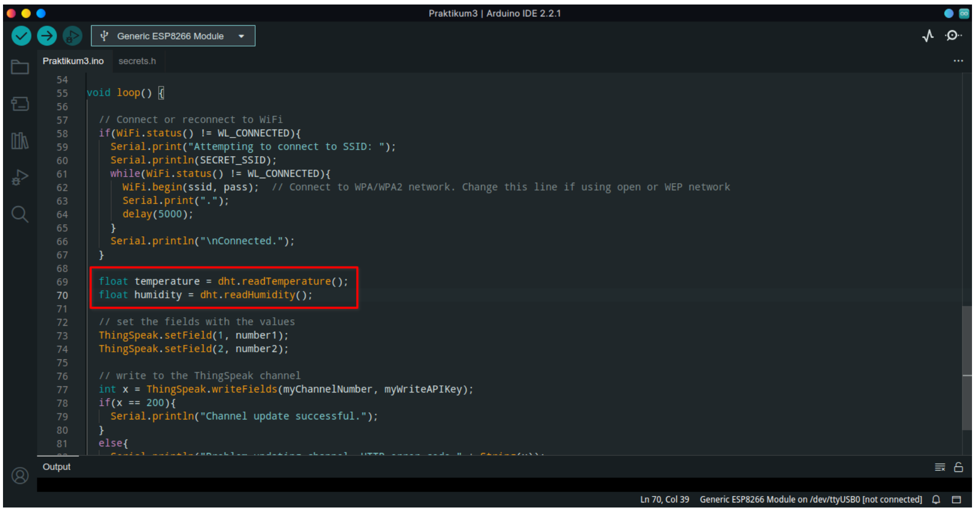
Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino
secrets.h
Output
Ln 80, Col 4
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```
void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open or WEP network
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

  // set the fields with the values
  ThingSpeak.setField(1, number1);
  ThingSpeak.setField(2, number2);

  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }

```

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

Potongan Kode

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



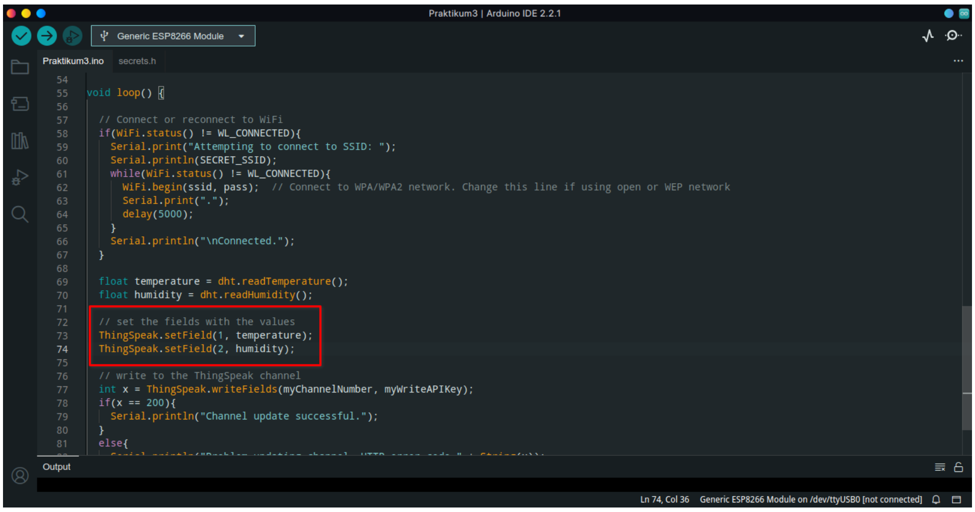
Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino
secrets.h
Output
Ln 70, Col 39
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```
void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open or WEP network
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

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

  // set the fields with the values
  ThingSpeak.setField(1, number1);
  ThingSpeak.setField(2, number2);

  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
```

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

Potongan Kode

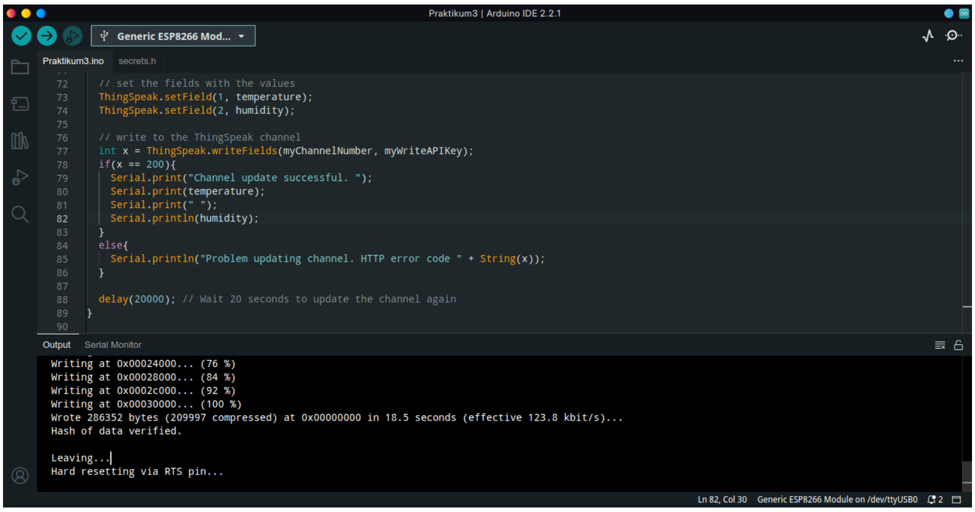
```
ThingSpeak.setField(1, temperature);
ThingSpeak.setField(2, humidity);
```



```
void loop() {

  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open or WEP network
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

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

  // set the fields with the values
  ThingSpeak.setField(1, temperature);
  ThingSpeak.setField(2, humidity);

  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
```

Praktikum3 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum3.ino
secrets.h
Output
Ln 74, Col 36
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

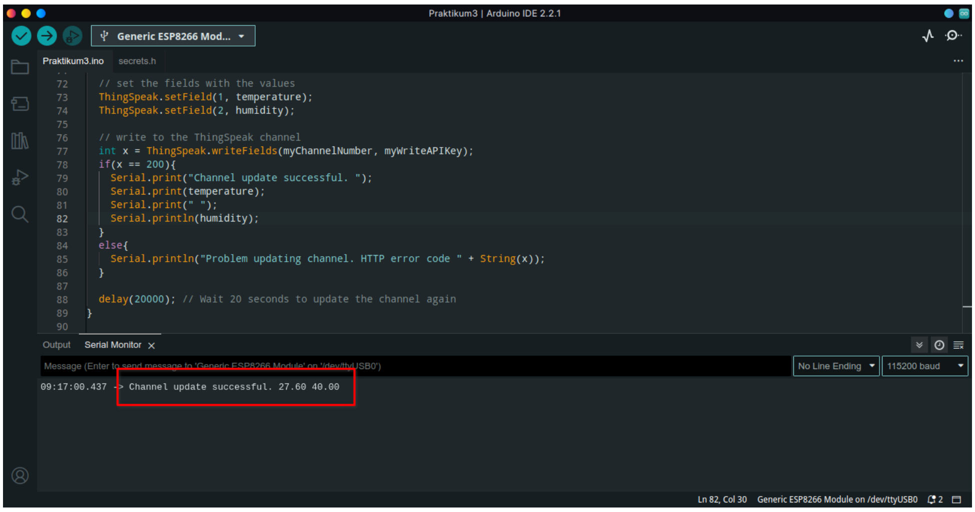
28. **Verifikasi** untuk memastikan kode sudah benar. Lalu klik **Upload**



```
  // set the fields with the values
  ThingSpeak.setField(1, temperature);
  ThingSpeak.setField(2, humidity);

  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.print("Channel update successful. ");
    Serial.print(temperature);
    Serial.print(" ");
    Serial.println(humidity);
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }

  delay(20000); // Wait 20 seconds to update the channel again
}
```

```
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



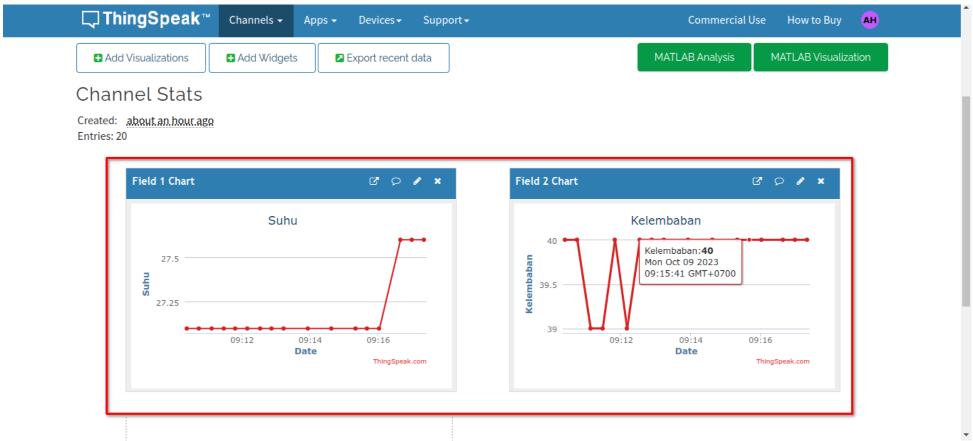
```
  // set the fields with the values
  ThingSpeak.setField(1, temperature);
  ThingSpeak.setField(2, humidity);

  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.print("Channel update successful. ");
    Serial.print(temperature);
    Serial.print(" ");
    Serial.println(humidity);
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }

  delay(20000); // Wait 20 seconds to update the channel again
}
```

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

No Line Ending

115200 baud

