



UNIVERSITAS SEMARANG
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
TEKNIK INFORMATIKA

Internet of Thing

Modul Praktikum Mahasiswa

Oleh:

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

Daftar Isi

Pendahuluan	2
0.1 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 Thonny, NodeMCU, dan MicroPython	7
1.2 Tutorial	7
2 Praktikum 2	14
2.1 NodeMCU, MicroPython, dan Wi-Fi	14
2.2 Tutorial	14
3 Praktikum 3	19
3.1 NodeMCU, MicroPython, dan Web Server	19
3.2 Tutorial	19
4 Praktikum 4	25
4.1 NodeMCU, MicroPython, dan MQTT	25
4.2 Tutorial	25
5 Praktikum 5	30
5.1 NodeMCU, MicroPython, dan Blynk	30
5.2 Tutorial	30
6 Praktikum 6	41
6.1 NodeMCU, MicroPython, dan Telegram Bot	41
6.2 Tutorial	41
7 Praktikum 7	48
7.1 NodeMCU, MicroPython, dan IFTTT	48
7.2 Tutorial	48
8 Praktikum 8	60
8.1 NodeMCU, MicroPython, dan Tugas Praktikum	60
8.2 Tugas Praktikum	60

Daftar Gambar

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

Pendahuluan

0.1 Mengenal *Internet of Things*

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

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



Gambar 1: Internet of Things

0.2 Perangkat Board IoT

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

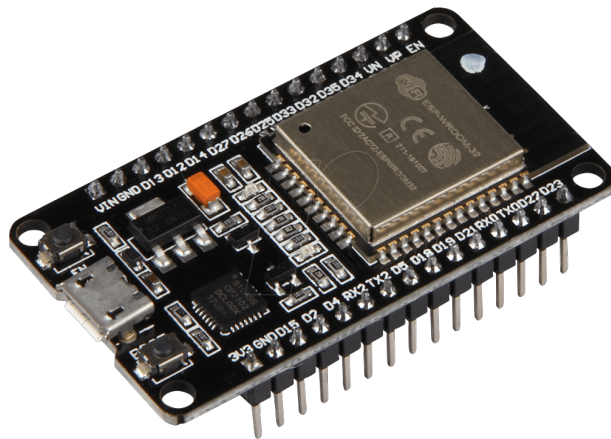
Berikut ini adalah daftar Board yang dapat dibeli dengan harga terjangkau:

1. Arduino



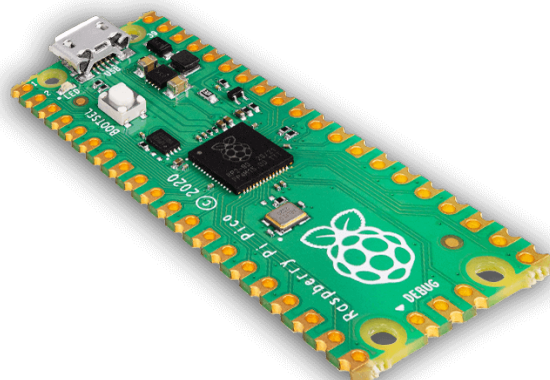
Gambar 2: Board Arduino

2. NodeMCU



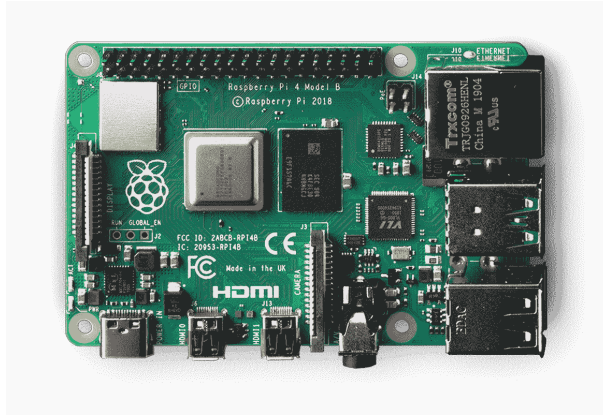
Gambar 3: Board NodeMCU

3. Raspberry Pi Pico



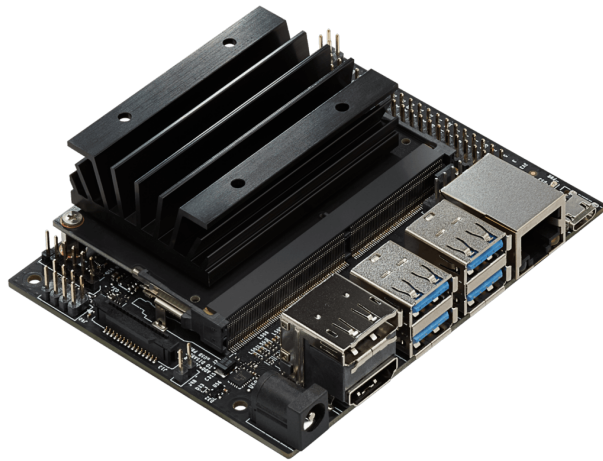
Gambar 4: Board Pico

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



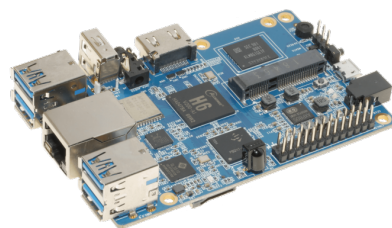
Gambar 5: Board Pi 4B

5. NVidia Jetson



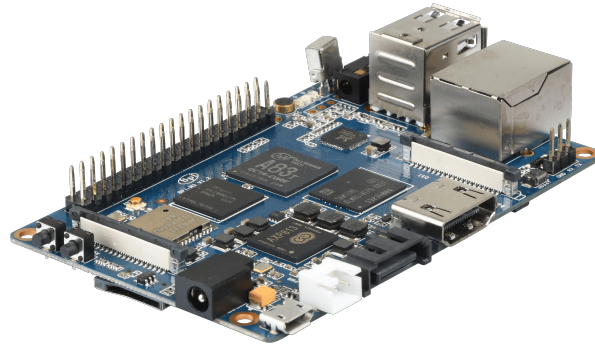
Gambar 6: Board NVidia Jetson

6. Orange Pi



Gambar 7: Board Orange Pi

7. Banana Pi



Gambar 8: Board Banana Pi

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

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

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

Persiapan Praktikum

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

0.3 Perangkat Keras

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

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

0.4 Perangkat Lunak

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

- Python IDE
 - Thonny Python Editor <https://thonny.org/>
- 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>
 - * Pilih **CP210x Windows Drivers**
 - Zadig (Universal) <https://zadig.akeo.ie/>
- Firmware MicroPython <https://micropython.org/download/esp8266/>
 - Bagian **Firmware**, Unduh **Latest**

Bab 1

Praktikum 1

1.1 Thonny, NodeMCU, dan MicroPython

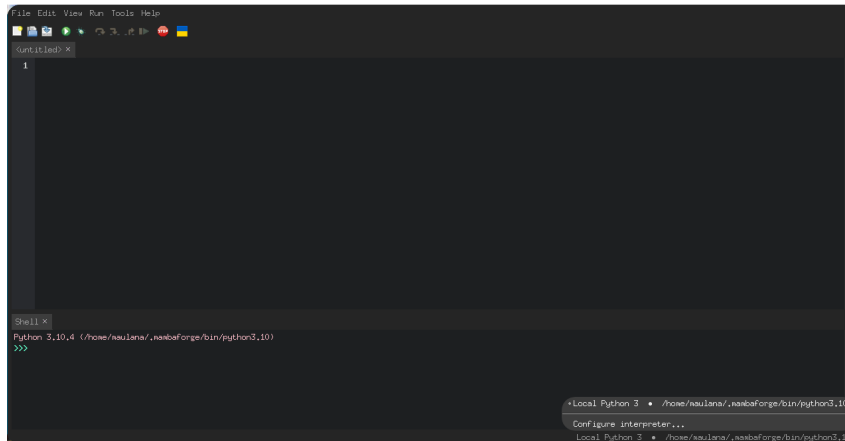
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke komputer beserta konfigurasinya hingga dapat dikenali oleh Thonny Python Editor. Mahasiswa diharapkan untuk membaca, dan memahami **Persiapan Praktikum** yang ada di halaman sebelumnya.

1.2 Tutorial

1. Pastikan **Thonny** sudah terinstall dan driver **Zadig** terunduh jika diperlukan.
2. Colokkan perangkat **NodeMCU** yang ada ke port USB komputer. Tunggu beberapa saat menunggu komputer menginstall driver yang diperlukan.
3. Untuk mengecek apakah perangkat sudah bisa digunakan dapat menggunakan dua cara:
 - Buka **Device Manager** dan pastikan tidak ada tanda **Warning** yang ada di perangkat **CDC Serial**. Jika ada, maka **perangkat tidak terdeteksi**



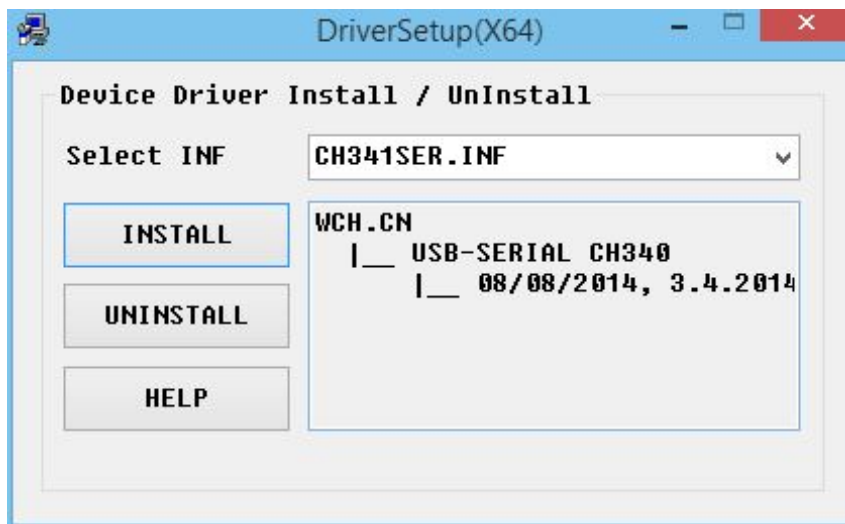
- Buka **Thonny** dan klik bagian **Kanan Bawah**. Ganti **Interpreter** dari **Local Python** ke **MicroPython**. Jika hanya ada **Local Python**, maka **perangkat tidak terdeteksi**.



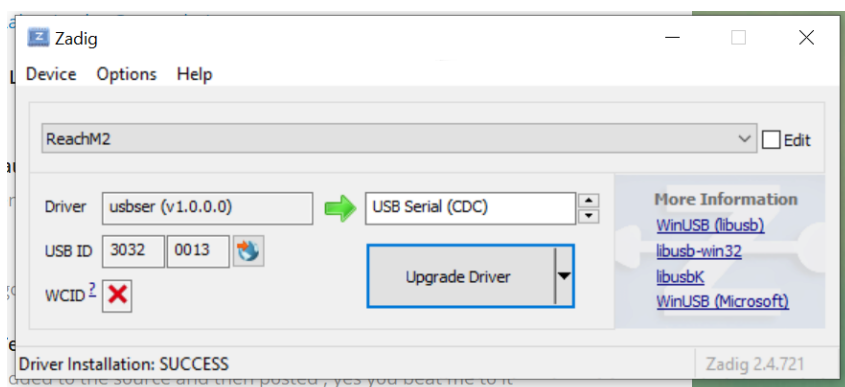
4. Jika:

- **Tidak Terdeteksi**

- **Model ESP8266** : Gunakan **Driver CH341** untuk NodeMCU model ESP8266 standar.

