



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 <i>Internet of Things</i>	2
0.2 Perangkat Board IoT	2
Persiapan Praktikum	6
0.3 Perangkat Keras	6
0.4 Perangkat Lunak	6
1 Praktikum 1	7
1.1 Konfigurasi Arduino IDE dan ESP8266	7
1.2 Tutorial	7
2 Praktikum 2	14
2.1 ESP8266, 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 ESP8266, DHT11, dan Telegram Bot	53
5.2 Tutorial	53
6 Praktikum 6	59
6.1 ESP8266, DHT11, dan Linear Regression	59
6.2 Tutorial	59
7 Praktikum 7	70
7.1 ESP8266, DHT11, dan Fuzzy Logic	70
7.2 Tutorial	70
8 Praktikum 8	82
8.1 Tugas Akhir Praktikum	82

Daftar Gambar

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

Pendahuluan

0.1 Mengenal *Internet of Things*

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

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



Gambar 1: Internet of Things

0.2 Perangkat Board IoT

Untuk membangun sebuah perangkat berbasis IoT, komponen dasar seperti **Board** sangatlah vital untuk dippunyai. 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.

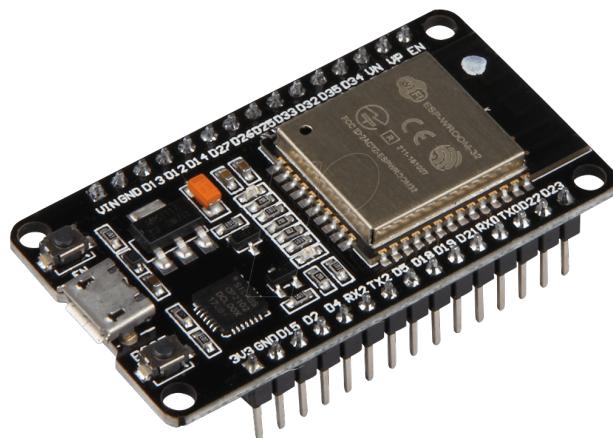
Berikut ini adalah daftar Board yang dapat dibeli dengan harga terjangkau:

1. Arduino



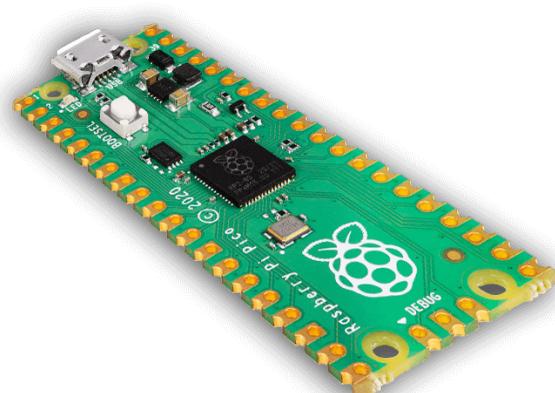
Gambar 2: Board Arduino

2. NodeMCU



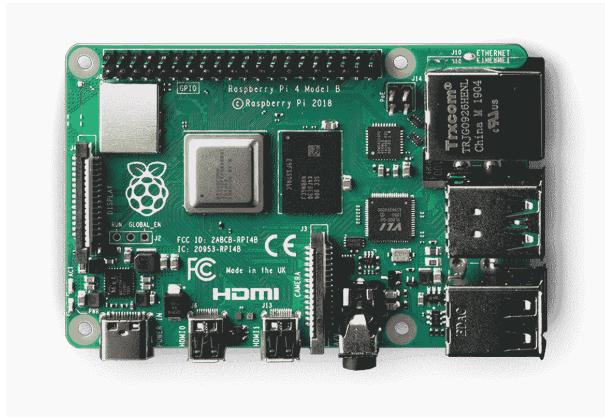
Gambar 3: Board NodeMCU

3. Raspberry Pi Pico



Gambar 4: Board Pico

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



Gambar 5: Board Pi 4B

5. NVidia Jetson



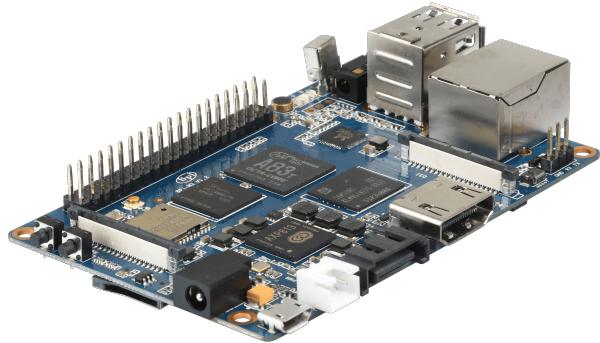
Gambar 6: Board NVidia Jetson

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

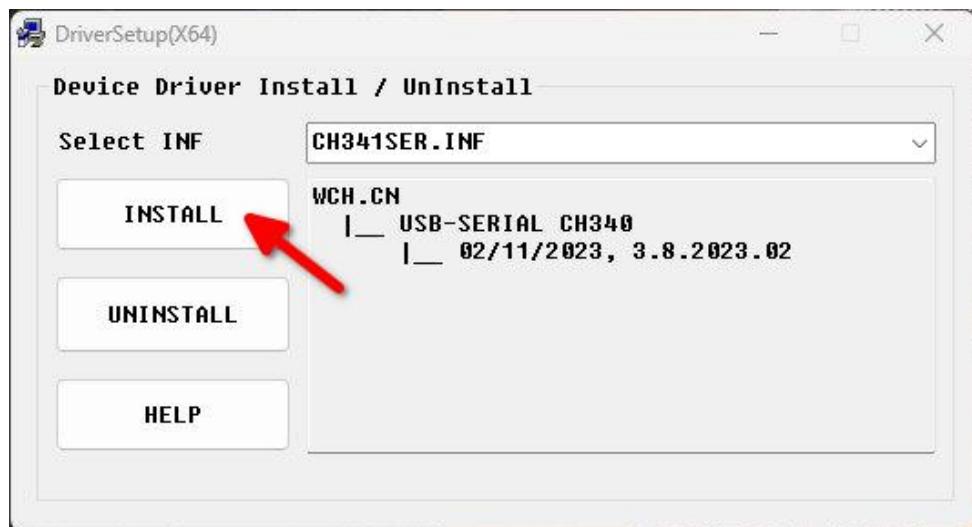
Praktikum 1

1.1 Konfigurasi Arduino IDE dan ESP8266

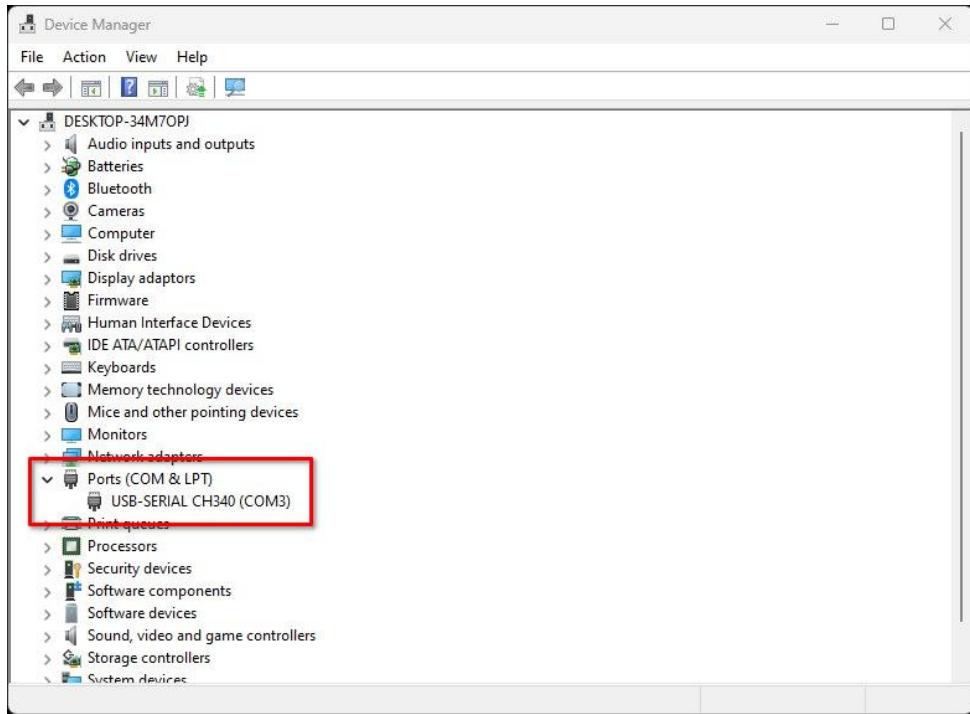
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke komputer beserta konfigurasinya hingga dapat dikenali oleh Arduino IDE. Mahasiswa diharapkan untuk membaca, dan memahami **Persiapan Praktikum** yang ada di halaman sebelumnya.

1.2 Tutorial

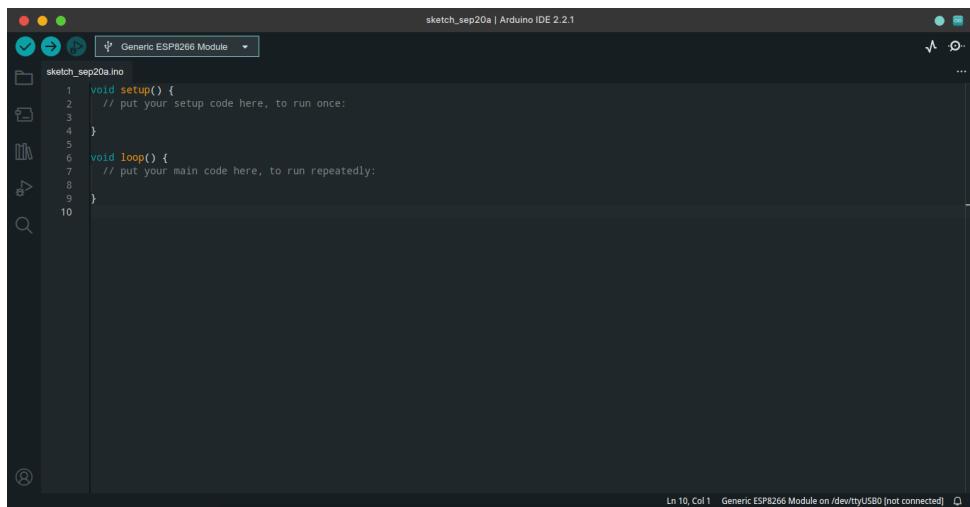
1. Setelah mahasiswa menyiapkan perangkat-perangkat yang diperlukan, maka langkah berikutnya adalah melakukan instalasi driver terlebih dahulu.
2. File driver **CH341SER** yang sudah diunduh, dibuka untuk diinstall. Cukup klik **Install** untuk memasang driver (Windows 10 ke bawah)



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

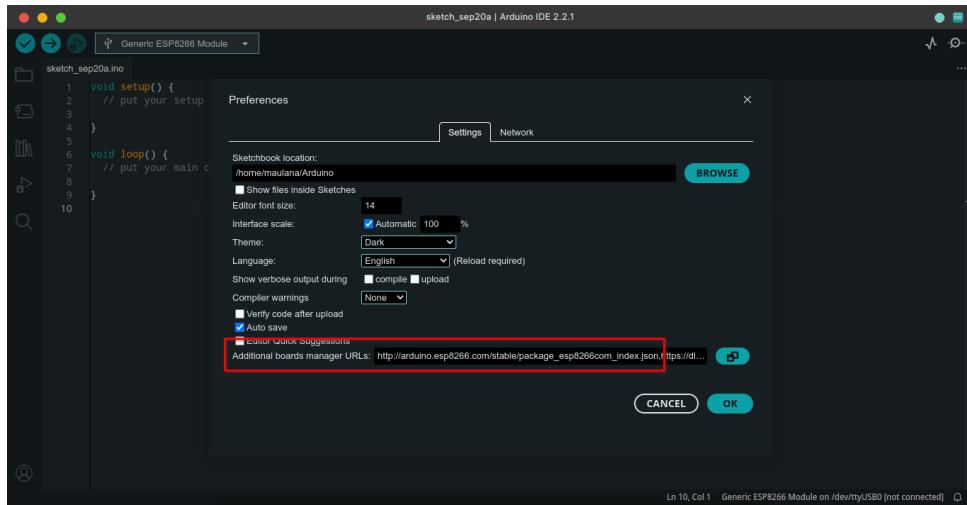


4. Langkah berikutnya adalah mengunduh **Arduino IDE**, usahakan untuk mendapatkan versi terbaru. Setelah unduh, buka aplikasi tersebut

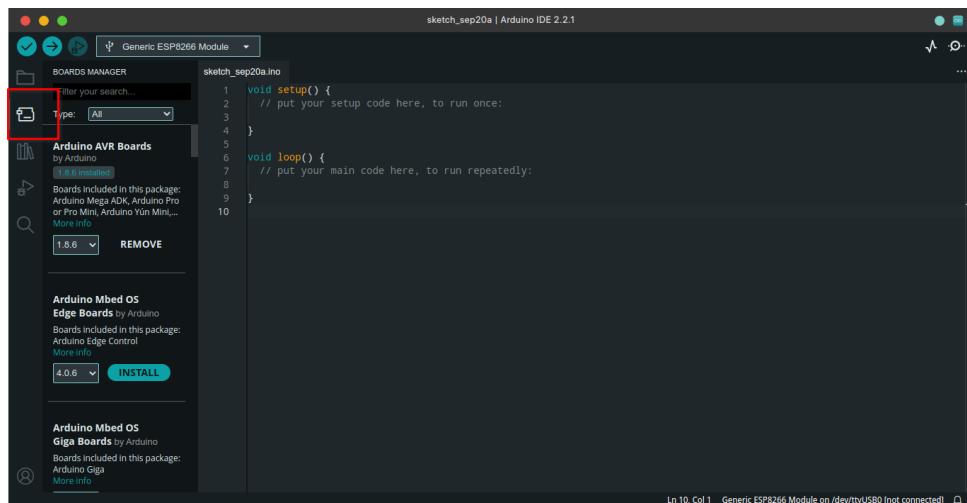


5. Namun **Arduino IDE** ini belum mendukung perangkat yang kita gunakan. Langkah berikutnya buka **File** → **Preferences** →. Tambahkan baris **Alamat URL** berikut ke **Additional board manager URLs**. Klik **OK** untuk mengupdate otomatis.

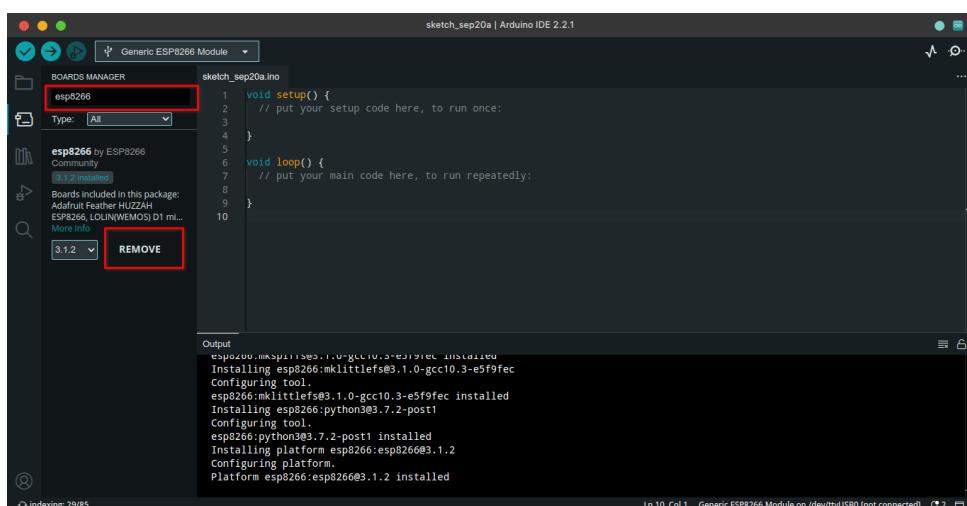
- http://arduino.esp8266.com/stable/package_esp8266com_index.json



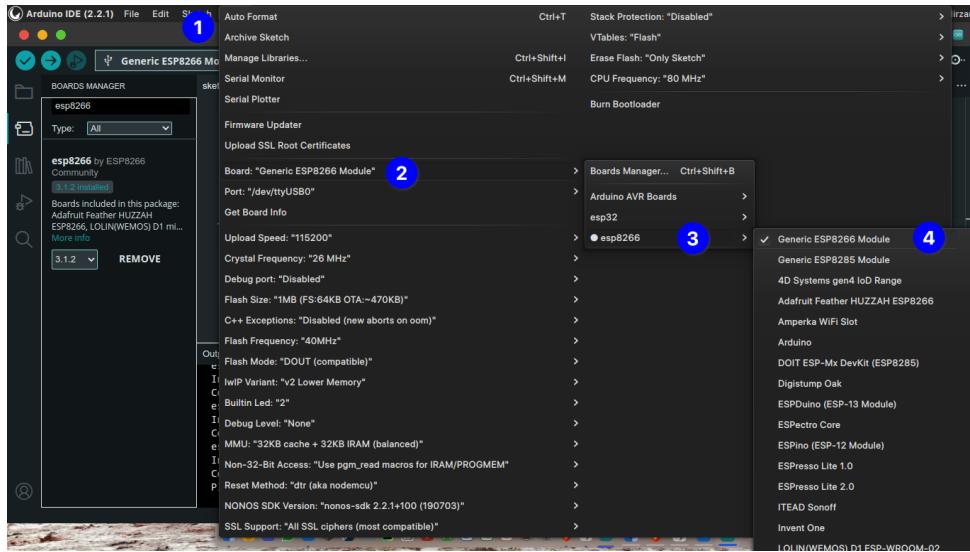
6. Jika sudah, install driver **ESP8266** dengan klik **Boards Manager** di Sidebar Kanan atau **Tools → Board: → Boards Manager**



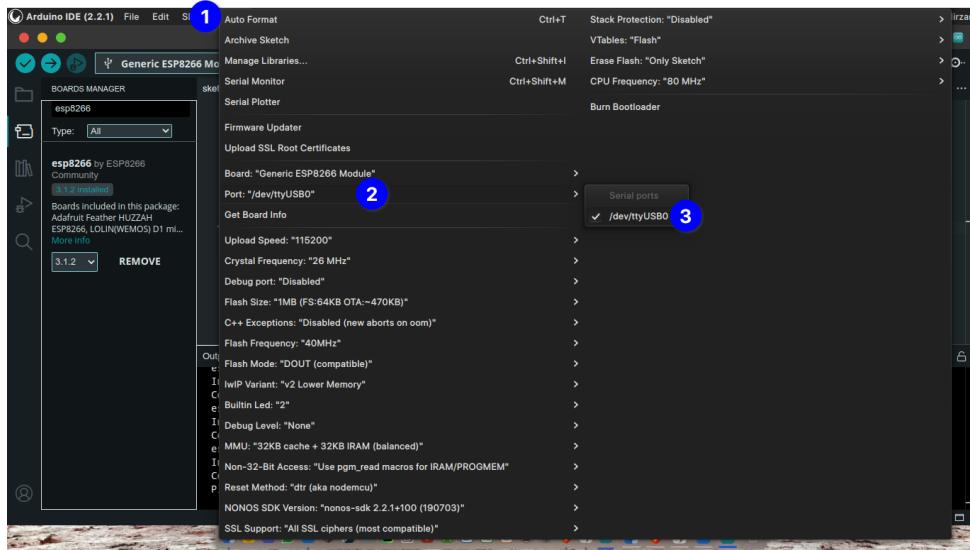
7. Di kolom **Pencarian**, ketik **ESP8266** dan klik **Install**



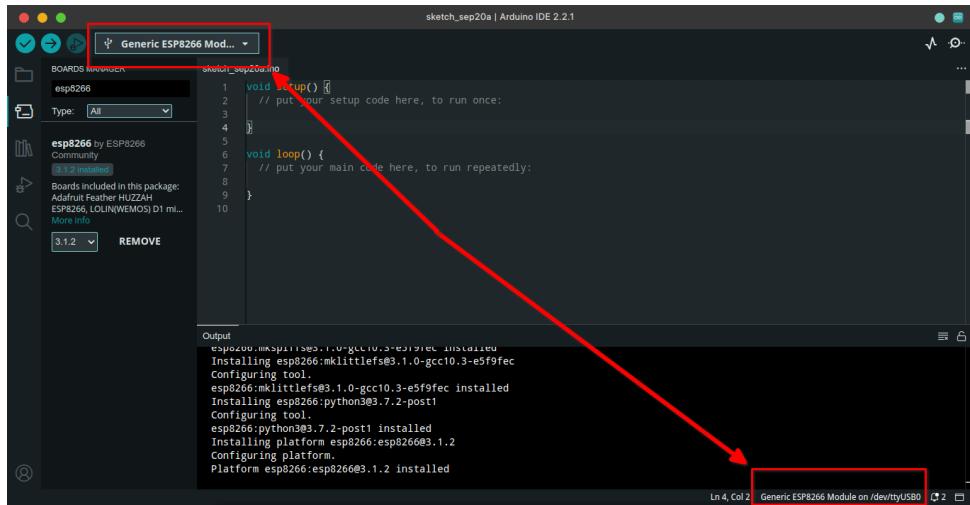
8. Arduino IDE sudah siap, namun belum terhubung ke perangkat. Untuk menghubungkan antara IDE dengan **ESP8266**, pilih **Tools → Board: → esp8266 → Generic ESP8266 Module**



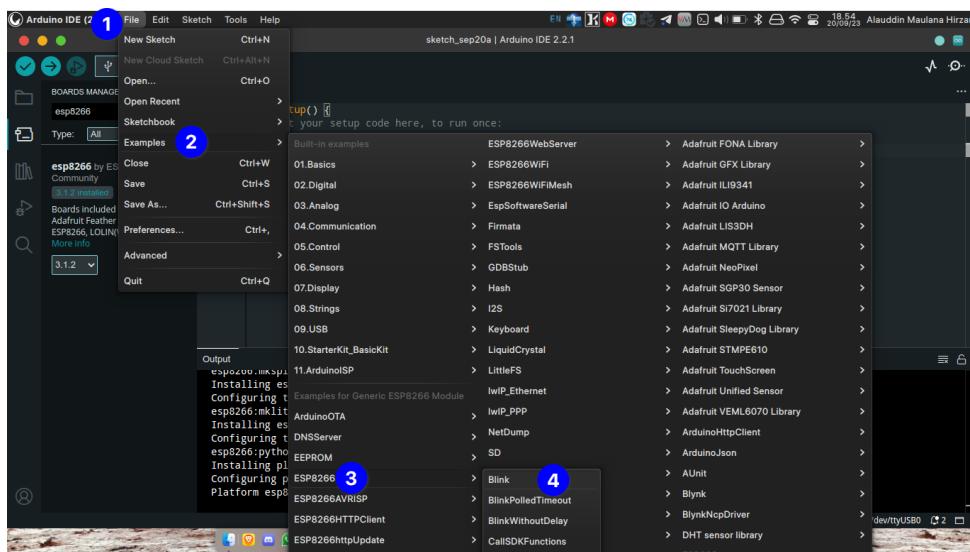
9. Kemudian pastikan **Port Serial** yang digunakan, sama dengan yang ada di **Device Manager**. Cek dengan menu **Tools → Port: → Pilih COM Sesuai Device Manager**



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



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



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

```

Blink | Arduino IDE 2.2.1
Generic ESP8266 Mod...
Blink.ino
1  /*
2   * ESP8266 Blink by Simon Peter
3   * Blink the blue LED on the ESP-01 module
4   * This example code is in the public domain
5
6   * The blue LED on the ESP-01 module is connected to GPIO1
7   * (which is also the TXD pin; so we cannot use Serial.print() at the same time)
8
9   * Note that this sketch uses LED_BUILTIN to find the pin with the internal LED
10 */
11
12 void setup() {
13     pinMode(LED_BUILTIN, OUTPUT); // Initialize the LED_BUILTIN pin as an output
14 }
15
16 // the loop function runs over and over again forever
17 void loop() {
18     digitalWrite(LED_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage level
19     // but actually the LED is on; this is because
20     // it is active low on the ESP-01)
21     delay(1000);
22     digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH
23     delay(2000);
24 }
25

```

Indexing: 48/85

Ln 12, Col 15 Generic ESP8266 Module on /dev/ttyUSB0

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

```

Ln 1, Col 1 Generic ESP8266 Module on /dev/ttyUSB0

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

The screenshot shows the Arduino IDE interface with the 'Blink' sketch open. A red arrow points from the top-left corner to the 'Upload' button in the toolbar. Another red box highlights the 'Output' window at the bottom, which displays a memory dump for the ESP8266 module. The dump includes sections for DATA, RODATA, BSS, and various memory segments like ICACHE and IRAM.