```
09:17:00.437 -> Channel update successful. 27.60 40.00
```

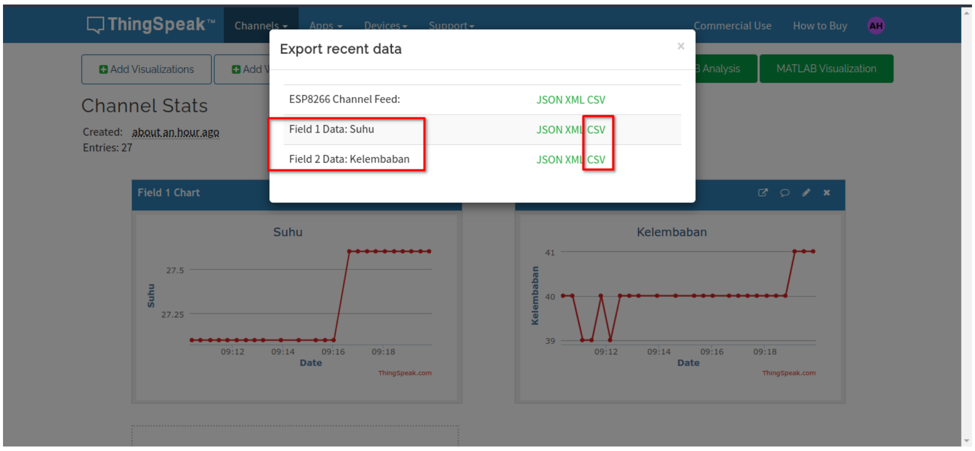


Channel Stats

Created: about an hour ago

Entries: 20

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



Channel Stats

Created: about an hour ago

Entries: 27

Export recent data

ESP8266 Channel Feed: JSON XML CSV

Field 1 Data: Suhu JSON XML CSV

Field 2 Data: Kelembaban JSON XML CSV

Bab 4

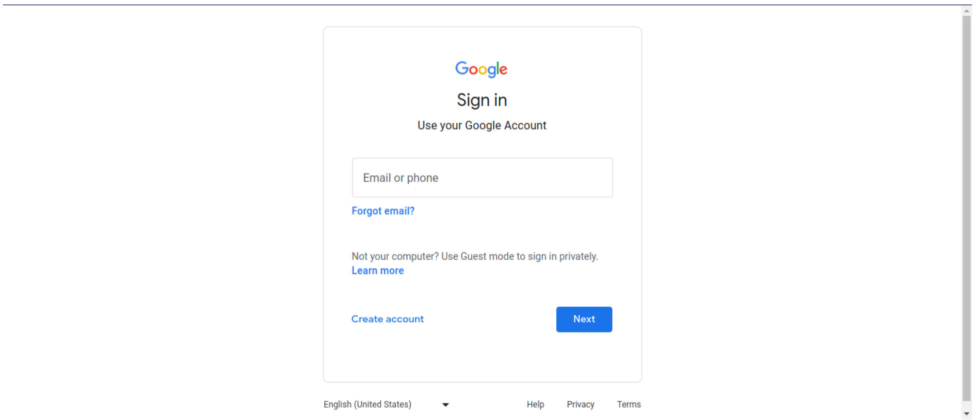
Praktikum 4

4.1 ESP8266, DHT11, dan Firebase Realtime

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

4.2 Tutorial

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

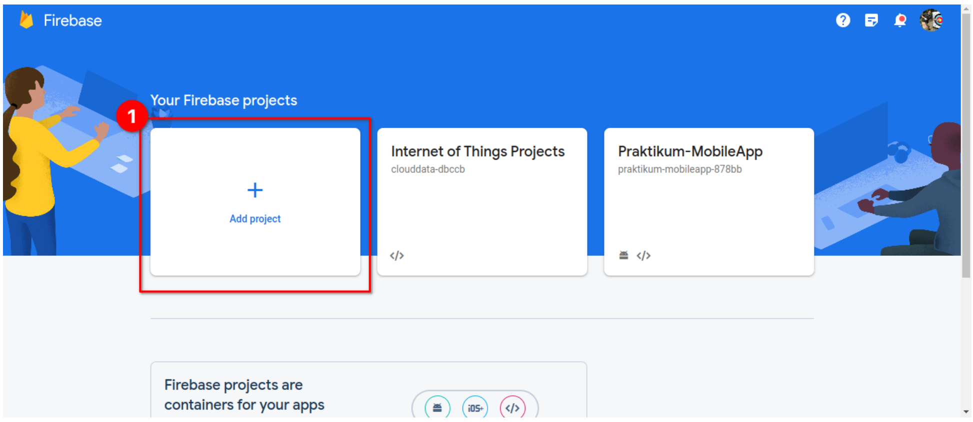


Sign in

Use your Google Account

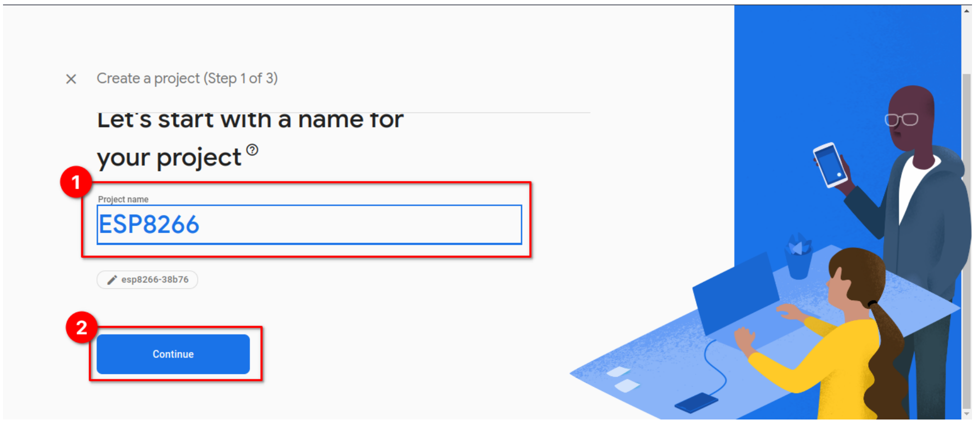
Not your computer? Use Guest mode to sign in privately.

2. Buat projek baru dengan melakukan klik tanda +



Firebase projects are containers for your apps

3. Isi nama projek



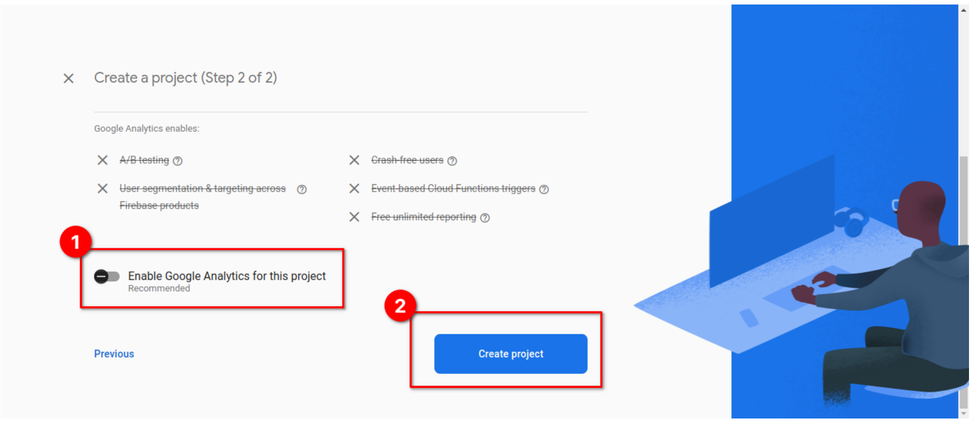
Create a project (Step 1 of 3)

Let's start with a name for your project

Project name: ESP8266

esp8266-38b76

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



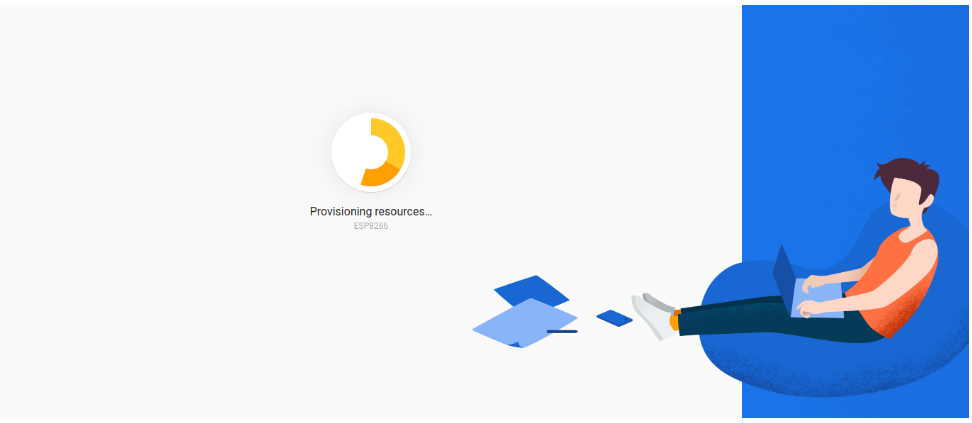
Create a project (Step 2 of 2)

Google Analytics enables:

- ~~A/B testing~~
- ~~User segmentation & targeting across Firebase products~~
- ~~Crash-free users~~
- ~~Event-based Cloud Functions triggers~~
- ~~Free unlimited reporting~~

○ Enable Google Analytics for this project
Recommended

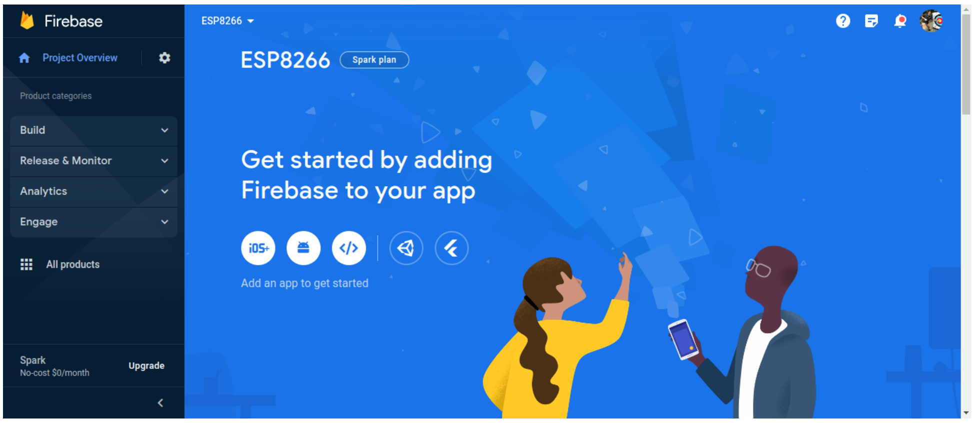
5. Tunggu proses berlangsung dan klik tombol apabila sudah muncul



Provisioning resources...

ESP8266

6. Firebase akan menampilkan dasbor sistem

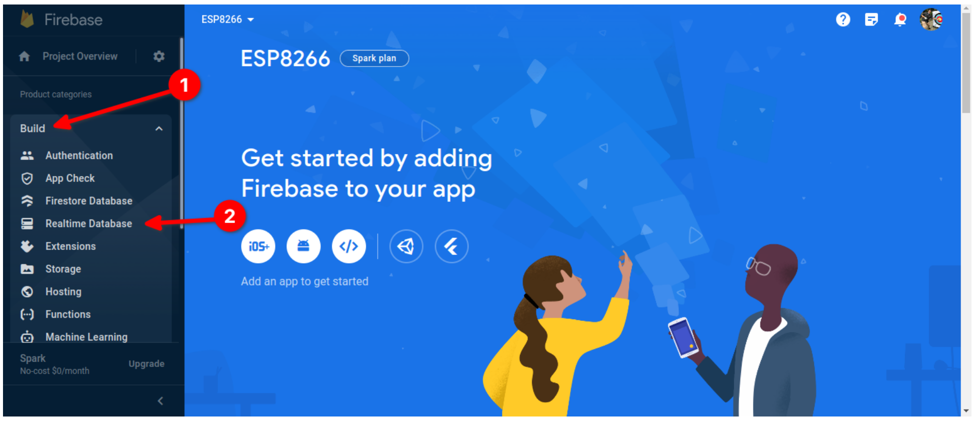


ESP8266

Get started by adding Firebase to your app

Add an app to get started

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

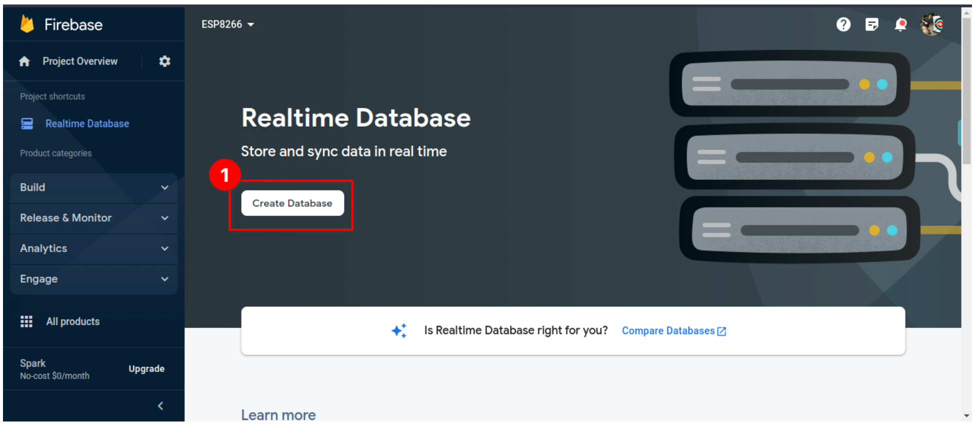


ESP8266

Get started by adding Firebase to your app

Add an app to get started

8. Klik **Create Database**

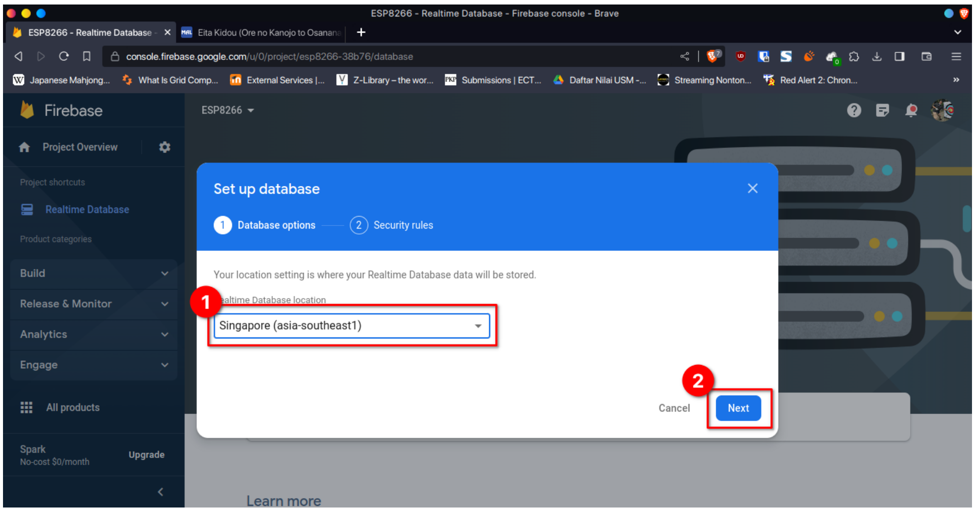


Realtime Database

Store and sync data in real time

Is Realtime Database right for you?

9. Pilih Lokasi dan Klik **Next**



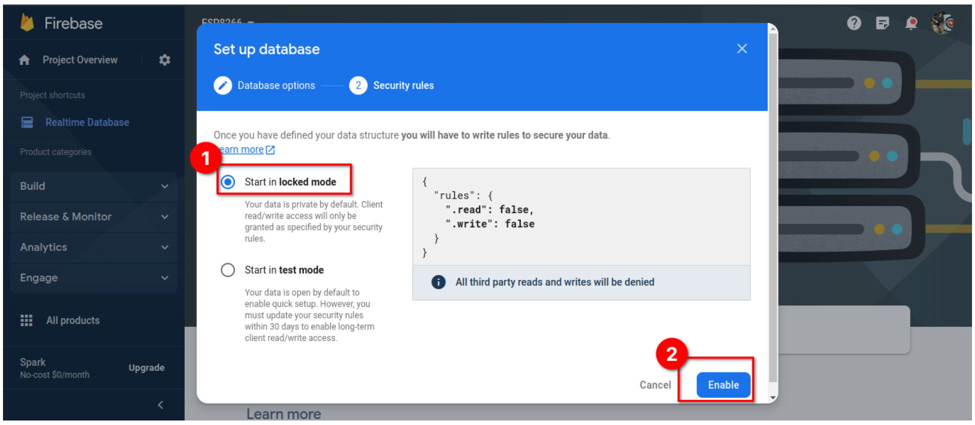
Set up database

1 Database options — 2 Security rules

Your location setting is where your Realtime Database data will be stored.

Realtime Database location: Singapore (asia-southeast1)

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



Set up database

Database options — 2 Security rules

Once you have defined your data structure **you will have to write rules to secure your data.** Learn more

● Start in **locked mode**

Your data is private by default. Client read/write access will only be granted as specified by your security rules.

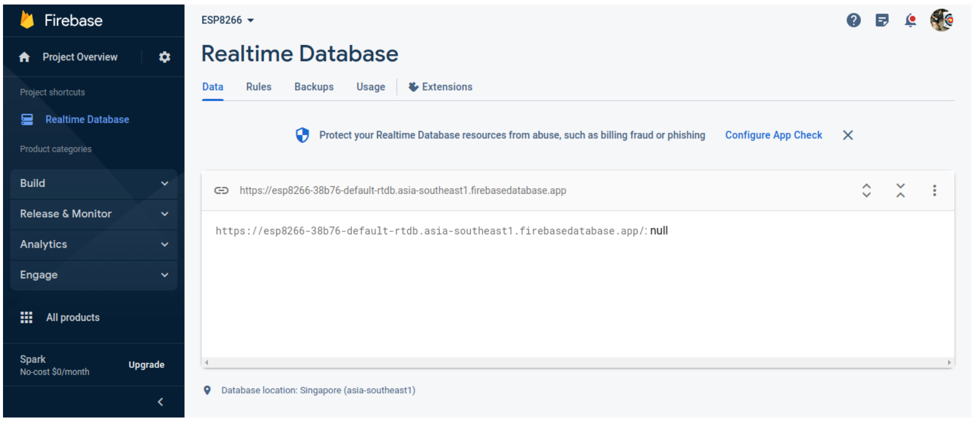
○ Start in **test mode**

Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{
  "rules": {
    ".read": false,
    ".write": false
  }
}
```

All third party reads and writes will be denied

11. Database sudah dibuat



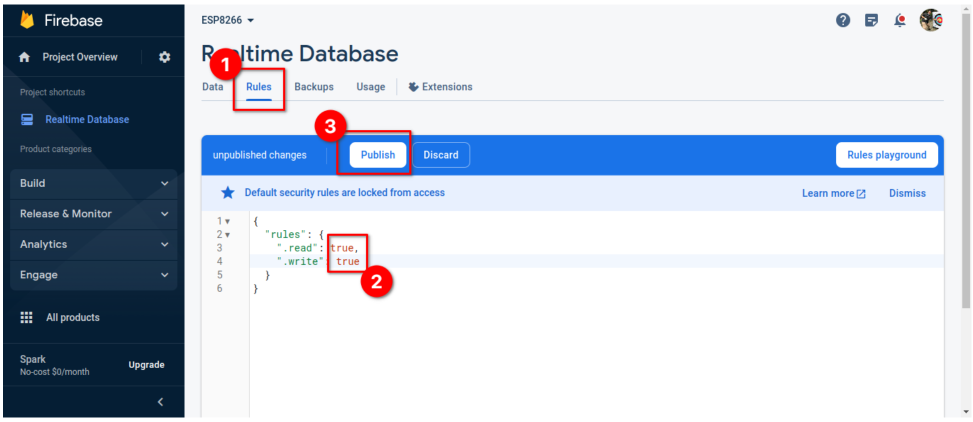
Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://esp8266-38b76-default-rtdb.asia-southeast1.firebasedatabase.app

```
https://esp8266-38b76-default-rtdb.asia-southeast1.firebasedatabase.app/: null
```

Database location: Singapore (asia-southeast1)

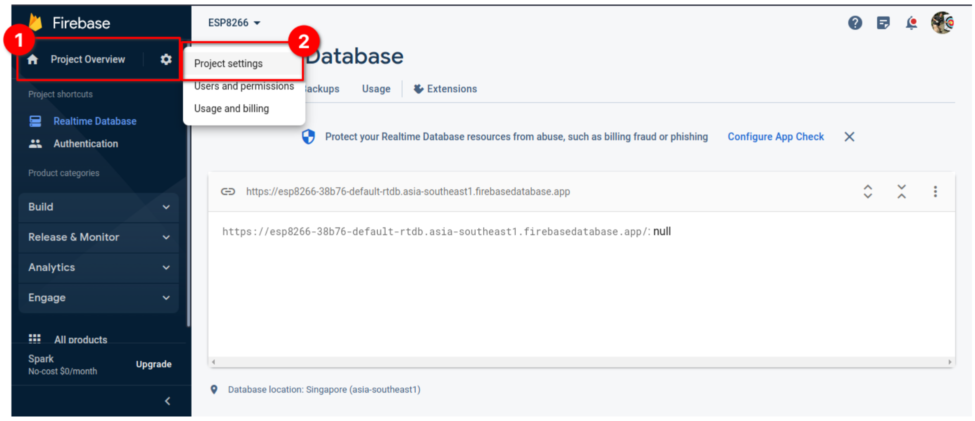
12. Sebelumnya ubah aturan database dengan klik **Rules**, dan ubah kata **false** menjadi **true**. dan klik **Publish**



Default security rules are locked from access

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

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



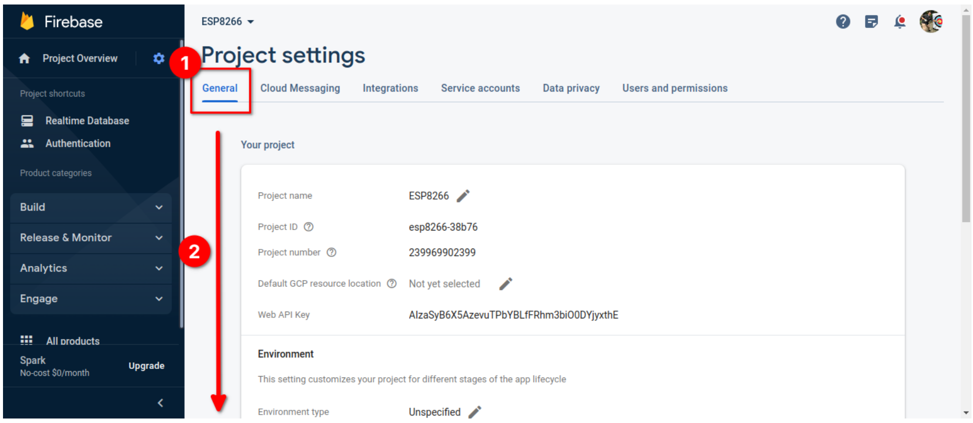
Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://esp8266-38b76-default-rtdb.asia-southeast1.firebasedatabase.app

```
https://esp8266-38b76-default-rtdb.asia-southeast1.firebasedatabase.app/: null
```

Database location: Singapore (asia-southeast1)

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



Project settings

Your project

Project name ESP8266

Project ID esp8266-38b76

Project number 239969902399

Default GCP resource location Not yet selected

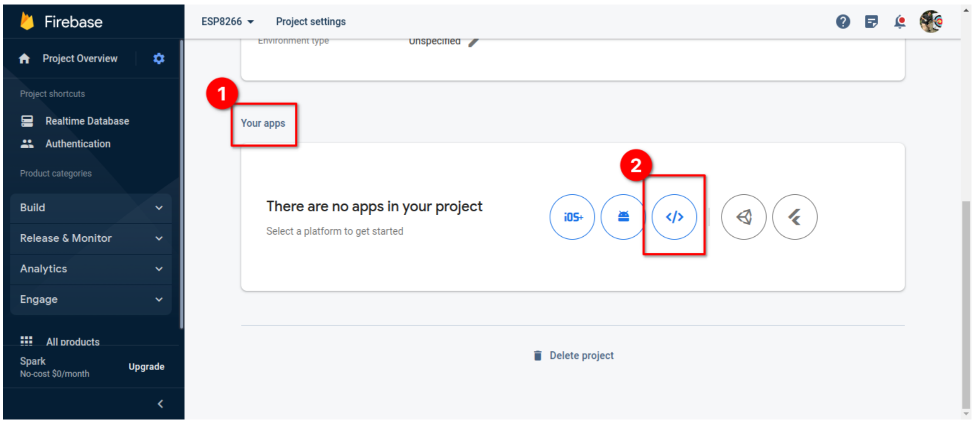
Web API Key AIzaSyB6X5AzevuTPbYBLfFRhm3biO0DYjyxthE

Environment

This setting customizes your project for different stages of the app lifecycle

Environment type Unspecified

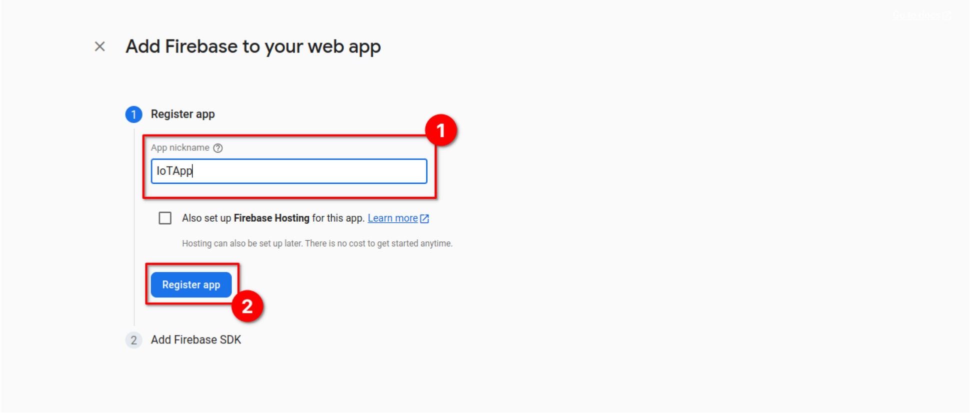
15. Di bagian **Your Apps** pilih **Web**



There are no apps in your project

Select a platform to get started

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

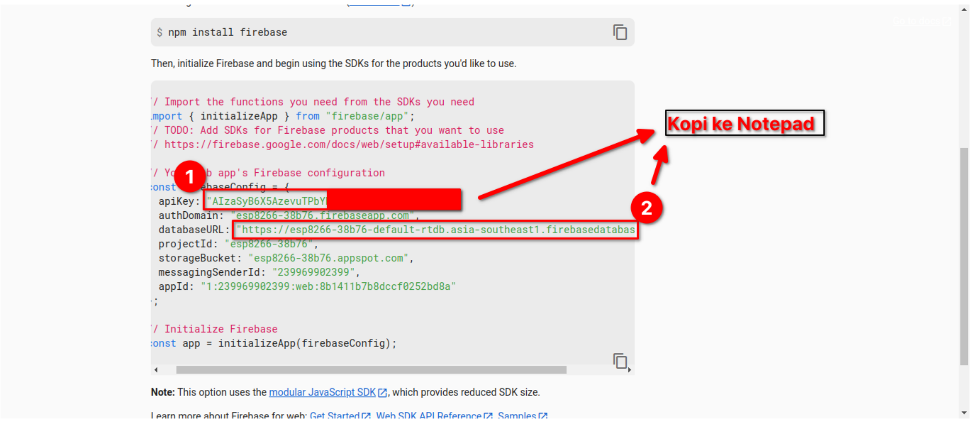


App nickname IoTApp

□ Also set up **Firebase Hosting** for this app. Learn more

Hosting can also be set up later. There is no cost to get started anytime.