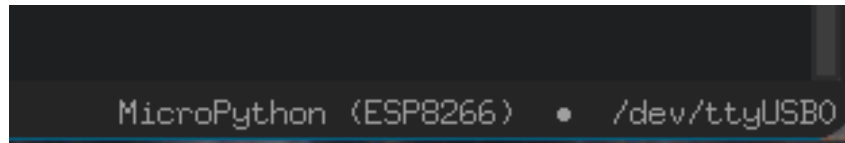


- **Model ESP8266Mod (Amica)** : Gunakan **Driver CP210X**
- **Universal Driver** : Gunakan **Zadig** untuk menginstall **USB Serial (CDC)**

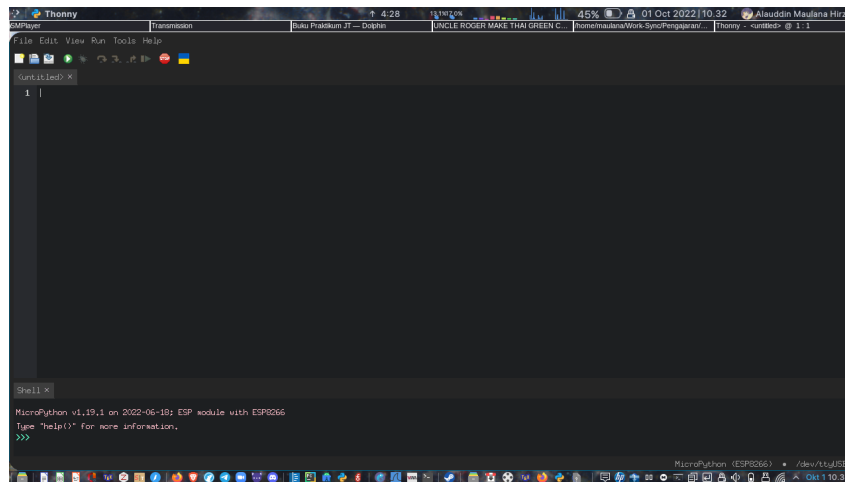


- **Terdeteksi**, maka lanjutkan tahap selanjutnya

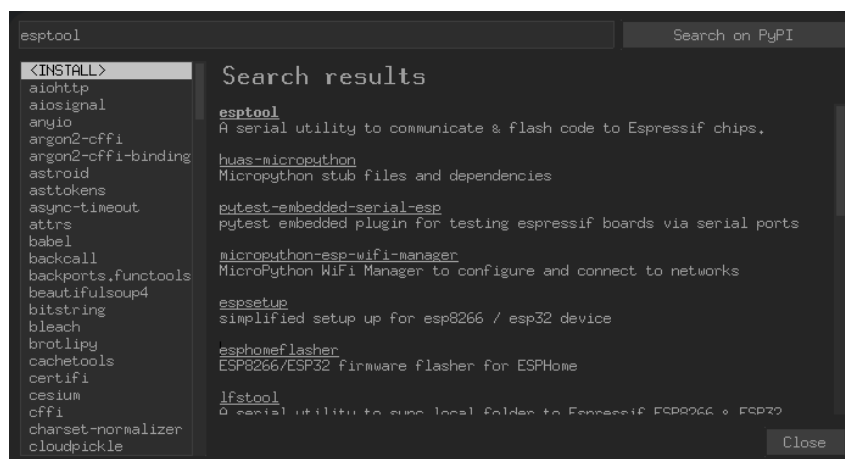
5. Cek NodeMCU dengan membuka **Thonny** untuk mendeteksi perangkat tersebut. Namun **Thonny** tidak bisa menggunakan perangkat tersebut.



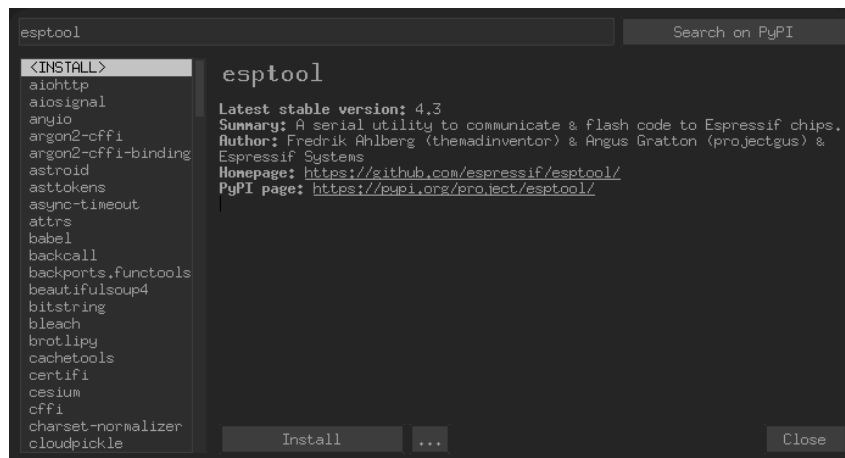
6. Untuk menginstall **MicroPython** ke perangkat **NodeMCU**. langkah berikutnya yang dilakukan adalah melakukan **flashing** ke **NodeMCU**. Buka **Thonny**.



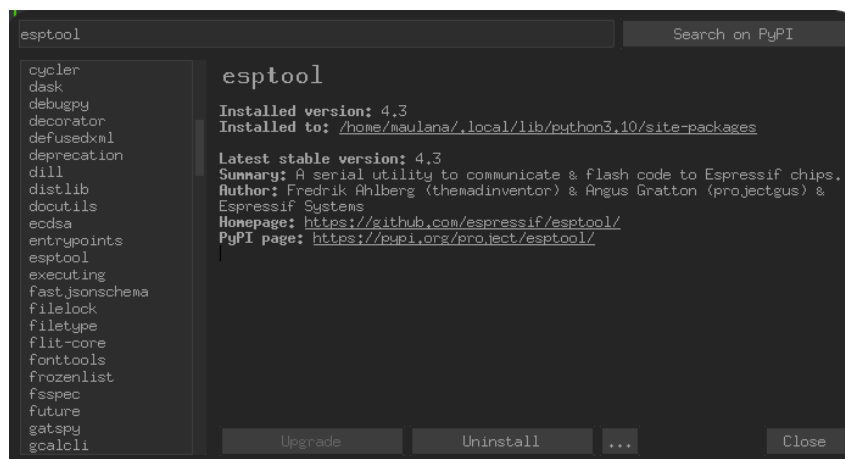
7. Sebelum melakukan **flashing**, **Thonny** harus memiliki paket tambahan dengan nama **esptool**. Paket ini dapat diinstall melalui **PIP** Python atau melalui **Thonny** sendiri.
8. Sebelum menginstall pastikan **Interpreter** di set ke **Local Python**. Cek **Kanan Bawah**. Lalu Klik menu **Tools > Manage packages**
9. Di window **Search**, masukkan **esptool** dan klik **Search on PyPI**



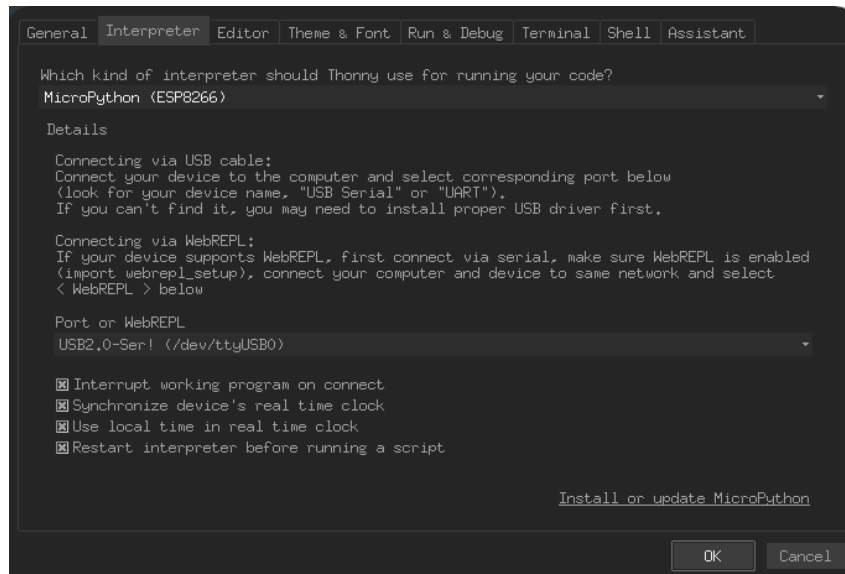
10. Klik **esptool** yang ada di atas, dan Klik **Install**. Jika **Button Install** tidak ada, maka **esptool** sudah terpasang.



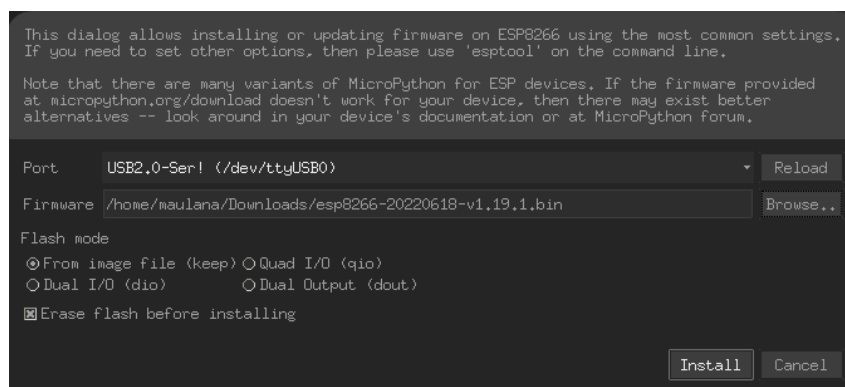
11. Tunggu hingga selesai dan **esptool** akan terinstall. Klik **Close**



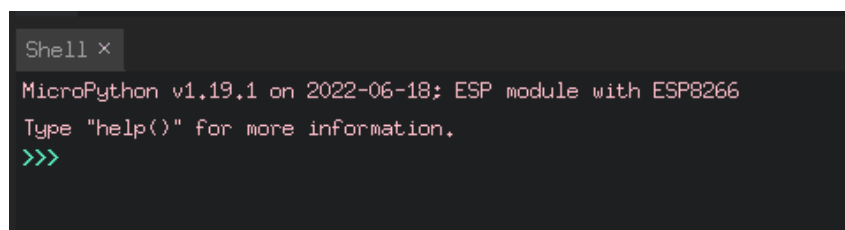
12. Langkah berikutnya adalah melakukan **flashing NodeMCU ESP8266**. Klik **Tools > Options > Interpreter > Pilih MicroPython (ESP8266) > Klik Install or update MicroPython**



13. **Thonny** akan menampilkan window baru untuk memilih **Target** dan **File**. Klik **Install** setelah memilih **Target** dan **File**. Tunggu hingga selesai.