```
DATA 1470 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.

The screenshot shows the Arduino IDE interface during the upload process. A red arrow points from the top-left corner to the 'Upload' button in the toolbar. Another red box highlights the 'Output' window at the bottom, which shows the progress of writing data to the ESP8266 module. The progress bar indicates the upload is nearly complete at 100%.

```
Writing at 0x00010000... (00 %)
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.
```

Bab 2

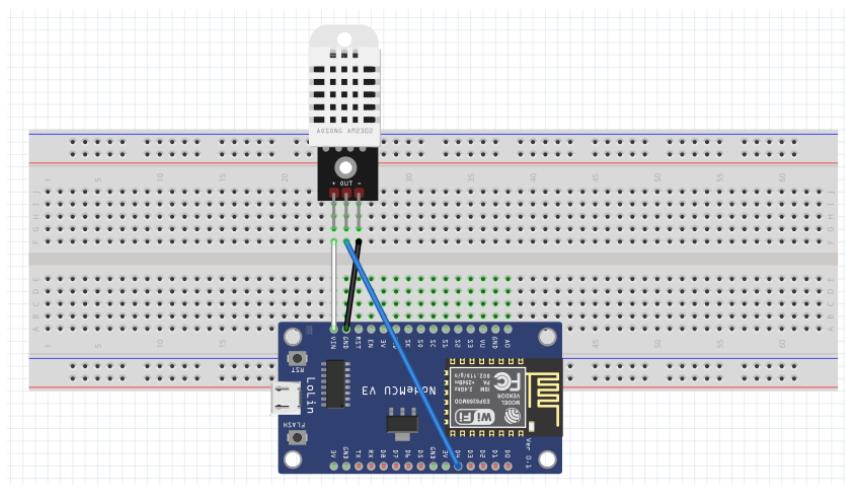
Praktikum 2

2.1 ESP8266, DHT11, dan AdafruitIO

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

2.2 Tutorial

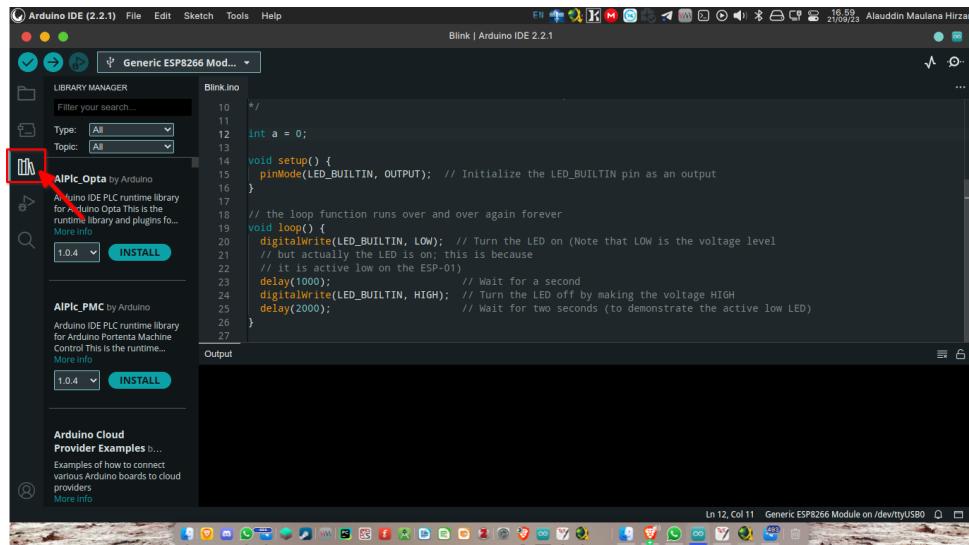
1. Langkah pertama yang perlu dilakukan adalah memasang sensor ke perangkat. Perlu diketahui bahwa dalam memasang sensor harus dalam keadaan **MATI/TIDAK TERTANCAP** untuk menghindari KORSLETING
2. Perhatikan sensor **DHT11**, di bagian kakinya ada tanda **Plus +**, **Minus -**, dan **Out**. Sambungkan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



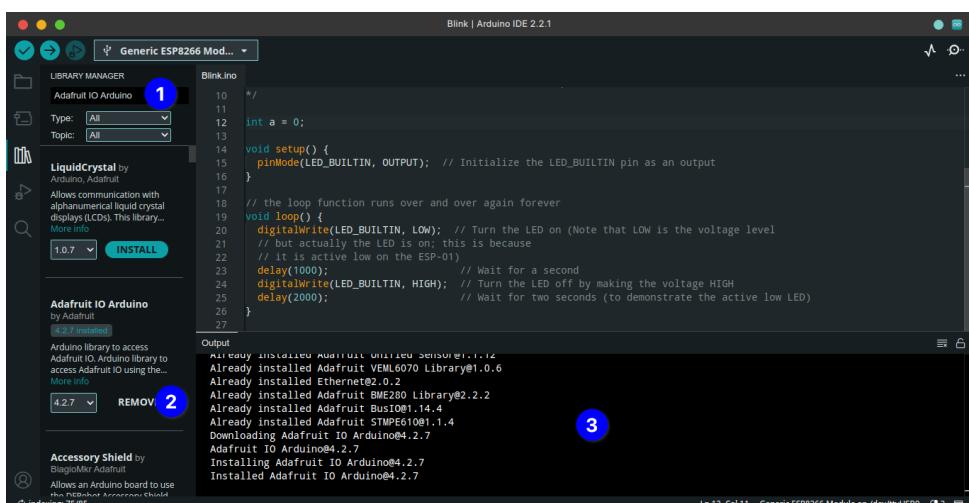
3. Setelah selesai menancapkan sensor, berikutnya adalah melakukan registrasi ke website AdafruitIO dengan link : <https://io.adafruit.com/>. Setelah teregistrasi akan terlihat dasbor seperti berikut:

The screenshot shows the Adafruit IO dashboard for user 'maulanhirzan'. It includes sections for Account Status (Devices: 0 of 2, Groups: 0 of 5, Feeds: 0 of 10, Dashboards: 0 of 5, Data Rate: 0 of 30), Live Errors (No errors since page load), and Live Data (No data since page load). A green banner at the top informs users about the Basic plan and encourages upgrading.

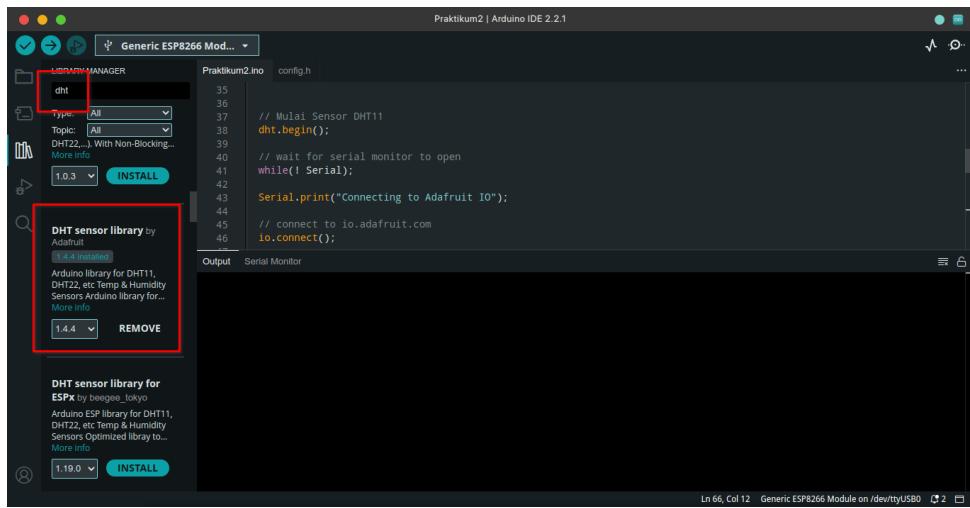
4. Kembali ke Arduino IDE, dan install Library dengan mengakses menu samping atau Sketch → Include Library → Manage Libraries



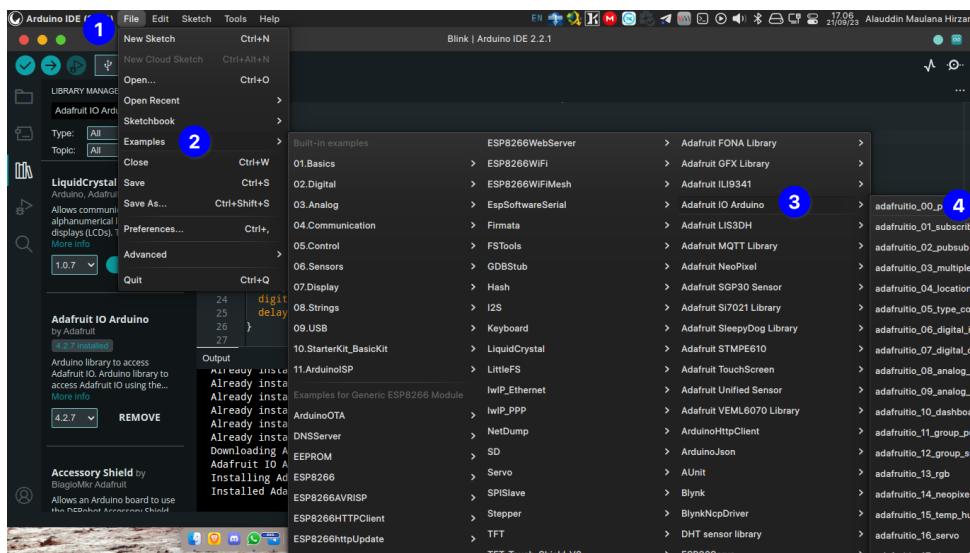
5. Cari Adafruit IO Arduino, klik INSTALL, lalu INSTALL ALL



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



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



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

The screenshot shows the Adafruit IO dashboard for the user 'maulanahirzan'. At the top right, there is a 'New Device' button with a key icon. A red arrow points to this button, indicating where to click to start creating a new device.

- 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**

The screenshot shows the 'YOUR ADAFRUIT IO KEY' modal window. It displays the Adafruit IO Key configuration details. The Arduino code section, which contains the #define IO_USERNAME and #define IO_KEY lines, is highlighted with a red box.

The screenshot shows the Arduino IDE with the 'config.h' file open. The code defines the Adafruit IO key and username. The lines '#define IO_USERNAME "maulanahirzan"' and '#define IO_KEY "aio_nzry79PnWNARw38JWUusj3Vebkh"' are highlighted with a red box.

```

1 //***** Adafruit IO Config *****/ 
2 // visit io.adafruit.com if you need to create an account,
3 // or go to https://www.adafruit.com/login
4 #define IO_USERNAME "maulanahirzan"
5 #define IO_KEY "aio_nzry79PnWNARw38JWUusj3Vebkh"
6 //***** WiFi *****/
7
8 // the AdafruitWiFi client will work with the following boards:
9 // - HUZZAH ESP8266 Breakout -> https://www.adafruit.com/products/2471
10 // - Feather HUZZAH ESP8266 -> https://www.adafruit.com/products/2821
11 // - Feather HUZZAH ESP32 -> https://www.adafruit.com/product/3405
12 // - Feather M0 WiFi -> https://www.adafruit.com/products/3010
13 // - Feather WiFi -> https://www.adafruit.com/products/3056
14 // - Adafruit WiFiCEP -> https://www.adafruit.com/product/4116
15 // - Adafruit Metro M4 Express Airlift Lite ->
16 // https://www.adafruit.com/product/4000
17 // - Adafruit Airlift Breakout -> https://www.adafruit.com/product/4201
18 // - Adafruit Airlift Shield -> https://www.adafruit.com/product/4285
19 // - Adafruit Airlift FeatherWing -> https://www.adafruit.com/product/4264
20
21 //***** WiFi *****/
22
23 #define WIFI_SSID "your_ssid"
24 #define WIFI_PASS "your_pass"
25
26 // uncomment the following line if you are using airlift
27 // #define USE_AIRLIFT
28
29 // uncomment the following line if you are using winc1500
30 // #define USE_WINC1500

```

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

maulanahirzan / Feeds

New Feed New Group

Feed Name	Key	Last value	Recorded
Default			

Loaded in 0.31 seconds.

Get Help Learn
Quick Guides IO Plus
API Documentation News

Create a new Feed

Name suhu
Maximum length: 128 characters. Used: 4

Description

Cancel Create

Loaded in 0.31 seconds.

Get Help Learn
Quick Guides IO Plus
API Documentation News

Create a new Feed

Name lembab
Maximum length: 128 characters. Used: 6

Description

Cancel Create

Loaded in 0.32 seconds.

Get Help Learn
Quick Guides IO Plus

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

```

Generic ESP8266 Mod... | config.h
adafruitio_00_publish.ino | config.h
17 // - Adafruit Metro M4 Express Airlift Lite ->
18 // https://www.adafruit.com/product/4000
19 // - Adafruit Airlift Breakout -> https://www.adafruit.com/product/4201
20 // - Adafruit Airlift Shield -> https://www.adafruit.com/product/4285
21 // - Adafruit Airlift Featherwing -> https://www.adafruit.com/product/4264
22
23 #define WIFI_SSID "Free Wifi USM 1"
24 #define WIFI_PASS ""
25
26 // uncomment the following line if you are using airlift
27 // #define USE_AIRLIFT
28
29 // uncomment the following line if you are using winc1500
30 // #define USE_WINC1500
31
32 // uncomment the following line if you are using mkr1010 or nano 33 iot
33 // #define ARDUINO_SAMD_MKR1010
34
35 // comment out the following lines if you are using fona or ethernet
36 #include "AdafruitIO_WiFi.h"
37
38 #if defined(USE_AIRLIFT) || defined(ADAFRUIT_METRO_M4_AIRLIFT_LITE) || \
39     defined(ADAFRUIT_PYPORTAL)
40 // Configure the pins used for the ESP32 connection
41 #if !defined(SPIWIFI_SS) // if the wifi definition isn't in the board variant
42 // Don't change the names of these #define's! they match the variant ones
43 #define SPIWIFI_SPI
44 #define SPIWIFI_SS 10 // Chip select pin
45 #define NINA_ACK 9 // a.k.a. BUSY or READY pin
46 #define NINA_RESETEN 6 // Reset pin

```

Ln 24, Col 20 Generic ESP8266 Module on /dev/ttyUSB0

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

```

Generic ESP8266 Mod... | arduino_00_publish.ino | config.h
1 adafruitio_00_publish.ino | config.h
2 // Example Starts Here ****
3 // this int will hold the current count for our sketch
4 int count = 0;
5
6 // set up the 'counter' feed
7 AdafruitIO_Feed *counter = io.feed("counter");
8
9 void setup() {
10
11     // start the serial connection
12     Serial.begin(115200);
13
14     // wait for serial monitor to open
15     while(! Serial);
16
17     Serial.print("Connecting to Adafruit IO");
18
19     // connect to io.adafruit.com
20     io.connect();
21
22     // wait for a connection
23     while(io.status() < AIO_CONNECTED) {
24         Serial.print(".");
25         delay(500);
26     }
27
28     // we are connected
29     Serial.println();
30     Serial.println(io.statusText());
31
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

```

Ln 1, Col 1 Generic ESP8266 Module on /dev/ttyUSB0

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

```

Ln 26, Col 1 Generic ESP8266 Module on /dev/ttyUSB0

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

```

Ln 26, Col 1 Generic ESP8266 Module on /dev/ttyUSB0

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
Praktikum2.ino config.h
22
23 // set up the 'counter' feed
24 AdafruitIO_Feed *suhu = io.feed("suhu");
25 AdafruitIO_Feed *lembab = io.feed("lembab");
26
27 #define DHTPIN 2
28 #define DHTTYPE DHT11
29 DHT dht(DHTPIN, DHTTYPE);
30
31 void setup() {
32
33 // start the serial connection
34 Serial.begin(115200);
35
36 // Mulai Sensor DHT11
37 dht.begin();
38
39 // wait for serial monitor to open
40 while(! Serial);
41
42 Serial.print("Connecting to Adafruit IO");
43
44 // connect to io.adafruit.com

```

Ln 59, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 [not connected] 2

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
Praktikum2.ino config.h
1 void setup() {
2
3 // start the serial connection
4 Serial.begin(115200);
5
6 // wait for serial monitor to open
7 while(! Serial);
8
9 Serial.print("Connecting to Adafruit IO");
10
11 // connect to io.adafruit.com
12 io.connect();
13
14 // wait for a connection
15 while(io.status() < AIO_CONNECTED) {
16   Serial.print(".");
17   delay(500);
18 }
19
20 // we are connected
21 Serial.println();
22 Serial.println(io.statusText());
23
24 // Mulai Sensor DHT11
25 dht.begin();
26
27 void loop() {

```

Ln 53, Col 24 Generic ESP8266 Module on /dev/ttyUSB0

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

Potongan Kode

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

```

Praktikum2.ino config.h
50   Serial.println();
51   Serial.println(io.statusText());
52
53   // Mulai Sensor DHT11
54   dht.begin();
55 }
56
57 void loop() {
58
59   // io.run(); is required for all sketches.
60   // it should always be present at the top of your loop
61   // function. It keeps the client connected to
62   // io.adafruit.com, and processes any incoming data.
63   io.run();
64
65   float temperature = dht.readTemperature();
66   float humidity = dht.readHumidity();
67
68   // save count to the 'counter' feed on Adafruit IO
69   Serial.print("sending -> ");
70   Serial.println(count);
71   counter->save(count);
72
73   // increment the count by 1
74   count++;
75
76   // Adafruit IO is rate limited for publishing, so a delay is required in
77   // between feed->save events. In this example, we will wait three seconds
78   // (1000 milliseconds == 1 second) during each loop.
79   // delay(3000);
80
81 }
82
83
84
85

```

Ln 66, Col 39 Generic ESP8266 Module on /dev/ttyUSB0

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

Potongan Kode

```

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

```

```

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

```

Ln 71, Col 1 Generic ESP8266 Module on /dev/ttyUSB0

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

Potongan Kode

```

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

```

```

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

```

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

```

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

```

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

```

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

Output
DATA    1304  initialized variables
└─ RODATA 1312  constants
BSS    25944  zeroed variables
. Instruction RAM (TRAM_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

Praktikum2 | Arduino IDE 2.2.1

Generic ESP8266 Mod...

Praktikum2.ino config.h

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

Output

```
Writing at 0x00024000... ( 92 %)
Writing at 0x00028000... ( 84 %)
Writing at 0x0002c000... ( 92 %)
Writing at 0x00030000... (100 %)
Wrote 290816 bytes (212964 compressed) at 0x00000000 in 18.7 seconds (effective 124.2 kbit/s)...
Hash of data verified.

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

Done uploading.

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

```
Praktikum2 | Arduino IDE 2.2.1

Generic ESP8266 Mod... ▾

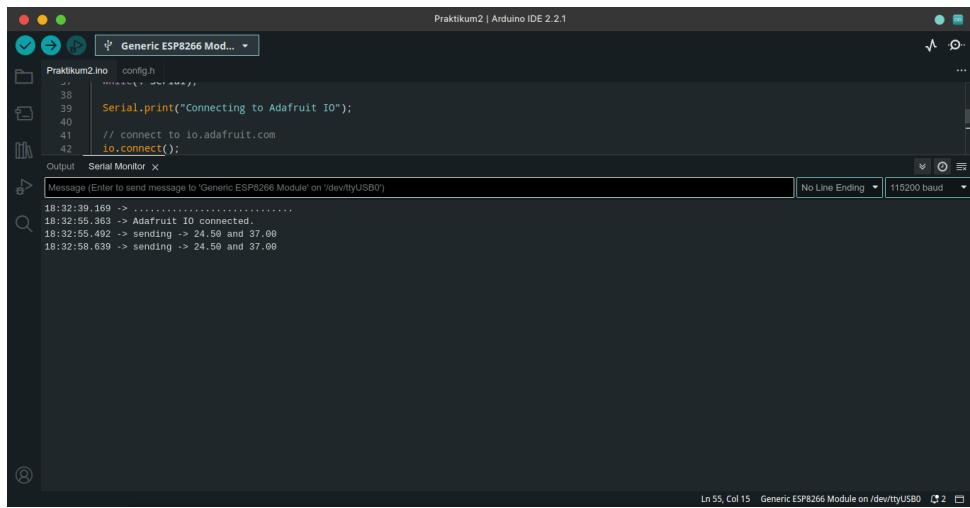
Praktikum2.ino config.h

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

Output: Serial Monitor X
Message: [redacted] to send message to 'Generic ESP8266 Module' on '/dev/ttyUSB0'
No Line Ending ▾ 115200 baud ▾
```

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

26. Alat terhubung dan berhasil mengirimkan data



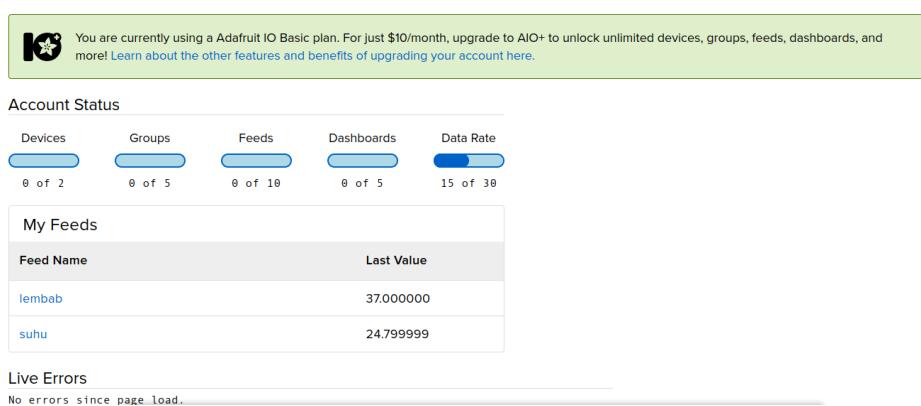
The screenshot shows the Arduino IDE interface with the title "Praktikum2 | Arduino IDE 2.2.1". The code in the Praktikum2.ino file is as follows:

```
Praktikum2.ino config.h
38
39 Serial.print("Connecting to Adafruit IO");
40
41 // connect to io.adafruit.com
42 io.connect();
43
44 Output
Serial Monitor x
Message (Enter to send message to "Generic ESP8266 Module" on "/dev/ttyUSB0")
18:32:39.169 -> .....
18:32:55.363 -> Adafruit IO connected.
18:32:55.492 -> sending -> 24.99 and 37.00
18:32:56.639 -> sending -> 24.99 and 37.00
```

The Serial Monitor window shows the following output:

```
18:32:39.169 -> .....
18:32:55.363 -> Adafruit IO connected.
18:32:55.492 -> sending -> 24.99 and 37.00
18:32:56.639 -> sending -> 24.99 and 37.00
```

27. Hasil di website Adafruit IO



The screenshot shows the Adafruit IO account status and my feeds page.