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



```
$ npm install firebase
```

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

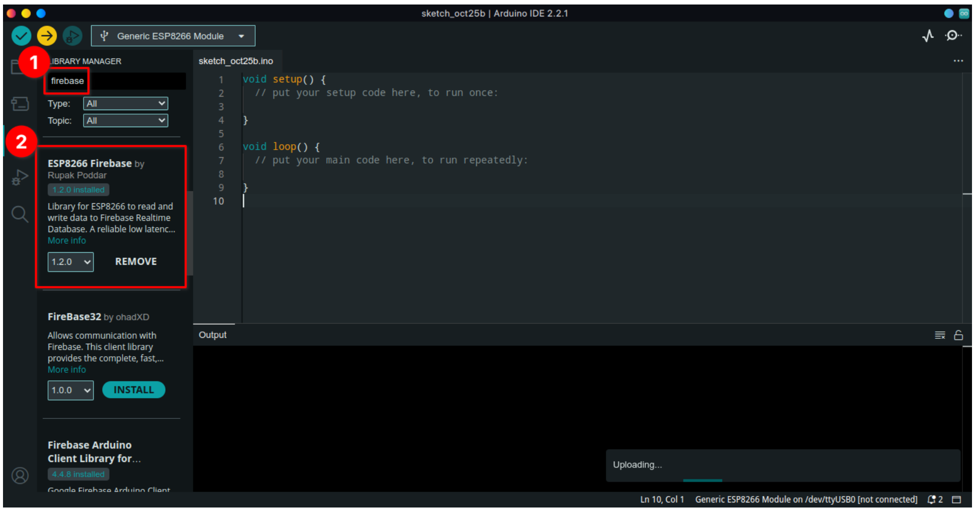
```
/ Import the functions you need from the SDKs you need
mport { initializeApp } from "firebase/app";
/ TODO: Add SDKs for Firebase products that you want to use
/ https://firebase.google.com/docs/web/setup#available-libraries

/ Yo   b app's Firebase configuration
onst   ebaseConfig = {
 apiKey: "AIzaSyB6X5AzevuTPbY[redacted]
 authDomain: "esp8266-38b76.firebaseapp.com",
 databaseURL: "https://esp8266-38b76-default-rtdb.asia-southeast1.firebasedatabas
 projectId: "esp8266-38b76",
 storageBucket: "esp8266-38b76.appspot.com",
 messagingSenderId: "239969902399",
 appId: "1:239969902399:web:8b1411b7b8dccf0252bd8a"
;

/ Initialize Firebase
onst app = initializeApp(firebaseConfig);
```

Note: This option uses the modular JavaScript SDK, which provides reduced SDK size.

Learn more about Firebase for web: Get Started Web SDK API Reference Samples

Kopi ke Notepad

18. Di **Arduino IDE**, buka **Libraries** dan install **ESP8266 Firebase** dan **Firebase Arduino Client Library**



sketch_oct25b | Arduino IDE 2.2.1

Generic ESP8266 Module

LIBRARY MANAGER

firebase

Type: All

Topic: All

ESP8266 Firebase by Rupak Poddar

1.2.0 installed

Library for ESP8266 to read and write data to Firebase Realtime Database. A reliable low latenc...

More info

1.2.0 REMOVE

FireBase32 by ohadXD

Allows communication with Firebase. This client library provides the complete, fast,...

More info

1.0.0 INSTALL

Firebase Arduino Client Library for...

4.4.8 installed

Google Firebase Arduino Client

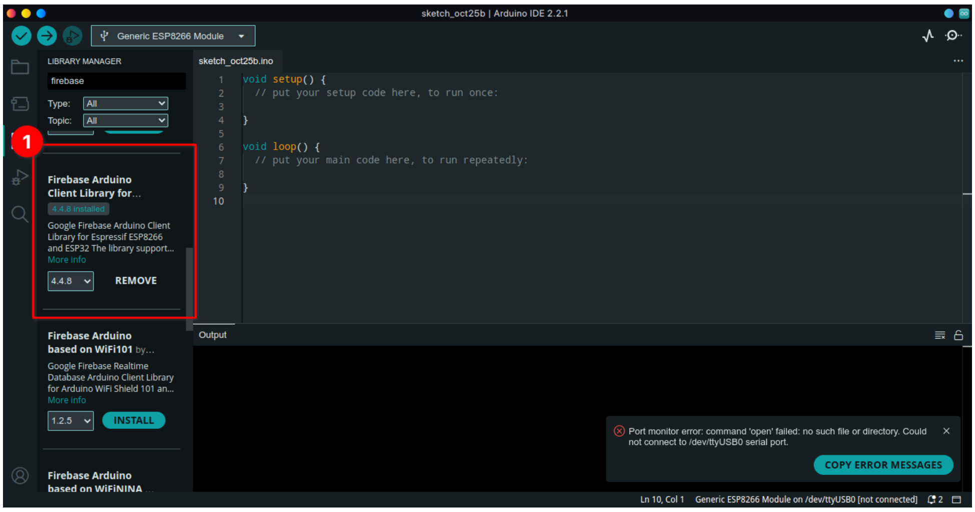
sketch_oct25b.ino

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Output

Uploading...

Ln 10, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]



sketch_oct25b | Arduino IDE 2.2.1

Generic ESP8266 Module

LIBRARY MANAGER

firebase

Type: All

Topic: All

Firebase Arduino Client Library for...

4.4.8 installed

Google Firebase Arduino Client Library for Espressif ESP8266 and ESP32 The library support...

More info

4.4.8 REMOVE

Firebase Arduino based on WiFi101 by...

Google Firebase Realtime Database Arduino Client Library for Arduino WiFi Shield 101 an...

More info

1.2.5 INSTALL

Firebase Arduino based on WiFiNINA ...

sketch_oct25b.ino

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

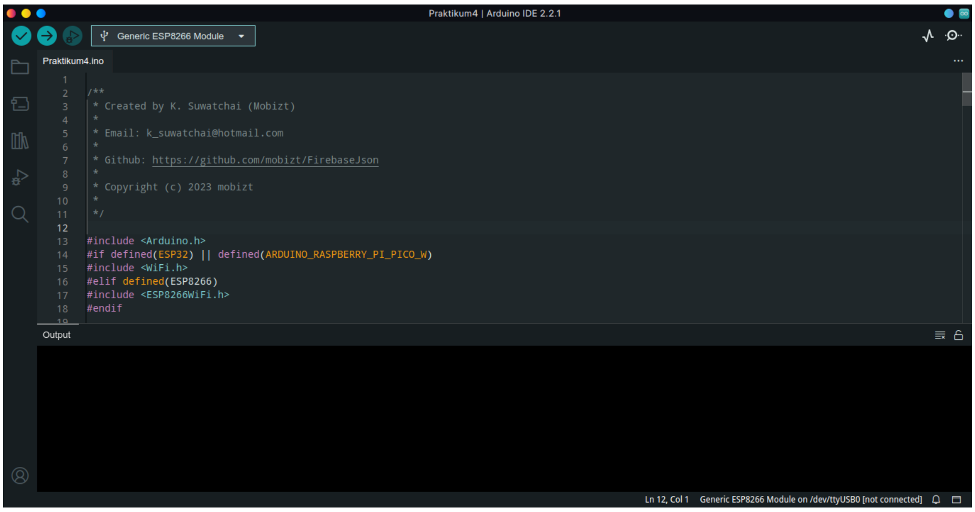
Output

Port monitor error: command 'open' failed: no such file or directory. Could not connect to /dev/ttyUSB0 serial port.

COPY ERROR MESSAGES

Ln 10, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

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

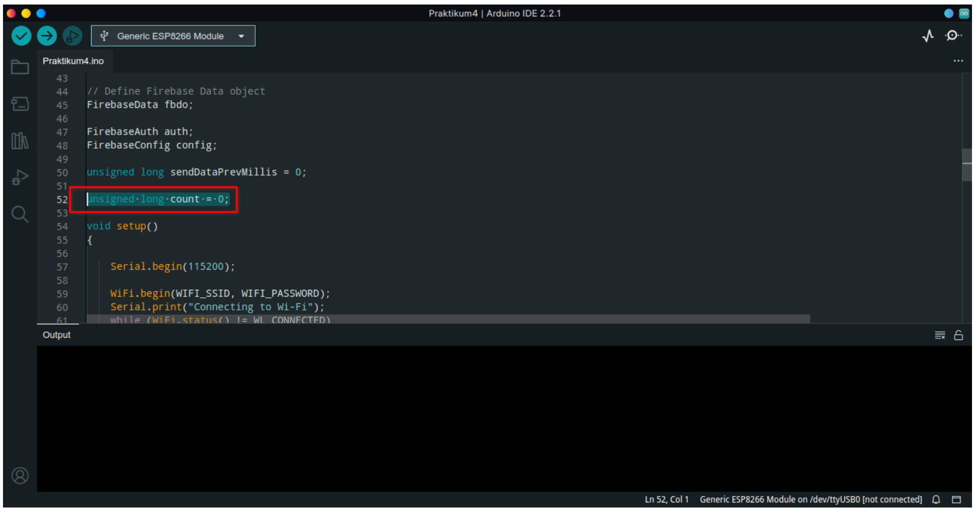


```
/**
 * Created by K. Suwatchai (Mobizt)
 *
 * Email: k_suwatchai@hotmail.com
 *
 * Github: https://github.com/mobizt/FirebaseJson
 *
 * Copyright (c) 2023 mobizt
 *
 */

#include <Arduino.h>
#if defined(ESP32) || defined(ARDUINO_RASPBERRY_PI_PICO_W)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
```

Ln 12, Col 1   Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

20. Hapus beberapa bagian kode berikut:

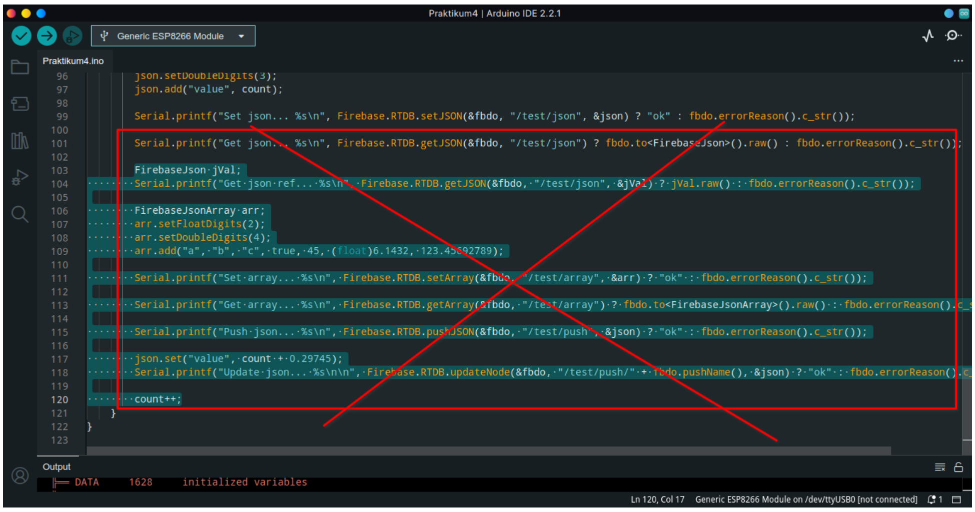
- Bagian 1



```

// Define Firebase Data object
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;

unsigned long count = 0;

void setup()
{

    Serial.begin(115200);

    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Connecting to Wi-Fi");
    while (WiFi.status() != WL_CONNECTED)
```

Ln 52, Col 1   Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

- Bagian 2

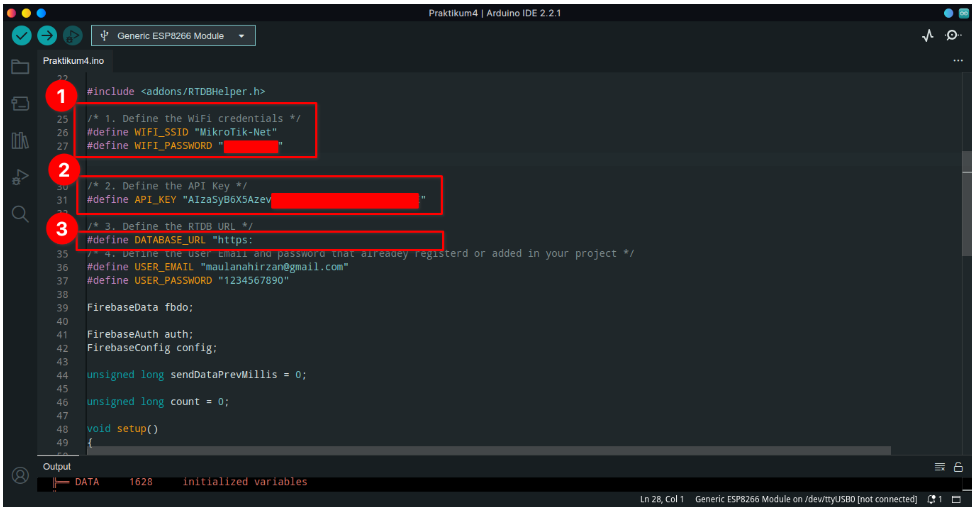


```
        json.setDoubleDigits(3);
        json.add("value", count);

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

        Serial.printf("Get json... %s\n", Firebase.RTDB.getJSON(&fbdo, "/test/json") ? fbdo.to<FirebaseJson>().raw() : fbdo.errorReason().c_str());

        FirebaseJson jVal;
        Serial.printf("Get json ref... %s\n", Firebase.RTDB.getJSON(&fbdo, "/test/json", &jVal) ? jVal.raw() : fbdo.errorReason().c_str());

        FirebaseJsonArray arr;
        arr.setFloatDigits(2);
        arr.setDoubleDigits(4);
        arr.add("a", "b", "c", true, 45, (float)6.1432, 123.45692789);

        Serial.printf("Set array... %s\n", Firebase.RTDB.setArray(&fbdo, "/test/array", &arr) ? "ok" : fbdo.errorReason().c_str());

        Serial.printf("Get array... %s\n", Firebase.RTDB.getArray(&fbdo, "/test/array") ? fbdo.to<FirebaseJsonArray>().raw() : fbdo.errorReason().c_

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

        json.set("value", count + 0.29745);
        Serial.printf("Update json... %s\n\n", Firebase.RTDB.updateNode(&fbdo, "/test/push/" + fbdo.pushName(), &json) ? "ok" : fbdo.errorReason().c_

        count++;
    }
}

```

```
DATA     1628     initialized variables
```

Ln 120, Col 17   Generic ESP8266 Module on /dev/ttyUSB0 [not connected]   1

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

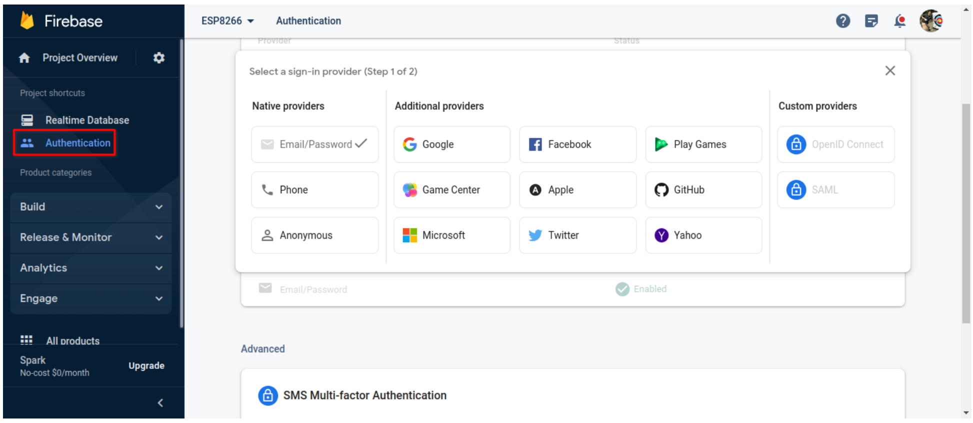


Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum4.ino
1
2
3
Output
Ln 28, Col 1   Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

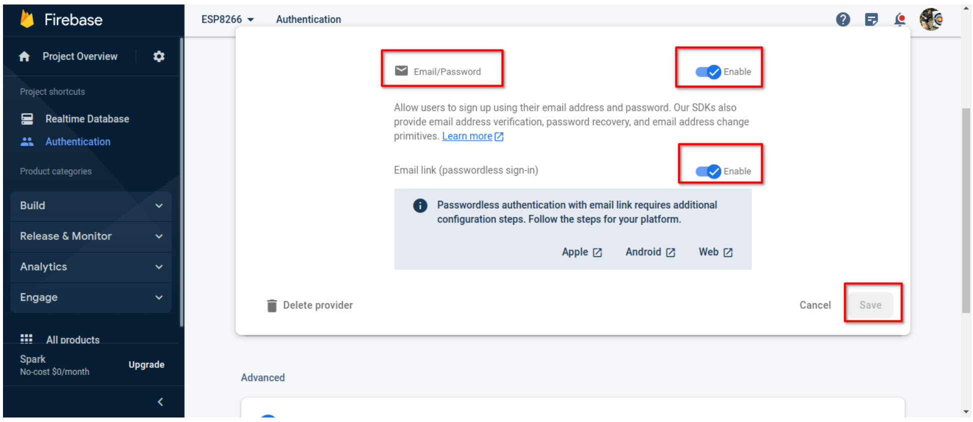
```
#include <addons/RTDBHelper.h>

/* 1. Define the WiFi credentials */
#define WIFI_SSID "MikroTik-Net"
#define WIFI_PASSWORD "[redacted]"

/* 2. Define the API Key */
#define API_KEY "AIzaSyB6X5Azev[redacted]"

/* 3. Define the RTDB URL */
#define DATABASE_URL "https:
/* 4. Define the user Email and password that alreadey registerd or added in your project */
#define USER_EMAIL "maulanahirzan@gmail.com"
#define USER_PASSWORD "1234567890"

FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;

unsigned long count = 0;

void setup()
{
```

```
DATA      1628      initialized variables
```