14. Jika sudah selesai, Klik **OK**.
15. Contoh **Interpreter** yang berhasil di install adalah sebagai berikut:



16. Untuk mengetes apakah **NodeMCU** berhasil. Masukkan kode berikut ke **Editor** dan **Run**.

Potongan Kode

```
from machine import Pin
import time
p0 = Pin(2, Pin.OUT)
p0.value(0)
time.sleep(1)
p0.value(1)
```

17. Jika di **Run**, **NodeMCU ESP8266** akan menyalakan LED nya. Jika tidak, cek kembali model perangkat beserta lokasi Pin LED nya.
18. Langkah berikutnya adalah menginstall **Library-library** yang diperlukan oleh perangkat untuk menjalani praktikum ini. **Buatlah File Baru** dengan nama **LibsInstaller.py**
19. Lalu masukkan kode berikut untuk membantu instalasi **Library** untuk praktikum

```

import network
import upip

SSID = "<SSID>"
Password = "<Password>"
reset = True
print(">> Memulai Installer Libs ESP8266 ... ")
print(">> Menghubungkan ... ")
wlan0 = network.WLAN(network.STA_IF)
if reset:
    wlan0.active(True)
    wlan0.connect(SSID, Password)
    while not wlan0.isconnected():
        wlan0.active(True)
        pass

status = wlan0.isconnected()
if(status == True):
    print("==>> Terhubung")
    print(">> Mengecek Libs urequests ...")
    try:
        import urequests
        print("==>> OK")
    except:
        print("==>> Gagal. Menginstall urequests ... ")
        upip.install('micropython-urequests')
    print(">> Mengecek Libs umqtt.robust ...")
    try:
        import umqtt.robust
        print("==>> OK")
    except:
        print("==>> Gagal. Menginstall umqtt.robust ... ")
        upip.install("micropython-umqtt.robust")
    print(">> Mengecek Libs umqtt.simple ...")
    try:
        import umqtt.simple
        print("==>> OK")
    except:
        print("==>> Gagal. Menginstall umqtt.simple ... ")
        upip.install("micropython-umqtt.simple")
    print("")
    print(">> Ulangi Installer untuk mengecek Libs")
else:
    print("==>> Koneksi Gagal")

```

Bab 2

Praktikum 2

2.1 NodeMCU, MicroPython, dan Wi-Fi

Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke jaringan nirkabel yang sudah disiapkan. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 1** yang ada di halaman sebelumnya.

2.2 Tutorial

1. Untuk memulai praktikum ini, mahasiswa diwajibkan membawa perangkat NodeMCU yang sudah dipasang **MicroPython**, Kabel Data, Thonny, USB Driver (Jika diperlukan), dan hotspot pribadi.
2. Tancapkan **NodeMCU** ke komputer, dan pastikan sudah bisa terdeteksi oleh **Thonny** sebagai **MicroPython (ESP8266)**.
3. Ganti **Interpreter** Thonny ke **MicroPython (ESP8266)** dan bukan **Local Python**



4. Untuk menyambungkan perangkat ke jaringan nirkabel, hal pertama yang perlu dilakukan adalah melakukan setup hotspot.
5. Mahasiswa dibebaskan mengatur **Akses Poin** dan **Password** Hotspot mereka sendiri
6. Jika sudah mengatur hotspot, berikutnya adalah memasukkan kode pemrograman ke perangkat. Masukkan **potongan kode berikut**

Potongan Kode

```
from machine import Pin
import time
import network
```

7. Kode tersebut digunakan untuk mengimpor fitur-fitur dasar yang ada di **NodeMCU**.
8. Sebelum masuk ke kode utama, masukkan **Variabel Global** yang akan digunakan untuk proses koneksi WiFi nanti:

Potongan Kode

```
SSID = "<SSID>"           <== Masukkan SSID
Password = "<Password>"       <== Masukkan Password
p0 = Pin(2, Pin.OUT)
blinker = 0
```

9. Setelah itu buatlah dua (2) jenis notifikasi LED yang bisa membantu ketika perangkat tidak terhubung ke komputer. Masukkan kode berikut setelah kode sebelumnya.

Potongan Kode

```
# Blink Normal 1s
def blinkNormal():
    p0.value(0)
    time.sleep(1)
    p0.value(1)
    time.sleep(1)

# Blink Pendek 50ms
def blinkPendek():
    p0.value(0)
    time.sleep_ms(100)
    p0.value(1)
    time.sleep_ms(100)
```

10. Proses berikutnya adalah membuat fungsi khusus untuk menghubungkan ke **Jaringan Nirkabel**. Fungsi ini akan mengembalikan status berhasil atau tidak

Potongan Kode

```
def connectNetwork():
    # Set Model Station / Klien
    sta_if = network.WLAN(network.STA_IF)

    # Cek Jika Belum Terkoneksi
    if not sta_if.isconnected():
        print("Menghubungkan ke ",SSID," ...")
        sta_if.active(True)

    # Masukkan SSID dan Password
    sta_if.connect(SSID, Password)

    while not sta_if.isconnected():
        sta_if.active(True)
        pass

    status = sta_if.isconnected()
    netconf = sta_if.ifconfig()

    return status,netconf
```

11. Kode fungsionalitas yang diperlukan untuk terhubung ke jaringan nirkabel sudah dibuat. Berikutnya adalah membuat fungsi utama alias **main** untuk eksekusi otomatisnya. Perhatikan **Potongan Kode** berikut

Potongan Kode

```
def main():
    global blinker,host
    print("Memulai ESP8266...")
    blinkNormal()
    print("Memulai Konfigurasi Wi-Fi...")
    blinkNormal()

    # Hubungkan ke Wi-Fi
    status,netconf = connectNetwork()

    print("")

    # Cek Network
    if(status == True):
        print("ESP8266 Terhubung")
        print("IP Address : ",netconf[0])
        print("Subnet Mask : ",netconf[1])
        print("Default Gateway : ",netconf[2])
        while(blinker<3):
            blinkPendek()
            blinker += 1
    else:
        print("ESP8266 Tidak Terhubung")
        while(blinker<1):
            blinkPendek()
            blinker += 1

main()
```

12. Jika perangkat dinyalakan, perangkat akan melakukan **2 Kali Blink Panjang** yang artinya **Normal**, dan **3 Kali Blink Pendek** jika tersambung ke **Jaringan Nirkabel**. Jika gagal, perangkat akan **1 Kali Blink Pendek**.
13. Selain itu, perangkat akan menampilkan pesan-pesan di Log nya
 - IP Address
 - Subnet Mask
 - Default Gateway
14. Jika perangkat sudah terhubung dengan sempurna, maka proses dilanjutkan dengan memberikan kemampuan PING. Unduh file **uping.py** yang ada di elearning, dan masukkan ke **NodeMCU** melalui **Thonny**.
15. Buka file **uping.py** dengan **Thonny**, lalu **Save As** ke **MicroPython** dengan nama **uping.py**
16. Untuk bisa menggunakan file **uping.py** tersebut, tambah kan **import** ke awal kode

Potongan Kode

```
from machine import Pin
import time
import network
import uping <== Kode ini
```

17. Lalu tambahkan **Variabel Global** baru seperti berikut:

Potongan Kode

```
SSID = "<SSID>"
Password = "<Password>"
p0 = Pin(2, Pin.OUT)
blinker = 0
host = "1.1.1.1" <== Kode ini, CloudFlare IP
```

18. Langkah terakhir adalah memodifikasi kode **main** dengan fungsi **PING**.

Potongan Kode

```
# Cek Network
if(status == True):
    print("ESP8266 Terhubung")
    print("IP Address : ",netconf[0])
    print("Subnet Mask : ",netconf[1])
    print("Default Gateway : ",netconf[2])
    while(blinker<3):
        blinkPendek()
        blinker += 1

# Kode Tambahan
# Ping Google / Lainnya
print("")
print(f"Pinging host")
result = uping.ping(host)

print("")
print("ESP8266 Kirim ",result[0]," dan Terima ",result[1])
```

19. Jika dilakukan dengan benar maka perangkat akan menghasilkan **output** sebagai berikut:

```
ESP8266 Terhubung
IP Address : 192.168.21.117
Subnet Mask : 255.255.255.0
Default Gateway : 192.168.21.46

Pinging 1.1.1.1
PING 1.1.1.1 (1.1.1.1): 64 data bytes
84 bytes from 1.1.1.1: icmp_seq=1, ttl=55, time=126.584983 ms
84 bytes from 1.1.1.1: icmp_seq=2, ttl=55, time=233.882999 ms
84 bytes from 1.1.1.1: icmp_seq=3, ttl=55, time=281.459951 ms
4 packets transmitted, 3 packets received

ESP8266 Sent 4 and Received 3

>>>
```

20. Praktikum 2 Selesai

Bab 3

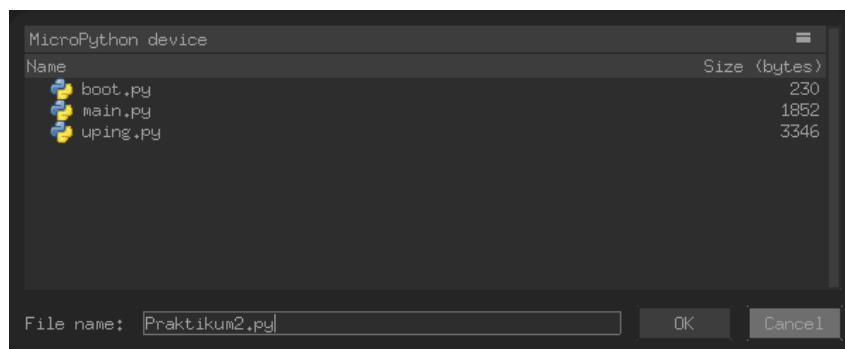
Praktikum 3

3.1 NodeMCU, MicroPython, dan Web Server

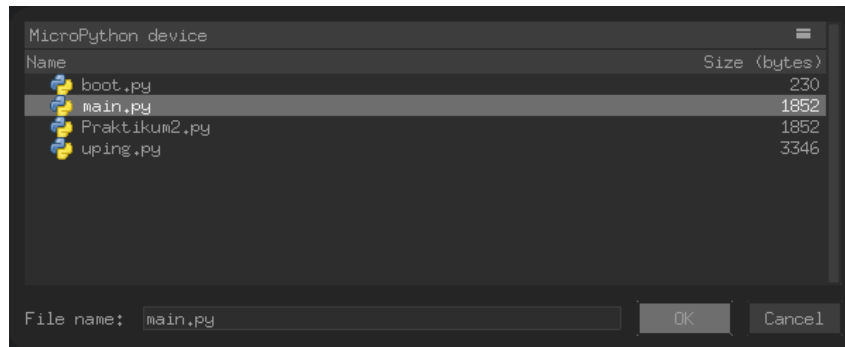
Di bagian ini mahasiswa diajarkan bagaimana membangun Web Server sederhana untuk menampilkan data sensor maupun lainnya. Mahasiswa diharapkan untuk membaca, dan memahami **Praktikum 2** yang ada di halaman sebelumnya.