Account Status

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

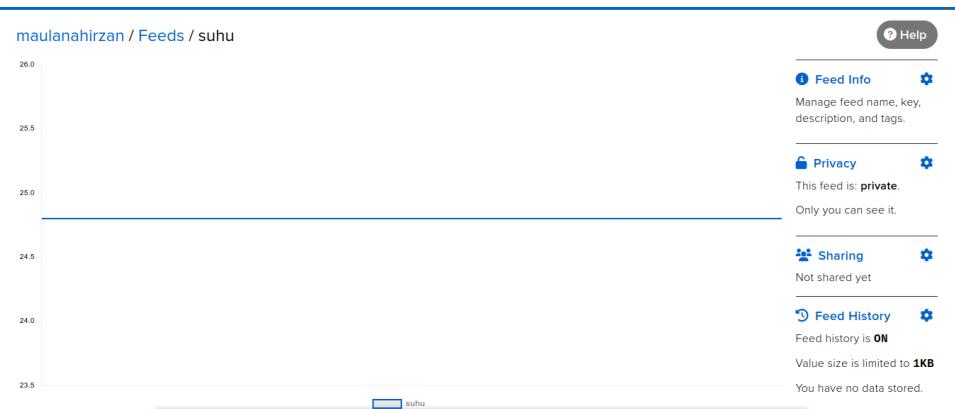
My Feeds

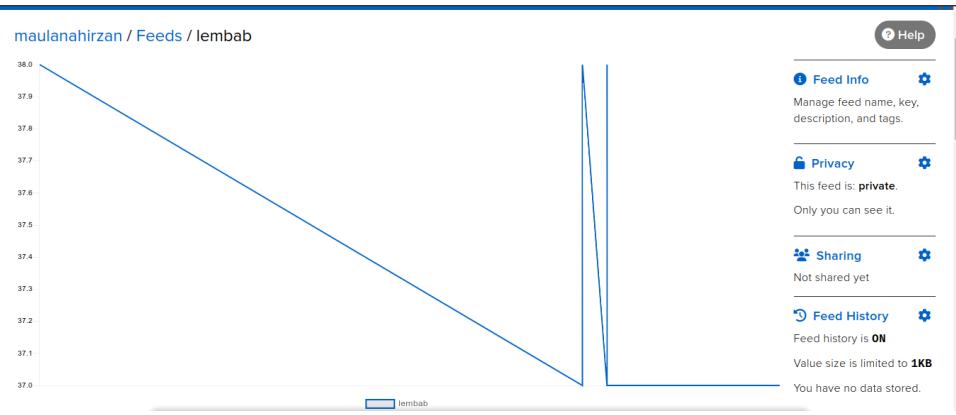
Feed Name	Last Value
lembab	37.000000
suhu	24.799999

Live Errors

No errors since page load.

28. Klik salah satu feed untuk melihat data





Potongan Kode

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

```

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

Ln 73, Col 28 Generic ESP8266 Module on /dev/ttyUSB0

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

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

Bab 3

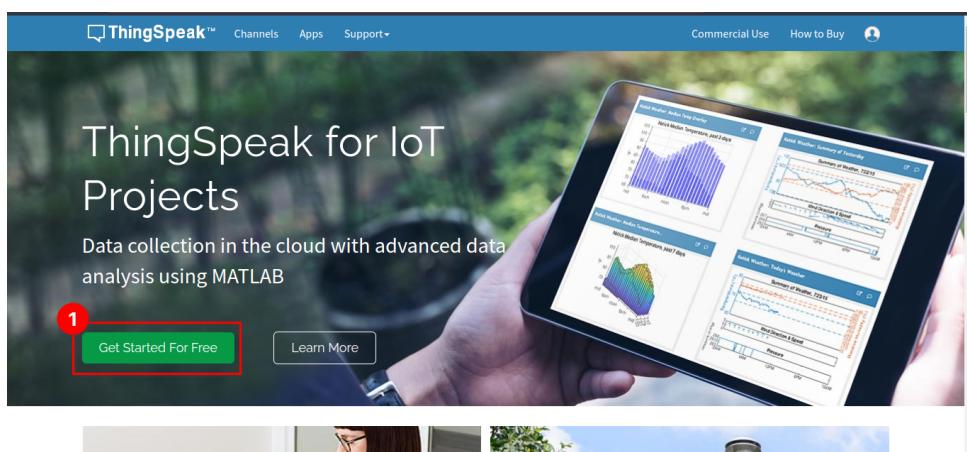
Praktikum 3

3.1 ESP8266, DHT11, dan Thingspeak

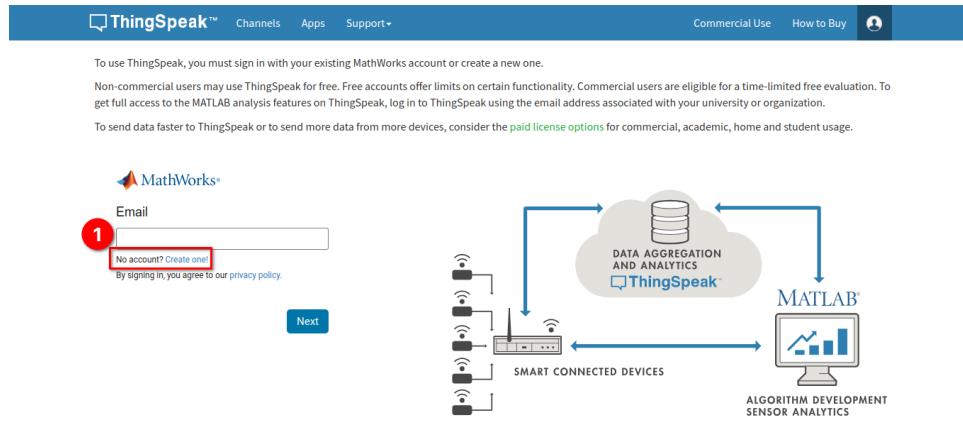
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Thingspeak. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 2** yang ada di halaman sebelumnya.

3.2 Tutorial

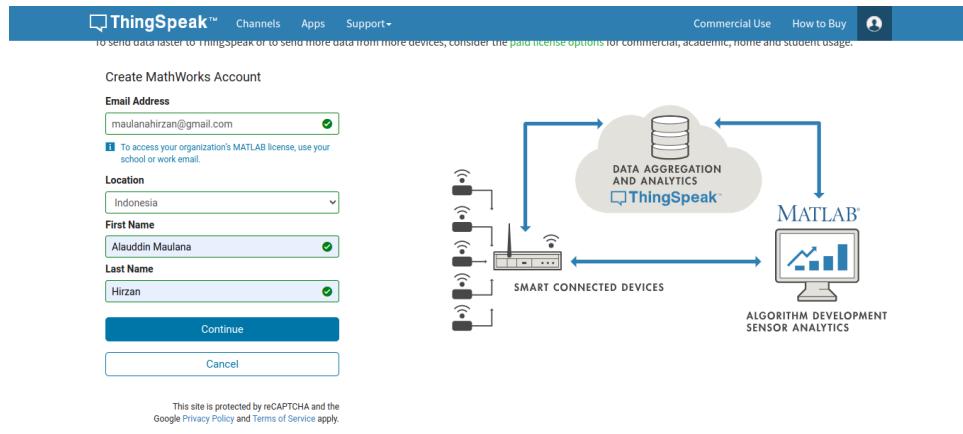
1. Untuk memulai praktikum ini, mahasiswa diwajibkan untuk membuat akun di <https://thingspeak.com/> secara gratis. Klik **Get started for free**



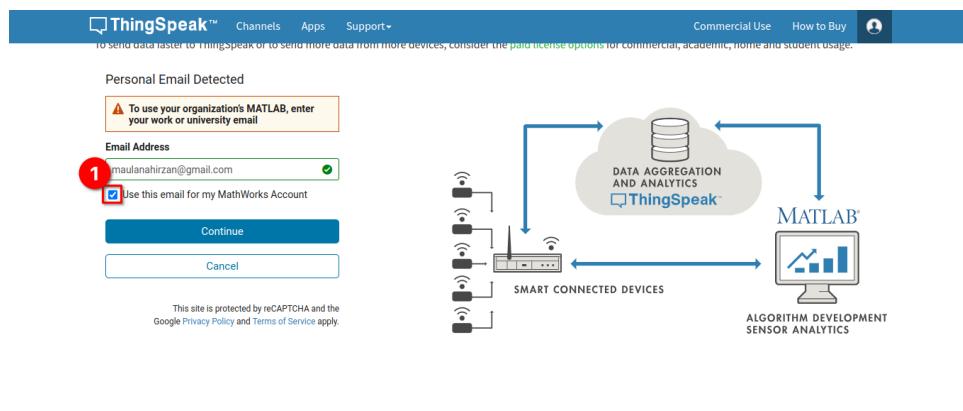
2. Klik **Create one!**



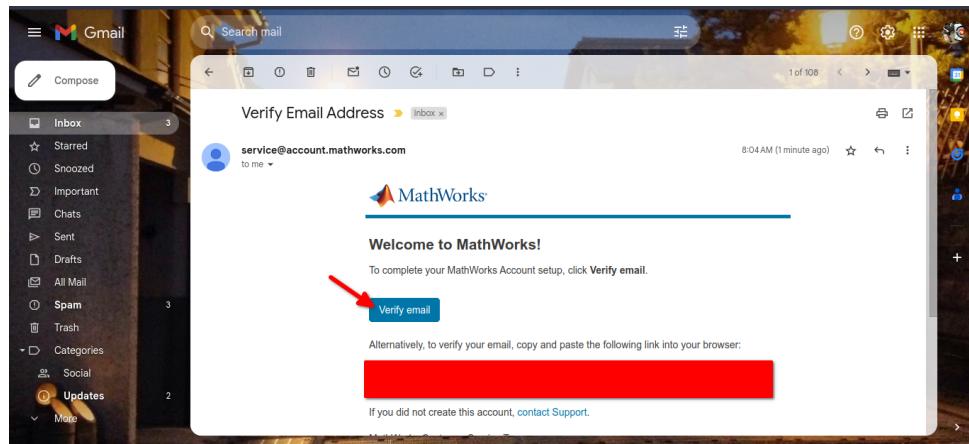
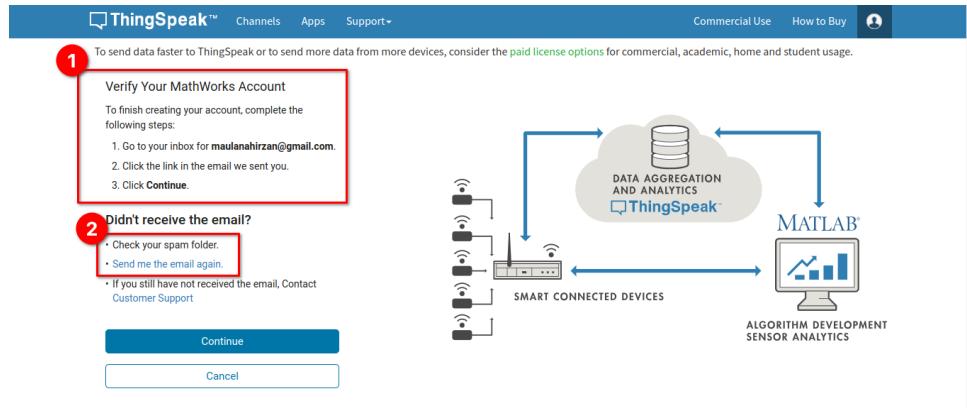
3. Isi informasi identitas



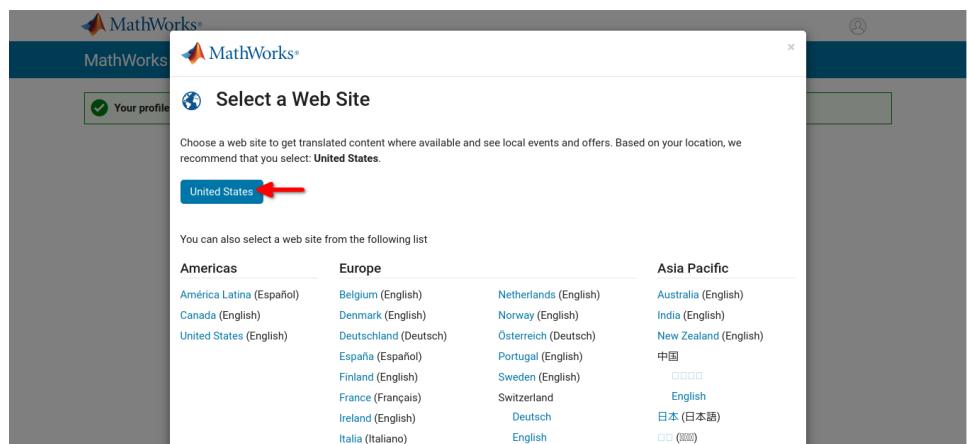
4. Centang untuk menggunakan email pribadi



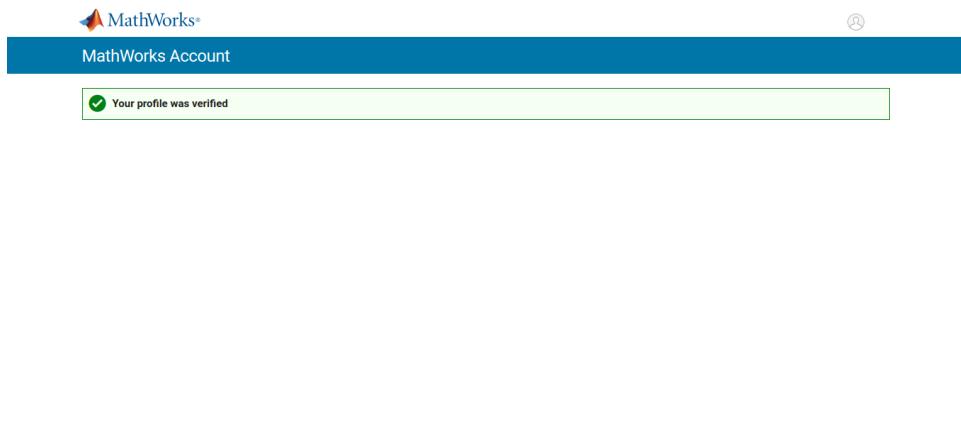
5. Cek email anda (termasuk SPAM) untuk verifikasi email. **JANGAN TUTUP WINDOW INI!!!**



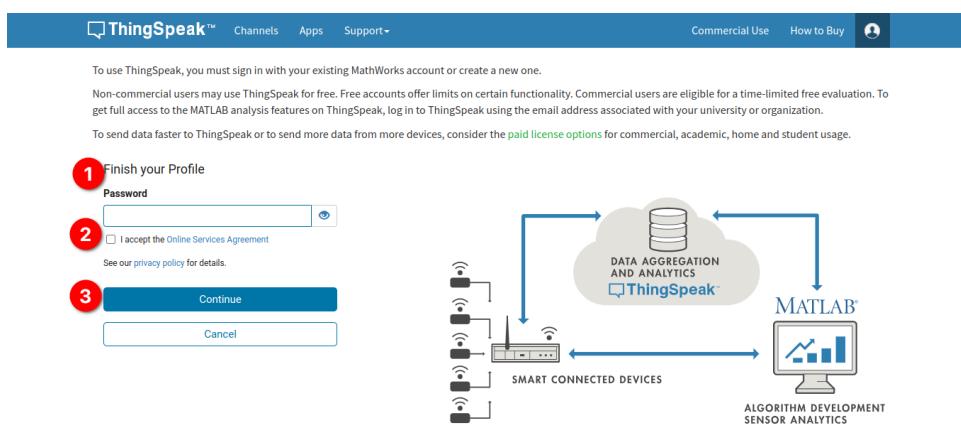
6. Pilih negara untuk website



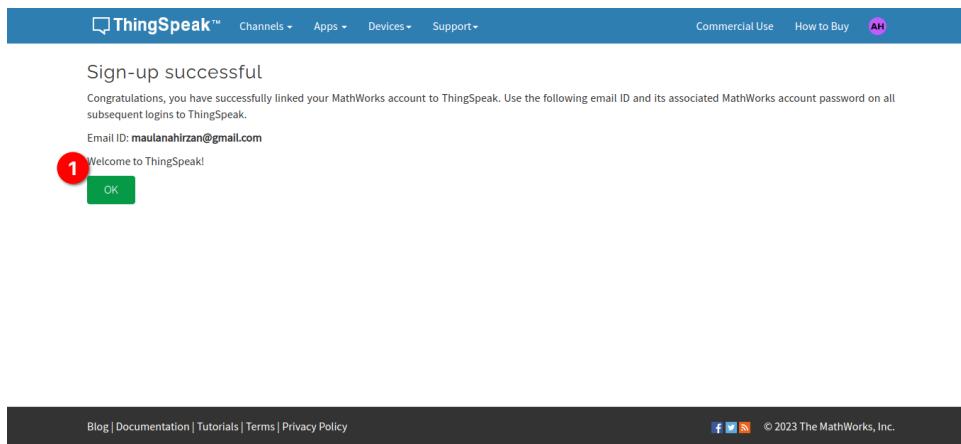
7. Akun sudah terverifikasi



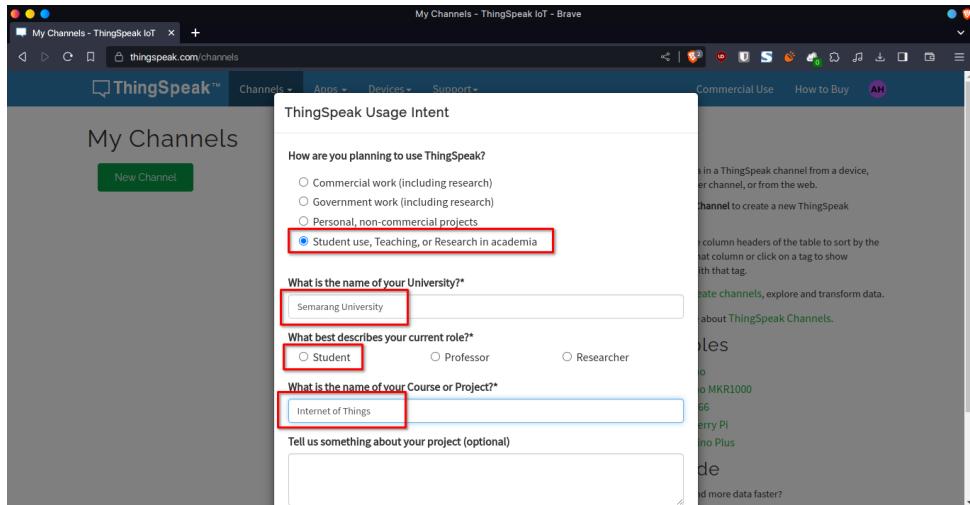
8. Ketika sudah selesai, kembali ke **WINDOW** di **Langkah 5.** dan klik **Continue**



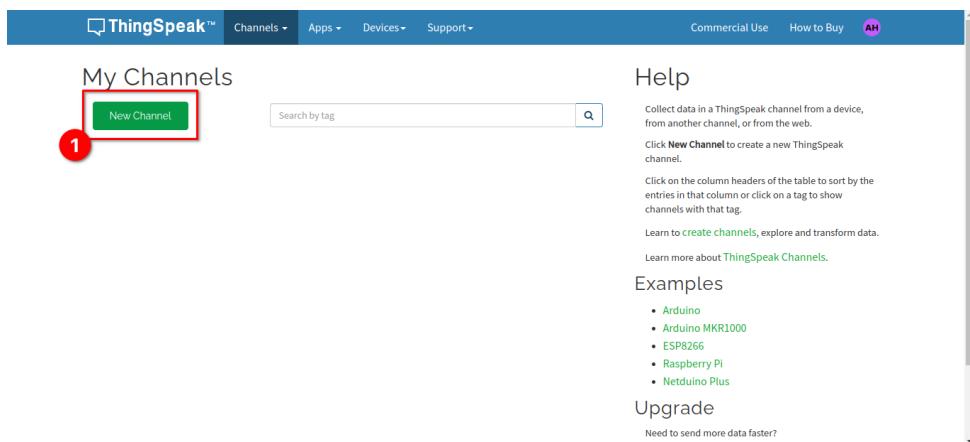
9. Klik **OK** untuk pindah ke **Dasbor**



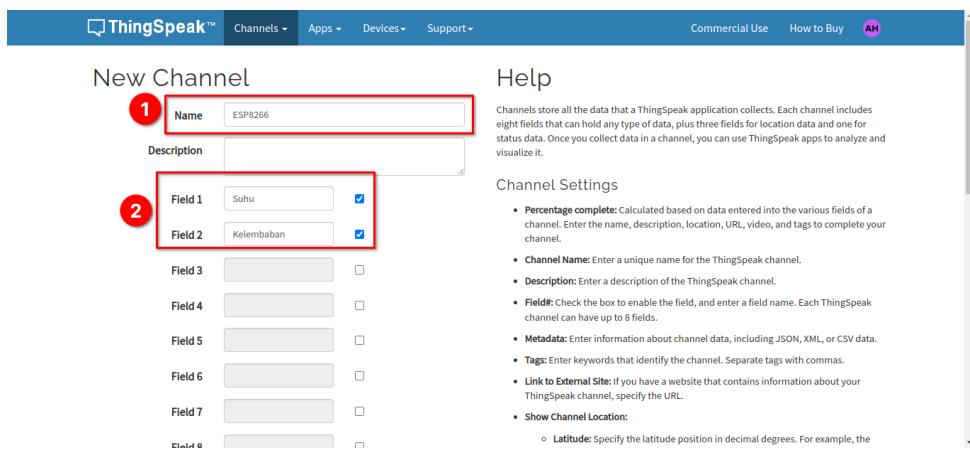
10. Di Dashboard akan ditanya penggunaan **Thingspeak**. Isi sesuai pertanyaan. Jangan lupa untuk klik **OK** atau **Continue**



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



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



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

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

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

16. Install Library **Thingspeak**

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

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

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

```
/*
 * 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 ESP8266 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.

```
// 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 "12345678" // replace MyPassword with your WiFi password
#define SECRET_CH_ID 229 // replace 000000 with your channel number
#define SECRET_WRITE_APIKEY "U9OQWVZP" // 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
Praktikum3.ino secrets.h

9
10 Note:
11 - Requires ESP8266WiFi library and ESP8622 board add-on. See https://github.com/esp8266/Arduino for details.
12 - Select the target hardware from the Tools->Board menu
13 - This example is written for a network using WPA encryption. For WEP or WPA, change the WiFi.begin() call accordingly.
14
15 ThingSpeak ( https://www.thingspeak.com ) is an analytic IoT platform service that allows you to aggregate, visualize, and
16 analyze live data streams in the cloud. Visit https://www.thingspeak.com to sign up for a free account and create a channel.
17 Documentation for the ThingSpeak Communication Library for Arduino is in the README.md folder where the library was installed.
18 See https://www.mathworks.com/help/thingspeak/index.html for the full ThingSpeak documentation.
19
20 For licensing information, see the accompanying license file.
21
22 Copyright 2020, The MathWorks, Inc.
23
24 */
25
26 #include <ESP8266WiFi.h>
27 #include "secrets.h"
28 //include "DHT.h" // always include thingspeak header file after other header files and custom macros
29 #include <DHT.h>
30
31 char ssid[] = SECRET_SSID; // your network SSID (name)
32 char pass[] = SECRET_PASS; // your network password
33 int keyIndex = 0; // your network key index number (needed only for WEP)
34 WiFiClient client;
35
36 unsigned long myChannelNumber = SECRET_CH_ID;

```

Ln 29, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 (not connected)

21. Hapus kode berikut

```

Praktikum3 | Arduino IDE 2.2.1
Praktikum3.ino secrets.h

25
26 #include <ESP8266WiFi.h>
27 #include "secrets.h"
28 //include "DHT.h" // always include thingspeak header file after other header files and custom macros
29
30 char ssid[] = SECRET_SSID; // your network SSID (name)
31 char pass[] = SECRET_PASS; // your network password
32 int keyIndex = 0; // your network key index number (needed only for WEP)
33 WiFiClient client;
34
35 unsigned long myChannelNumber = SECRET_CH_ID;
36 const char * myWriteAPIKey = SECRET_WRITE_APIKEY;

37 // Initialize our values
38 int number1 = 0;
39 int number2 = random(0,100);
40 int number3 = random(0,100);
41 int number4 = random(0,100);
42 String myStatus = "";
43
44 void setup() {
45   Serial.begin(115200); // Initialize serial
46   while (!Serial) {
47     ; // wait for serial port to connect. Needed for Leonardo native USB port only
48   }
49
50   WiFi.mode(WIFI_STA);
51   ThingSpeak.begin(client); // Initialize ThingSpeak
52 }


```

Ln 44, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 (not connected)

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

Potongan Kode

```

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

```

```

Praktikum3.ino secrets.h
22
23 Copyright 2020, The MathWorks, Inc.
24 /*
25
26 #include <ESP8266WiFi.h>
27 #include "secrets.h"
28 #include "thingspeak.h" // always include thingspeak header file after other header files and custom macros
29 #include <DHT.h>
30
31 char ssid[] = SECRET_SSID; // your network SSID (name)
32 char pass[] = SECRET_PASS; // your network password
33 int keyIndex = 0; // your network key Index number (needed only for WEP)
34 WiFiClient client;
35
36 unsigned long myChannelNumber = SECRET_CH_ID;
37 const char * myWriteAPIKey = SECRET_WRITE_APIKEY;
38
1 #define DHTPIN 2
#define DHTTYPE DHT11
39 DHT dht(DHTPIN, DHTTYPE);
40
41 void setup() {
42   Serial.begin(115200); // Initialize serial
43   while (!Serial) {
44     ; // wait for serial port to connect. Needed for Leonardo native USB port only
45   }
46
47   WiFi.mode(WIFI_STA);
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

Ln 42, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 (not connected)

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

Potongan Kode

```

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

```

Praktikum3.ino secrets.h
38
39 #define DHTPIN 2
40 #define DHTTYPE DHT11
41 DHT dht(DHTPIN, DHTTYPE);
42
43 void setup() {
44   Serial.begin(115200); // Initialize serial
45   while (!Serial) {
46     ; // wait for serial port to connect. Needed for Leonardo native USB port only
47   }
48
49   WiFi.mode(WIFI_STA);
50   ThingSpeak.begin(client); // Initialize ThingSpeak
1 // Mulai Sensor DHT11
52 dht.begin();
53
54
55
56
57
58
59
60
61
62
63
64
65

```

Ln 52, Col 15 Generic ESP8266 Module on /dev/ttyUSB0 (not connected)

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

Praktikum3 | Arduino IDE 2.2.1

```

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

```

Ln 52, Col 15 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

Praktikum3 | Arduino IDE 2.2.1

```

Praktikum3.ino secrets.h
55
56
57 // set the fields with the values
58 ThingSpeak.setField(1, number1);
59 ThingSpeak.setField(2, number2);
60 ThingSpeak.setField(3, number3);
61 ThingSpeak.setField(4, number4);
62
63 // figure out the status message
64 if(number1 > number2){
65     myStatus = String("field1 is greater than field2");
66 }
67 else if(number1 < number2){
68     myStatus = String("field1 is less than field2");
69 }
70 else{
71     myStatus = String("field1 equals field2");
72 }
73
74 // set the status
75 ThingSpeak.setStatus(myStatus);
76
77 // write to the ThingSpeak channel
78 int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
79 if(x == 200){
80     WiFi.println("Channel update successful.");
81 }
82 else{
83     WiFi.println("Problem updating channel. HTTP error code " + String(x));
84 }
85

```

Ln 52, Col 15 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

Praktikum3 | Arduino IDE 2.2.1

```

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

```

Ln 52, Col 15 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

25. Hasil AKHIR SEHARUSNYA:

The screenshot shows the Arduino IDE interface with the title "Praktikum3 | Arduino IDE 2.2.1". The code editor contains the following C++ code:

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

The status bar at the bottom indicates "Ln 80, Col 4 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]".