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



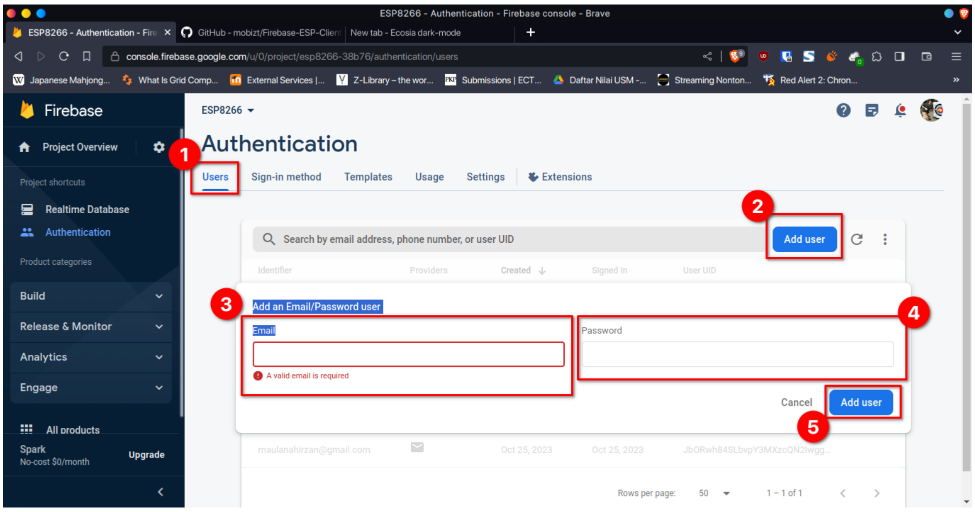
23. Pilih **Email/Password**, klik semua menjadi **Enable**, dan **Save**



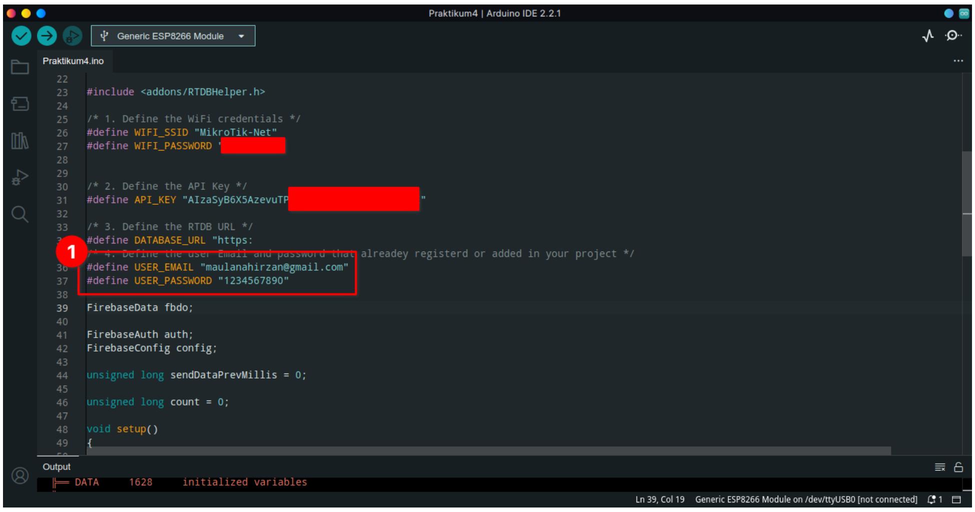
Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. Learn more

Passwordless authentication with email link requires additional configuration steps. Follow the steps for your platform.

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**


Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum4.ino
1
Output
Ln 39, Col 19
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

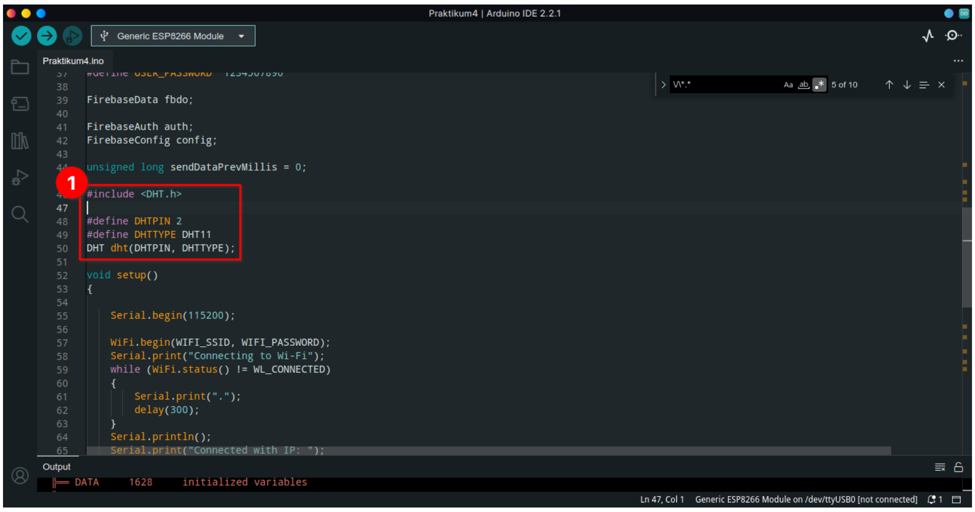
```
#include <addons/RTDBHelper.h>

/* 1. Define the WiFi credentials */
#define WIFI_SSID "MikroTik-Net"
#define WIFI_PASSWORD "[redacted]


/* 2. Define the API Key */
#define API_KEY "AIzaSyB6X5AzevuTP[redacted]"

/* 3. Define the RTDB URL */
#define DATABASE_URL "https:
/* 4. Define the user Email and password that alreadey registerd or added in your project */
#define USER_EMAIL "maulanahirzan@gmail.com"
#define USER_PASSWORD "1234567890"

FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;

unsigned long count = 0;

void setup()
{
```

```
DATA      1628      initialized variables
```

26. Berikutnya adalah menambahkan kode untuk sensor DHT

Potongan Kode

```
#include <DHT.h>

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



Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum4.ino
\/*.*
5 of 10
1
Output
Ln 47, Col 1
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

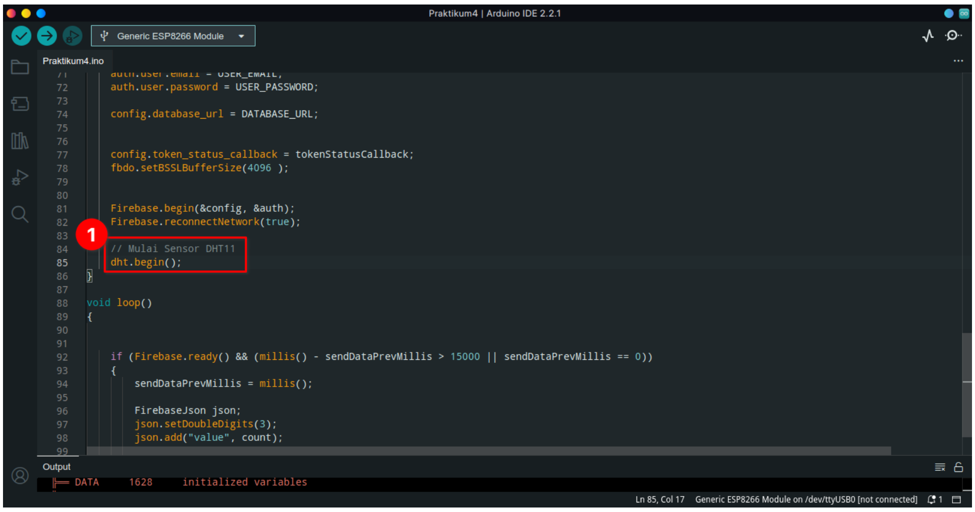
```
#define USER_PASSWORD "1234567890"

FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;

#include <DHT.h>

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

void setup()
{

    Serial.begin(115200);

    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Connecting to Wi-Fi");
    while (WiFi.status() != WL_CONNECTED)
    {
        Serial.print(".");
        delay(300);
    }
    Serial.println();
    Serial.print("Connected with IP: ");
```

```
DATA     1628     initialized variables
```

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

Potongan Kode

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



Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum4.ino
1
Output
Ln 85, Col 17
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

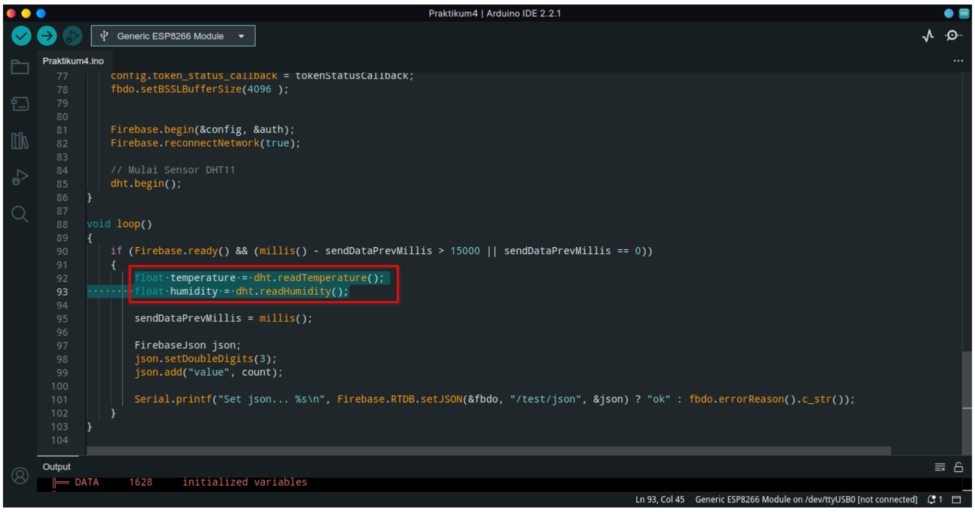
```
    auth.user.email = USER_EMAIL;
    auth.user.password = USER_PASSWORD;

    config.database_url = DATABASE_URL;


    config.token_status_callback = tokenStatusCallback;
    fbdo.setBSSLBufferSize(4096 );


    Firebase.begin(&config, &auth);
    Firebase.reconnectNetwork(true);

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

void loop()
{


    if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
    {
        sendDataPrevMillis = millis();

        FirebaseJson json;
        json.setDoubleDigits(3);
        json.add("value", count);
```

```
DATA     1628     initialized variables
```

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

Potongan Kode

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



Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum4.ino
Output
Ln 93, Col 45 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

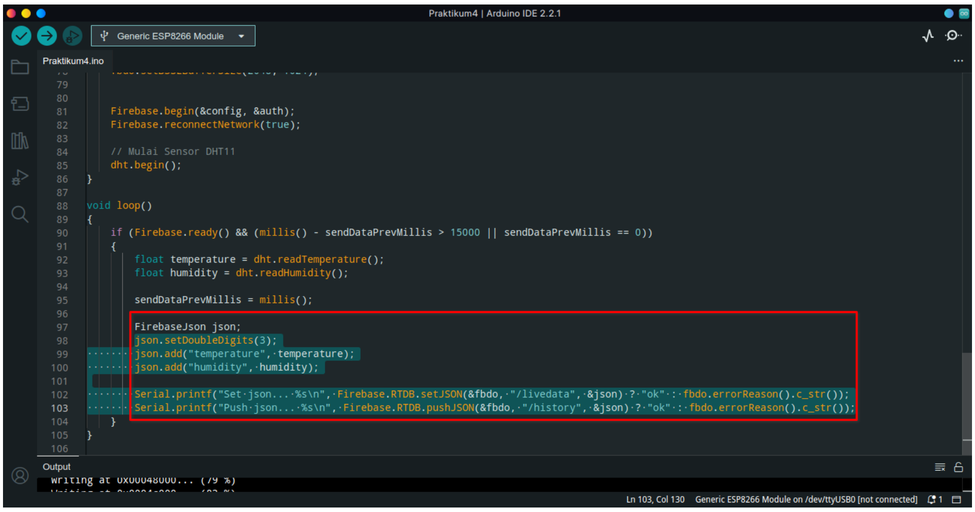
```
    config.token_status_callback = tokenStatusCallback;
    fbdo.setBSSLBufferSize(4096 );


    Firebase.begin(&config, &auth);
    Firebase.reconnectNetwork(true);

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

void loop()
{
    if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
    {
        float temperature = dht.readTemperature();
        float humidity = dht.readHumidity();

        sendDataPrevMillis = millis();

        FirebaseJson json;
        json.setDoubleDigits(3);
        json.add("value", count);

        Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo, "/test/json", &json) ? "ok" : fbdo.errorReason().c_str());
    }
}
```

```
╟── DATA      1628      initialized variables
```

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());
```

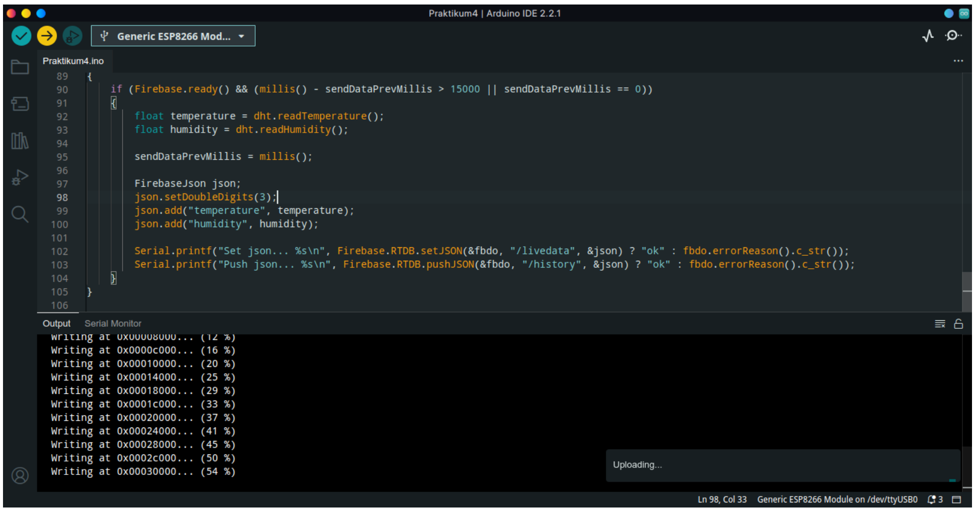


Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum4.ino
Output
Ln 103, Col 130 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```


    Firebase.begin(&config, &auth);
    Firebase.reconnectNetwork(true);

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

void loop()
{
    if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
    {
        float temperature = dht.readTemperature();
        float humidity = dht.readHumidity();

        sendDataPrevMillis = millis();

        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());
    }
}
```

```
Writing at 0x00048000... (79 %)
```

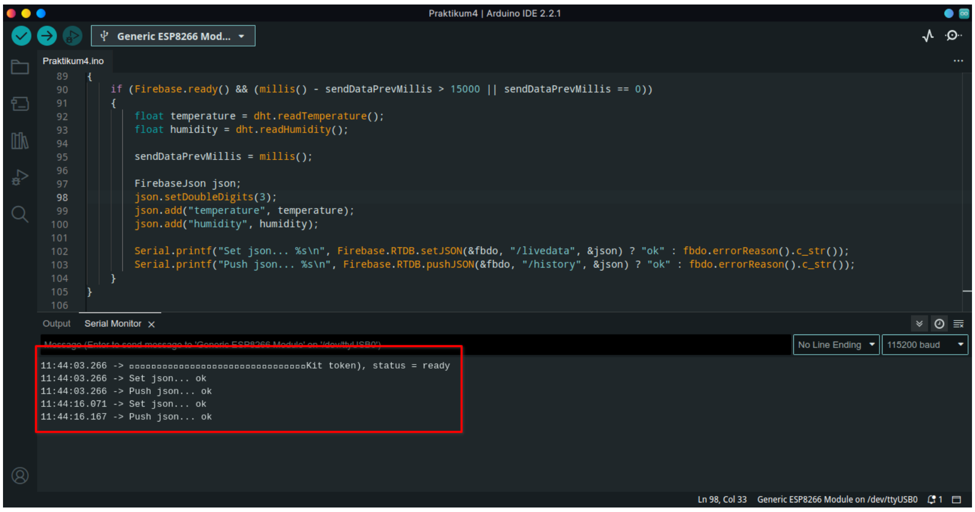
30. Verifikasi dan Upload aplikasi



```
{
    if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
    {
        float temperature = dht.readTemperature();
        float humidity = dht.readHumidity();

        sendDataPrevMillis = millis();

        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());
    }
}
```

```
Writing at 0x00008000... (12 %)
Writing at 0x0000c000... (16 %)
Writing at 0x00010000... (20 %)
Writing at 0x00014000... (25 %)
Writing at 0x00018000... (29 %)
Writing at 0x0001c000... (33 %)
Writing at 0x00020000... (37 %)
Writing at 0x00024000... (41 %)
Writing at 0x00028000... (45 %)
Writing at 0x0002c000... (50 %)
Writing at 0x00030000... (54 %)
```

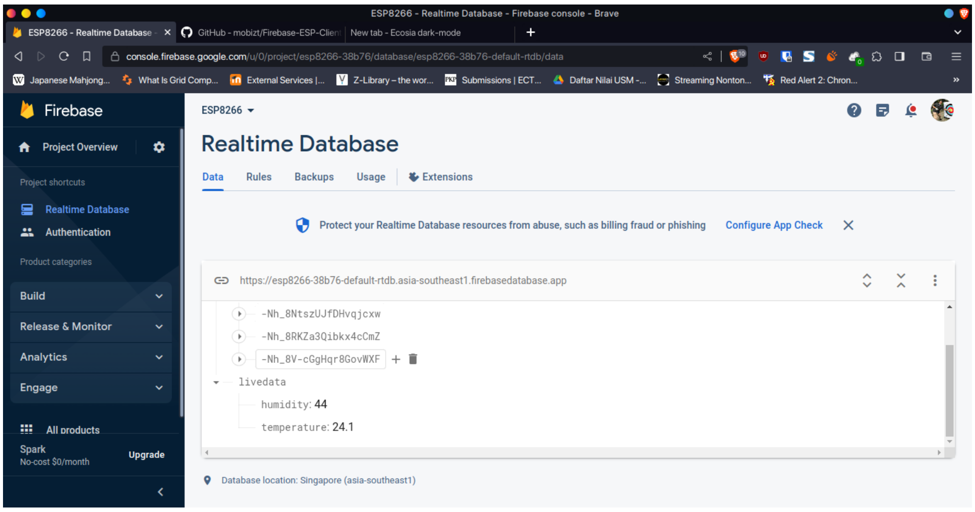
31. Data sukses diunggah


Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Praktikum4.ino
Output
Serial Monitor
Message (Enter to send message to 'Generic ESP8266 Module' on '/dev/ttyUSB0')
No Line Ending
115200 baud
Ln 98, Col 33
Generic ESP8266 Module on /dev/ttyUSB0