3.2 Tutorial

1. Untuk melaksanakan praktikum ini, perangkat wajib memiliki file koneksi ke WiFi sebagai syarat dasar. Tanpa akses jaringan nirkabel, perangkat tidak akan bisa diakses.
2. Buka file koneksi WiFi sebelumnya (Praktikum 2) lalu simpan dengan nama baru **Praktikum2.py** di NodeMCU.



3. Buat file baru dengan Klik **New** di **Thonny**, lalu **Simpan** sebagai **Praktikum3.py**



4. Masukkan potongan kode berikut untuk menambahkan fungsi dasar:

(a) Impor Module

Potongan Kode

```
from machine import Pin
import time
import network
import ntptime
try:
    import usocket as socket
except:
    import socket
```

(b) Variabel Global

Potongan Kode

```
SSID = "SSID"
Password = "Password"
p0 = Pin(2, Pin.OUT)
blinker = 0
reset = False
current_time = None
```

(c) Blinker Pender

Potongan Kode

```
# Blink Pendek 50ms
def blinkPendek():
    p0.value(0)
    time.sleep_ms(100)
    p0.value(1)
    time.sleep_ms(100)
```

(d) Blinker Normal

Potongan Kode

```
# Blink Normal 1s
def blinkNormal():
    p0.value(0)
    time.sleep(1)
    p0.value(1)
    time.sleep(1)
```

(e) Koneksi Wi-Fi

Potongan Kode

```
# Koneksi Wi-Fi
def connectWLAN():
    wlan0 = network.WLAN(network.STA_IF)
    if reset:
        wlan0.active(True)
        wlan0.connect(SSID, Password)

        while not wlan0.isconnected():
            wlan0.active(True)
            pass

    status = wlan0.isconnected()
    ip_addr = wlan0.ifconfig()
    return status, ip_addr
```

(f) Waktu dan Tanggal

Potongan Kode

```
# Set Tanggal
def setDateTime():
    ntptime.settime()

# Get Tanggal
def getDateTime():
    datetime = time.localtime()
    waktu = str(datetime[3]+7)+"."+str(datetime[4])+"."+str(datetime[5])
    tanggal = str(datetime[2])+"/"+str(datetime[1])+"/"+str(datetime[0])
    return waktu, tanggal
```

5. Jika fungsi dasar sudah dimasukkan ke dalam script, berikutnya adalah mengunduh file halaman **Web** dengan nama **webpage.py**. Lalu masukkan ke dalam **ESP8266**.
6. Jika sudah memasukkan file **webpage.py** ke dalam **ESP8266**, lalu **import** dengan menambahkan kode berikut ke bagian atas kode (Barisan **Import**)

Potongan Kode

```
import webpage
```

7. Langkah berikutnya adalah membuat fungsi **main**. Perhatikan kode berikut:

Potongan Kode

```
def main():
    global blinker, reset

    # Boot Sequence
    print(">> Memulai ESP8266 ... ")
    blinkNormal()

    print(">> Menghubungkan ke Wi-Fi ... ")
    blinkNormal()
    status, ip_addr = connectWLAN()

    if(status == True):
        print("==>> Sukses")
        while(blinker < 3):
            blinkPendek()
            blinker += 1

    print("")
    print(">> Mengkonfigurasi Waktu dan Tanggal ... ")
    blinkNormal()
    setDateTime()
    waktu, tanggal = getDateTime()
    print(">> Jam : ", waktu, ", dan Tanggal : ", tanggal)

    print("")
    print(">> Memulai Web Server ... ")
    print(">> Akses Web via ", ip_addr[0], "... ")
    blinkNormal()

    # Kode Web Server
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.bind(('', 80))
    s.listen(5)
```

8. Potongan Lanjutan (Cek Indentasi)

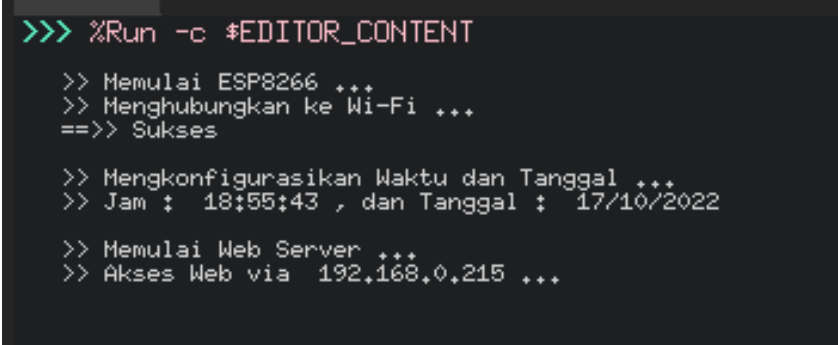
Potongan Kode

```
while True:
    blinkPendek()
    waktu,tanggal = getDateTIme()
    try:
        if gc.mem_free() < 102000:
            gc.collect()
            conn, addr = s.accept()
            conn.settimeout(3.0)
            print("")
            print('>> Mendapatkan Koneksi Dari %s' % str(addr))
            request = conn.recv(1024)
            conn.settimeout(None)
            request = str(request)
            print('==>> Content = %s' % request)
            response = webpage.webPage(waktu,tanggal,ip_addr)
            conn.send('HTTP/1.1 200 OK\n')
            conn.send('Content-Type: text/html\n')
            conn.send('Connection: close\n\n')
            conn.sendall(response)
            conn.close()
        except OSError as e:
            conn.close()
            print('Koneksi Tertutup')

    else:
        print("==>> Gagal")
        print("==>> Sukses")
        while(blinker<1):
            blinkPendek()
            blinker += 1

main()
```

9. Jalankan kode tersebut, dan perhatikan hasil keluaran dari terminal. Juga bisa menggunakan lampu LED.



```
>>> %Run -c $EDITOR_CONTENT
>> Memulai ESP8266 ***
>> Menghubungkan ke Wi-Fi ***
==>> Sukses

>> Mengkonfigurasi Waktu dan Tanggal ***
>> Jam : 18:55:43 , dan Tanggal : 17/10/2022

>> Memulai Web Server ***
>> Akses Web via 192,168,0,215 ***
```

10. Akses dapat dibuka melalui Komputer maupun HP dengan jaringan yang sama

ESP8266 Web Server

KATEGORI	NILAI
Waktu	19:0:57
Tanggal	17/10/2022

KATEGORI	NILAI
IP Address	192.168.0.215
Subnet Mask	255.255.255.0
Gateway	192.168.0.1
DNS Server	192.168.0.1

11. Data di web akan merefresh setiap 5 detik dan ditandai dengan blink pendek di perangkat satu kali
12. Praktikum 3 Selesai

Bab 4

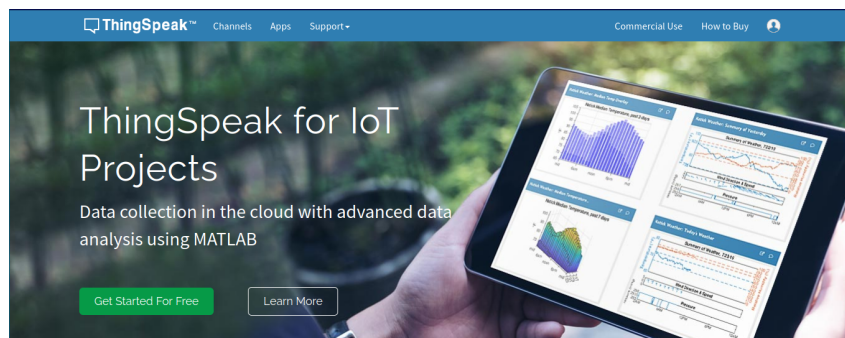
Praktikum 4

4.1 NodeMCU, MicroPython, dan MQTT

Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke Cloud MQTT **ThingSpeak** sehingga dapat mengirimkan data ke Cloud secara otomatis. Mahasiswa diharapkan untuk membaca, dan memahami **Persiapan Praktikum** yang ada di halaman sebelumnya.

4.2 Tutorial

1. Praktikum ini memerlukan **Thonny** sebagai Editor nya, dan pastikan **Interpreter (Kanan Bawah)** sudah mengarah ke **MicroPython ESP8266**
2. Langkah berikutnya adalah mendaftar ke <https://thingspeak.com/>. Klik **Get Started for Free**



3. Lalu klik **Create One** untuk mendaftar akun gratis



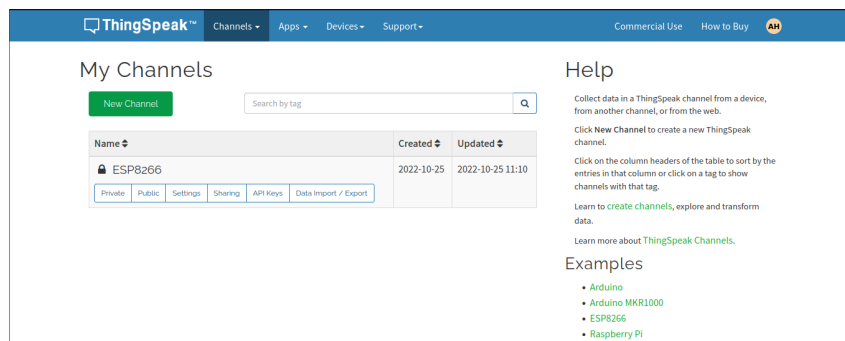
Email

No account? [Create one!](#)

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

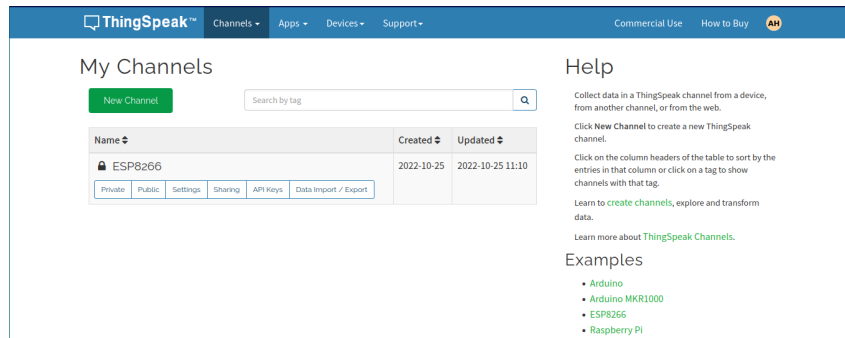
Next

4. Lanjutkan proses-proses berikutnya, dan pastikan untuk melakukan **Verifikasi E-Mail**. Jika sudah login ke web, dan tampilan akan seperti berikut.

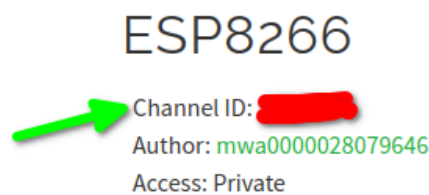


5. Klik **New Channel** untuk membuat Channel MQTT baru. Beri **Nama** (Contoh: ESP8266), dan **Field 1** dengan **Data**. Lalu klik **Save Channel** di bawah halaman.