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

Potongan Kode

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

The screenshot shows the Arduino IDE interface with the title "Praktikum3 | Arduino IDE 2.2.1". The code editor contains the following C++ code, with the new lines highlighted in a red box:

```
Praktikum3.ino secrets.h
54
55 void loop() {
56
57     // Connect or reconnect to WiFi
58     if(WiFi.status() != WL_CONNECTED){
59         Serial.println("Attempting to connect to SSID: ");
60         Serial.println(SECRET_SSID);
61         while(WiFi.status() != WL_CONNECTED){
62             WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using open or WEP network
63             Serial.print(".");
64             delay(5000);
65         }
66         Serial.println("\nConnected.");
67     }
68
69     float temperature = dht.readTemperature();
70     float humidity = dht.readHumidity();
71
72     // set the fields with the values
73     ThingSpeak.setField(1, number1);
74     ThingSpeak.setField(2, number2);
75
76     // write to the ThingSpeak channel
77     int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
78     if(x == 200){
79         Serial.println("Channel update successful.");
80     } else{
81         Serial.println("Problem updating channel. HTTP error code " + String(x));
82     }
83
84 }
```

The status bar at the bottom indicates "Ln 70, Col 39 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]".

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

Potongan Kode

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

```

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

```

Ln 74, Col 36 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

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

```

Praktikum3 | Arduino IDE 2.2.1
Praktikum3.ino secrets.h
54
55 void loop() {
56
57     // set the fields with the values
58     ThingSpeak.setField(1, temperature);
59     ThingSpeak.setField(2, humidity);
60
61     // write to the ThingSpeak channel
62     int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
63     if(x == 200){
64         WiFi.print("Channel update successful. ");
65         WiFi.print(temperature);
66         WiFi.print(" ");
67         WiFi.println(humidity);
68     } else{
69         WiFi.println("Problem updating channel. HTTP error code " + String(x));
70     }
71
72     WiFi.println("delay(20000); // Wait 20 seconds to update the channel again");
73
74 }
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90

```

Serial Monitor

Writing at 0x00024000... (76 %)
 Writing at 0x00028000... (84 %)
 Writing at 0x0002c000... (92 %)
 Writing at 0x00030000... (100 %)
 Wrote 286352 bytes (20997 compressed) at 0x00000000 in 18.5 seconds (effective 123.8 kbit/s)...
 Hash of data verified.

Leaving...|
 Hard resetting via RTS pin...

Ln 82, Col 30 Generic ESP8266 Module on /dev/ttyUSB0

29. Data terkirim dan terunggah

```

Praktikum3 | Arduino IDE 2.2.1
Praktikum3.ino secrets.h
54
55 void loop() {
56
57     // set the fields with the values
58     ThingSpeak.setField(1, temperature);
59     ThingSpeak.setField(2, humidity);
60
61     // write to the ThingSpeak channel
62     int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
63     if(x == 200){
64         WiFi.print("Channel update successful. ");
65         WiFi.print(temperature);
66         WiFi.print(" ");
67         WiFi.println(humidity);
68     } else{
69         WiFi.println("Problem updating channel. HTTP error code " + String(x));
70     }
71
72     WiFi.println("delay(20000); // Wait 20 seconds to update the channel again");
73
74 }
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90

```

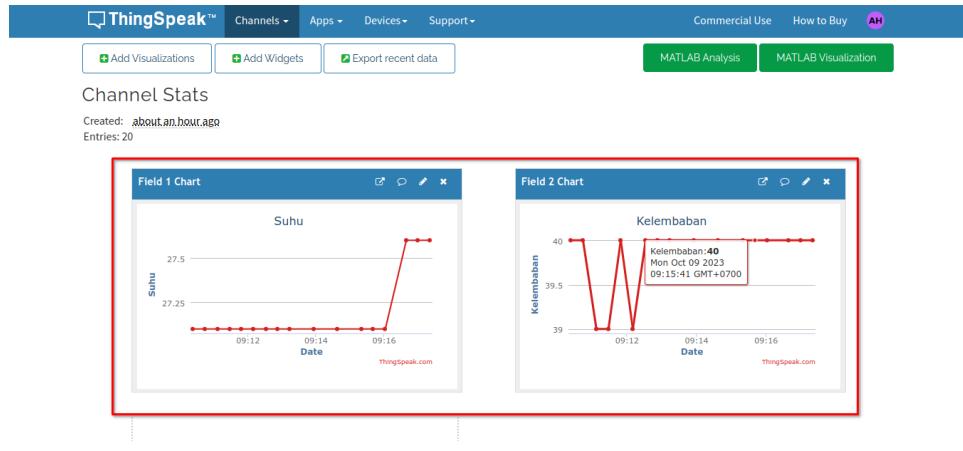
Serial Monitor

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

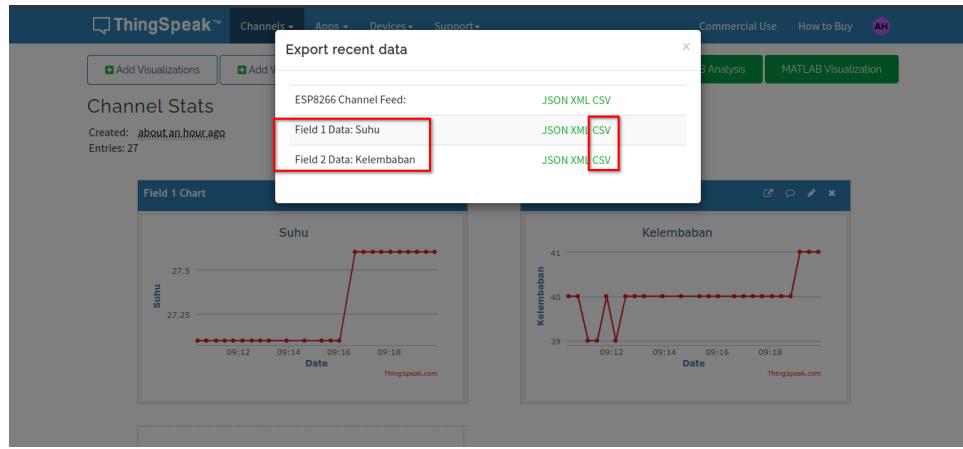
09:17:00.437 -> Channel update successful. 27.60 40.00

No Line Ending | 115200 baud

Ln 82, Col 30 Generic ESP8266 Module on /dev/ttyUSB0



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



Bab 4

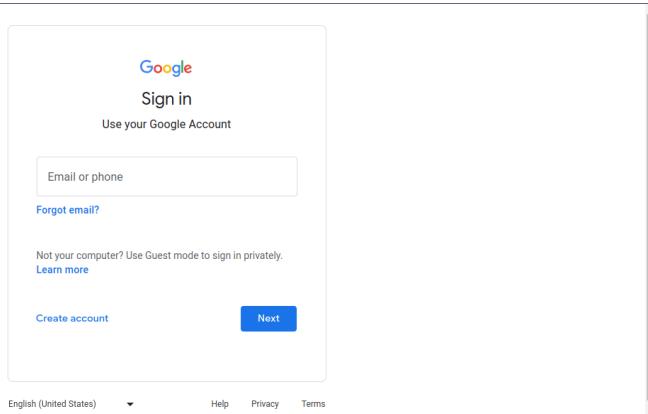
Praktikum 4

4.1 ESP8266, DHT11, dan Firebase Realtime

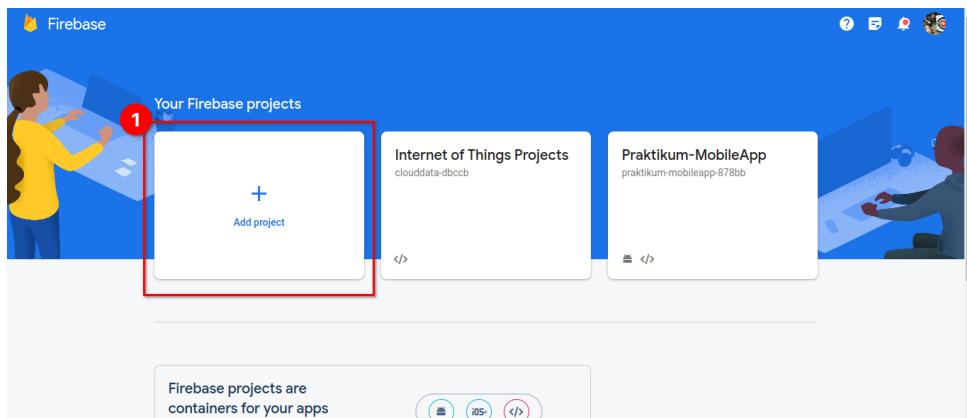
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Firebase Realtime. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 3** yang ada di halaman sebelumnya.

4.2 Tutorial

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



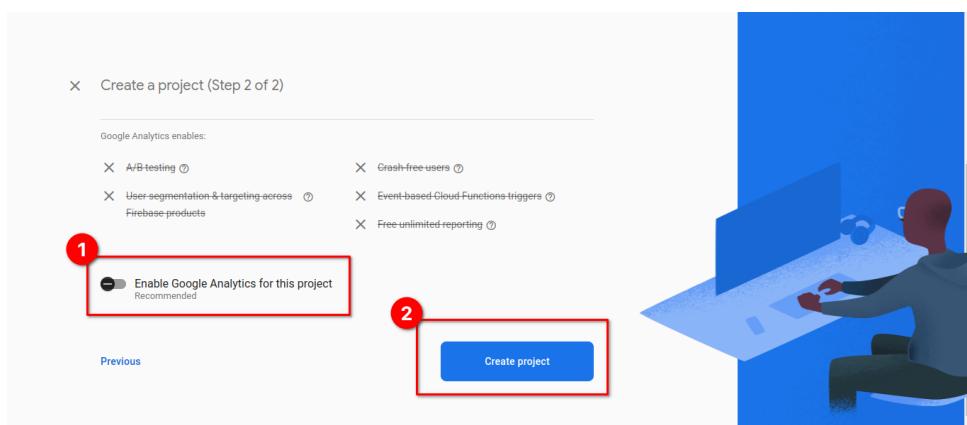
2. Buat projek baru dengan melakukan klik tanda +



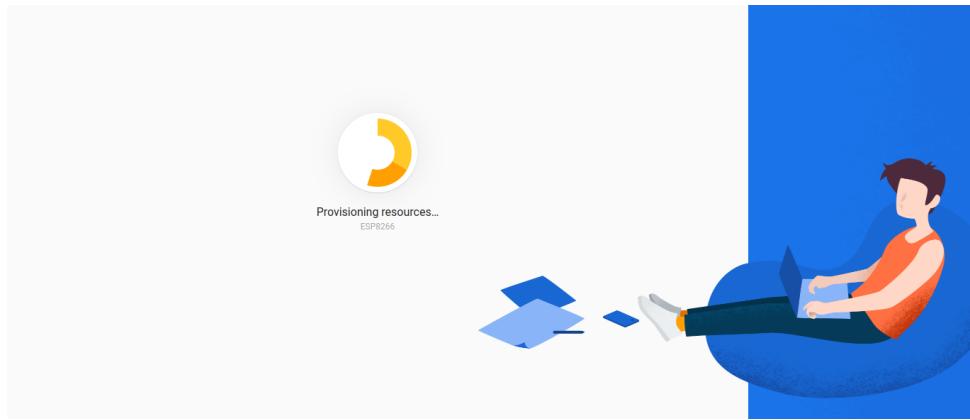
3. Isi nama projek



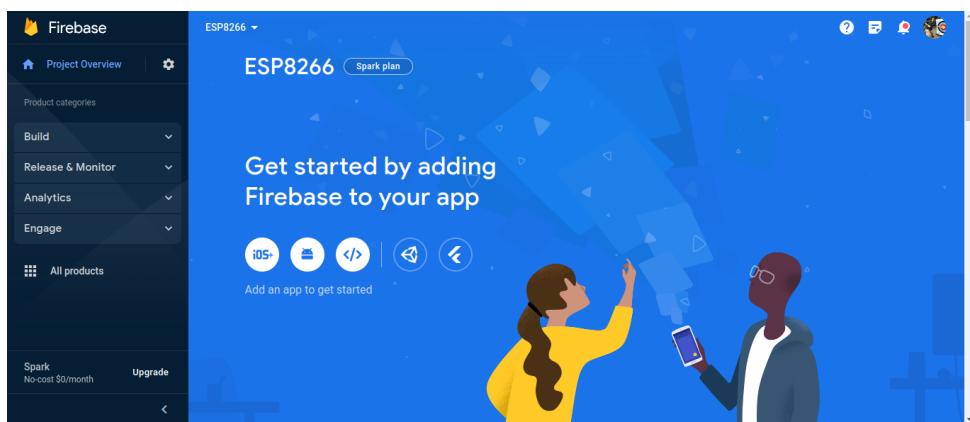
4. Matikan Google Analytic dan klik Create Project



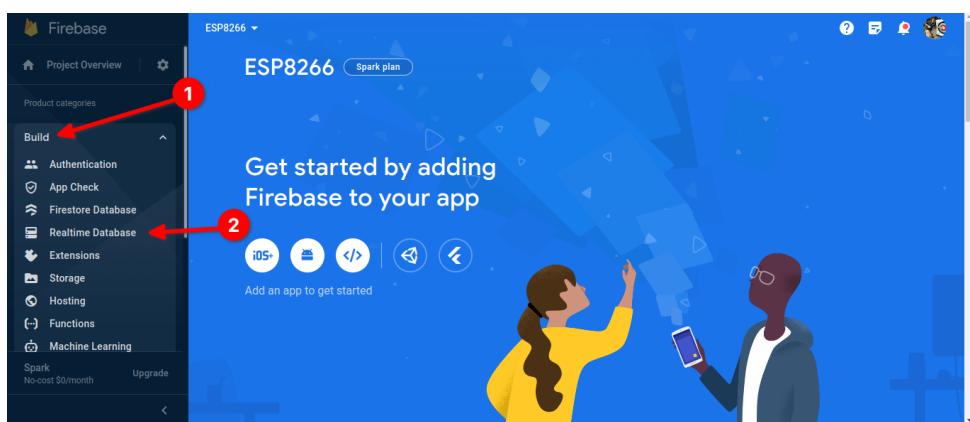
5. Tunggu proses berlangsung dan klik tombol apabila sudah muncul



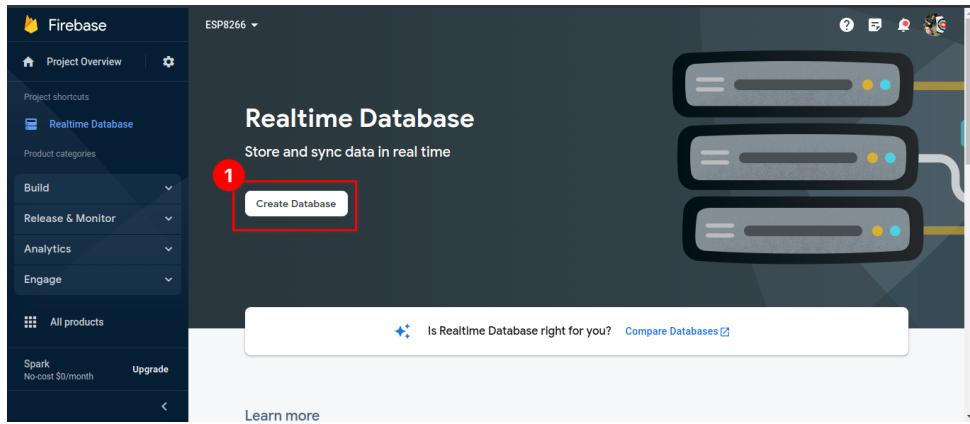
6. Firebase akan menampilkan dasbor sistem



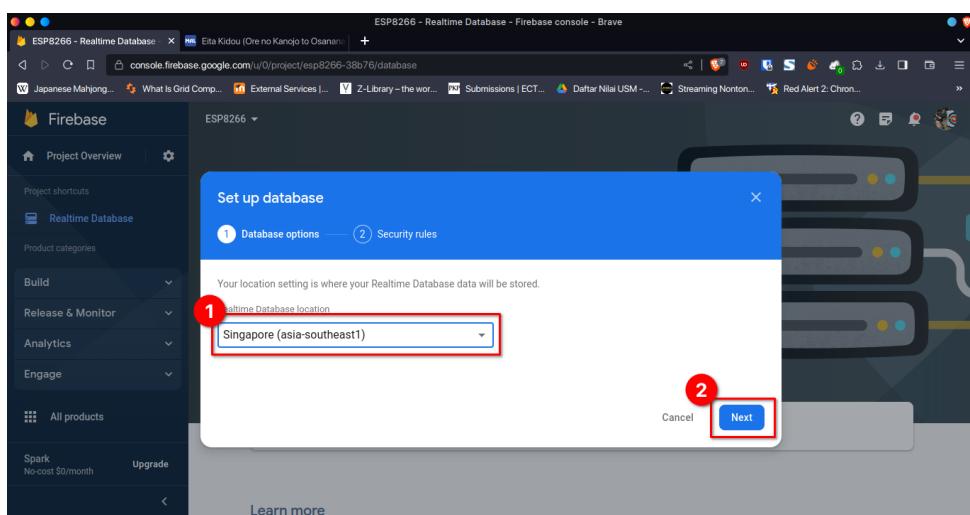
7. Klik Build dan pilih Realtime Database



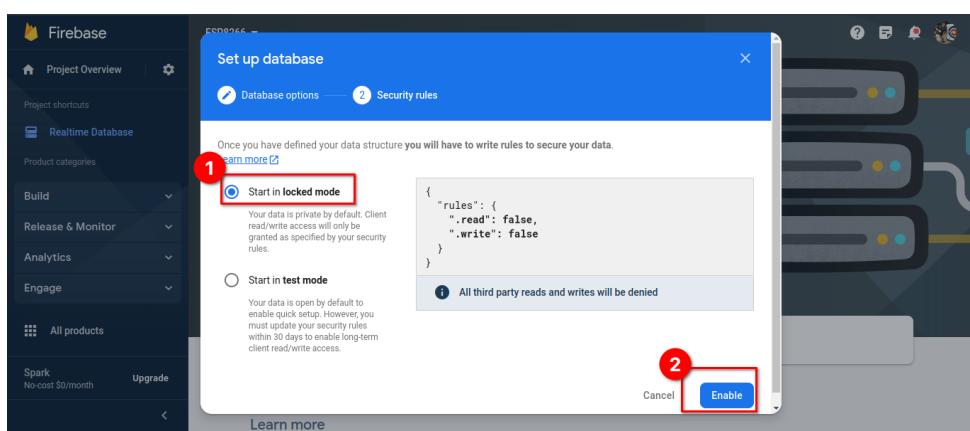
8. Klik Create Database



9. Pilih Lokasi dan Klik Next



10. Pilih Locked Mode dan klik Enable



11. Database sudah dibuat

The screenshot shows the Firebase Realtime Database interface. On the left, there's a sidebar with project settings like Project Overview, Build, Release & Monitor, Analytics, Engage, and All products. The main area is titled 'Realtime Database' and has tabs for Data, Rules, Backups, Usage, and Extensions. Under the Data tab, there's a URL field containing 'https://esp8266-38b76-default-rtdb.firebaseio.com/' and a note 'Protect your Realtime Database resources from abuse, such as billing fraud or phishing'. Below the URL is a text input field with 'https://esp8266-38b76-default-rtdb.firebaseio.com/:null'. At the bottom, it says 'Database location: Singapore (asia-southeast1)'.

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

This screenshot shows the 'Realtime Database' rules editor. It features a sidebar with 'Project Overview' and 'Realtime Database' selected. The main area has tabs for Data, Rules (which is highlighted with a red box and a circled '1'), Backups, Usage, and Extensions. Below the tabs is a blue bar with 'unpublished changes', a 'Publish' button (highlighted with a red box and circled '3'), and a 'Discard' button. A 'Rules playground' button is also present. The code editor shows the following rules:

```

1  {
2    "rules": {
3      ".read": true,
4      ".write": true
5    }
6  }

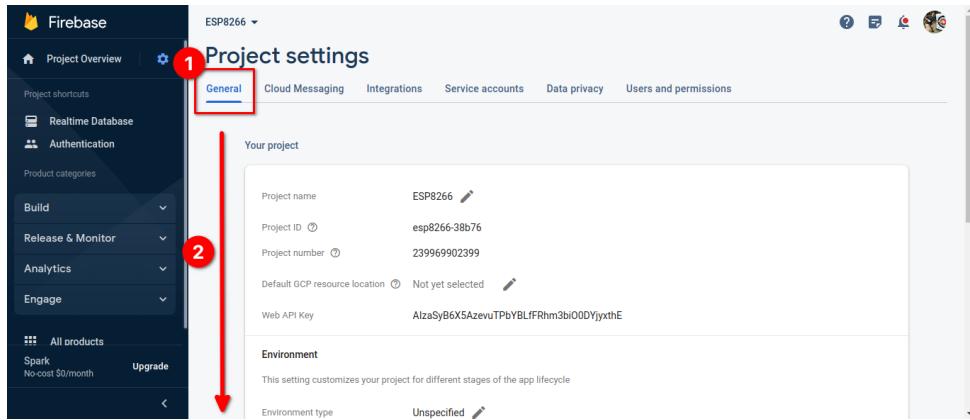
```

The lines '.read: true' and '.write: true' are highlighted with a red box and circled '2'.