```
{
    if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
    {
        float temperature = dht.readTemperature();
        float humidity = dht.readHumidity();

        sendDataPrevMillis = millis();

        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());
    }
}
```

```
11:44:03.266 -> ▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫▫Kit token), status = ready
11:44:03.266 -> Set json... ok
11:44:03.266 -> Push json... ok
11:44:16.071 -> Set json... ok
11:44:16.167 -> Push json... ok
```

32. Hasil



```
https://esp8266-38b76-default-rtdb.asia-southeast1.firebasedatabase.app
    -Nh_8NtszUJfDHvqjcxw
    -Nh_8RKZa3Qibkx4cCmZ
    -Nh_8V-cGgHqr8GovWXF
livedata
    humidity: 44
    temperature: 24.1
```

Bab 5

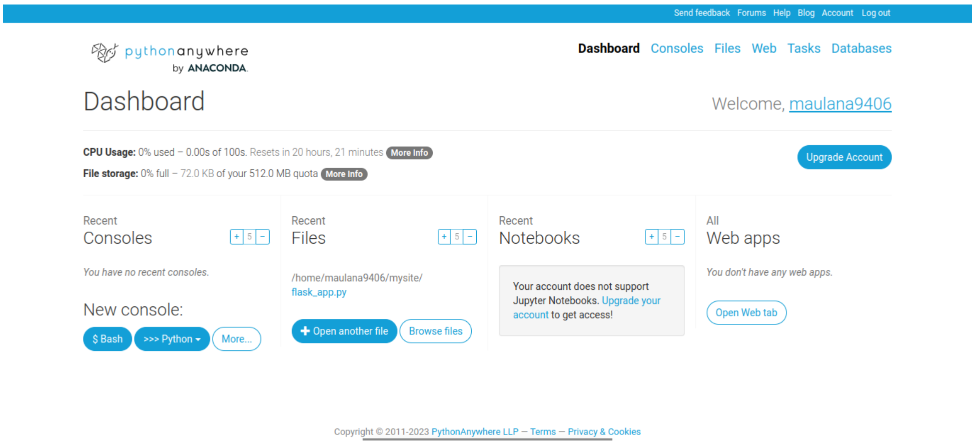
Praktikum 5

5.1 NodeMCU, DHT11, dan Web App

Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Web App sehingga dapat dipantau dan unduh secara daring secara bersamaan. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 4** yang ada di halaman sebelumnya.

5.2 Tutorial

1. Untuk memulai praktikum ini, mahasiswa diwajibkan menyelesaikan **Praktikum 4**
2. Jika sudah, buka `https://pythonanywhere.com/`. Dan buatlah satu akun di website tersebut.
3. Jika sudah buka halaman dasbor seperti gambar berikut:


Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA
Dashboard Consoles Files Web Tasks Databases
Upgrade Account
More Info
Recent Consoles
+ 5 −
New console:
$ Bash
>>> Python
More...
Recent Files
Open another file
Browse files
Recent Notebooks
All Web apps
Open Web tab
Copyright © 2011-2023 PythonAnywhere LLP — Terms — Privacy & Cookies

Dashboard

Welcome, maulana9406

CPU Usage: 0% used – 0.00s of 100s. Resets in 20 hours, 21 minutes

File storage: 0% full – 72.0 KB of your 512.0 MB quota

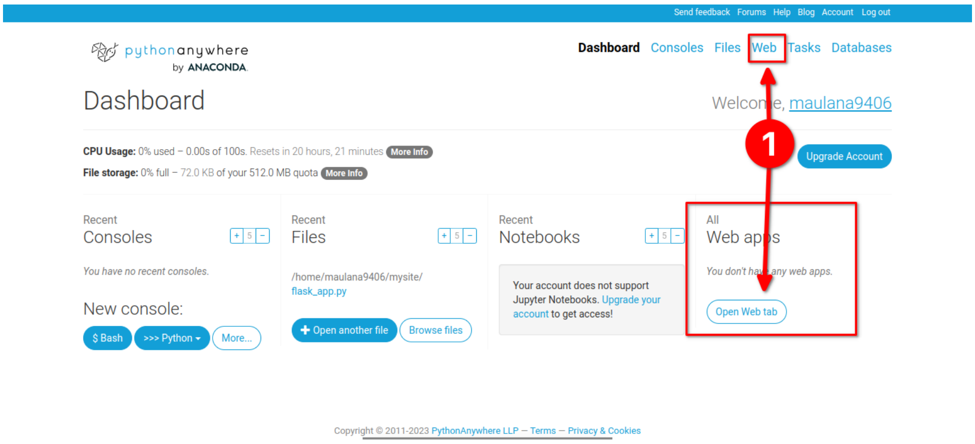
You have no recent consoles.

/home/maulana9406/mysite/flask_app.py

Your account does not support Jupyter Notebooks. Upgrade your account to get access!

You don't have any web apps.

4. Untuk memulai membuat Web apps, klik **Open Web tab**



CPU Usage: 0% used – 0.00s of 100s. Resets in 20 hours, 21 minutes More Info

File storage: 0% full – 72.0 KB of your 512.0 MB quota More Info

You have no recent consoles.

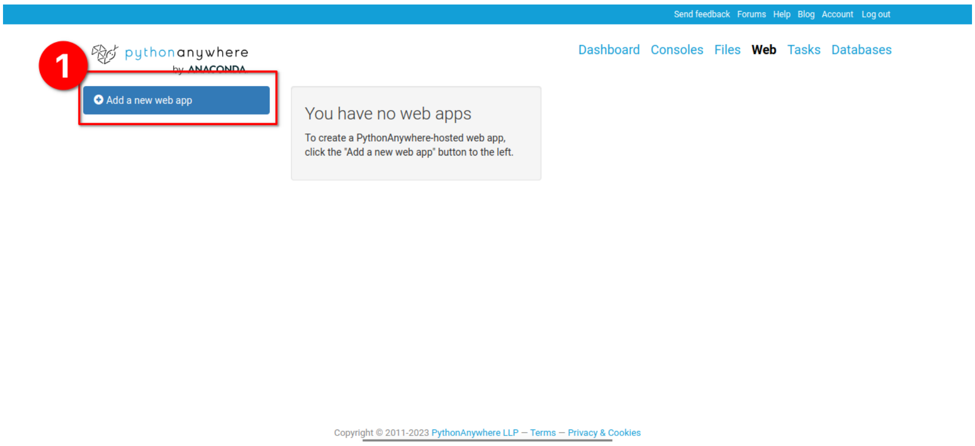
/home/maulana9406/mysite/flask_app.py

Your account does not support Jupyter Notebooks. Upgrade your account to get access!

You don't have any web apps.

Copyright © 2011-2023 PythonAnywhere LLP — Terms — Privacy & Cookies

5. Di halaman berikutnya buatlah satu **Web app** dengan klik **Add a new web app**

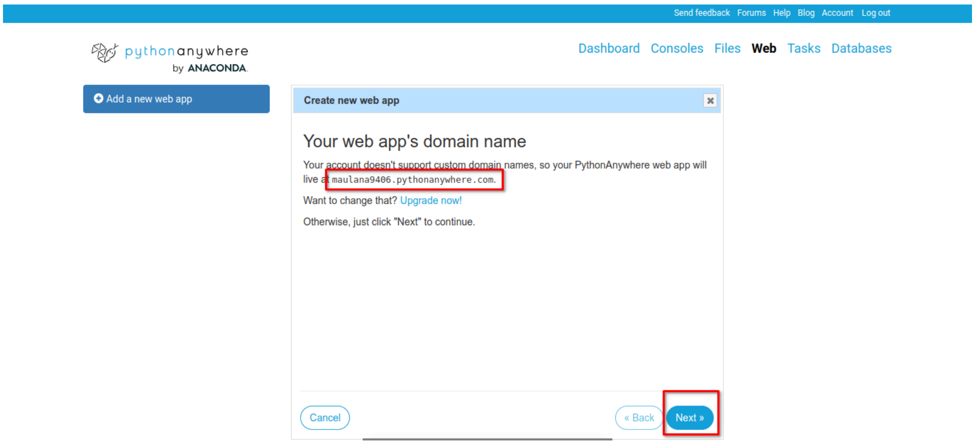

Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA.
Dashboard Consoles Files Web Tasks Databases
1
Add a new web app

You have no web apps

To create a PythonAnywhere-hosted web app, click the "Add a new web app" button to the left.

Copyright © 2011-2023 PythonAnywhere LLP — Terms — Privacy & Cookies

6. Berikutnya klik **Next** saja karena nama web akan default ke username


Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA.
Dashboard Consoles Files Web Tasks Databases
Add a new web app
Create new web app
Cancel
« Back
Next »

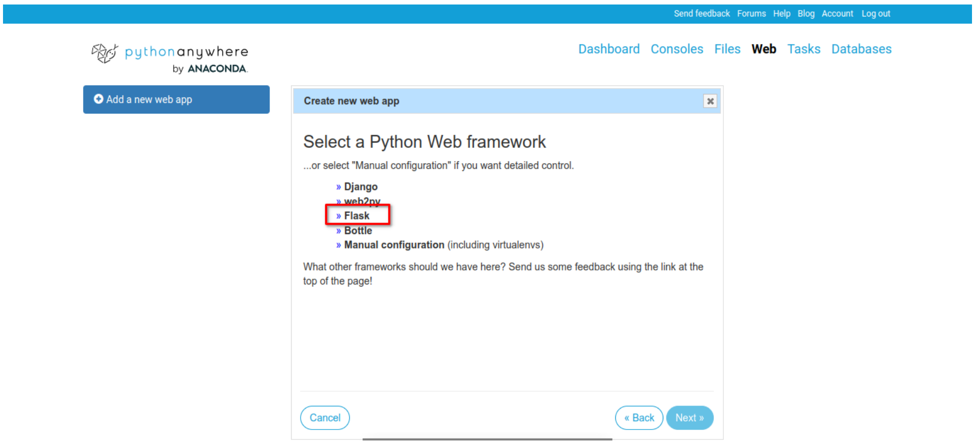
Your web app's domain name

Your account doesn't support custom domain names, so your PythonAnywhere web app will live at maulana9406.pythonanywhere.com.

Want to change that? Upgrade now!

Otherwise, just click "Next" to continue.

7. Berikutnya adalah memilih **Engine API**. Klik **Flask**



Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA
Dashboard Consoles Files Web Tasks Databases
Add a new web app
Cancel
« Back
Next »

Create new web app

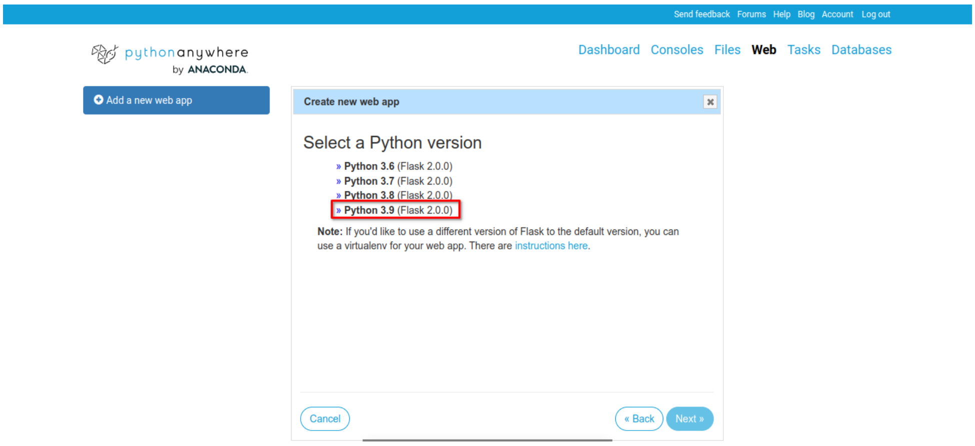
Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

- » Django
- » web2py
- » Flask
- » Bottle
- » Manual configuration (including virtualenvs)

What other frameworks should we have here? Send us some feedback using the link at the top of the page!

8. Berikutnya pilih **Python 3.9**



Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA
Dashboard Consoles Files Web Tasks Databases
Add a new web app
Cancel
« Back
Next »

Create new web app

Select a Python version

- » Python 3.6 (Flask 2.0.0)
- » Python 3.7 (Flask 2.0.0)
- » Python 3.8 (Flask 2.0.0)
- » Python 3.9 (Flask 2.0.0)

Note: If you'd like to use a different version of Flask to the default version, you can use a virtualenv for your web app. There are instructions here.

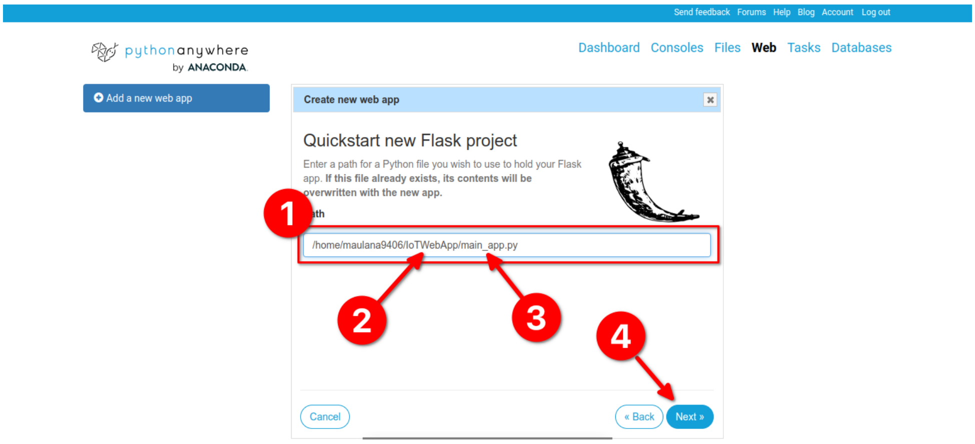
9. Ubah target direktori dari (JANGAN DIKOPI DAN TEMPEL):

Sebelum

```
/home/maulana9406/mysite/flask_app.py
```

Sesudah

```
/home/maulana9406/IoTWebApp/main_app.py
```



Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA
Dashboard Consoles Files Web Tasks Databases
Add a new web app
1
2
3
4
Cancel
« Back
Next »

Create new web app

Quickstart new Flask project

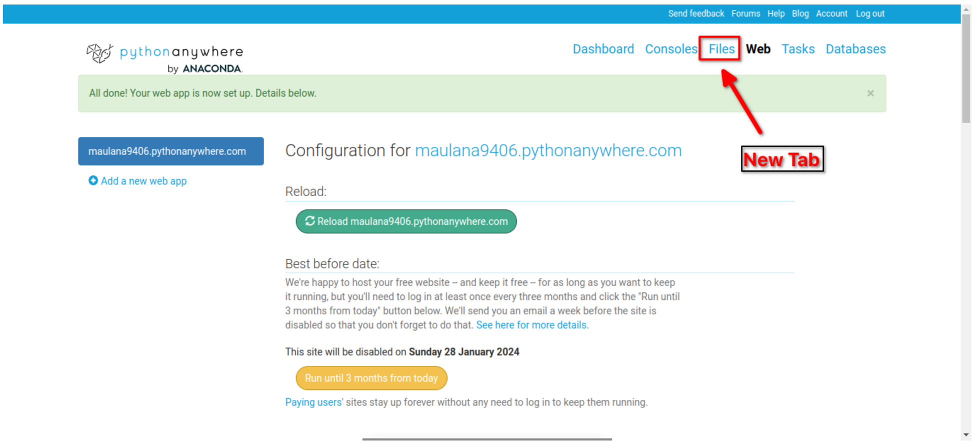
Enter a path for a Python file you wish to use to hold your Flask app. **If this file already exists, its contents will be overwritten with the new app.**

Path

/home/maulana9406/IoTWebApp/main_app.py

10. Jika sudah, website akan membawa ke **Configuration Web App**

11. Buka **Files** di **Tab Baru**


Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA
Dashboard Consoles Files Web Tasks Databases
New Tab
maulana9406.pythonanywhere.com
Add a new web app
Reload maulana9406.pythonanywhere.com
Run until 3 months from today

All done! Your web app is now set up. Details below.

Configuration for maulana9406.pythonanywhere.com

Reload:

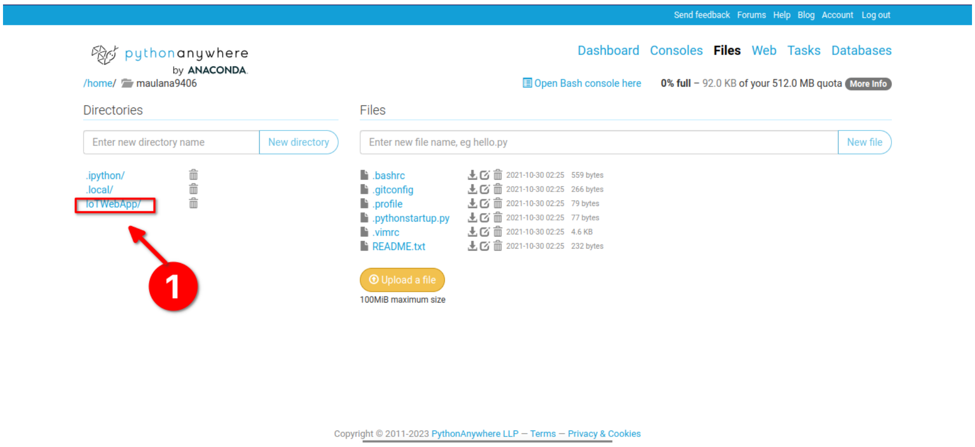
Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. See here for more details.

This site will be disabled on **Sunday 28 January 2024**

Paying users' sites stay up forever without any need to log in to keep them running.

12. Di bagian ini mahasiswa dapat melihat struktur direktori **Web App**. Buka folder **IoTWebApp** di bagian kiri


Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA
Dashboard Consoles Files Web Tasks Databases
Open Bash console here
More Info
Enter new directory name
New directory
Enter new file name, eg hello.py
New file
1
Upload a file

/home/ maulana9406

0% full – 92.0 KB of your 512.0 MB quota

Directories

.ipython/
.local/
IoTWebApp/

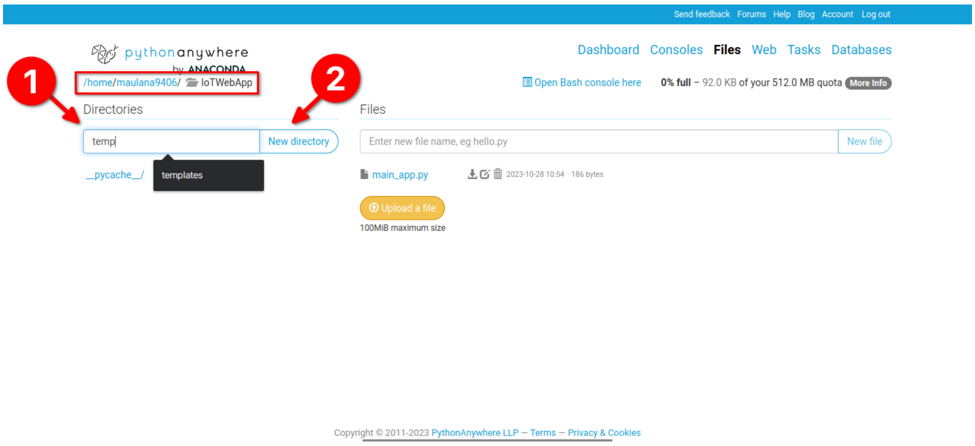
Files