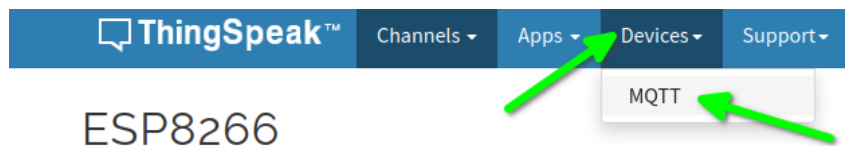
6. Jika berhasil dibuat maka ThingSpeak akan menampilkan channel baru seperti berikut:



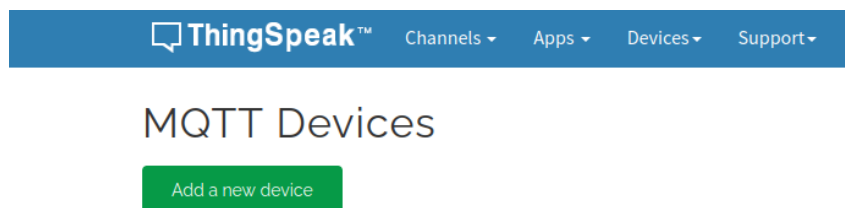
7. Klik **Channel** tersebut untuk melihat **Channel ID** yang diperlukan untuk **MicroPython** mengirim data. Simpan **ID** tersebut



8. Langkah berikutnya adalah mendaftarkan perangkat yang digunakan untuk perangkat IoT. Klik **Devices** yang ada di bagian atas, kemudian pilih **MQTT**.



9. Langkah setelah itu adalah membuat perangkat yang akan terhubung dengan MQTT. Klik **Add a new device**.



10. Masukkan **Nama**, lalu pilih **Channel ESP8266** tadi. Klik **Add Channel**. Lalu klik **Add Device**

11. Nanti akan muncul **Drop Down** yang berisikan **Credentials** yang harus disimpan. Kopi masing-masing ke sebuah file notepad dan amankan.

12. Konfigurasi ThingSpeak selesai. Berikutnya adalah membuat program untuk NodemCU. Dengan mengunduh file dari link berikut ke **Thonny Editor**: <https://raw.githubusercontent.com/miketeachman/micropython-mqtt-thingspeak/master/publishThingspeak.py>
13. Setelah kode ada di dalam **Thonny Editor**, ubah kode bagian berikut sesuai dengan kode yang sudah diberikan oleh **ThingSpeak**

```

# Configuration parameters
#=====
WIFI_SSID = 'ENTER_WIFI_SSID'
WIFI_PASSWORD = 'ENTER_WIFI_PASSWORD'
THINGSPEAK_MQTT_CLIENT_ID = b"ENTER_THINGSPEAK_MQTT_CLIENT_ID"
THINGSPEAK_MQTT_USERNAME = b"ENTER_THINGSPEAK_MQTT_USERNAME"
THINGSPEAK_MQTT_PASSWORD = b"ENTER_THINGSPEAK_MQTT_PASSWORD"
THINGSPEAK_CHANNEL_ID = b'ENTER_THINGSPEAK_CHANNEL_ID'
#=====

```

14. Lalu bagian berikut ubah menjadi **5 detik**. Sehingga NodeMCU akan mengirimkan data setiap 5 detik.

```
PUBLISH_PERIOD_IN_SEC = 5
```

15. Tambahkan fungsi **blinker pendek** untuk membantu apakah data dikirimkan atau tidak.
16. Masukkan **Import** agar **LED** dapat dikenali **Board**. Letakkan di barisan pertama sebelum **import network**

```
from machine import Pin
```

17. Atur Pin daripada **LED** dengan kode berikut. Letakkan di barisan **sesudah import**

```
p0 = Pin(2, Pin.OUT)
```

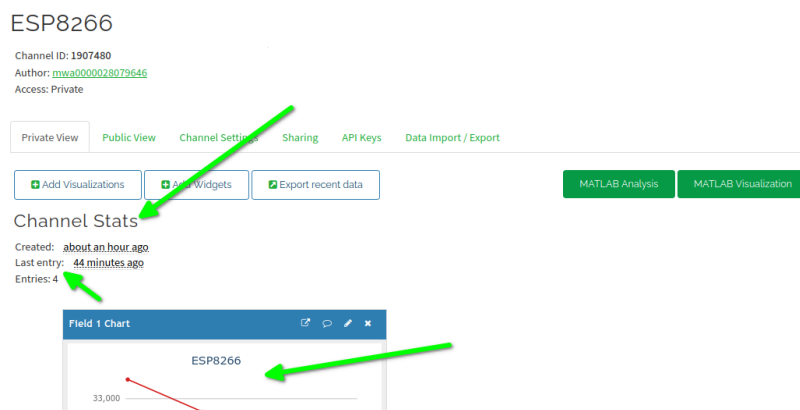
18. Letakkan **SESUDAH** Pin dan **SEBELUM** Configuration Parameters

```
def blinkPendek():  
    p0.value(0)  
    time.sleep_ms(100)  
    p0.value(1)  
    time.sleep_ms(100)
```

19. Lalu sisipkan pemanggil **blinkPendek** di bagian berikut:

```
client.publish(credentials, payload)  
blinkPendek() <=== Kode Baru  
time.sleep(PUBLISH_PERIOD_IN_SEC)
```

20. Jalankan kode, dan perhatikan **Blink Pendek** dari perangkat serta **Channel** yang dibuat.



21. Praktikum 4 Selesai

Bab 5

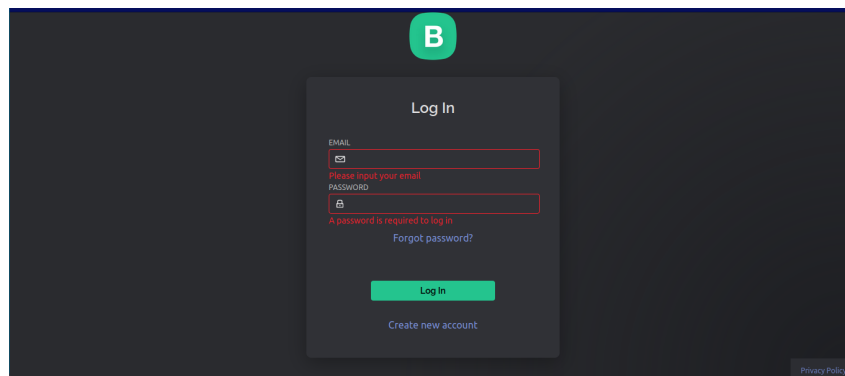
Praktikum 5

5.1 NodeMCU, MicroPython, dan Blynk

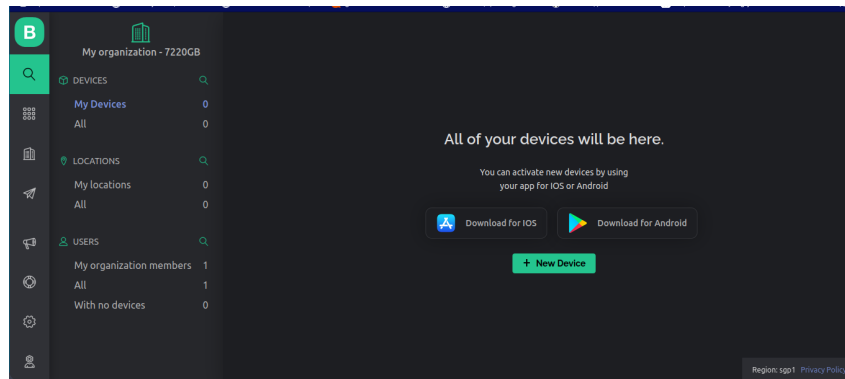
Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke layanan **Blynk** sehingga dapat mengirimkan data ke Cloud secara otomatis. Mahasiswa diharapkan untuk membaca, dan memahami **Persiapan Praktikum** yang ada di halaman sebelumnya.

5.2 Tutorial

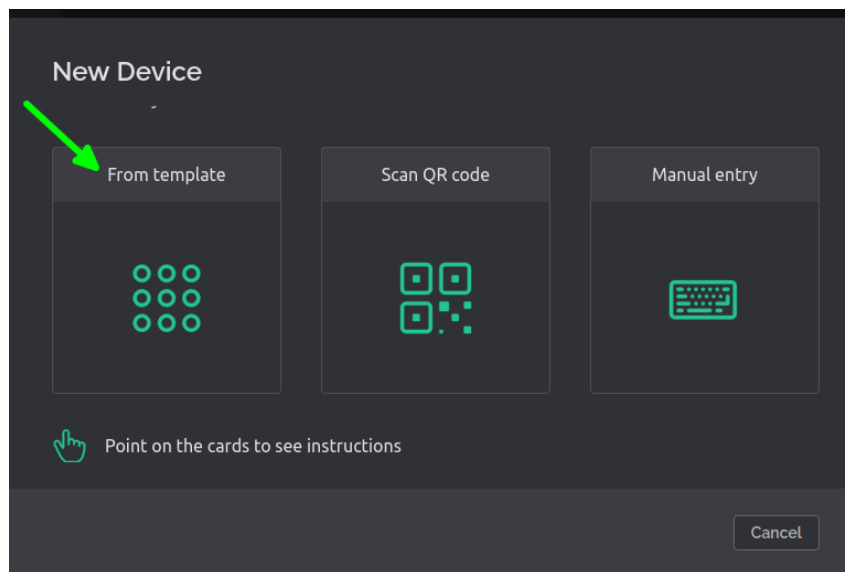
1. Sebelum masuk ke tahapan pemrograman perangkat, hal pertama yang harus dilakukan adalah mendaftarkan akun ke **Blynk** melalui link blynk.cloud



2. Setelah selesai mendaftar, masuklah ke sistem Blynk, dan Blynk akan menampilkan halaman dasbarnya. Jika yang muncul adalah **Add New Device**, maka ikuti langkah selanjutnya. Jika tidak, buat **Template** terlebih dahulu dengan klik **Add New Template**