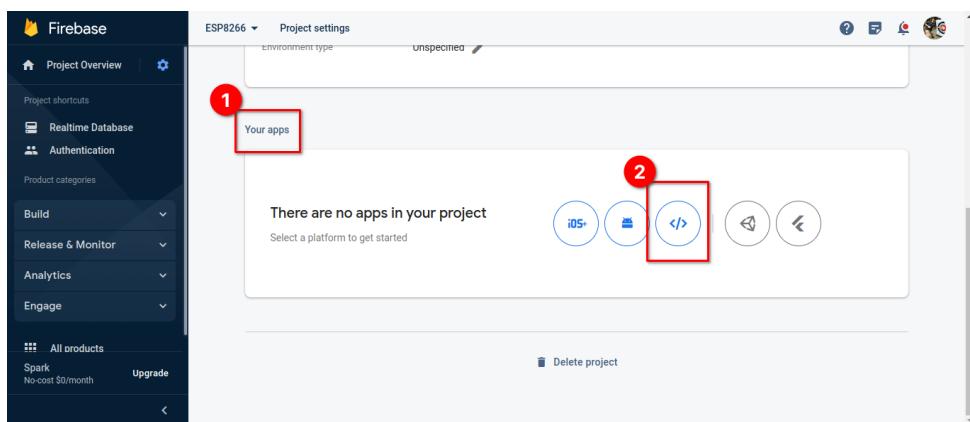
13. Untuk membuat kunci, klik **Roda Gigi Project Overview**, pilih **Project Settings**

This screenshot shows the 'Project Overview' page. A red circle labeled '1' highlights the gear icon in the top right corner. A red circle labeled '2' highlights the 'Project settings' link in the dropdown menu that appears when the gear icon is clicked. The main area shows 'Realtime Database' selected under 'Project shortcuts'.

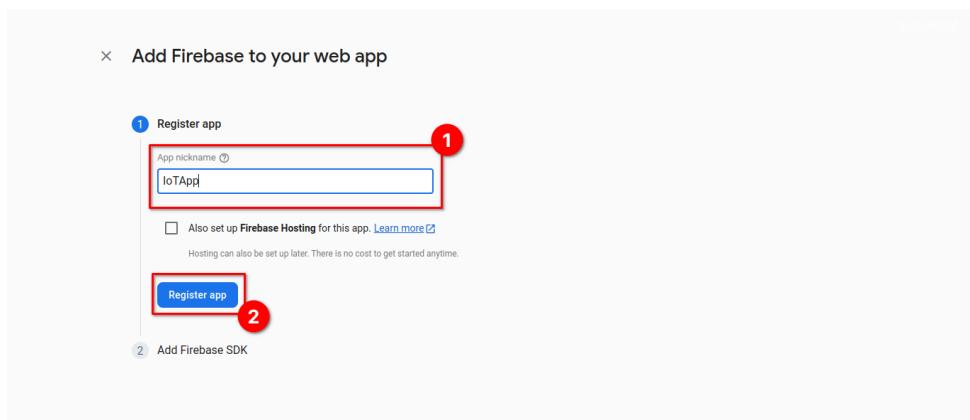
14. Di bagian **General**, scroll turun hingga menemukan **Apps**



15. Di bagian Your Apps pilih Web



16. Isikan nama app, dan pilih Register app



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

```

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

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

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyB6X5azevuTPY[REDACTED]", // 1
  authDomain: "esp8266-38b76.firebaseioapp.com",
  databaseURL: "https://esp8266-38b76-default-rtdb.firebaseio.com", // 2
  projectId: "esp8266-38b76",
  storageBucket: "esp8266-38b76.appspot.com",
  messagingSenderId: "239969902399",
  appId: "1:239969902399:web:8b1411b7b8dccf0252bd8a"
};

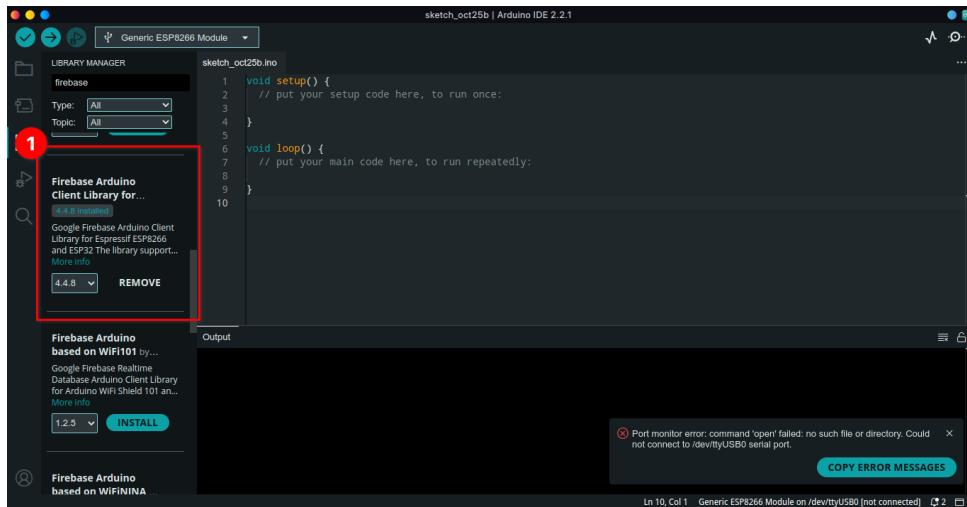
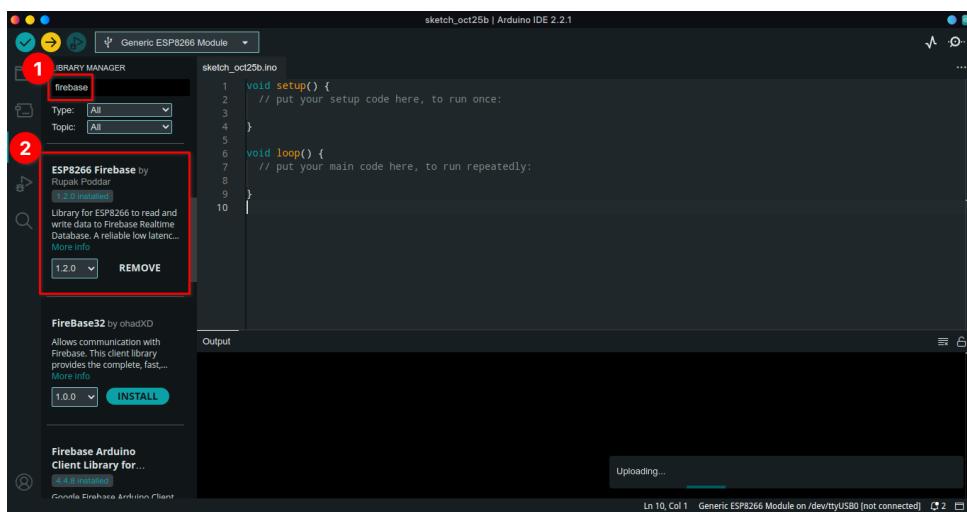
// Initialize Firebase
const app = initializeApp(firebaseConfig);

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

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

```

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



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

```

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

```

20. Hapus beberapa bagian kode berikut:

- Bagian 1

```

Praktikum4 | Arduino IDE 2.2.1
Praktikum4.ino
43 // Define Firebase Data object
44 FirebaseDatabase fbdo;
45 FirebaseAuth auth;
46 FirebaseConfig config;
47
48 unsigned long sendDataPrevMillis = 0;
49
50 if(signed-long:count==0;)
51 {
52     void setup()
53     {
54         Serial.begin(115200);
55         WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
56         Serial.print("Connecting to Wi-Fi");
57         while (WiFi.status() != WL_CONNECTED)
58     }
59
60
61
Output
Ln 52, Col 1 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```

- Bagian 2

```

Praktikum4 | Arduino IDE 2.2.1
Praktikum4.ino
96 json.setDoubleDigits(3);
97 json.add("value", count);
98
99 Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo, "/test/json", &json) ? "ok" : fbdo.errorReason().c_str());
100
101 Serial.printf("Get json... %s\n", Firebase.RTDB.getJSON(&fbdo, "/test/json") ? fbdo.to<FirebaseJson>().raw() : fbdo.errorReason().c_str());
102
103 FirebaseJson jval;
104 Serial.printf("Get json-ref... %s\n", Firebase.RTDB.getJSON(&fbdo, "/test/json", &jVal) ? jVal.raw() : fbdo.errorReason().c_str());
105
106 FirebaseJsonArray arr;
107 arr.setFloatDigits(2);
108 arr.setDoubleDigits(4);
109 arr.add("a", "b", "c", true, -45, -(float)6.1432, 123.45, 9789);
110
111 Serial.printf("Set-array... %s\n", Firebase.RTDB.setArray(&fbdo, "/test/array", &arr) ? "ok" : fbdo.errorReason().c_str());
112
113 Serial.printf("Get-array... %s\n", Firebase.RTDB.getArray(&fbdo, "/test/array") ? fbdo.to<FirebaseJsonArray>().raw() : fbdo.errorReason().c_str());
114
115 Serial.printf("Push-json... %s\n", Firebase.RTDB.pushJSON(&fbdo, "/test/push", &json) ? "ok" : fbdo.errorReason().c_str());
116
117 json.set("value", count + 0.29745);
118 Serial.printf("Update-json... %s\n", Firebase.RTDB.updateNode(&fbdo, "/test/push/" + i, &jdo.pushName(), &json) ? "ok" : fbdo.errorReason().c_str());
119
120
121
122
123
Output
Ln 120, Col 17 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```

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

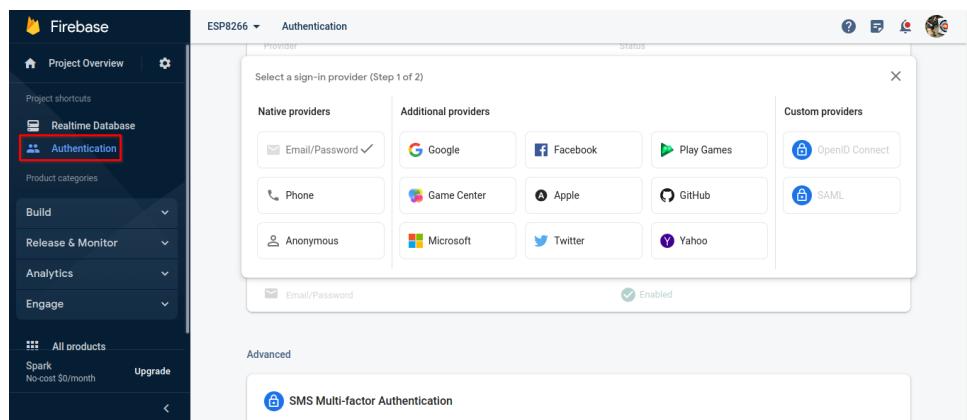
```

Praktikum4 | Arduino IDE 2.2.1
Generic ESP8266 Module
Praktikum4.ino
1 #include <addons/RTDBHelper.h>
2 /* 1. Define the WiFi credentials */
3 #define WIFI_SSID "MikroTik-Net"
4 #define WIFI_PASSWORD "XXXXXXXXXX"
5
6 /* 2. Define the API Key */
7 #define API_KEY "AIzaSyB6X5AzevXXXXXXXXXX"
8
9 /* 3. Define the RTDB URL */
10 #define DATABASE_URL "https://XXXXXXXXXX.firebaseio.com"
11
12 // Define the user email and password that already registered or added in your project
13 #define USER_EMAIL "maulanaahirzani@gmail.com"
14 #define USER_PASSWORD "1234567890"
15
16 FirebaseData fbd;
17 FirebaseAuth auth;
18 FirebaseConfig config;
19
20 unsigned long sendDataPrevMillis = 0;
21
22 unsigned long count = 0;
23
24 void setup()
25 {
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
}

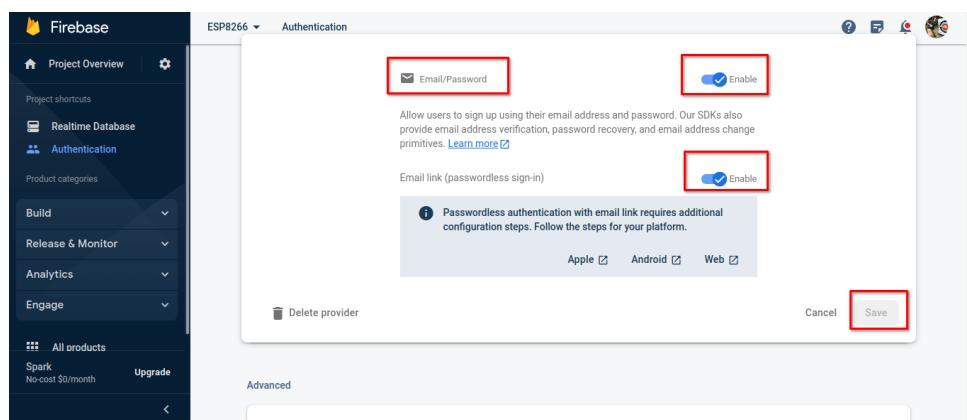
```

Output
DATA 1628 initialized variables

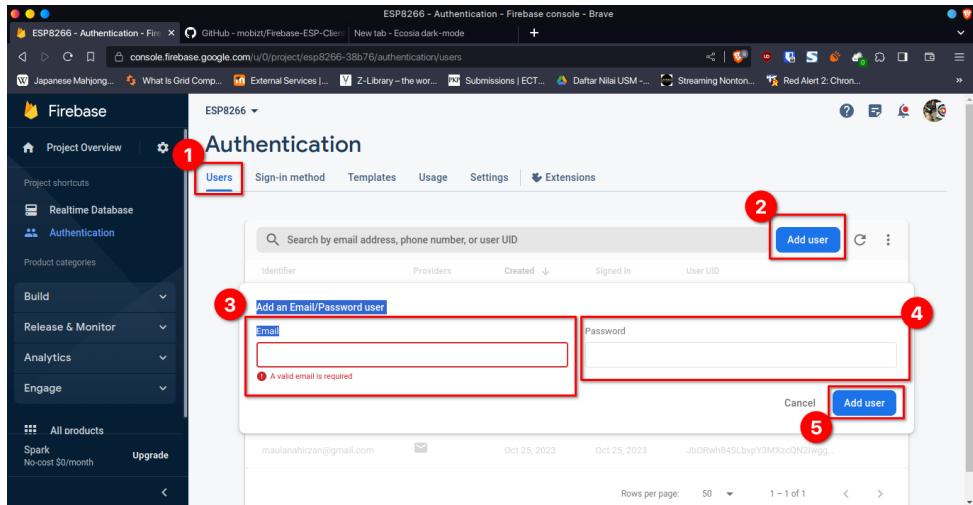
22. Bagian berikutnya adalah akun. Buka kembali **Firebase**, buka menu **Build** lalu **Firebase Authentication**



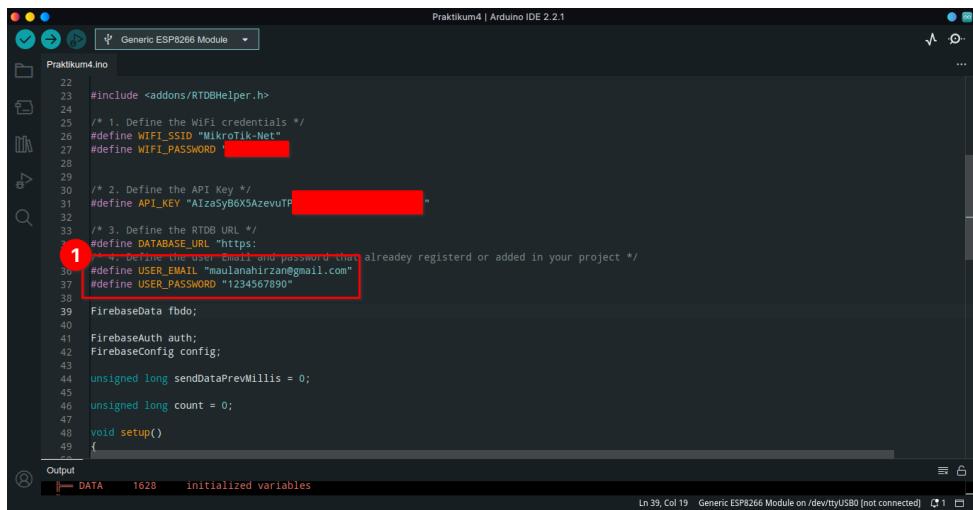
23. Pilih **Email/Password**, klik semua menjadi **Enable**, dan **Save**



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



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



26. Berikutnya adalah menambahkan kode untuk sensor DHT

Potongan Kode

```
#include <DHT.h>

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

```

Praktikum4 | Arduino IDE 2.2.1
Praktikum.ino
1 //include "ESP8266WiFi.h"
2 WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
3 Serial.print("Connecting to Wi-Fi");
4 while (WiFi.status() != WL_CONNECTED)
5 {
6   Serial.print(".");
7   delay(300);
8 }
9 Serial.println();
10 Serial.print("Connected with IP: ");
11
12 #include <DHT.h>
13
14 #define DHTPIN 2
15 #define DHTTYPE DHT11
16 DHT dht(DHTPIN, DHTTYPE);
17
18 void setup()
19 {
20
21   Serial.begin(115200);
22
23   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
24   Serial.print("Connecting to Wi-Fi");
25   while (WiFi.status() != WL_CONNECTED)
26   {
27     Serial.print(".");
28     delay(300);
29   }
30   Serial.println();
31   Serial.print("Connected with IP: ");
32
33   dht.begin();
34
35 }
36
37 void loop()
38 {
39
40   if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
41   {
42     sendDataPrevMillis = millis();
43
44     FirebaseJson json;
45     json.setDoubleDigits(3);
46     json.add("value", count);
47
48     Firebase.publish("data", json);
49   }
50
51   delay(1000);
52 }

```

Output
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
Praktikum.ino
1 //include "ESP8266WiFi.h"
2 WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
3 Serial.print("Connecting to Wi-Fi");
4 while (WiFi.status() != WL_CONNECTED)
5 {
6   Serial.print(".");
7   delay(300);
8 }
9 Serial.println();
10 Serial.print("Connected with IP: ");
11
12 #include <DHT.h>
13
14 #define DHTPIN 2
15 #define DHTTYPE DHT11
16 DHT dht(DHTPIN, DHTTYPE);
17
18 void setup()
19 {
20
21   Serial.begin(115200);
22
23   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
24   Serial.print("Connecting to Wi-Fi");
25   while (WiFi.status() != WL_CONNECTED)
26   {
27     Serial.print(".");
28     delay(300);
29   }
30   Serial.println();
31   Serial.print("Connected with IP: ");
32
33   // Mulai Sensor DHT11
34   dht.begin();
35
36 }
37
38 void loop()
39 {
40
41   if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
42   {
43     sendDataPrevMillis = millis();
44
45     FirebaseJson json;
46     json.setDoubleDigits(3);
47     json.add("value", count);
48
49     Firebase.publish("data", json);
50   }
51
52   delay(1000);
53 }

```

Output
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
Praktikum4.ino
77 config.token_status_callback = tokenStatusCallback;
78 fbdo.setBSSLBufferSize(4096);
79
80
81 Firebase.begin(&config, &auth);
82 Firebase.reconnectNetwork(true);
83
84 // Mulai Sensor DHT11
85 dht.begin();
86
87 void loop()
88 {
89     if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
90     {
91         float temperature = dht.readTemperature();
92         float humidity = dht.readHumidity();
93
94         sendDataPrevMillis = millis();
95
96         FirebaseJson json;
97         json.setDoubleDigits(3);
98         json.add("value", count);
99
100        Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo, "/test/json", &json) ? "ok" : fbdo.errorReason().c_str());
101    }
102 }
103
104
105
106

```

Output
DATA 1628 initialized variables
Ln 93, Col 45 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

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
Praktikum4.ino
79
80
81 Firebase.begin(&config, &auth);
82 Firebase.reconnectNetwork(true);
83
84 // Mulai Sensor DHT11
85 dht.begin();
86
87 void loop()
88 {
89     if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0))
90     {
91         float temperature = dht.readTemperature();
92         float humidity = dht.readHumidity();
93
94         sendDataPrevMillis = millis();
95
96         FirebaseJson json;
97         json.setDoubleDigits(3);
98         json.add("temperature", temperature);
99         json.add("humidity", humidity);
100
101        Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo, "/livedata", &json) ? "ok" : fbdo.errorReason().c_str());
102        Serial.printf("Push json... %s\n", Firebase.RTDB.pushJSON(&fbdo, "/history", &json) ? "ok" : fbdo.errorReason().c_str());
103    }
104 }
105
106

```

Output
writing at uxuuu48uuu... (/9 %)
Ln 103, Col 130 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

30. Verifikasi dan Upload aplikasi

```

Praktikum4 | Arduino IDE 2.2.1
File Generic ESP8266 Mod...
Praktikum4.ino
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106

Output Serial Monitor
Writing at 0x00000000... (1 %)
Writing at 0x00000000... (16 %)
Writing at 0x00001000... (20 %)
Writing at 0x00001000... (35 %)
Writing at 0x00001800... (29 %)
Writing at 0x00001c00... (33 %)
Writing at 0x00002000... (37 %)
Writing at 0x00002400... (41 %)
Writing at 0x00002800... (45 %)
Writing at 0x00002c00... (50 %)
Writing at 0x00003000... (54 %)

Uploading...
Ln 98, Col 33 Generic ESP8266 Module on /dev/ttyUSB0 3

```

31. Data sukses diunggah

```

Praktikum4 | Arduino IDE 2.2.1
File Generic ESP8266 Mod...
Praktikum4.ino
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106

Output Serial Monitor X
Message (Copy to send message to Generic ESP8266 Module on /dev/ttyUSB0)
11:44:03.266 -> onesp8266apssssssssssssssssssssssssKit 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
No Line Ending 115200 baud
Ln 98, Col 33 Generic ESP8266 Module on /dev/ttyUSB0 1

```

32. Hasil

ESP8266 - Realtime Database - Firebase console - Brave

Realtime Database

Data Rules Backups Usage Extensions

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

https://esp8266-38b76-default.firebaseio.com/.json

```

{
  "livedata": {
    "humidity": 44,
    "temperature": 24.1
  }
}
Database location: Singapore (asia-southeast1)

```

Bab 5

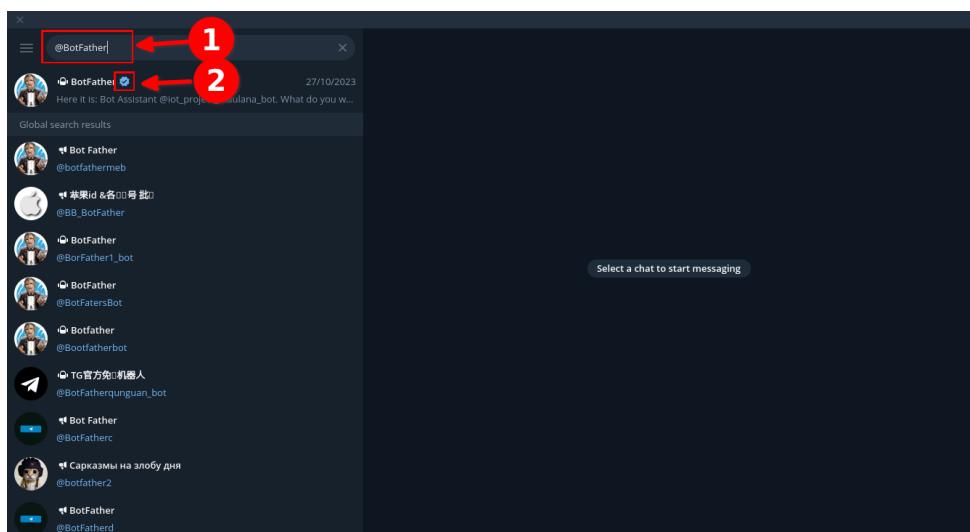
Praktikum 5

5.1 ESP8266, DHT11, dan Telegram Bot

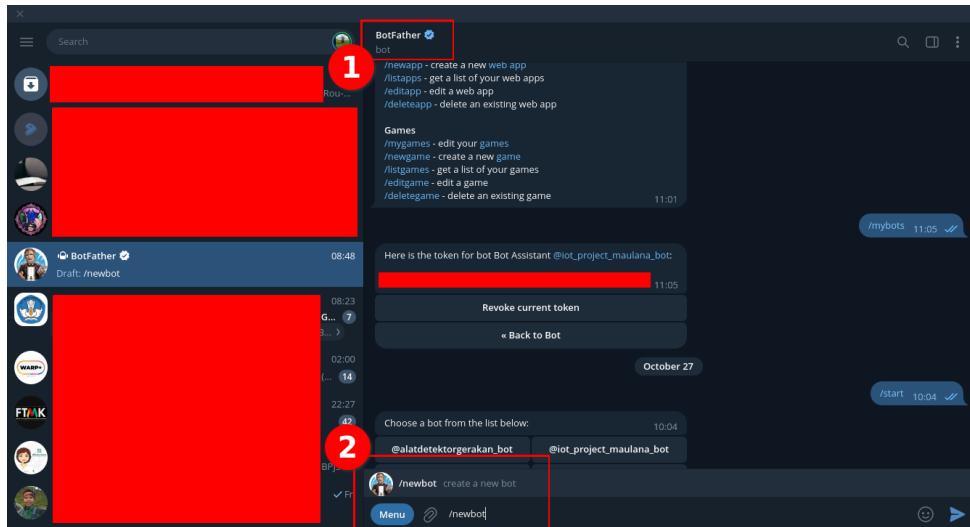
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Telegram Bot. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 4** yang ada di halaman sebelumnya.