```
.bashrc              2021-10-30 02:25  559 bytes
.gitconfig           2021-10-30 02:25  266 bytes
.profile             2021-10-30 02:25  79 bytes
.pythonstartup.py    2021-10-30 02:25  77 bytes
.vimrc               2021-10-30 02:25  4.6 KB
README.txt           2021-10-30 02:25  232 bytes
```

100MiB maximum size

Copyright © 2011-2023 PythonAnywhere LLP — Terms — Privacy & Cookies

13. Pastikan mahasiswa sudah membuka folder **IoTWebApp**. Jika sudah, buatlah satu folder dengan nama **templates**. Masukkan kata **templates** lalu **Enter**


Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA
Dashboard Consoles Files Web Tasks Databases
1
2
Open Bash console here
More Info
New directory
templates
Enter new file name, eg hello.py
New file
Upload a file

/home/maulana9406/ IoTWebApp

0% full – 92.0 KB of your 512.0 MB quota

Directories

temp

__pycache__/

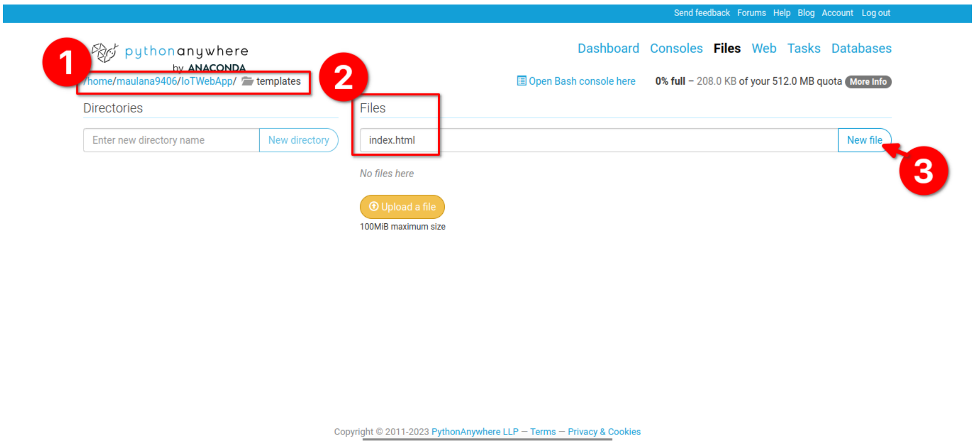
Files

```
main_app.py    2023-10-28 10:54  186 bytes
```

100MiB maximum size

Copyright © 2011-2023 PythonAnywhere LLP — Terms — Privacy & Cookies

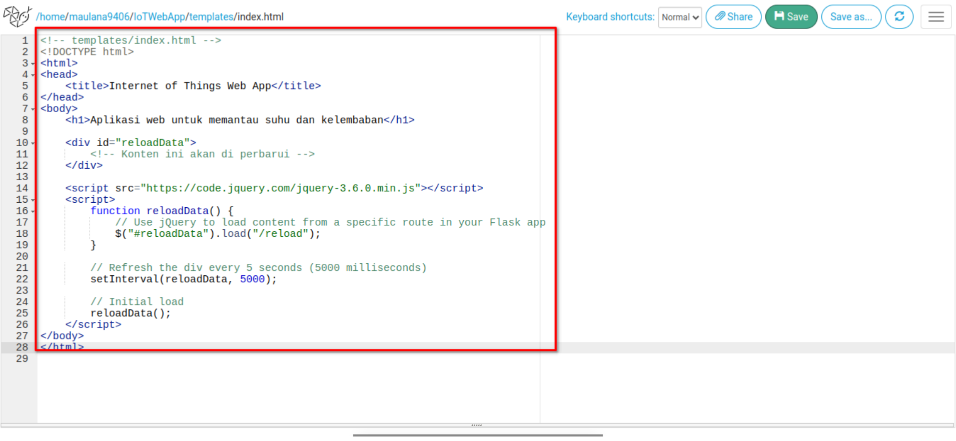
14. Jika folder sudah di buat. Berikutnya adalah membuat file dengan nama **index.html** di bagian kanan.


Send feedback Forums Help Blog Account Log out
pythonanywhere by ANACONDA
Dashboard Consoles Files Web Tasks Databases
1
/home/maulana9406/IoTWebApp/ templates
2
Open Bash console here
0% full – 208.0 KB of your 512.0 MB quota More Info
Directories
Enter new directory name
New directory
Files
index.html
New file
3
No files here
Upload a file
100MiB maximum size
Copyright © 2011-2023 PythonAnywhere LLP — Terms — Privacy & Cookies

15. Jika sudah, buka file **index.html**, masukkan kode berikut, dan simpan

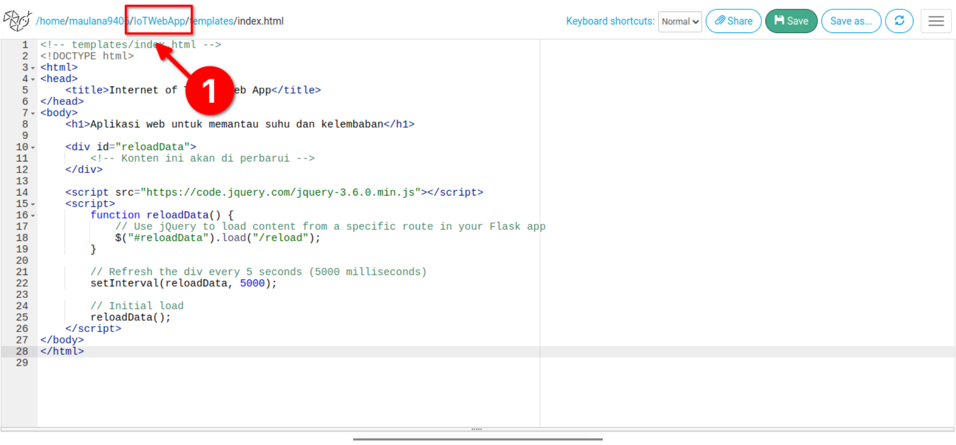
Sesudah

```
<!-- templates/index.html -->
<!DOCTYPE html>
<html>
    <head>
        <title>Internet of Things Web App</title>
    </head>
    <body>
        <h1>Aplikasi web untuk memantau suhu dan kelembaban</h1>

        <div id=”reloadData” style=”border: 2px solid #000;
            outline: 2px solid #f00; padding: 20px;”>
            <!-- Konten ini akan di perbarui -->
        </div>

        <script src=”https://code.jquery.com/jquery-3.6.0.min.js”></script>
        <script>
            function reloadData() {
                    $(”#reloadData”).load(”/reload”);
            }

            function openLink() {
                    var urlToOpen = '/download';
                    window.open(urlToOpen, '_blank');
            }

            // Refresh the div every 5 seconds (5000 milliseconds)
            setInterval(reloadData, 5000);

            // Initial load
            reloadData();
        </script>
        <div>
            <button onclick=”openLink()”>Unduh Data</button>
        </div>
    </body>
</html>
```



```
<!-- templates/index.html -->
<!DOCTYPE html>
<html>
<head>
    <title>Internet of Things Web App</title>
</head>
<body>
    <h1>Aplikasi web untuk memantau suhu dan kelembaban</h1>

    <div id="reloadData">
        <!-- Konten ini akan di perbarui -->
    </div>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script>
        function reloadData() {
            // Use jQuery to load content from a specific route in your Flask app
            $("#reloadData").load("/reload");
        }

        // Refresh the div every 5 seconds (5000 milliseconds)
        setInterval(reloadData, 5000);

        // Initial load
        reloadData();
    </script>
</body>
</html>
```

16. Kembali ke folder atas dengan klik **IoTWebApp** di bagian atas. Lalu buka file **main_app.py**



```
<!-- templates/inde html -->
<!DOCTYPE html>
<html>
<head>
    <title>Internet of        eb App</title>
</head>
<body>
    <h1>Aplikasi web untuk memantau suhu dan kelembaban</h1>

    <div id="reloadData">
        <!-- Konten ini akan di perbarui -->
    </div>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script>
        function reloadData() {
            // Use jQuery to load content from a specific route in your Flask app
            $("#reloadData").load("/reload");
        }

        // Refresh the div every 5 seconds (5000 milliseconds)
        setInterval(reloadData, 5000);

        // Initial load
        reloadData();
    </script>
</body>
</html>
```

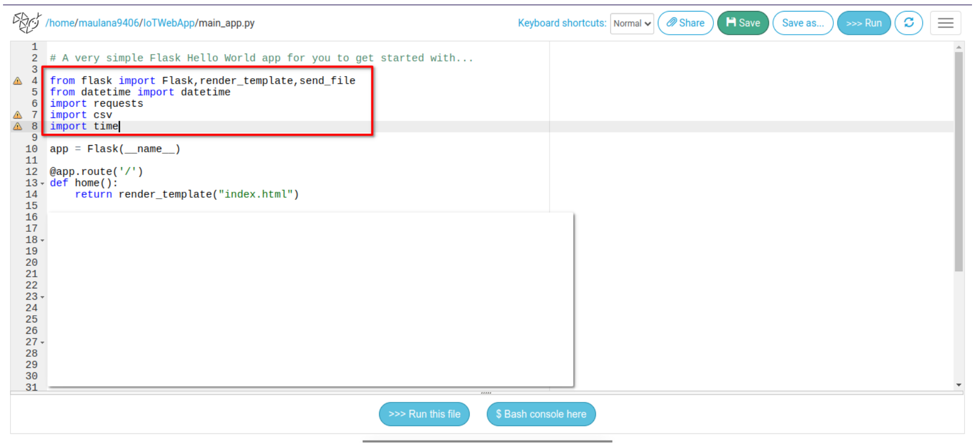
17. Di dalam file **main_app.py** ini. Di bagian paling atas ada perubahan kode seperti berikut:

Sebelum

```
from flask import Flask
```

Sesudah

```
from flask import Flask,render_template, send_file
from datetime import datetime
import requests
import csv
import time
```


/home/maulana9406/IoTWebApp/main_app.py
Keyboard shortcuts: Normal
Share
Save
Save as...
>>> Run
>>> Run this file
$ Bash console here

```
# A very simple Flask Hello World app for you to get started with...

from flask import Flask,render_template,send_file
from datetime import datetime
import requests
import csv
import time

app = Flask(__name__)

@app.route('/')
def home():
    return render_template("index.html")
```

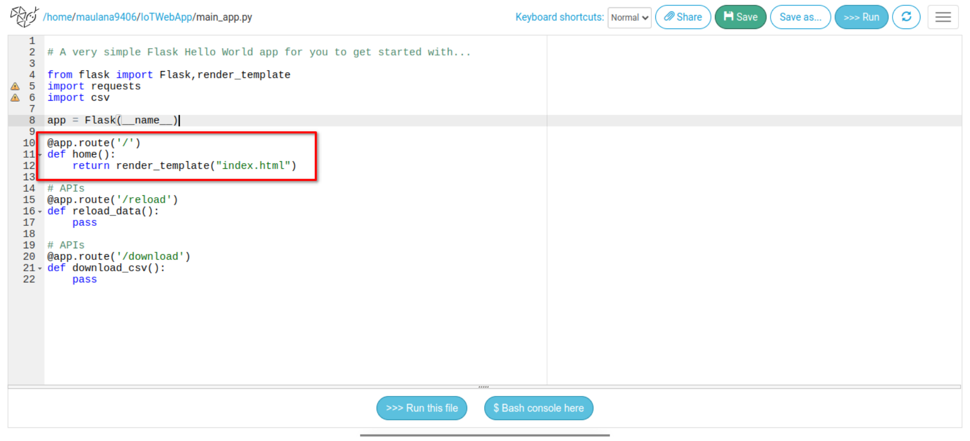
18. Lalu ubah kode untuk akses **Home**. Perhatikan perubahan kode berikut:

Sebelum

```
@app.route('/')
def hello_world():
    return 'Hello from Flask!'
```

Sesudah

```
@app.route('/')
def home():
    return render_template(”index.html”)
```

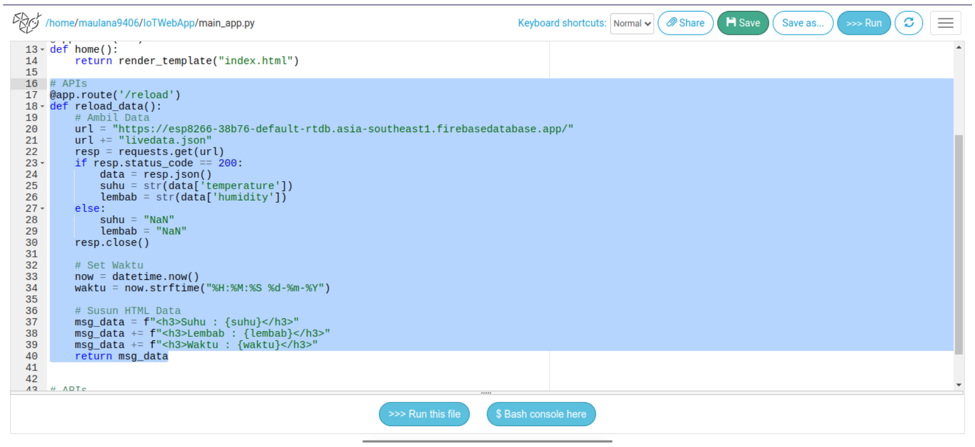

/home/maulana9406/IoTWebApp/main_app.py
Keyboard shortcuts: Normal
Share
Save
Save as...
>>> Run
>>> Run this file
$ Bash console here

```
# A very simple Flask Hello World app for you to get started with...

from flask import Flask,render_template
import requests
import csv

app = Flask(__name__)

@app.route('/')
def home():
    return render_template("index.html")

# APIs
@app.route('/reload')
def reload_data():
    pass

# APIs
@app.route('/download')
def download_csv():
    pass
```

19. Setelah itu di baris bawah lagi, tambahkan kode untuk melakukan **reloading** untuk menampilkan data dari database dan mengunduh file. GANTI <URL> dengan Database masing-masing

Sesudah

```
# APIs
@app.route('/reload')
def reload_data():
    # Ambil Data
    url = "<URL REALTIME DATABASE>"
    url += "livedata.json"
    resp = requests.get(url)
    if resp.status_code == 200:
        data = resp.json()
        suhu = str(data['temperature'])
        lembab = str(data['humidity'])
    else:
        suhu = "NaN"
        lembab = "NaN"
    resp.close()

    # Set Waktu
    now = datetime.now()
    waktu = now.strftime("%H:%M:%S %d-%m-%Y")

    # Susun HTML Data
    msg_data = f"<h3>Suhu : {suhu}</h3>"
    msg_data += f"<h3>Lembab : {lembab}</h3>"
    msg_data += f"<h3>Waktu : {waktu}</h3>"
    return msg_data
```

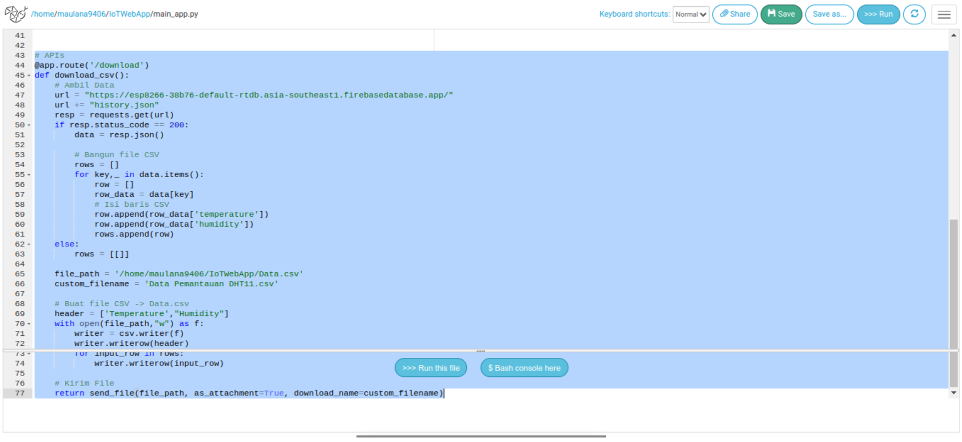

/home/maulana9406/IoTWebApp/main_app.py
Keyboard shortcuts: Normal
Share
Save
Save as...
>>> Run
>>> Run this file
$ Bash console here

```
def home():
    return render_template("index.html")

# APIs
@app.route('/reload')
def reload_data():
    # Ambil Data
    url = "https://esp8266-38b76-default-rtdb.asia-southeast1.firebasedatabase.app/"
    url += "livedata.json"
    resp = requests.get(url)
    if resp.status_code == 200:
        data = resp.json()
        suhu = str(data['temperature'])
        lembab = str(data['humidity'])
    else:
        suhu = "NaN"
        lembab = "NaN"
    resp.close()