3. Pilih **From Template** untuk mempermudah pembuatan sistem Blynk untuk ESP8266



4. Untuk template, Pilih **Quickstart Template**, dan Nama perangkat **ESP8266**.
Lalu klik **Create**

New Device

Create new device by filling in the form below

TEMPLATE

Quickstart Template

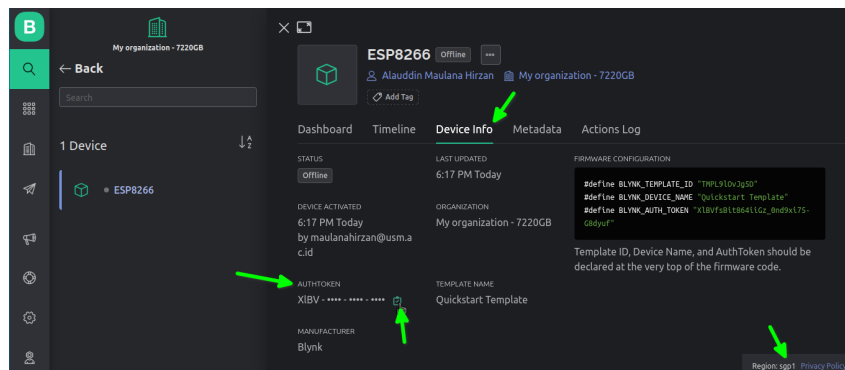
DEVICE NAME

ESP8266

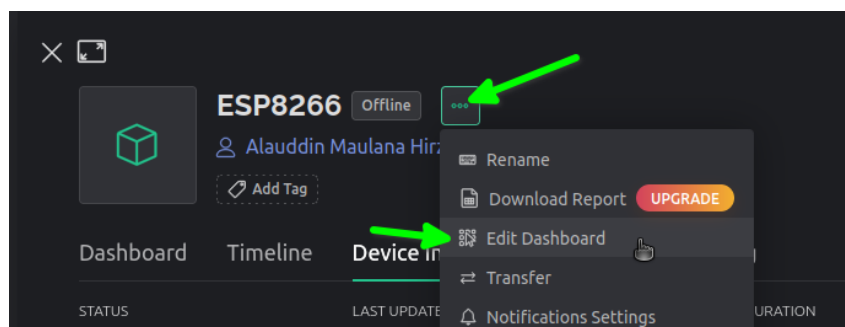
Use letters, digits and spaces only

Cancel Create

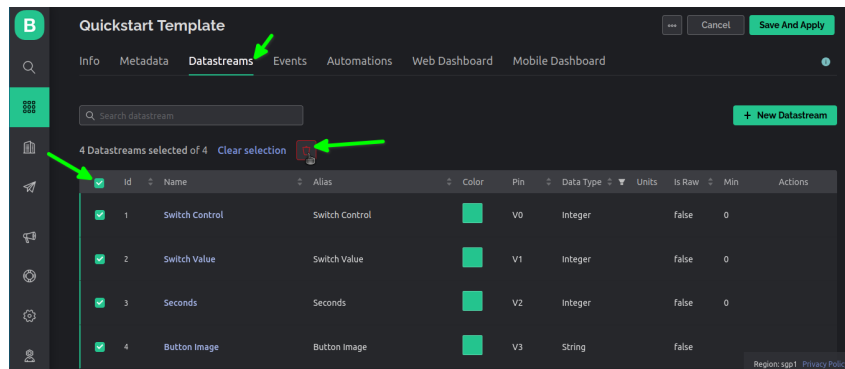
- Untuk bisa mendapatkan **Token** dan **Lokasi Server**, klik **Device Info**, di bagian **Authorization** klik **Icon Copy**. Simpan ke dalam notepad beserta **Lokasi Server** yang ada di bagian **Bawah Kanan**. Contoh Gambar: Lokasi **sgp1**



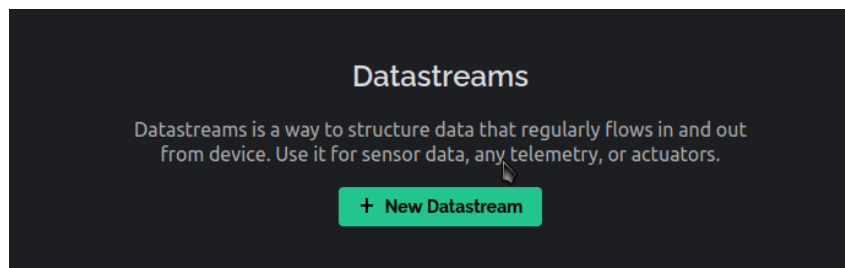
- Simpan baik-baik **Token** dan **Lokasi Server**. Jangan sampai diketahui orang lain.
- Berikutnya adalah memodifikasi **Dasbor** untuk memonitoring data dari sensor perangkat. Klik **Tiga Titik**, lalu klik **Edit Dashboard**.



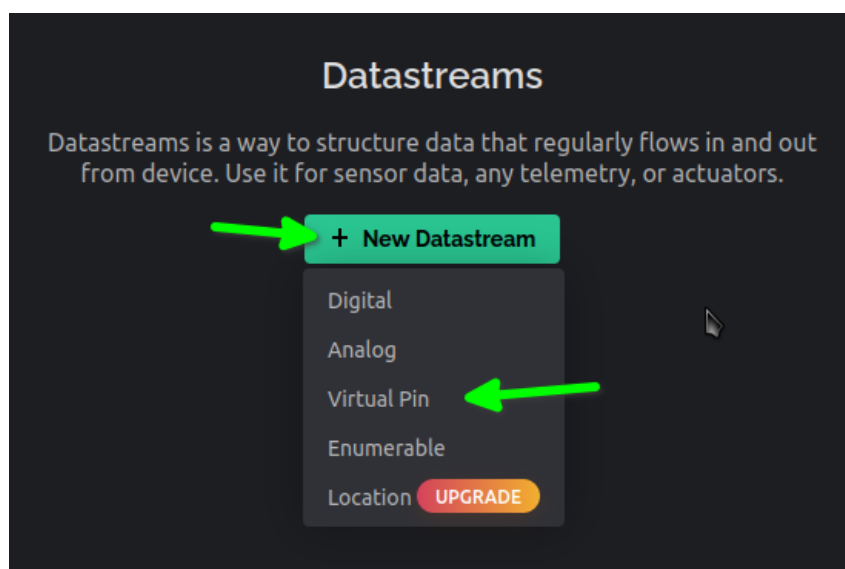
- Sebelum menambahkan **Item Dashboard**, sebaiknya untuk menambahkan **DataStream** yang digunakan untuk menerima data dari perangkat.
- Klik **DataStream** di bagian atas. Pilih **Semua DataStream**, lalu klik **Hapus**.



- Kini **Datastream** sudah kosong



- Berikutnya adalah membuat **DataStream** baru, klik **New Datastream** => **Virtual Pin**



- Contoh Pertama untuk membuat Data Stream Jam adalah sebagai berikut: Isi Nama dengan **Jam** => Pin **V0** => Units **Hr** (untuk Jam) => Nilai **Max** 24 Jam.

Virtual Pin Datastream

NAME: Jam ALIAS: Jam

PIN: V0 DATA TYPE: Integer

UNITS: Hour(s), hr

MIN: 0 MAX: 24 DEFAULT VALUE: 0

ADVANCED SETTINGS

Cancel Create

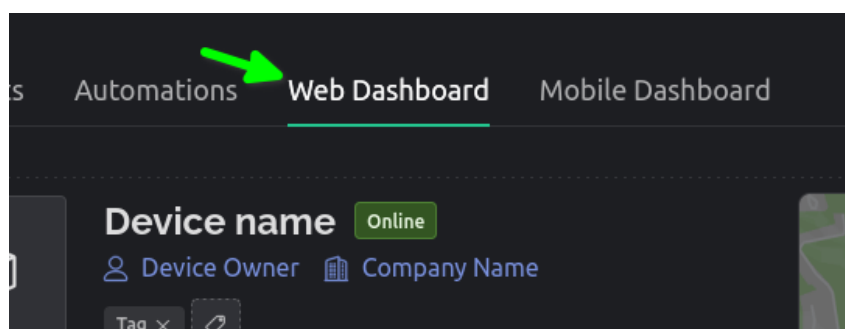
13. Ulangi pembuatan **DataStream** untuk:

- (a) **Jam** : Value **0-24** Satuan **hr** Pin **V0**
- (b) **Menit** : Value **0-60** Satuan **min** Pin **V1**
- (c) **Detik** : Value **0-60** Satuan **s** Pin **V2**
- (d) **PH** : Value **0-14** Satuan **None** Pin **V3**
- (e) **Suhu** : Value **0-38** Satuan **Celcius** Pin **V4**
- (f) **RH** : Value **0-100** Satuan **%** Pin **V5**

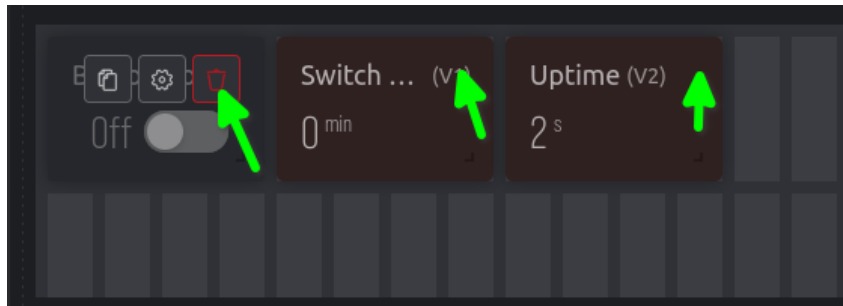
14. Hasil akhir

Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min	Max	Decimals	Default
1	Jam	Jam	Light Green	V0	Integer	hr	false	0	24	-	0
2	Menit	Menit	Orange	V1	Integer	min	false	0	60	-	0
3	Detik	Detik	Light Green	V2	Integer	s	false	0	60	-	0
4	PH	PH	Blue	V3	Integer		false	0	14	-	0
5	Suhu	Suhu	Light Green	V4	Integer	°C	false	0	38	-	0
6	RH	RH	Grey	V5	Integer	%	false	0	100	-	0

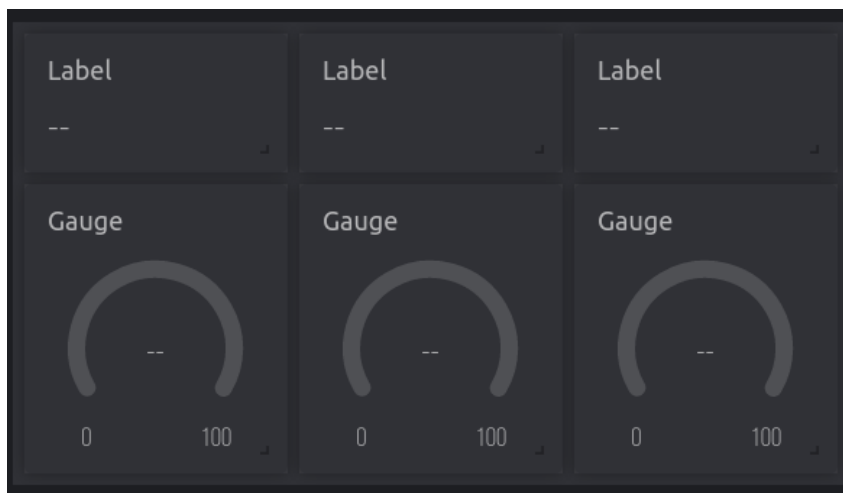
15. Langkah berikutnya adalah mengatur tata letak **Dashbor**. Klik **Web Dashboard**



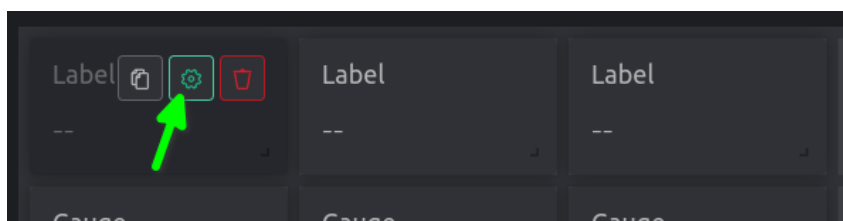
16. Hapus semua elemen yang ada di dasbor dengan mengklik tombol-tombol yang ada di masing-masing elemen.