5.2 Tutorial

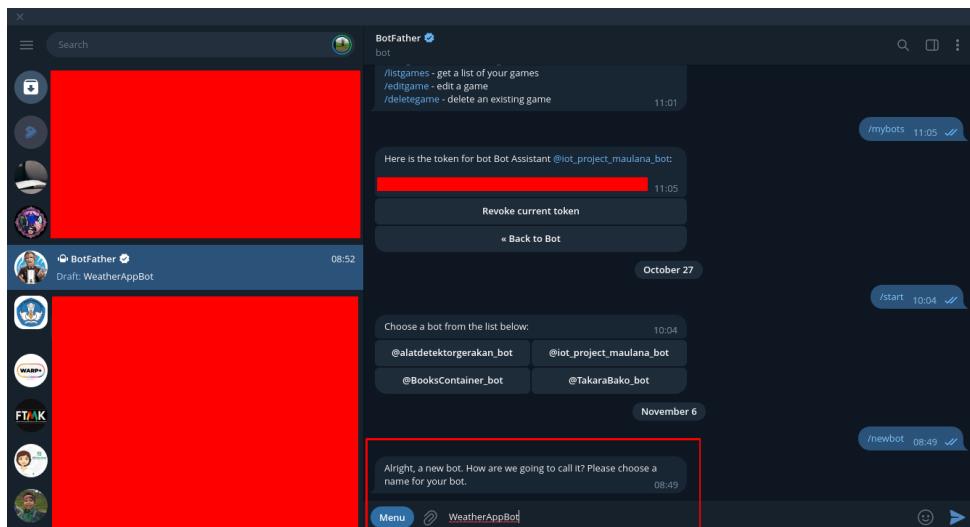
1. Tahap pertama yang dilakukan adalah membuat **Telegram Bot**. Pastikan untuk memiliki Akun Telegram untuk bisa memulai langkah ini
2. Cari **Bog Manager** dengan **@BotFather**



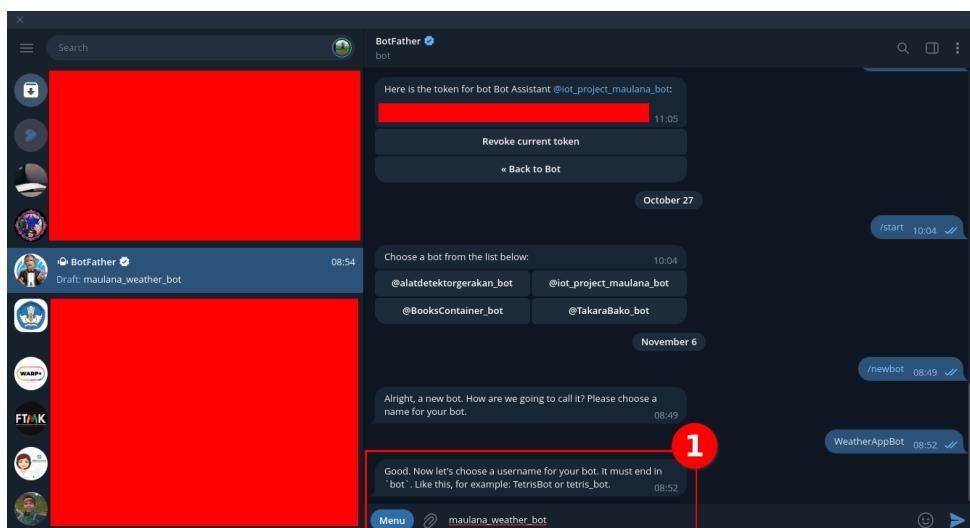
3. Gunakan perintah /newbot untuk membuat **Telegram Bot** baru



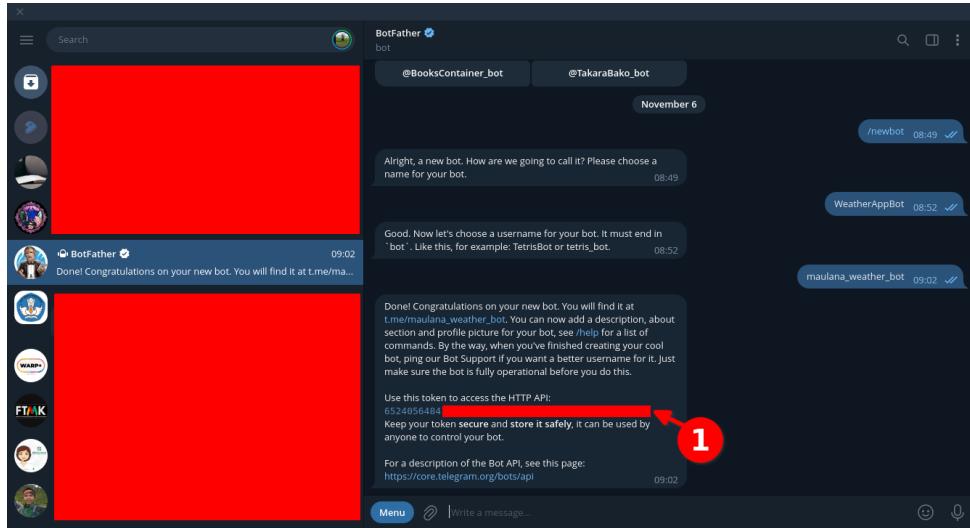
4. Lalu masukkan nama dari **Telegram Bot**



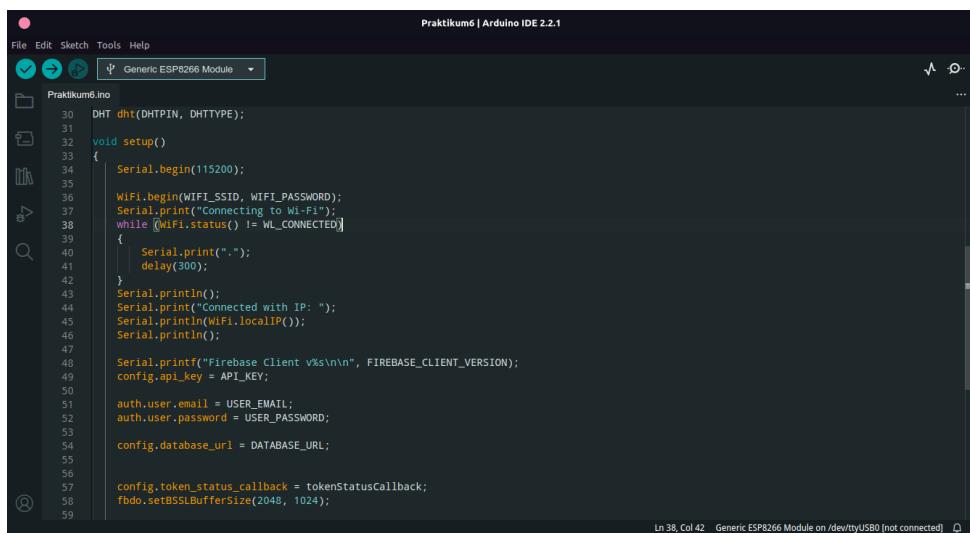
5. Lalu masukkan **username** untuk mempermudah pencarian **Telegram Bot**. Pastikan memiliki akhiran _bot



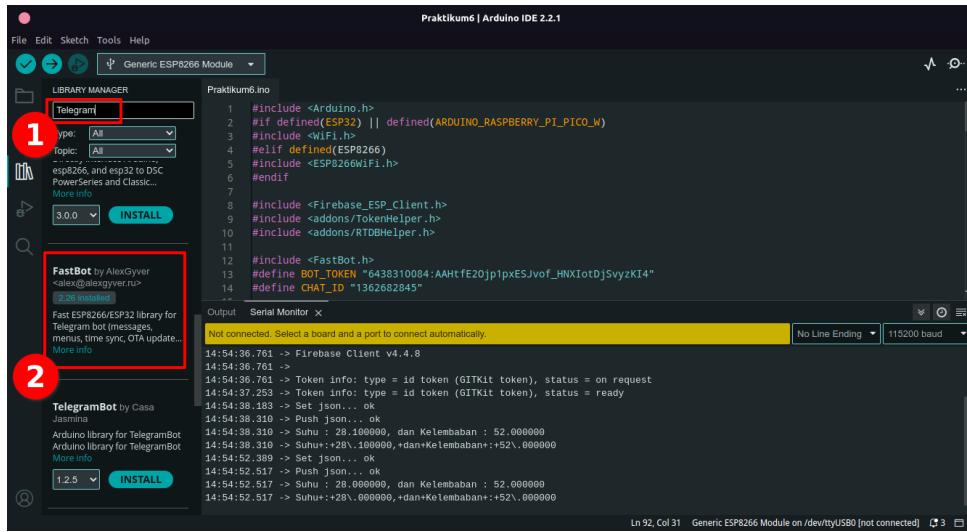
6. Telegram Bot sudah jadi dan **Token API** akan ditampilkan. Simpan baik-baik kode tersebut



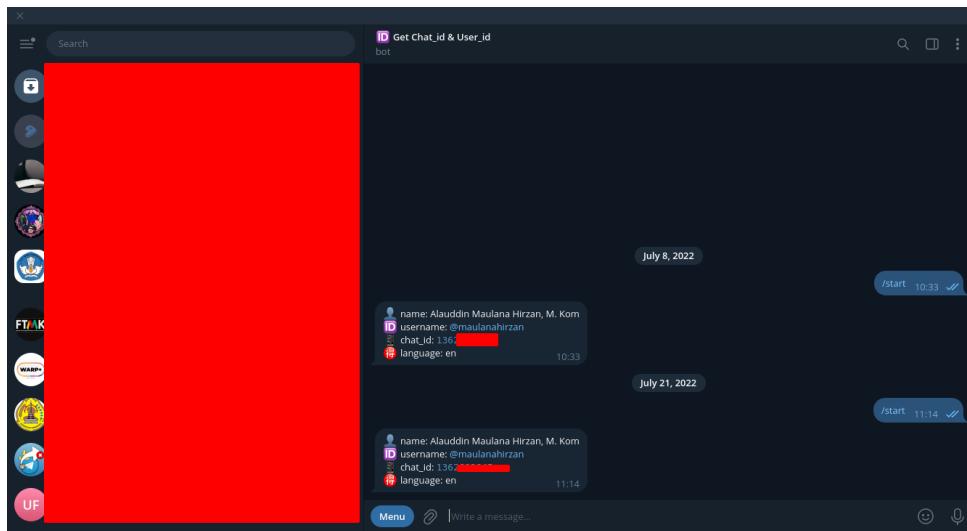
7. Berikutnya adalah membuka kembali **Praktikum 4** dengan menggunakan **Arduino IDE**. Lakukan **Save As** untuk menyimpan sebagai **Praktikum 6**



8. Install Library dengan nama **FastBot**



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



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

Sesudah

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

```

Praktikum6 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Generic ESP8266 Module
Praktikum6.ino
1 #include <Arduino.h>
2 #if defined(ESP32) || defined(ARDUINO_RASPBERRY_PI_PICO_W)
3 #include <WiFi.h>
4 #elif defined(ESP8266)
5 #include <ESP8266WiFi.h>
6 #endif
7
8 #include <Firebase_ESP_Client.h>
9 #include <addons/TokenHelper.h>
10 #include <addons/RTDBHelper.h>
11
12 #include <FastBot.h>
13 #define BOT_TOKEN "6438310084 [REDACTED]"
14 #define CHAT_ID "1362 [REDACTED]"
Output Serial Monitor
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 55496 bytes (401648 compressed) at 0x00000000 in 35.3 seconds (effective 125.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
Ln 92, Col 31 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```

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
27 unsigned long sendDataRecvillis = 0;
28
29 #include <DHT.h>
30 #define DHTPIN 2
31 #define DHTTYPE DHT11
32 DHT dht(DHTPIN, DHTTYPE);
33
34 FastBot bot(BOT_TOKEN);
35
36 void setup()
37 {
38     Serial.begin(115200);
39
40     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
41     Serial.print("Connecting to Wi-Fi");
42     while (WiFi.status() != WL_CONNECTED)
43     {
44         Serial.print(".");
45         delay(300);
46     }
47     Serial.println();
48     Serial.print("Connected with ID: ");
49
50     Writing at 0x00044000... (72 %)
51     Writing at 0x00048000... (88 %)
52     Writing at 0x00050000... (92 %)
53     Writing at 0x00054000... (96 %)
54     Writing at 0x00060000... (100 %)
55     Wrote 55496 bytes (401648 compressed) at 0x00000000 in 35.3 seconds (effective 125.7 kbit/s)...
56     Hash of data verified.
Output Serial Monitor
Ln 8, Col 33 Generic ESP8266 Module on /dev/ttyUSB0 [not connected]

```

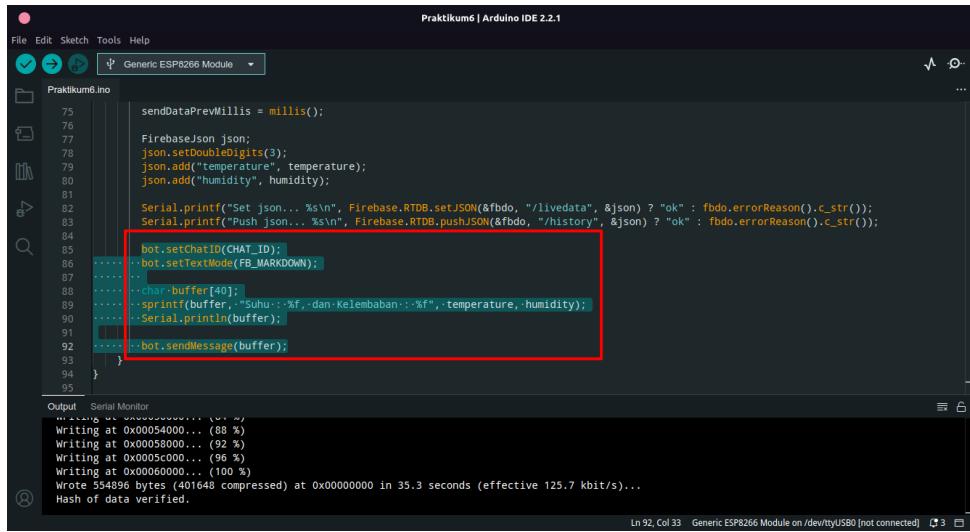
12. Setelah itu, cukup tambahkan kode berikut tepat di akhir fungsi **void loop()**

Sesudah

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

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

bot.sendMessage(buffer);

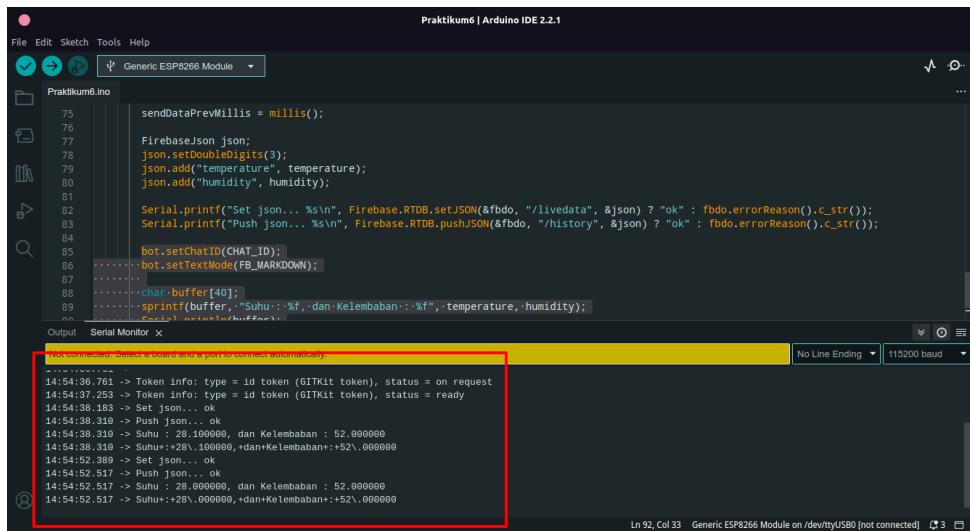


```

Praktikum6 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Generic ESP8266 Module
Praktikum6.ino
75 sendDataPrevMillis = millis();
76
77 FirebaseJson json;
78 json.setDoubleDigits(3);
79 json.add("temperature", temperature);
80 json.add("humidity", humidity);
81
82 Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo, "/livedata", &json) ? "ok" : fbdo.errorReason().c_str());
83 Serial.printf("Push json... %s\n", Firebase.RTDB.pushJSON(&fbdo, "/history", &json) ? "ok" : fbdo.errorReason().c_str());
84
85 bot.setChatID(CHAT_ID);
86 .....bot.setTextMode(FB_MARKDOWN);
87 .....char buffer[40];
88 .....sprintf(buffer, "Suhu::%f,.dan Kelembaban::%f,.temperature,.humidity");
89 .....Serial.println(buffer);
90
91 .....bot.sendMessage(buffer);
92
93 }
94
95
Output Serial Monitor
Writing at 0x00054000... (88 %)
Writing at 0x00058000... (92 %)
Writing at 0x0005c000... (96 %)
Writing at 0x00060000... (100 %)
Wrote 554896 bytes (401648 compressed) at 0x00000000 in 35.3 seconds (effective 125.7 kbit/s)...
Hash of data verified.
Ln 92, Col 33 Generic ESP8266 Module on /dev/ttyUSB0 [not connected] 3

```

13. Verifikasi dan Upload kode ke Perangkat



```

Praktikum6 | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Generic ESP8266 Module
Praktikum6.ino
75 sendDataPrevMillis = millis();
76
77 FirebaseJson json;
78 json.setDoubleDigits(3);
79 json.add("temperature", temperature);
80 json.add("humidity", humidity);
81
82 Serial.printf("Set json... %s\n", Firebase.RTDB.setJSON(&fbdo, "/livedata", &json) ? "ok" : fbdo.errorReason().c_str());
83 Serial.printf("Push json... %s\n", Firebase.RTDB.pushJSON(&fbdo, "/history", &json) ? "ok" : fbdo.errorReason().c_str());
84
85 bot.setChatID(CHAT_ID);
86 .....bot.setTextMode(FB_MARKDOWN);
87 .....char buffer[40];
88 .....sprintf(buffer, "Suhu::%f,.dan Kelembaban::%f,.temperature,.humidity");
89 .....Serial.println(buffer);
90
91 .....bot.sendMessage(buffer);
92
93 }
94
95
Output Serial Monitor x
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
14:54:38.310 -> Set json... ok
14:54:38.310 -> Push json... ok
14:54:39.517 -> Suhu : 29.000000, dan Kelembaban : 52.000000
14:54:39.517 -> Suhu:+29\,000000,dan+Kelembaban:+52\,000000
14:54:39.517 -> Set json... ok
14:54:39.517 -> Push json... ok
14:54:52.517 -> Suhu : 29.000000, dan Kelembaban : 52.000000
14:54:52.517 -> Suhu:+29\,000000,dan+Kelembaban:+52\,000000
14:54:52.517 -> Set json... ok
14:54:52.517 -> Push json... ok
Ln 92, Col 33 Generic ESP8266 Module on /dev/ttyUSB0 [not connected] 3

```

Bab 6

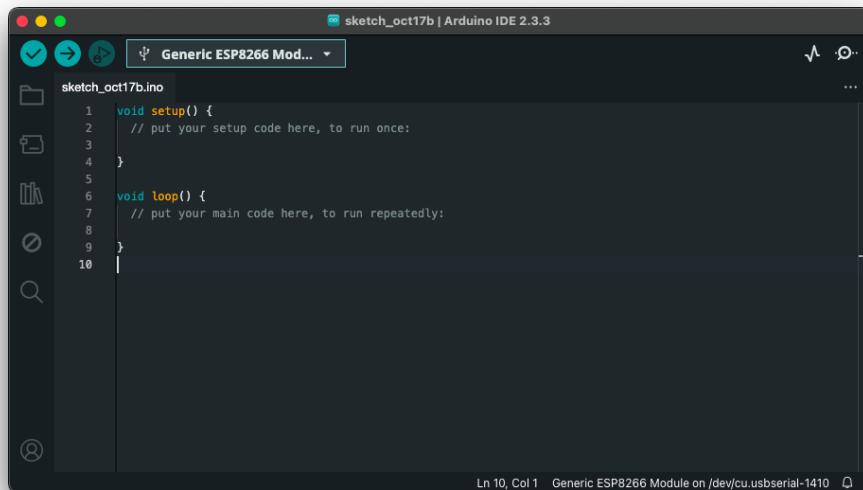
Praktikum 6

6.1 ESP8266, DHT11, dan Linear Regression

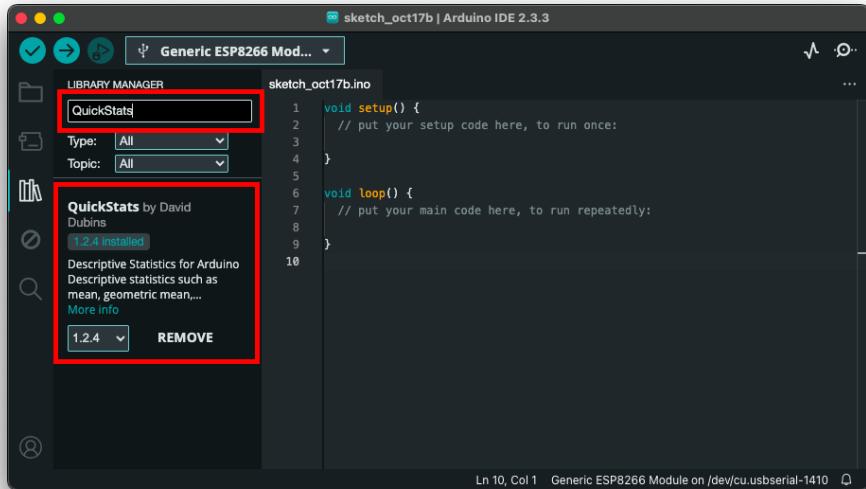
Di bagian ini mahasiswa diajarkan bagaimana mengimplementasikan algoritma *Linear Regression* sederhana dengan perangkat **ESP8266**, dan **DHT11**. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 5** yang ada di halaman sebelumnya.