    # Set Waktu
    now = datetime.now()
    waktu = now.strftime("%H:%M:%S %d-%m-%Y")

    # Susun HTML Data
    msg_data = f"<h3>Suhu : {suhu}</h3>"
    msg_data += f"<h3>Lembab : {lembab}</h3>"
    msg_data += f"<h3>Waktu : {waktu}</h3>"
    return msg_data


# APIs
```

20. Terakhir adalah membuat kode unduh. Tambahkan kode berikut tepat di bawah kode reload. GANTI <LINK URL> dan <USERNAME> sesuai masing-masing

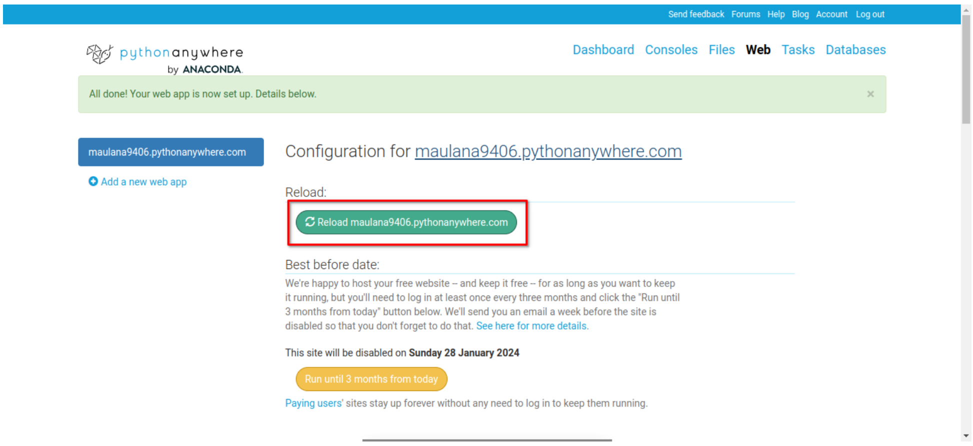
Sesudah

```
# APIs
@app.route('/download')
def download_csv():
    # Ambil Data
    url = ”<URL REALTIME DATABASE>”
    url += ”history.json”
    resp = requests.get(url)
    if resp.status_code == 200:
        data = resp.json()

        # Bangun file CSV
        rows = []
        for key,_ in data.items():
            row = []
            row_data = data[key]
            # Isi baris CSV
            row.append(row_data['temperature'])
            row.append(row_data['humidity'])
            rows.append(row)
    else:
        rows = [[]]

    file_path = '/home/<USERNAME>/IoTWebApp/Data.csv'
    custom_filename = 'Data Pemantauan DHT11.csv'

    # Buat file CSV -> Data.csv
    header = ['Temperature',”Humidity”]
    with open(file_path,”w”) as f:
        writer = csv.writer(f)
        writer.writerow(header)
        for input_row in rows:
            writer.writerow(input_row)

    # Kirim File
    return send_file(file_path, as_attachment=True,
        download_name=custom_filename)
```


/home/maulana9406/IoTWebApp/main_app.py
Keyboard shortcuts: Normal
Share
Save
Save as...
>>> Run
>>> Run this file
$ Bash console here

```python


# APIs
@app.route('/download')
def download_csv():
    # Ambil Data
    url = "https://esp8266-38b76-default-rtdb.asia-southeast1.firebasedatabase.app/"
    url += "history.json"
    resp = requests.get(url)
    if resp.status_code == 200:
        data = resp.json()

        # Bangun file CSV
        rows = []
        for key,_ in data.items():
            row = []
            row_data = data[key]
            # Isi baris CSV
            row.append(row_data['temperature'])
            row.append(row_data['humidity'])
            rows.append(row)
    else:
        rows = [[]]

    file_path = '/home/maulana9406/IoTWebApp/Data.csv'
    custom_filename = 'Data Pemantauan DHT11.csv'

    # Buat file CSV -> Data.csv
    header = ['Temperature',"Humidity"]
    with open(file_path,"w") as f:
        writer = csv.writer(f)
        writer.writerow(header)
        for input_row in rows:
            writer.writerow(input_row)

    # Kirim File
    return send_file(file_path, as_attachment=True, download_name=custom_filename)
```

21. Langkah terakhir, melakukan **Reloading Web App** dengan kembali ke **Tab Web** dengan tampilan di **Langkah 11**. Tunggu hingga selesai


Send feedback
Forums
Help
Blog
Account
Log out
pythonanywhere by ANACONDA
Dashboard
Consoles
Files
Web
Tasks
Databases
maulana9406.pythonanywhere.com
Add a new web app
Reload maulana9406.pythonanywhere.com
Run until 3 months from today

All done! Your web app is now set up. Details below.

Configuration for maulana9406.pythonanywhere.com

Reload:

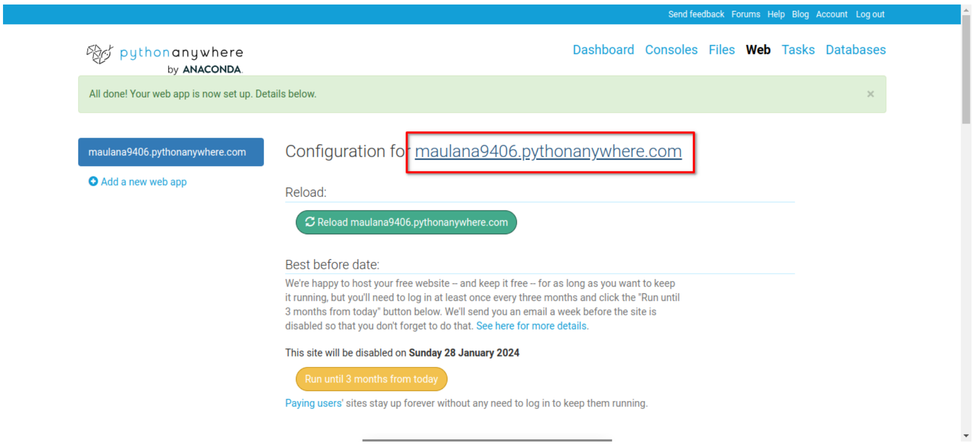
Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. See here for more details.

This site will be disabled on **Sunday 28 January 2024**

Paying users' sites stay up forever without any need to log in to keep them running.

22. Klik nama website untuk membuak Web


Send feedback
Forums
Help
Blog
Account
Log out
pythonanywhere by ANACONDA
Dashboard
Consoles
Files
Web
Tasks
Databases
maulana9406.pythonanywhere.com
Add a new web app
Reload maulana9406.pythonanywhere.com
Run until 3 months from today

All done! Your web app is now set up. Details below.

Configuration for maulana9406.pythonanywhere.com

Reload:

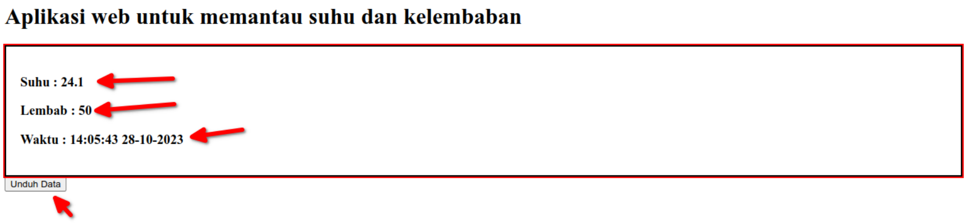
Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. See here for more details.

This site will be disabled on **Sunday 28 January 2024**

Paying users' sites stay up forever without any need to log in to keep them running.

23. Lihat dan coba Web App



Aplikasi web untuk memantau suhu dan kelembaban

Suhu : 24.1

Lembab : 50

Waktu : 14:05:43 28-10-2023

24. Klik Unduh Data untuk mengambil data CSV

Bab 6

Praktikum 6

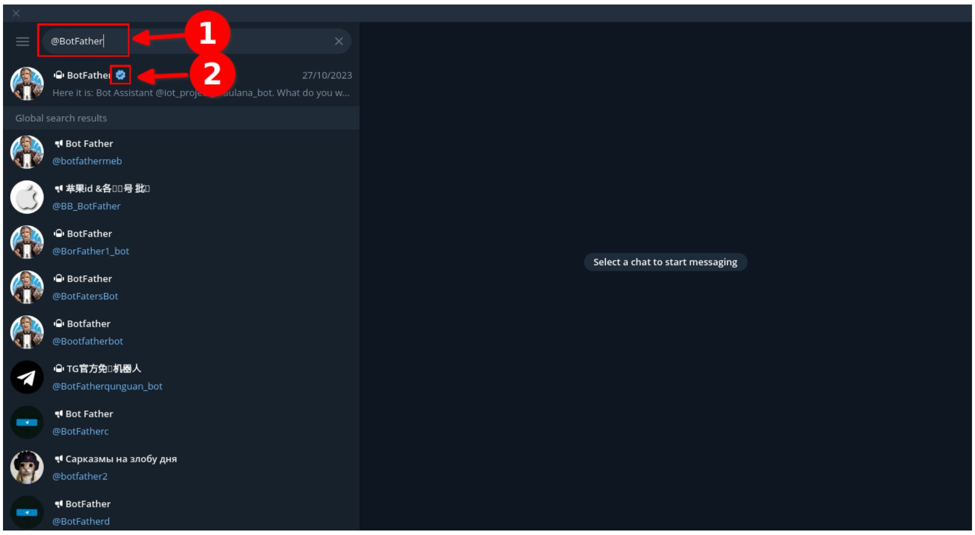
6.1 ESP8266, DHT11, dan Telegram Bot

Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Telegram Bot. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 5** yang ada di halaman sebelumnya.

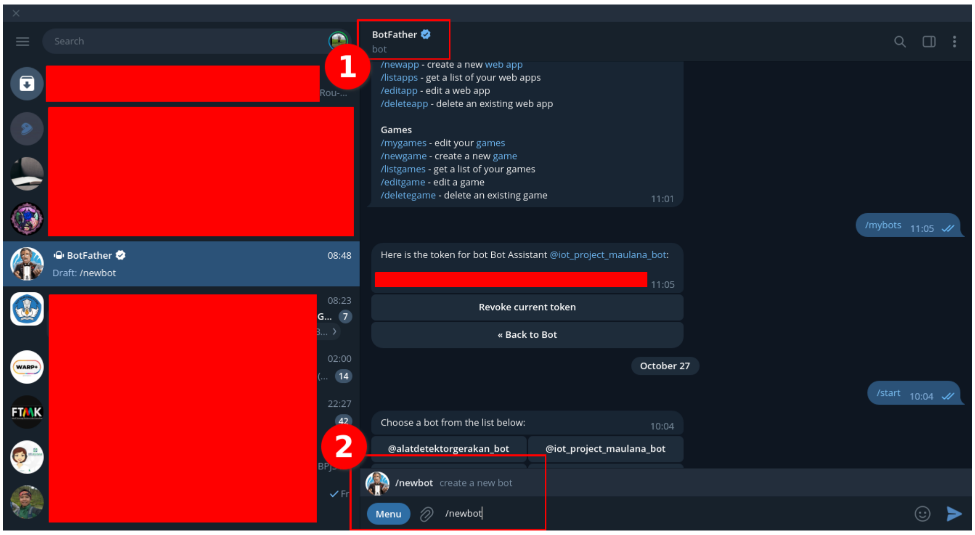
6.2 Tutorial

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



[redacted]

[redacted]

BotFather Draft: /newbot 08:48

[redacted]

/newapp - create a new web app
/listapps - get a list of your web apps
/editapp - edit a web app
/deleteapp - delete an existing web app

Games
/mygames - edit your games
/newgame - create a new game
/listgames - get a list of your games
/editgame - edit a game
/deletegame - delete an existing game 11:01

/mybots 11:05

Here is the token for bot Bot Assistant @iot_project_maulana_bot:
[redacted] 11:05

October 27

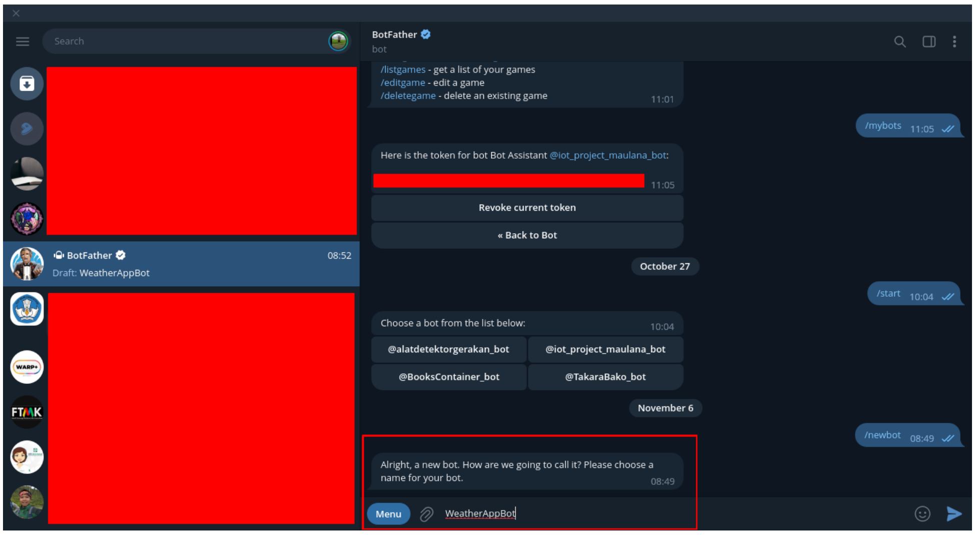
/start 10:04

Choose a bot from the list below: 10:04
@alatdetektorgerakan_bot @iot_project_maulana_bot

/newbot create a new bot

/newbot

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



[redacted]

BotFather Draft: WeatherAppBot 08:52

[redacted]

/listgames - get a list of your games
/editgame - edit a game
/deletegame - delete an existing game 11:01

/mybots 11:05

Here is the token for bot Bot Assistant @iot_project_maulana_bot:
[redacted] 11:05

October 27

/start 10:04

Choose a bot from the list below: 10:04
@alatdetektorgerakan_bot @iot_project_maulana_bot
@BooksContainer_bot @TakaraBako_bot

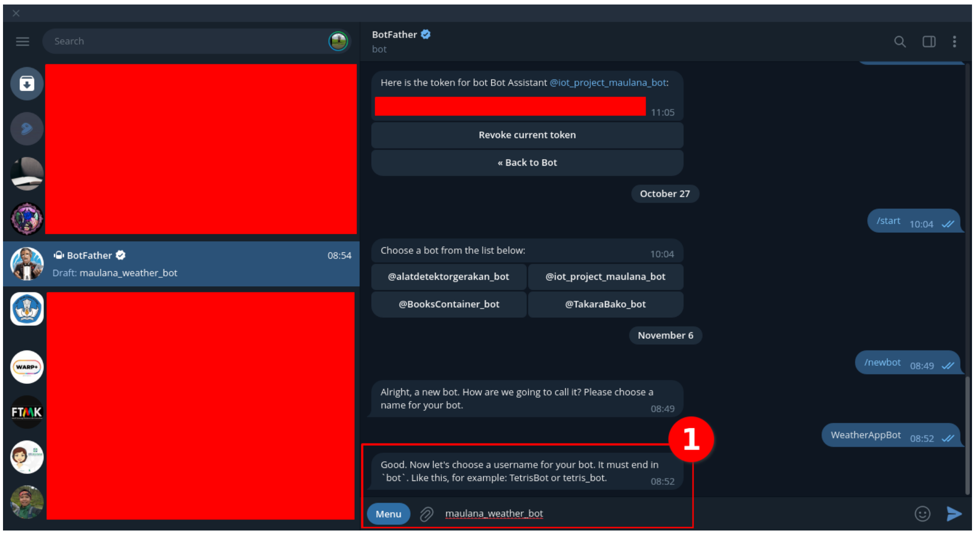
November 6

/newbot 08:49

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 08:49

WeatherAppBot

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



[redacted]

BotFather Draft: maulana_weather_bot 08:54

[redacted]

Here is the token for bot Bot Assistant @iot_project_maulana_bot:
[redacted] 11:05

October 27

/start 10:04

Choose a bot from the list below: 10:04
@alatdetektorgerakan_bot @iot_project_maulana_bot
@BooksContainer_bot @TakaraBako_bot

November 6

/newbot 08:49

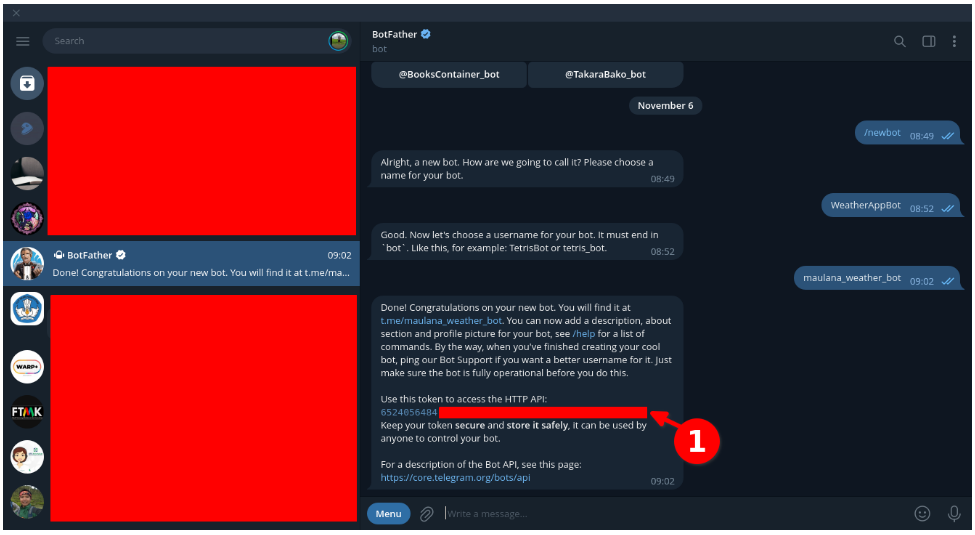
Alright, a new bot. How are we going to call it? Please choose a name for your bot. 08:49

WeatherAppBot 08:52

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 08:52

maulana_weather_bot

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



[redacted]

BotFather 09:02
Done! Congratulations on your new bot. You will find it at t.me/ma...

[redacted]

November 6

/newbot 08:49

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 08:49

WeatherAppBot 08:52

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 08:52

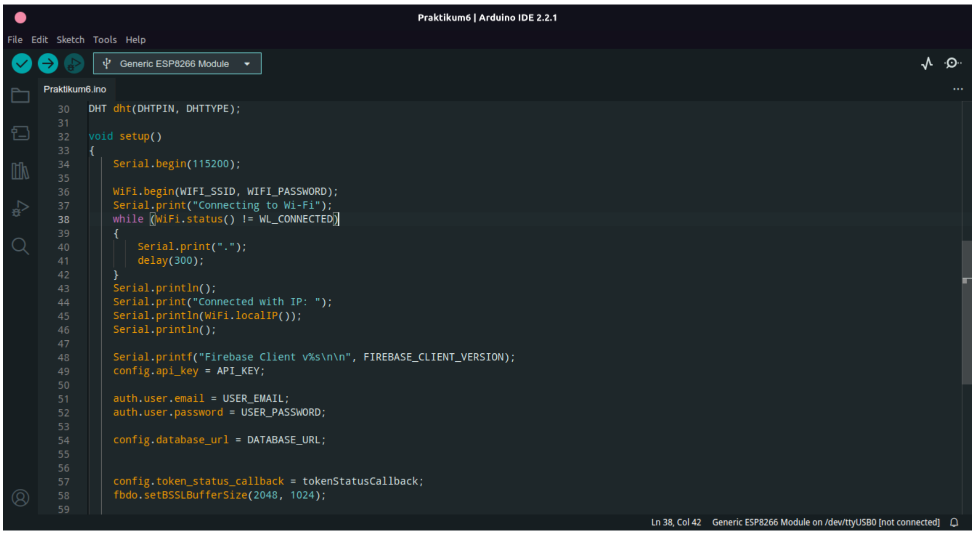
maulana_weather_bot 09:02

Done! Congratulations on your new bot. You will find it at t.me/maulana_weather_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
6524056484[redacted]
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:
https://core.telegram.org/bots/api 09:02

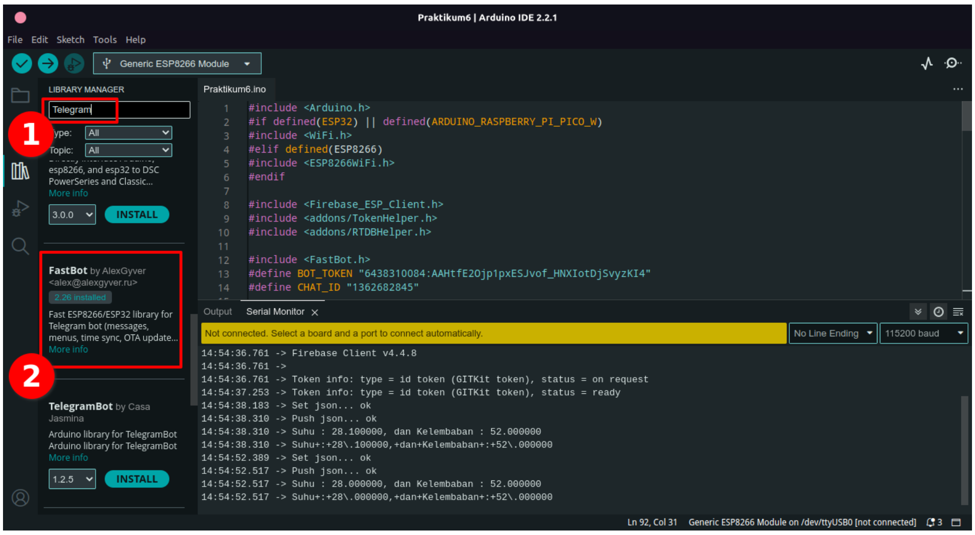
7. Berikutnya adalah membuka kembali **Praktikum 4** dengan menggunakan **Arduino IDE**. Lakukan **Save As** untuk menyimpan sebagai **Praktikum 6**