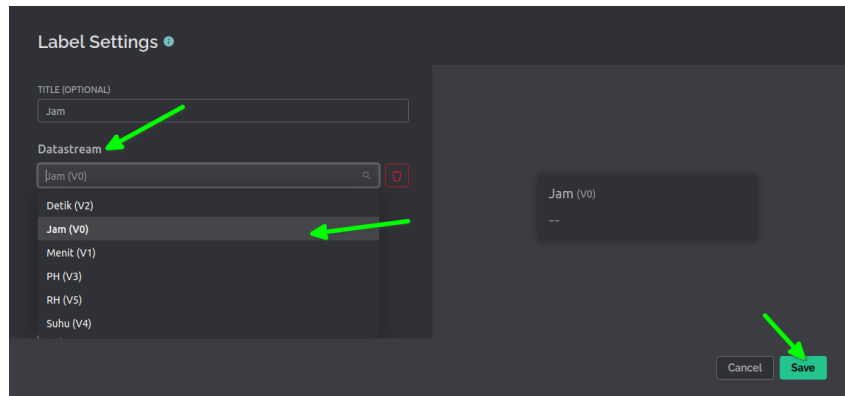
17. Masukkan **3 Label** dan **3 Gauge** seperti berikut ini:



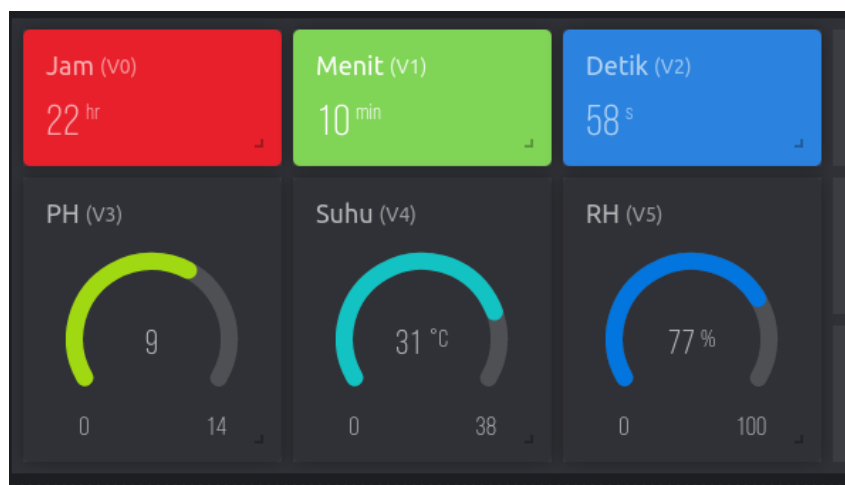
18. Lalu edit masing-masing elemen dimulai dari **Label Kanan**



19. Cukup dengan memilih **Datastream**, maka **Nama Label** akan secara otomatis muncul. Lalu klik **Save** dan ulangi semua elemen.



20. Berikut ini adalah hasil dari perubahan elemen termasuk kustomisasi warna agar lebih menarik.



21. Langkah terakhir berikutnya adalah memberikan pemrograman ke perangkat. Masukkan potongan-potongan kode berikut:

- Kode Import

Potongan Kode

```
from machine import Pin
import time
import network
import ntptime
import urandom as random
import urequests as requestsw
```

- Kode Variabel Global

Potongan Kode

```
# Variabel Wi-Fi
SSID = "<SSID>"
Password = "<Password>"
reset = True
current_time = None

# Variabel Mesin
p0 = Pin(2, Pin.OUT)
blinker = 0

# Variabel Blynk
token = "<Token>"
api = "https://<server>.blynk.cloud/external/api/update?token="
# api = "https://sgp1.blynk.cloud/external/api/update?token="
```

- Kode Blink LED

Potongan Kode

```
# Blink Pendek 50ms
def blinkPendek():
    p0.value(0)
    time.sleep_ms(100)
    p0.value(1)
    time.sleep_ms(100)

# Blink Normal 1s
def blinkNormal():
    p0.value(0)
    time.sleep(1)
    p0.value(1)
    time.sleep(1)
```

- Kode Wi-Fi

Potongan Kode

```
# Koneksi Wi-Fi
def connectWLAN():
    wlan0 = network.WLAN(network.STA_IF)
    if reset:
        wlan0.active(True)

    # Masukkan SSID dan Password
    wlan0.connect(SSID, Password)

    while not wlan0.isconnected():
        wlan0.active(True)
        pass

    status = wlan0.isconnected()
    ip_addr = wlan0.ifconfig()

    return status, ip_addr
```

- Waktu dan Tanggal

Potongan Kode

```
# Set Tanggal
def setDateTime():
    ntptime.settime()

# Get Tanggal
def getDateTime():
    datetime = time.localtime()
    waktu = str(datetime[3]+7)+":"+str(datetime[4])+":"+str(datetime[5])
    tanggal = str(datetime[2])+"/"+str(datetime[1])+"/"+str(datetime[0])
    return waktu, tanggal
```

- Kode main #1

Potongan Kode

```
def main():
    global blinker
    print(">> Memulai ESP8266 ... ")
    blinkNormal()

    print(">> Menghubungkan ke '{}'.format(SSID))
    blinkNormal()
    status,ip_addr = connectWLAN()

    if(status==True):
        print("==>> Terhubung")
        while(blinker<3):
            blinkPendek()
            blinker += 1
        print("==>> IP Address : {}".format(ip_addr[0]))
        print("==>> DNS Address : {}".format(ip_addr[3]))

        # Konfigurasikan Waktu dan Tanggal
        print("")
        print(">> Mengkonfigurasikan Waktu dan Tanggal ...")
        blinkNormal()
        setDateTime()
        waktu,tanggal = getDateTime()
        print("==>> Waktu : {} dan Tanggal : {}".format(waktu,tanggal))

        # Susunan VPIN Blynk
        # v0 = Jam
        # v1 = Menit
        # v2 = Detik
        # v3 = PH
        # v4 = Suhu
        # v5 = RH
        while True:
            print("")
            waktu,tanggal = getDateTime()
            print("==>> Waktu : {} dan Tanggal : {}".format(waktu,tanggal))

            jam = str.split(waktu,":")[0]
            menit = str.split(waktu,":")[1]
            detik = str.split(waktu,":")[2]

            print(">> Mengirim Waktu ke Blynk")
            requests.get(api+token+"&v0="+jam).close()
            requests.get(api+token+"&v1="+menit).close()
            requests.get(api+token+"&v2="+detik).close()
```

- Kode main #2

```

Potongan Kode
# Buat Data Acak sebagai Sampel Sensor
ph = random.getrandbits(12)%14
suhu = random.getrandbits(12)%38
rh = random.getrandbits(12)%100

print(">> Mengirim Suhu ke Blynk")
requests.get(api+token+"&v3="+str(int(ph))).close()
requests.get(api+token+"&v4="+str(int(suhu))).close()
requests.get(api+token+"&v5="+str(int(rh))).close()
blinkPendek()

else:
print("==>> Gagal Terhubung")
blinkPendek()
print("")
print(">> Membatalkan ESP8266 ... ")

main()

```

22. Jalankan kode tersebut dan cek Log

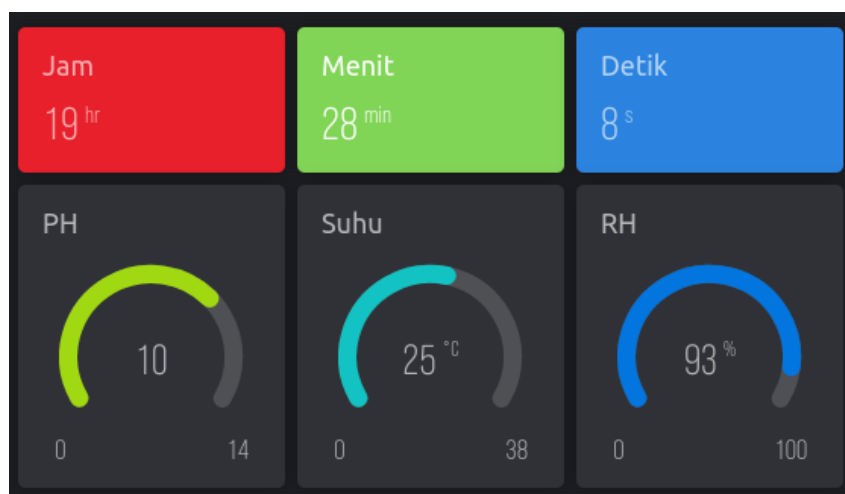
```

>>> %Run -c #EDITOR_CONTENT
>> Memulai ESP8266 ...
>> Menghubungkan ke 'Free Wifi USM 1'
==>> Terhubung
==>> IP Address : 10.20.17.232
==>> DNS Address : 172.16.1.1

>> Mengkonfigurasi Waktu dan Tanggal ...
==>> Waktu : 19:27:40 dan Tanggal : 27/10/2022

==>> Waktu : 19:27:40 dan Tanggal : 27/10/2022
>> Mengirim Waktu ke Blynk
>> Mengirim Suhu ke Blynk

```



23. Praktikum 5 Selesai

Bab 6

Praktikum 6

6.1 NodeMCU, MicroPython, dan Telegram Bot

Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke layanan **Telegram Bot** sehingga dapat mengirimkan data ke Cloud secara otomatis. Mahasiswa diharapkan untuk membaca, dan memahami **Persiapan Praktikum** yang ada di halaman sebelumnya.

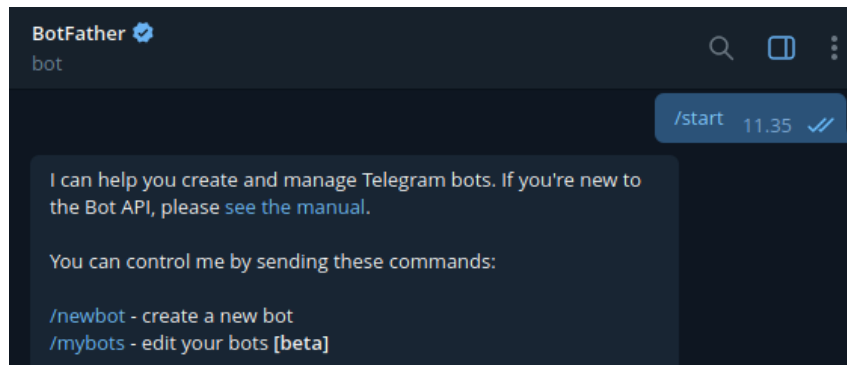
6.2 Tutorial

1. Buka Telegram dan pastikan untuk memiliki **Akun Telegram**. Jika belum memiliki akun Telegram, harap mendaftar terlebih dahulu.
2. Cari akun resmi **@BotFather** untuk dapat membuat Bot.