6.2 Tutorial

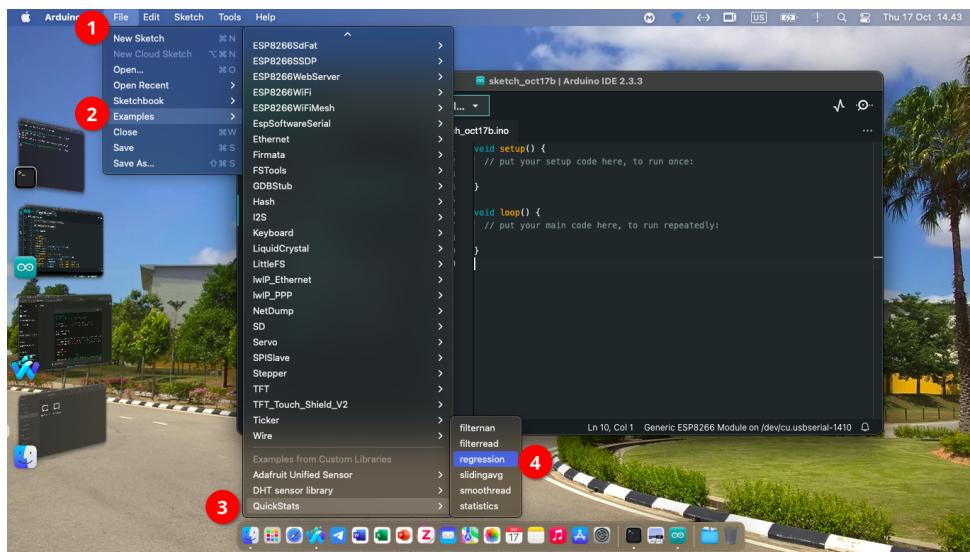
1. Buka kembali Arduino IDE untuk memulai projek baru. ESP8266 tidak harus tercolok.



2. Install Library **QuickStats**



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



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

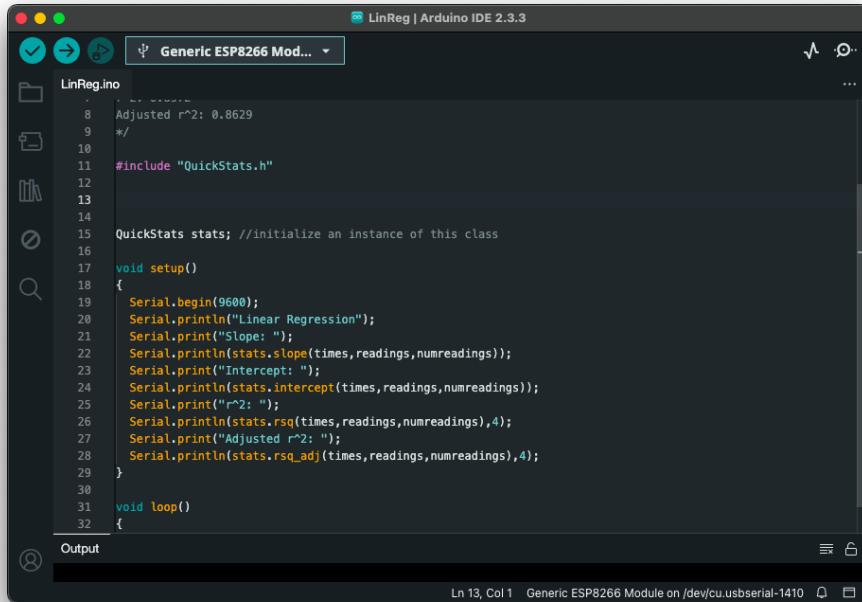
```
LinReg | Arduino IDE 2.3.3
Generic ESP8266 Mod...
LinReg.ino
8     Adjusted r^2: 0.8629
9     */
10
11 #include "QuickStats.h"
12
13 int numreadings = 5;
14 float readings[]={2.15, 3.25, 2.55, 4.64, 7.43};
15 float times[]={0.00, 0.10, 0.20, 0.41, 0.62};
16
17 QuickStats stats; //initialize an instance of this class
18
19 void setup()
20 {
21     Serial.begin(9600);
22     Serial.println("Linear Regression");
23     Serial.print("Slope: ");
24     Serial.println(stats.slope(times,readings,numreadings));
25     Serial.print("Intercept: ");
26     Serial.println(stats.intercept(times,readings,numreadings));
27     Serial.print("r^2: ");
28     Serial.println(stats.rsq(times,readings,numreadings),4);
29     Serial.print("Adjusted r^2: ");
30     Serial.println(stats.rsq_adj(times,readings,numreadings),4);
31 }
32
```

Ln 11, Col 24 Generic ESP8266 Module on /dev/cu.usbserial-1410

5. Hapus kode bagian yang ditandai

```
LinReg | Arduino IDE 2.3.3
Generic ESP8266 Mod...
LinReg.ino
8     Adjusted r^2: 0.8629
9     */
10
11 #include "QuickStats.h"
12
13 int numreadings = 5; // Lines 13-15 are highlighted by a red box
14 float readings[]={2.15, 3.25, 2.55, 4.64, 7.43}; // Lines 13-15 are highlighted by a red box
15 float times[]={0.00, 0.10, 0.20, 0.41, 0.62}; // Lines 13-15 are highlighted by a red box
16
17 QuickStats stats; //initialize an instance of this class
18
19 void setup()
20 {
21     Serial.begin(9600);
22     Serial.println("Linear Regression");
23     Serial.print("Slope: ");
24     Serial.println(stats.slope(times,readings,numreadings));
25     Serial.print("Intercept: ");
26     Serial.println(stats.intercept(times,readings,numreadings));
27     Serial.print("r^2: ");
28     Serial.println(stats.rsq(times,readings,numreadings),4);
29     Serial.print("Adjusted r^2: ");
30     Serial.println(stats.rsq_adj(times,readings,numreadings),4);
31 }
32
```

Ln 15, Col 46 Generic ESP8266 Module on /dev/cu.usbserial-1410



```
LinReg | Arduino IDE 2.3.3
Generic ESP8266 Mod...
LinReg.ino
8     Adjusted r^2: 0.8629
9     */
10
11 #include "QuickStats.h"
12
13
14 QuickStats stats; //initialize an instance of this class
15
16 void setup()
17 {
18     Serial.begin(9600);
19     Serial.println("Linear Regression");
20     Serial.print("Slope: ");
21     Serial.println(stats.slope(times,readings,numreadings));
22     Serial.print("Intercept: ");
23     Serial.println(stats.intercept(times,readings,numreadings));
24     Serial.print("r^2: ");
25     Serial.println(stats.rsq(times,readings,numreadings),4);
26     Serial.print("Adjusted r^2: ");
27     Serial.println(stats.rsq_adj(times,readings,numreadings),4);
28 }
29
30 void loop()
31 {
32 }
```

Ln 13, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410

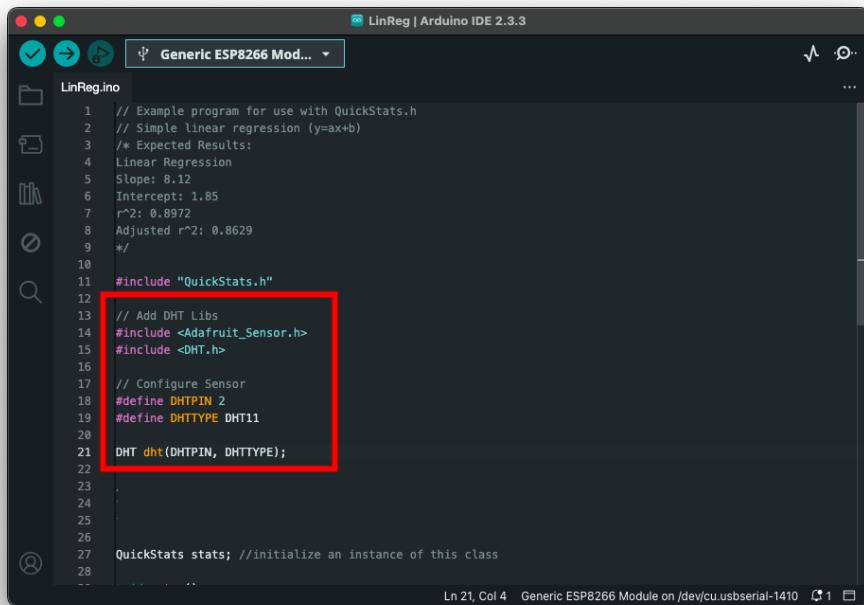
6. Di bagian bawah dari `#include "QuickStats.h"`, tambahkan kode berikut untuk sensor DHT11:

Kode Program

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

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

DHT dht(DHTPIN, DHTTYPE);
```



```

LinReg | Arduino IDE 2.3.3
Generic ESP8266 Mod...
LinReg.ino
1 // Example program for use with QuickStats.h
2 // Simple linear regression (y=ax+b)
3 /* Expected Results:
4 Linear Regression
5 Slope: 8.12
6 Intercept: 1.85
7 r^2: 0.8972
8 Adjusted r^2: 0.8629
9 */
10
11 #include "QuickStats.h"
12
13 // Add DHT Libs
14 #include <Adafruit_Sensor.h>
15 #include <DHT.h>
16
17 // Configure Sensor
18 #define DHTPIN 2
19 #define DHTTYPE DHT11
20
21 DHT dht(DHTPIN, DHTTYPE);
22
23
24
25
26
27 QuickStats stats; //initialize an instance of this class
28

```

Ln 21, Col 4 Generic ESP8266 Module on /dev/cu.usbserial-1410 ⚡ 1

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

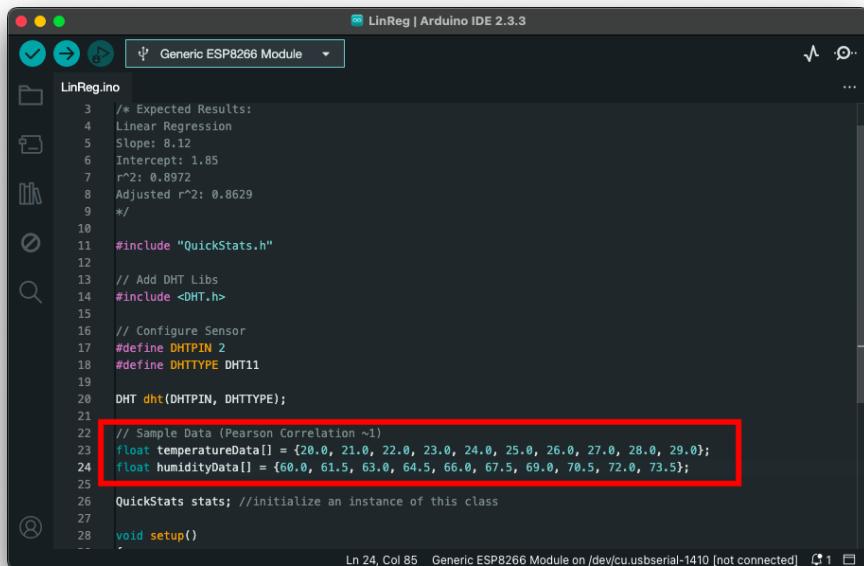
Kode Program

```

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

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

```



```

LinReg | Arduino IDE 2.3.3
LinReg.ino
3  /* Expected Results:
4  Linear Regression
5  Slope: 8.12
6  Intercept: 1.85
7  r^2: 0.8972
8  Adjusted r^2: 0.8629
9  */
10
11 #include "QuickStats.h"
12
13 // Add DHT Libs
14 #include <DHT.h>
15
16 // Configure Sensor
17 #define DHTPIN 2
18 #define DHTTYPE DHT11
19
20 DHT dht(DHTPIN, DHTTYPE);
21
22 // Sample Data (Pearson Correlation ~1)
23 float temperatureData[] = {20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0};
24 float humidityData[] = {60.0, 61.5, 63.0, 64.5, 66.0, 67.5, 69.0, 70.5, 72.0, 73.5};
25
26 QuickStats stats; //initialize an instance of this class
27
28 void setup()
29 {
30
31 }
32
33 void loop()
34 {
35
36 }

```

Ln 24, Col 85 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected] ↻ 1

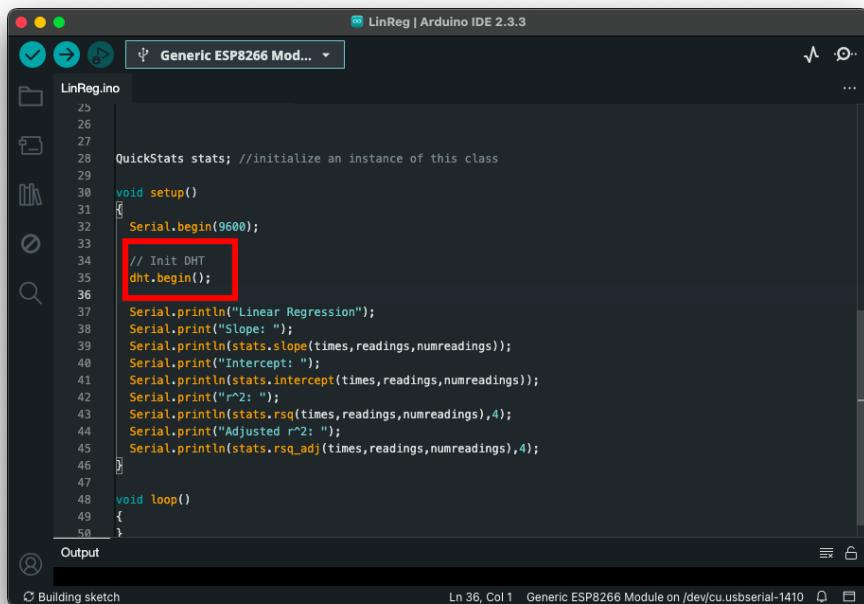
8. Setelah mendapatkan dana *training* untuk **Regresi Linier**, maka langkah berikutnya adalah menambahkan kode inisialisasi untuk sensor **DHT 11** dengan kode berikut setelah kode **Serial.begin(9600);**:

Kode Program

```

// Init DHT
dht.begin();

```



```

LinReg | Arduino IDE 2.3.3
LinReg.ino
25
26
27
28 QuickStats stats; //initialize an instance of this class
29
30 void setup()
31 {
32   Serial.begin(9600);
33
34 // Init DHT
35   dht.begin();
36
37   Serial.println("Linear Regression");
38   Serial.print("Slope: ");
39   Serial.println(stats.slope(times,readings,numreadings));
40   Serial.print("Intercept: ");
41   Serial.println(stats.intercept(times,readings,numreadings));
42   Serial.print("r^2: ");
43   Serial.println(stats.rsq(times,readings,numreadings),4);
44   Serial.print("Adjusted r^2: ");
45   Serial.println(stats.rsq_adj(times,readings,numreadings),4);
46 }
47
48 void loop()
49 {
50 }

```

Output

Building sketch

Ln 36, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 ↻ 1

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

```
LinReg | Arduino IDE 2.3.3
LinReg.ino
25
26 QuickStats stats; //initialize an instance of this class
27
28 void setup()
29 {
30     Serial.begin(9600);
31
32     // Init DHT
33     dht.begin();
34
35     Serial.println("Linear-Regression");
36     Serial.print("Slope:.");
37     Serial.println(stats.slope(times,readings,numreadings));
38     Serial.print("Intercept:.");
39     Serial.println(stats.intercept(times,readings,numreadings));
40     Serial.print("r^2:.");
41     Serial.println(stats.rsq(times,readings,numreadings),4);
42     Serial.print("Adjusted-r^2:.");
43     Serial.println(stats.rsq_ad(times,readings,numreadings),4)];
44 }
45
46 void loop()
47 {
48
49 }
```

Ln 43, Col 63 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected] ↻ 1

```
LinReg | Arduino IDE 2.3.3
LinReg.ino
17 #define DHTPIN 2
18 #define DHTTYPE DHT11
19
20 DHT dht(DHTPIN, DHTTYPE);
21
22 // Sample Data (Pearson Correlation ~1)
23 float temperatureData[10] = {20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0};
24 float humidityData[10] = {60.0, 61.5, 63.0, 64.5, 66.0, 67.5, 69.0, 70.5, 72.0, 73.5};
25
26 QuickStats stats; //initialize an instance of this class
27
28 void setup()
29 {
30     Serial.begin(9600);
31
32     // Init DHT
33     dht.begin();
34
35 }
36
37
38 void loop()
39 {
40
41 }
```

Ln 35, Col 3 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected] ↻ 1

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

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

```

LinReg | Arduino IDE 2.3.3
LinReg.ino
11 // DHT Library
12
13 // Add DHT Libs
14 #include <DHT.h>
15
16 // Configure Sensor
17 #define DHTPIN 2
18 #define DHTTYPE DHT11
19
20 DHT dht(DHTPIN, DHTTYPE);
21
22 // Sample Data (Pearson Correlation ~1)
23 float temperatureData[] = {20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0};
24 float humidityData[] = {60.0, 61.5, 63.0, 64.5, 66.0, 67.5, 69.0, 70.5, 72.0, 73.5};
25
26 QuickStats stats; //initialize an instance of this class
27
28 float slope = 0.0;
29 float intercept = 0.0;
30
31 void setup()
32 {
33   Serial.begin(9600);
34
35   // Init DHT
36   dht.begin();
37
38   // Calculate Slope of Temperature using Humidity Data
39   slope = stats.slope(humidityData, temperatureData, 10);
40   intercept = stats.intercept(humidityData, temperatureData, 10);
41 }
42
43 void loop()
44 {
45
46 }

```

Ln 29, Col 23 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected] 1 2

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

Kode Program