```
DHT dht(DHTPIN, DHTTYPE);

void setup()
{
    Serial.begin(115200);

    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Connecting to Wi-Fi");
    while (WiFi.status() != WL_CONNECTED)
    {
        Serial.print(".");
        delay(300);
    }
    Serial.println();
    Serial.print("Connected with IP: ");
    Serial.println(WiFi.localIP());
    Serial.println();

    Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);
    config.api_key = API_KEY;

    auth.user.email = USER_EMAIL;
    auth.user.password = USER_PASSWORD;

    config.database_url = DATABASE_URL;


    config.token_status_callback = tokenStatusCallback;
    fbdo.setBSSLBufferSize(2048, 1024);

```

Ln 38, Col 42 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

8. Install Library dengan nama **FastBot**


Praktikum6 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Generic ESP8266 Module
LIBRARY MANAGER
Telegram
Type: All
Topic: All
1
2
Praktikum6.ino
Output
Serial Monitor
No Line Ending
115200 baud
Ln 92, Col 31 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

esp8266, and esp32 to DSC PowerSeries and Classic...
More info
3.0.0 INSTALL

FastBot by AlexGyver <alex@alexgyver.ru>
2.26 installed
Fast ESP8266/ESP32 library for Telegram bot (messages, menus, time sync, OTA update...
More info

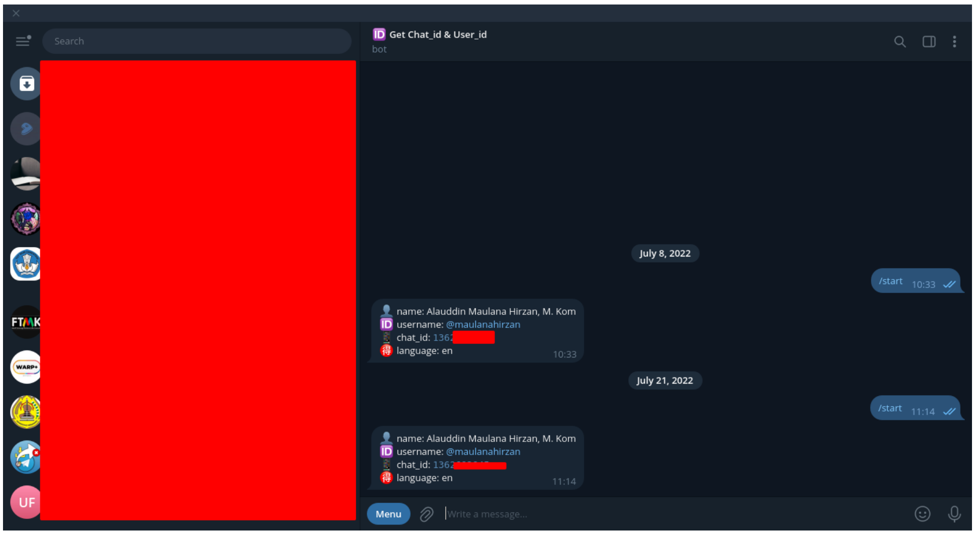
TelegramBot by Casa Jasmina
Arduino library for TelegramBot Arduino library for TelegramBot
More info
1.2.5 INSTALL

```
#include <Arduino.h>
#if defined(ESP32) || defined(ARDUINO_RASPBERRY_PI_PICO_W)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif

#include <Firebase_ESP_Client.h>
#include <addons/TokenHelper.h>
#include <addons/RTDBHelper.h>

#include <FastBot.h>
#define BOT_TOKEN "6438310084:AAHtfE2Ojp1pxESJvof_HNXIotDjSvyzKI4"
#define CHAT_ID "1362682845"
```

Not connected. Select a board and a port to connect automatically.

```
14:54:36.761 -> Firebase Client v4.4.8
14:54:36.761 ->
14:54:36.761 -> 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
```

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



[redacted]

July 8, 2022

/start 10:33

name: Alauddin Maulana Hirzan, M. Kom
username: @maulanahirzan
chat_id: 1362[redacted]
language: en
10:33

July 21, 2022

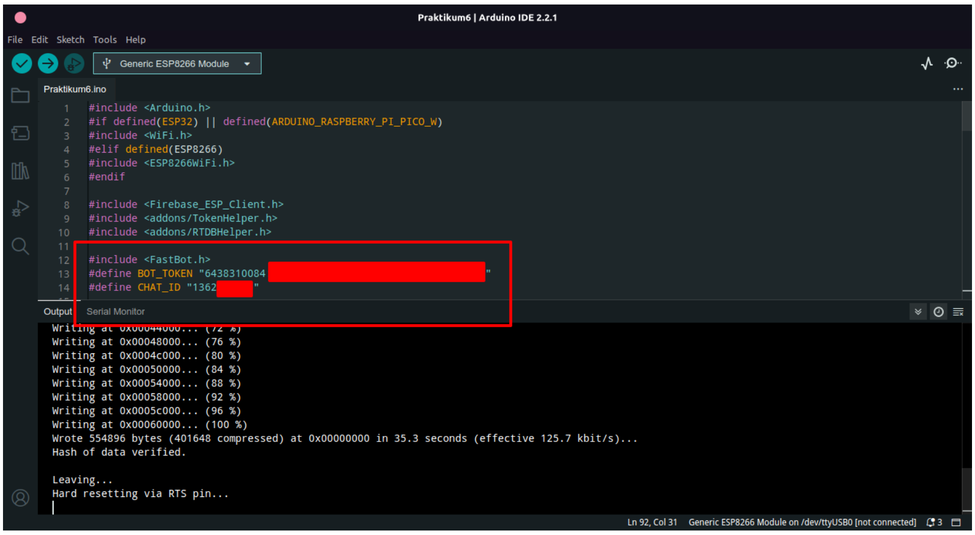
/start 11:14

name: Alauddin Maulana Hirzan, M. Kom
username: @maulanahirzan
chat_id: 1362[redacted]
language: en
11:14

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>"
```


Praktikum6 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Generic ESP8266 Module
Praktikum6.ino
Output Serial Monitor
Ln 92, Col 31 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

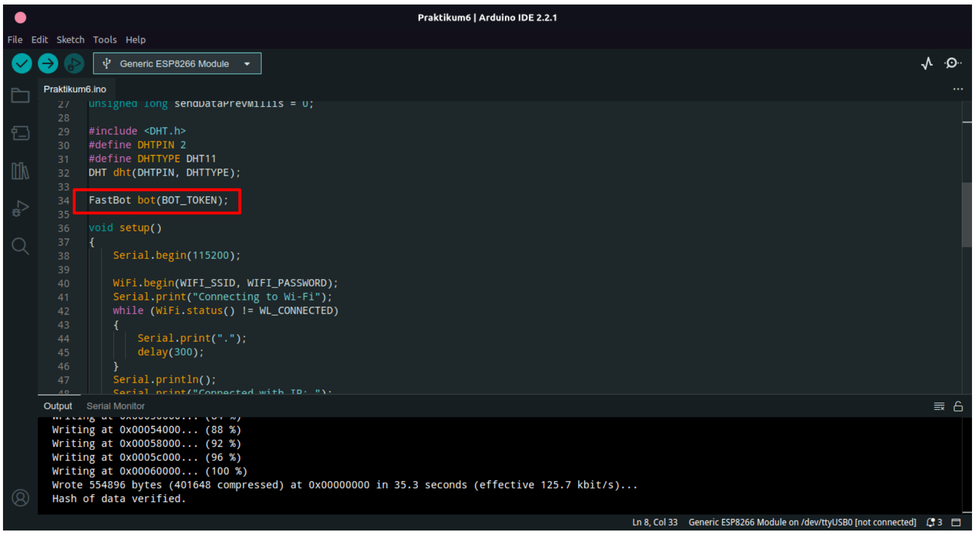
```
#include <Arduino.h>
#if defined(ESP32) || defined(ARDUINO_RASPBERRY_PI_PICO_W)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif

#include <Firebase_ESP_Client.h>
#include <addons/TokenHelper.h>
#include <addons/RTDBHelper.h>

#include <FastBot.h>
#define BOT_TOKEN "6438310084[redacted]"
#define CHAT_ID "1362[redacted]"
```

```
Writing at 0x00044000... (72 %)
Writing at 0x00048000... (76 %)
Writing at 0x0004c000... (80 %)
Writing at 0x00050000... (84 %)
Writing at 0x00054000... (88 %)
Writing at 0x00058000... (92 %)
Writing at 0x0005c000... (96 %)
Writing at 0x00060000... (100 %)
Wrote 554896 bytes (401648 compressed) at 0x00000000 in 35.3 seconds (effective 125.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

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

Sesudah

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


Praktikum6 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Generic ESP8266 Module
Praktikum6.ino
Output Serial Monitor
Ln 8, Col 33 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

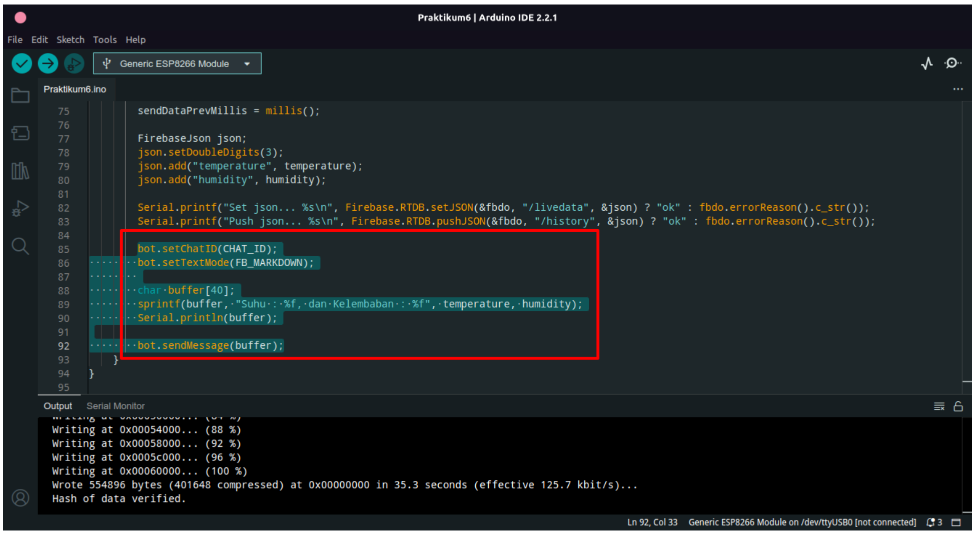
```
unsigned long sendDataPrevMillis = 0;

#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

FastBot bot(BOT_TOKEN);

void setup()
{
    Serial.begin(115200);

    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Connecting to Wi-Fi");
    while (WiFi.status() != WL_CONNECTED)
    {
        Serial.print(".");
        delay(300);
    }
    Serial.println();
    Serial.print("Connected with IP: ");
```

```
Writing at 0x00050000... (84 %)
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.
```

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

Sesudah

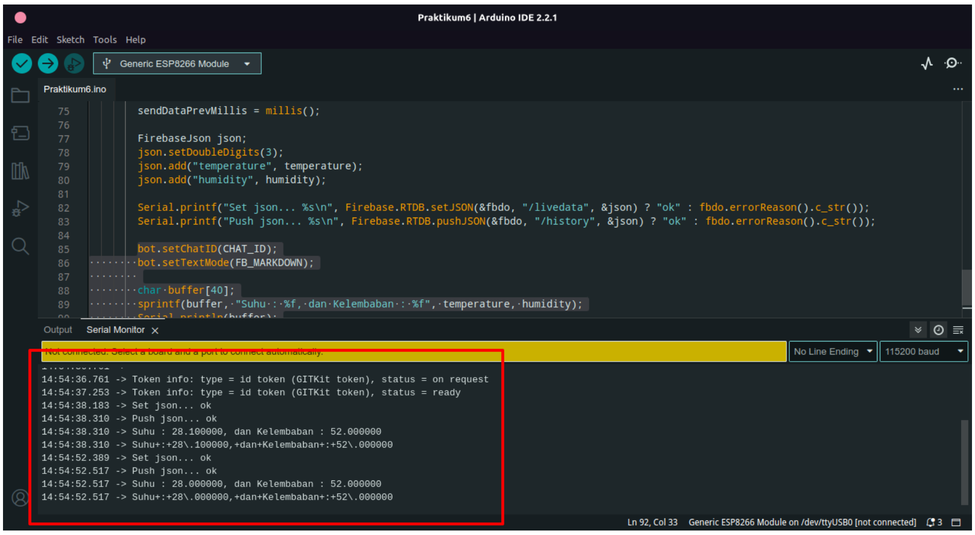
```
bot.setChatID(CHAT_ID);
bot.setTextMode(FB_MARKDOWN);

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

bot.sendMessage(buffer);
```


Praktikum6 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Generic ESP8266 Module
Praktikum6.ino
Output
Serial Monitor
Ln 92, Col 33
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]
3

```
        sendDataPrevMillis = millis();

        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());

        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);
    }
}
```

```
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 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Generic ESP8266 Module
Praktikum6.ino
Output
Serial Monitor
No Line Ending
115200 baud
Ln 92, Col 33
Generic ESP8266 Module on /dev/ttyUSB0 [not connected]
3

```
        sendDataPrevMillis = millis();

        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());

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

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

Not connected. Select a board and a port to connect automatically.

```
14:54:36.761 -> 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 7

Praktikum 7

7.1 Observasi dengan Internet of Things

Di bagian ini mahasiswa diajarkan bagaimana melakukan pengambilan data lingkungan dengan menggunakan Internet of Things. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 6** yang ada di halaman sebelumnya.

7.2 Tutorial

1. Mahasiswa perlu menyiapkan perlengkapan berupa:
   - Perangkat dari Praktikum 6 yang sudah dilengkapi dengan Firebase dan Telegram Bot
   - Charger HP dan Kabel MicroUSB
   - Akses Internet
2. Pastikan Akses Poin sudah sesuai dengan kode perangkat Internet of Things
3. Jika semua sudah berjalan dengan baik, Telegram Bot akan mengirimkan data dan Firebase Realtime DB akan merekam semua data.
4. Setelah satu jam, data yang terkumpul dapat diunduh melalui Web App.
5. Kirim data **CSV** ke **Praktikum 7**
6. Buat laporan sesuai dengan template yang ada di berikutnya dan kirim ke **Praktikum 8**

Bab 8

Praktikum 8

- Laporan hasil mengikuti 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 (Berupa Tabel Sampel Data - 15 baris data)
  6. Hasil Observasi #2 (Berupa Grafik masing-masing data, Suhu dan Kelembaban diurutkan berdasarkan waktunya)
  7. Analisis Hasil Observasi (Jelaskan hasil observasi yang didapatkan)
  8. Kesimpulan
- Laporan dikirimkan ke Praktikum 8
- Format File hanya **PDF**