```

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

```

```

LinReg | Arduino IDE 2.3.3
LinReg.ino
11 // DHT Library
12
13 // Add DHT Libs
14 #include <DHT.h>
15
16 // Configure Sensor
17 #define DHTPIN 2
18 #define DHTTYPE DHT11
19
20 DHT dht(DHTPIN, DHTTYPE);
21
22 // Sample Data (Pearson Correlation ~1)
23 float temperatureData[] = {20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0};
24 float humidityData[] = {60.0, 61.5, 63.0, 64.5, 66.0, 67.5, 69.0, 70.5, 72.0, 73.5};
25
26 QuickStats stats; //initialize an instance of this class
27
28 float slope = 0.0;
29 float intercept = 0.0;
30
31 void setup()
32 {
33   Serial.begin(9600);
34
35   // Init DHT
36   dht.begin();
37
38   // Calculate Slope of Temperature using Humidity Data
39   slope = stats.slope(humidityData, temperatureData, 10);
40   intercept = stats.intercept(humidityData, temperatureData, 10);
41 }
42
43 void loop()
44 {
45
46 }

```

2 New Notifications

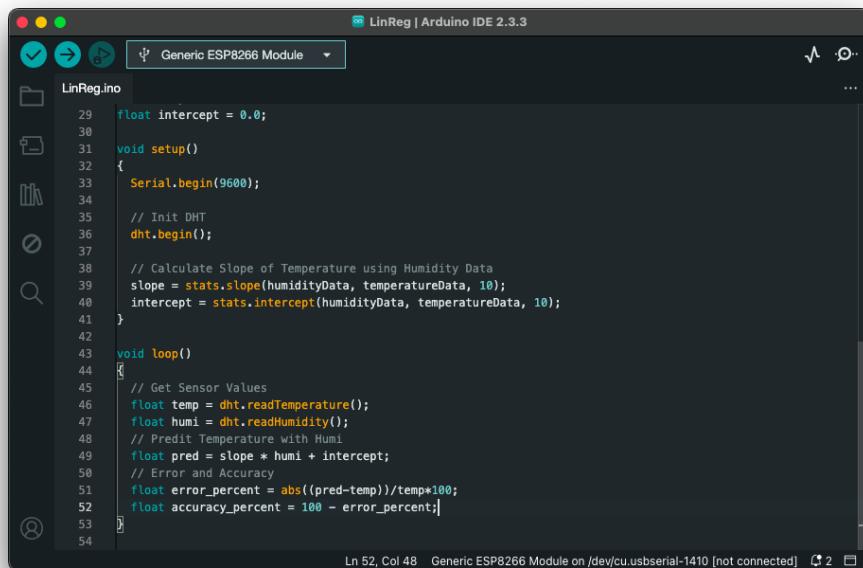
Ln 45, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 1 2

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

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

Kode Program

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

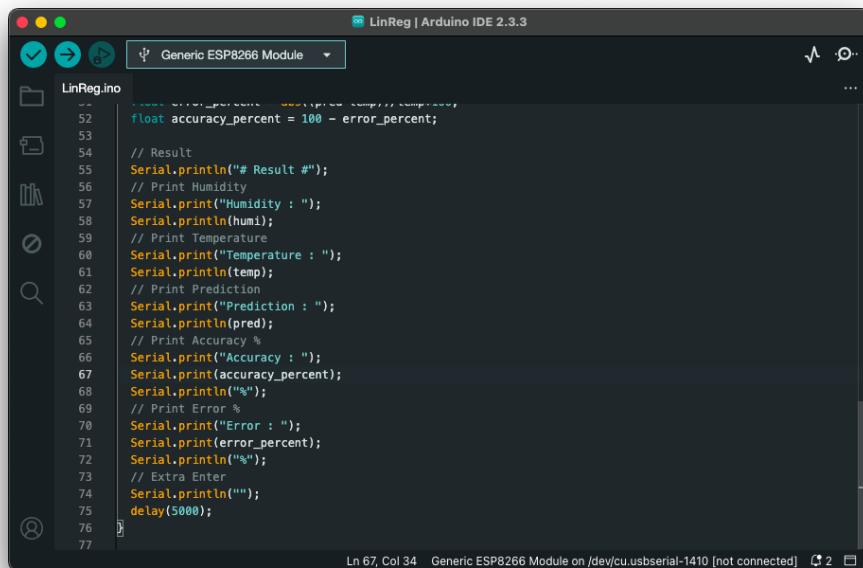


The screenshot shows the Arduino IDE interface with the title bar "LinReg | Arduino IDE 2.3.3". The central area displays the code for "LinReg.ino". The code is a C++ program using the DHT library to read temperature and humidity from a sensor. It calculates a linear regression line (slope and intercept) and then uses this line to predict temperature based on humidity. It also calculates the percentage error of the prediction relative to the actual temperature reading. The code is well-formatted with line numbers and color-coded keywords.

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

Kode Program

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



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

LinReg | Arduino IDE 2.3.3

LinReg.ino

62 // Print Prediction

Output Serial Monitor

Message (Enter to send message to 'Generic ESP8266 Module' on '/dev/cu.usbserial-1410')

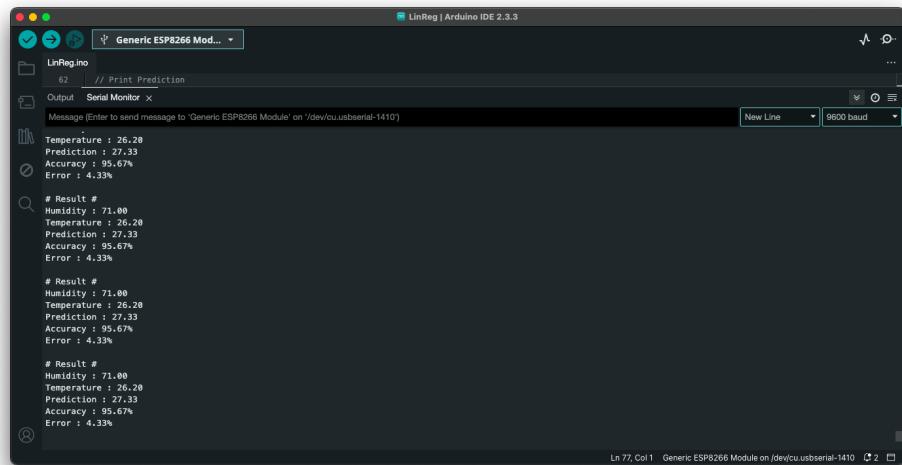
Temperature : 26.20
Prediction : 27.33
Accuracy : 95.67%
Error : 4.33%

Result #
Humidity : 71.00
Temperature : 26.20
Prediction : 27.33
Accuracy : 95.67%
Error : 4.33%

Result #
Humidity : 71.00
Temperature : 26.20
Prediction : 27.33
Accuracy : 95.67%
Error : 4.33%

Result #
Humidity : 71.00
Temperature : 26.20
Prediction : 27.33
Accuracy : 95.67%
Error : 4.33%

Ln 77, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410



Bab 7

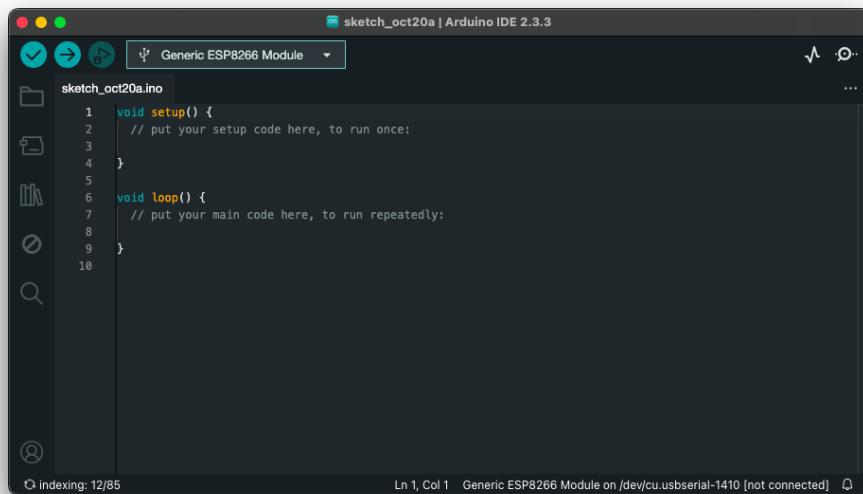
Praktikum 7

7.1 ESP8266, DHT11, dan Fuzzy Logic

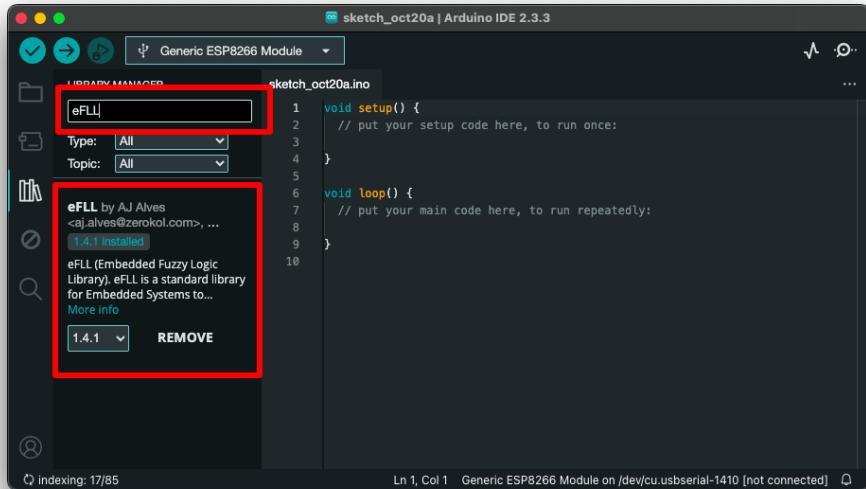
Di bagian ini mahasiswa diajarkan bagaimana mengimplementasikan algoritma *Fuzzy Logic* sederhana dengan perangkat **ESP8266**, dan **DHT11**. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 6** yang ada di halaman sebelumnya.

7.2 Tutorial

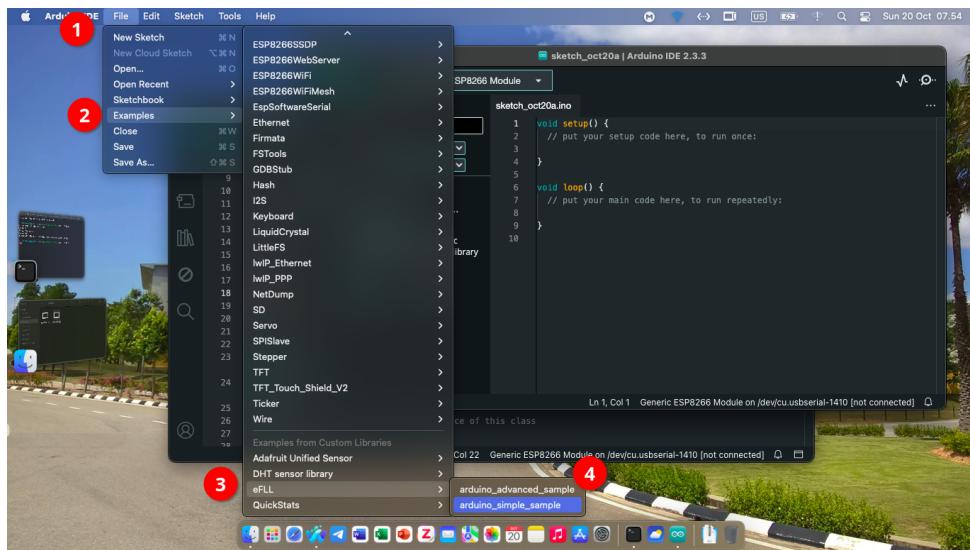
1. Buka Arduino IDE kembali



2. Install Library eFLL



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



```

1 #include <Fuzzy.h>
2
3 // Instantiating a Fuzzy object
4 Fuzzy *fuzzy = new Fuzzy();
5
6 void setup()
{
7     // Set the Serial output
8     Serial.begin(9600);
9     // Set a random seed
10    randomSeed(analogRead(0));
11
12    // Instantiating a FuzzyInput object
13    FuzzyInput *distance = new FuzzyInput(1);
14    // Instantiating a FuzzySet object
15    FuzzySet *small = new FuzzySet(0, 20, 20, 40);
16    // Including the FuzzySet into FuzzyInput
17    distance->addFuzzySet(small);
18    // Instantiating a Fuzzyset object
19    FuzzySet *safe = new FuzzySet(30, 50, 50, 70);
20    // Including the FuzzySet into FuzzyInput
21    distance->addFuzzySet(safe);
22

```

indexing: 10/85 Ln 1, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]

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

Kode Program

```

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

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

DHT dht(DHTPIN, DHTTYPE);

```

```

1 #include <Fuzzy.h>
2
3 // Add DHT Libs
4 #include <DHT.h>
5
6 // Configure Sensor
7 #define DHTPIN 2
8 #define DHTTYPE DHT11
9
10 DHT dht(DHTPIN, DHTTYPE);
11
12 // Instantiating a Fuzzy object
13 Fuzzy *fuzzy = new Fuzzy();
14
15 void setup()
{
16     // Set the Serial output
17     Serial.begin(9600);
18     // Set a random seed
19     randomSeed(analogRead(0));
20
21     // Instantiating a FuzzyInput object
22

```

Ln 3, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]

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

```

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

```

Ln 20, Col 29 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]

- Kemudian hapus lagi bagian kode yang ditunjuk oleh gambar

```

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

```

Ln 35, Col 34 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]

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

Kode Program

```

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

```

```
FL.ino
13 Fuzzy *fuzzy = new Fuzzy();
14
15 void setup()
16 {
17     // Set the Serial output
18     Serial.begin(9600);
19
20     // Define Temperature Range (Cold, Warm, and Hot)
21     FuzzyInput *temp = new FuzzyInput();
22     FuzzySet *cold = new FuzzySet(0, -15, -20);
23     temp->addFuzzySet(cold);
24     FuzzySet *warm = new FuzzySet(15, -20, -30, -35);
25     temp->addFuzzySet(warm);
26     FuzzySet *hot = new FuzzySet(30, -35, -100, -100);
27     temp->addFuzzySet(hot);
28     fuzzy->addFuzzyInput(temp);
29
30     // Instantiating a FuzzyOutput objects
31     FuzzyOutput *speed = new FuzzyOutput(1);
32     // Instantiating a FuzzySet object
33     FuzzySet *slow = new FuzzySet(0, 10, 10, 20);
34     // Including the FuzzySet into FuzzyOutput
35     speed->addFuzzySet(slow);
36
37     // Instantiating a FuzzySet object
38     FuzzySet *average = new FuzzySet(10, -20, -30, -40);
39     // Including the FuzzySet into FuzzyOutput
40     speed->addFuzzySet(average);
41
42     // Instantiating a FuzzySet object
43     FuzzySet *fast = new FuzzySet(30, -40, -40, -50);
44     // Including the FuzzySet into FuzzyOutput
45     speed->addFuzzySet(fast);
46
47     fuzzy->addFuzzyOutput(speed);
48 }
```

Ln 20, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]

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

```
FL.ino
13 Fuzzy *fuzzy = new Fuzzy();
14
15 void setup()
16 {
17     // Set the Serial output
18     Serial.begin(9600);
19
20     // Define Temperature Range (Cold, Warm, and Hot)
21     FuzzyInput *temp = new FuzzyInput();
22     FuzzySet *cold = new FuzzySet(0, -15, -20);
23     temp->addFuzzySet(cold);
24     FuzzySet *warm = new FuzzySet(15, -20, -30, -35);
25     temp->addFuzzySet(warm);
26     FuzzySet *hot = new FuzzySet(30, -35, -100, -100);
27     temp->addFuzzySet(hot);
28     fuzzy->addFuzzyInput(temp);
29
30     // Instantiating a FuzzyOutput objects
31     FuzzyOutput *speed = new FuzzyOutput(1);
32     // Instantiating a FuzzySet object
33     FuzzySet *slow = new FuzzySet(0, 10, 10, 20);
34     // Including the FuzzySet into FuzzyOutput
35     speed->addFuzzySet(slow);
36
37     // Instantiating a FuzzySet object
38     FuzzySet *average = new FuzzySet(10, -20, -30, -40);
39     // Including the FuzzySet into FuzzyOutput
40     speed->addFuzzySet(average);
41
42     // Instantiating a FuzzySet object
43     FuzzySet *fast = new FuzzySet(30, -40, -40, -50);
44     // Including the FuzzySet into FuzzyOutput
45     speed->addFuzzySet(fast);
46
47     fuzzy->addFuzzyOutput(speed);
48 }
```

Ln 30, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]

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

Kode Program

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

```
FuzzyInput *temp = new FuzzyInput(1);
FuzzySet *cold = new FuzzySet(0, 0, 15, 20);
temp->addFuzzySet(cold);
FuzzySet *warm = new FuzzySet(15, 20, 30, 35);
temp->addFuzzySet(warm);
FuzzySet *hot = new FuzzySet(30, 35, 100, 100);
temp->addFuzzySet(hot);
fuzzy->addFuzzyInput(temp);

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

// Building FuzzyRule "IF distance = small THEN speed = slow"
// Instantiating a FuzzyRuleAntecedent objects
FuzzyRuleAntecedent *ifDistanceSmall = new FuzzyRuleAntecedent();
Ln 30, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]
```

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

```
status->addFuzzySet(status_warm);
FuzzySet *status_hot = new FuzzySet(2, 3, 3);
status->addFuzzySet(status_hot);
fuzzy->addFuzzyOutput(status);

// Building FuzzyRule "If distance=-small-THEN-speed=-slow"
// Instantiating-a-FuzzyRuleAntecedent-objects
.FuzzyRuleAntecedent *ifDistanceSmall = new FuzzyRuleAntecedent();
// Creating-a-FuzzyRuleAntecedent-with-just-a-single-FuzzySet
.ifDistanceSmall->joinSingle(small);
// Instantiating-a-FuzzyRuleConsequent-objects
.FuzzyRuleConsequent *thenSpeedSlow = new FuzzyRuleConsequent();
// Including-a-FuzzySet-to-this-FuzzyRuleConsequent
.thenSpeedSlow->addOutput(slow);
// Instantiating-a-FuzzyRule-objects
.FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifDistanceSmall, thenSpeedSlow);
// Including-the-FuzzyRule-into-Fuzzy
.fuzzy->addFuzzyRule(fuzzyRule01);

// Building FuzzyRule "IF distance = safe THEN speed = average"
// Instantiating a FuzzyRuleAntecedent objects
FuzzyRuleAntecedent *ifDistanceSafe = new FuzzyRuleAntecedent();
```

```

FL.ino
50 FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifDistanceSmall, thenSpeedSlow);
51 // Including the FuzzyRule into Fuzzy
52 fuzzy->addFuzzyRule(fuzzyRule01);
53
54 // Building FuzzyRule "IF distance == safe THEN speed == average"
55 // Instantiating a FuzzyRuleAntecedent objects
56 FuzzyRuleAntecedent *ifDistanceSafe = new FuzzyRuleAntecedent();
57 ifDistanceSafe->joinSingle(safe);
58
59 // Instantiating a FuzzyRuleConsequent objects
60 FuzzyRuleConsequent *thenSpeedAverage = new FuzzyRuleConsequent();
61
62 // Including a FuzzySet to this FuzzyRuleConsequent
63 thenSpeedAverage->addOutput(average);
64
65 // Instantiating a FuzzyRule objects
66 FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifDistanceSafe, thenSpeedAverage);
67
68 // Including the FuzzyRule into Fuzzy
69 FuzzyRuleAntecedent *ifDistanceBig = new FuzzyRuleAntecedent();
70
71 fuzzy->addFuzzyRule(fuzzyRule02);
72
73 // Building FuzzyRule "IF distance = big THEN speed = high"
74 // Instantiating a FuzzyRuleAntecedent objects
75 FuzzyRuleAntecedent *ifDistanceBig = new FuzzyRuleAntecedent();
76 ifDistanceBig->joinSingle(big);
77
78 // Instantiating a FuzzyRuleConsequent objects
79 FuzzyRuleConsequent *thenSpeedFast = new FuzzyRuleConsequent();
80
81 // Including a FuzzySet to this FuzzyRuleConsequent
82 thenSpeedFast->addOutput(fast);
83
84 // Instantiating a FuzzyRule objects
85 FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifDistanceBig, thenSpeedFast);
86
87 // Including the FuzzyRule into Fuzzy
88
89 fuzzy->addFuzzyRule(fuzzyRule03);
90
91 void loop()
92 {
93
94 }

```

Ln 54, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]

```

FL.ino
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85

```

Ln 68, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]

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

(a) **Fuzzy Rule 1**

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

```

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

```

Ln 40, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]

(b) Fuzzy Rule 2

Kode Program

```

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

```

```

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

```

Ln 48, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410 [not connected]

(c) Fuzzy Rule 3

Kode Program

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

```
FL.ino
50     ifTempWarm->joinSingle(warm);
51     FuzzyRuleConsequent *thenStatusWarm = new FuzzyRuleConsequent();
52     thenStatusWarm->addOutput(status_warm);
53     FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifTempWarm, thenStatusWarm);
54     fuzzy->addFuzzyRule(fuzzyRule02);
55
56 // Define Fuzzy Rule #3. IF Temp = HOT, THEN Status = HOT
57 FuzzyRuleAntecedent *ifTempHot = new FuzzyRuleAntecedent();
58 ifTempHot->joinSingle(hot);
59 FuzzyRuleConsequent *thenStatusHot = new FuzzyRuleConsequent();
60 thenStatusHot->addOutput(status_hot);
61 FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifTempHot, thenStatusHot);
62 fuzzy->addFuzzyRule(fuzzyRule03);
63
64 void loop()
65 {
66     // Getting a random value
67     int input = random(0, 80);
68     // Printing something
69     Serial.println("\n\n\nEntrance: ");
70     Serial.print("\t\t\t\tDistance: ");
```

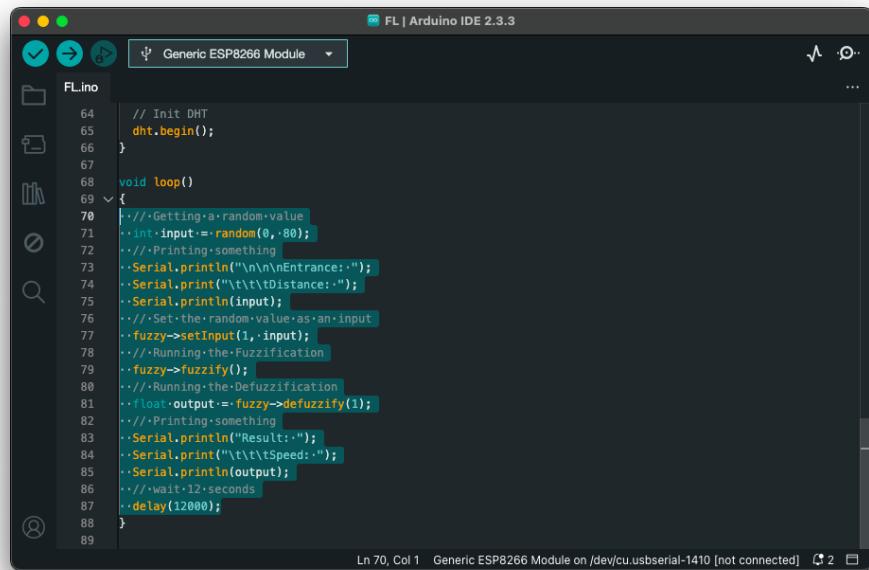
12. Bagian terakhir dari fungsi `setup()` adalah kode inisialisasi dht

Kode Program

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

```
FL.ino
49     FuzzyRuleAntecedent *ifTempWarm = new FuzzyRuleAntecedent();
50     ifTempWarm->joinSingle(warm);
51     FuzzyRuleConsequent *thenStatusWarm = new FuzzyRuleConsequent();
52     thenStatusWarm->addOutput(status_warm);
53     FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifTempWarm, thenStatusWarm);
54     fuzzy->addFuzzyRule(fuzzyRule02);
55
56 // Define Fuzzy Rule #3. IF Temp = HOT, THEN Status = HOT
57 FuzzyRuleAntecedent *ifTempHot = new FuzzyRuleAntecedent();
58 ifTempHot->joinSingle(hot);
59 FuzzyRuleConsequent *thenStatusHot = new FuzzyRuleConsequent();
60 thenStatusHot->addOutput(status_hot);
61 FuzzyRule *fuzzyRule03 = new FuzzyRule(3, ifTempHot, thenStatusHot);
62 fuzzy->addFuzzyRule(fuzzyRule03);
63
64 // Init DHT
65 dht.begin();
66
67 void loop()
68 {
69     // Getting a random value
70     int input = random(0, 80);
71     // Printing something
72     Serial.println("\n\n\nEntrance: ");
73     Serial.print("\t\t\t\tDistance: ");
```

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



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

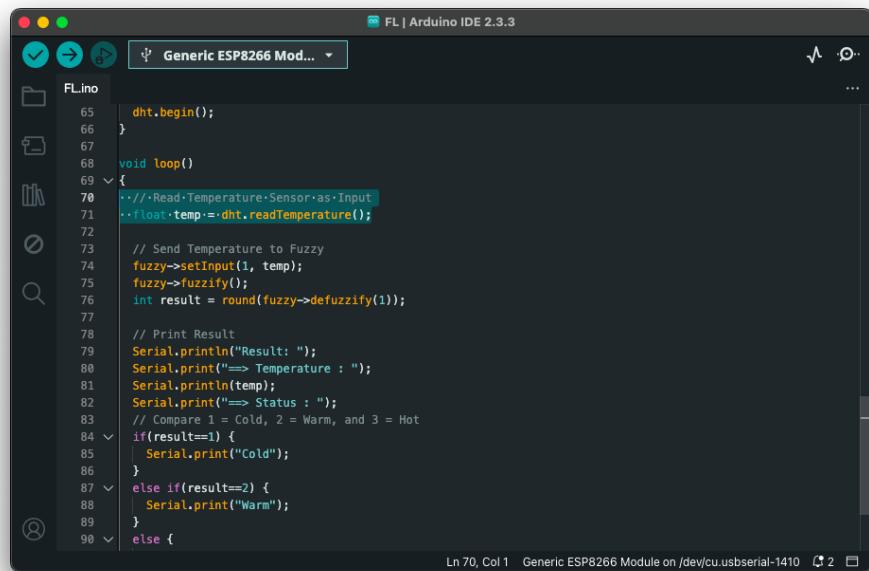
void loop()
{
    // Getting a random value
    int input = random(0, 80);
    // Printing something
    Serial.println("\n\nEntrance:");
    Serial.print("Distance:");
    Serial.println(input);
    // Set the random value as an input
    fuzzy->setInput(1, input);
    // Running the Fuzzification
    fuzzy->fuzzify();
    // Running the Defuzzification
    float output = fuzzy->defuzzify(1);
    // Printing something
    Serial.println("Result:");
    Serial.print("Speed:");
    Serial.println(output);
    // Wait 12 seconds
    delay(12000);
}
```

14. Lalu masukkan satu per satu kode berikut:

(a) Baris kode pembaca sensor suhu

Kode Program

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



```
dht.begin();

void loop()
{
    // Read Temperature Sensor as Input
    float temp = dht.readTemperature();

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

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

```

(b) Berikutnya kode Fuzzifikasi dan Defuzzifikasi

Kode Program

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

The screenshot shows the Arduino IDE interface with a dark theme. The central window displays the code for a Generic ESP8266 Module. The code includes a loop that reads temperature from a DHT sensor, sends it to a fuzzy logic system, and prints the result. The Arduino IDE status bar at the bottom indicates "Ln 73, Col 1 Generic ESP8266 Module on /dev/cu.usbserial-1410".

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

- (c) Terakhir, kode cetak hasil dan delay

Kode Program

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

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

The screenshot shows the Arduino IDE interface with the title bar "FL | Arduino IDE 2.3.3". The central area displays a code editor for a file named "FL.ino". The code implements a fuzzy logic system to map temperature inputs to status outputs. It includes functions for setting input values, fuzzifying them, and then defuzzifying to get a result. The result is then printed to the serial monitor. The code uses the Serial library for communication.

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

//>>>Result:
//>>>Serial.println("Result:");
//>>>Serial.print("==>Temperature : ");
//>>>Serial.println(temp);
//>>>Serial.print("==>Status : ");
//>>>Compare: 1.0 = Cold, 2.0 = Warm, and 3.0 = Hot
if(result==1){
    Serial.print("Cold");
}
else-if(result==2){
    Serial.print("Warm");
}
else{
    Serial.print("Hot");
}
Serial.println("\n");
//>>>Delay .5s
delay(5000);
```

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

The screenshot shows the Arduino IDE interface with the title bar "FL | Arduino IDE 2.3.3". The central area displays a code editor for a file named "FL.ino". The right side of the interface features a "Serial Monitor" window. The monitor shows a series of messages being sent from the ESP8266 module to the computer. These messages report the current temperature and categorize it as "Warm". The baud rate is set to 9600.

Message (Enter to send message to 'Generic ESP8266 Module' on '/dev/cu.usbserial-1410')
Result:
==> Temperature : 21.70
==> Status : Warm
Result:
==> Temperature : 21.60
==> Status : Warm
Result:
==> Temperature : 21.60
==> Status : Warm
Result:
==> Temperature : 21.50
==> Status : Warm
Result:
==> Temperature : 21.50
==> Status : Warm

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

Bab 8

Praktikum 8

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

8.1 Tugas Akhir Praktikum

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

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