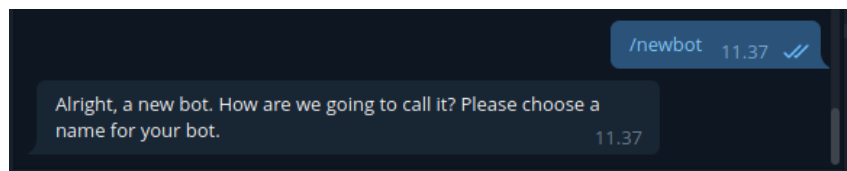
3. Setelah itu, mulai registrasi bot dengan mengirim pesan

`/start` Potongan Kode

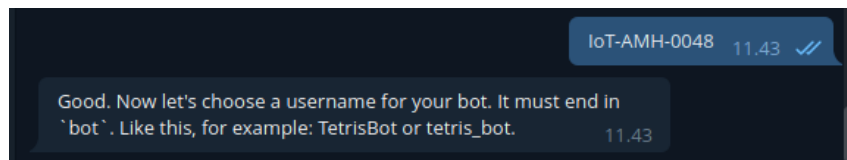


4. Balas dengan pesan

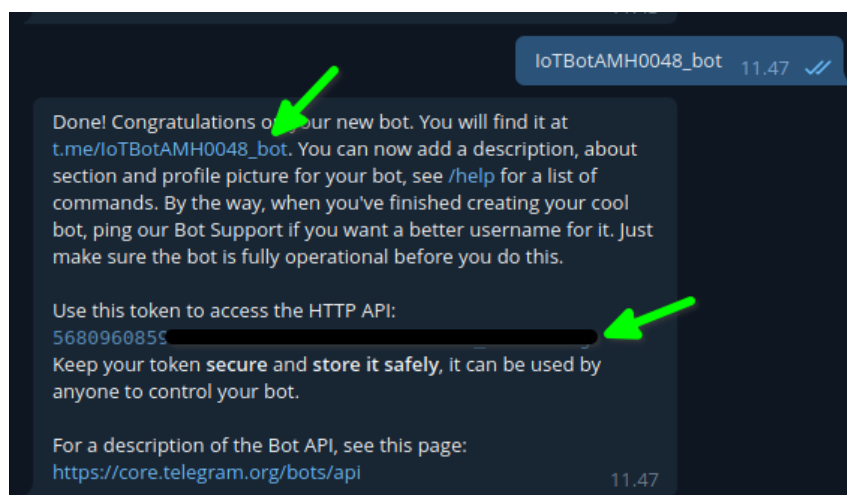
`/newbot` Potongan Kode



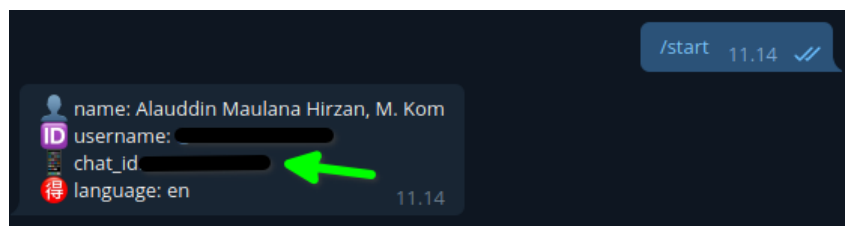
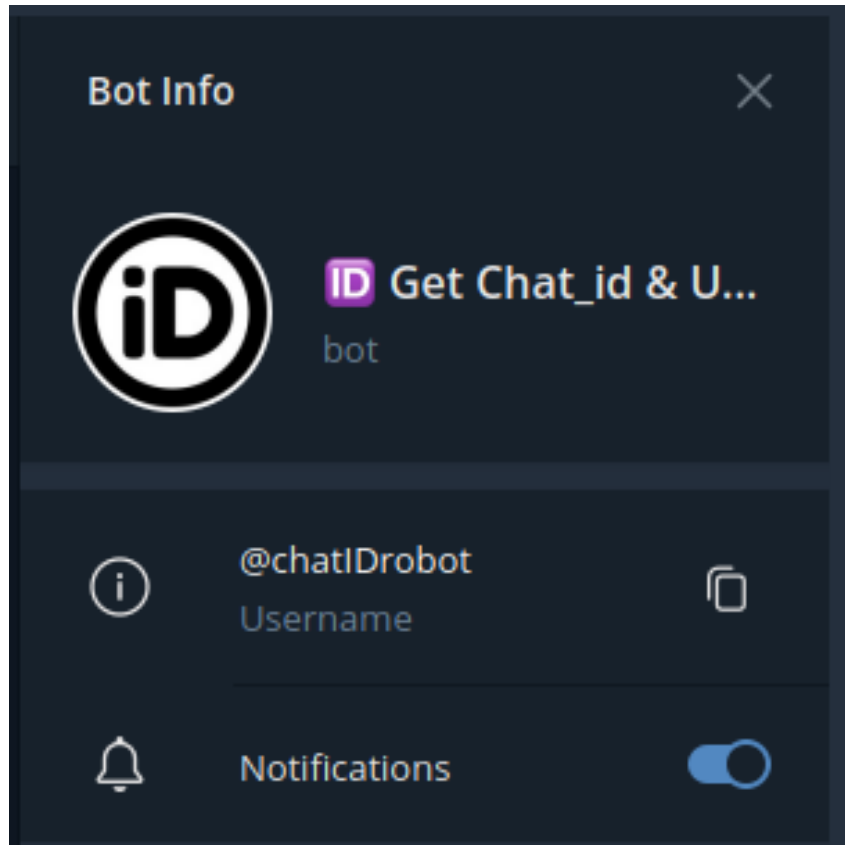
5. Berikan nama yang unik dan berbeda dari yang lain. Contoh IoT-[Inisial Nama]-[4 Digit NIM]



6. Berikutnya adalah memberikan username untuk bot nya. Beri nama seperti berikut: **IoTBot[Inisial Nama][4 Digit NIM]_bot**



7. Gunakan **Link** untuk membuka chat, dan **Token API** untuk akses via NodeMCU. Kini perangkat sudah bisa diakses. Namun sebelum melanjutkan, proses berikutnya adalah mencari **Chat ID** daripada user yang akan **diberi pesan**.
8. Cari Bot **@chatIDrobot** dan kirim pesan **/start** untuk mendapatkan **Chat ID** masing-masing



9. Simpan **Token** dan **Chat ID** baik-baik karena akan digunakan untuk NodeMCU.
10. Langkah berikutnya adalah membuka **Thonny Editor**, dan Hubungkan **NodeMCU** ke **Thonny**
11. Masukkan potongan-potongan kode berikut ini:
 - (a) Kode Import

Potongan Kode

```
from machine import Pin
import time
import network
import ntptime
import urandom as random
import urequests as requests
```

(b) Variabel Global

Potongan Kode

```
# Variabel Wi-Fi
SSID = "<SSID>"
Password = "<Password>"
reset = True
current_time = None

# Variabel Mesin
p0 = Pin(2, Pin.OUT)
blinker = 0

token = "<Token>"
chat_id = "<Chat ID>"
url = "https://maulanahirzan.pythonanywhere.com/sendMessage"
```

(c) Kode Blink LED Notifikasi

Potongan Kode

```
# Blink Pendek 50ms
def blinkPendek():
    p0.value(0)
    time.sleep_ms(100)
    p0.value(1)
    time.sleep_ms(100)

# Blink Normal 1s
def blinkNormal():
    p0.value(0)
    time.sleep(1)
    p0.value(1)
    time.sleep(1)
```

(d) Kode Koneksi Wi-Fi

Potongan Kode

```
# Koneksi Wi-Fi
def connectWLAN():
    wlan0 = network.WLAN(network.STA_IF)
    if reset:
        wlan0.active(True)

        # Masukkan SSID dan Password
        wlan0.connect(SSID, Password)

        while not wlan0.isconnected():
            wlan0.active(True)
            pass

    status = wlan0.isconnected()
    ip_addr = wlan0.ifconfig()
    return status, ip_addr
```

(e) Kode Waktu dan Tanggal

Potongan Kode

```
# Set Tanggal
def setDateTime():
    ntptime.settime()

# Get Tanggal
def getDateTime():
    datetime = time.localtime()
    waktu = str(datetime[3]+7)+":"+str(datetime[4])+":"+str(datetime[5])
    tanggal = str(datetime[2])+"/"+str(datetime[1])+"/"+str(datetime[0])
    return waktu, tanggal
```

(f) Kode terakhir yang perlu ditambahkan adalah **Kode Main #1**

Potongan Kode

```
def main():
    global blinker
    print(">> Memulai ESP8266 ... ")
    blinkNormal()

    print(">> Menghubungkan ke '{}'.format(SSID))
    blinkNormal()
    status,ip_addr = connectWLAN()

    if(status==True):
        print("==>> Terhubung")
        while(blinker<3):
            blinkPendek()
            blinker += 1

        print("==>> IP Address : {}".format(ip_addr[0]))
        print("==>> DNS Address : {}".format(ip_addr[3]))

        # Konfigurasi Waktu dan Tanggal
        print("")
        print(">> Mengkonfigurasi Waktu dan Tanggal ...")
        blinkNormal()
        setDateTime()
        waktu,tanggal = getDateTime()
        print("==>> Sukses")

        print("")
        print(">> ESP8266 Memulai Pengiriman Data ... ")
        while True:
            print(">> Mengirim Data ke Telegram Bot")

            # Ambil Data Waktu dan Tanggal
            waktu,tanggal = getDateTime()
            print("==>> Waktu : {} dan Tanggal : {}".format(waktu,tanggal))

            # Buat Data Acak sebagai Sampel Sensor
            ph = random.getrandbits(12)%14
            suhu = random.getrandbits(12)%38
            rh = random.getrandbits(12)%100
            print("==>> PH : {}, Suhu : {}, dan RH {}".format(ph,suhu,rh))
```

(g) Kode Main #2

Potongan Kode

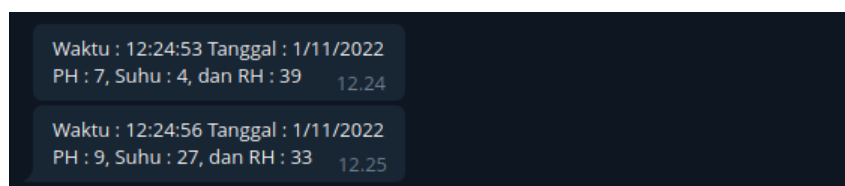
```
# Susun URL Web Service
post_token = "?token="+token
post_chatid = "&chatid="+chat_id
post_waktu = "&waktu="+waktu
post_tanggal = "&tanggal="+tanggal
post_ph = "&ph="+str(int(ph))
post_suhu = "&suhu="+str(int(suhu))
post_rh = "&rh="+str(int(rh))
url_api1 = url+post_token+post_chatid+post_waktu
url_api2 = url_api1+post_tanggal+post_ph
url_api = url_api2+post_suhu+post_rh
req = requests.get(url_api)
result = req.status_code

if(result == 200):
    print("==>> Sukses")
    req.close()
    blinkPendek()
else:
    print("==>> ESP8266 gagal mengirim ... ")

else:
    print("==>> Gagal Terhubung")
    blinkPendek()
    print("")
    print(">> Membatalkan ESP8266 ... ")

main()
```

12. Jalankan NodeMCU dan perhatikan **Telegram Bot** apakah Bot menerima data yang dikirimkan



13. Praktikum 6 Selesai

Bab 7

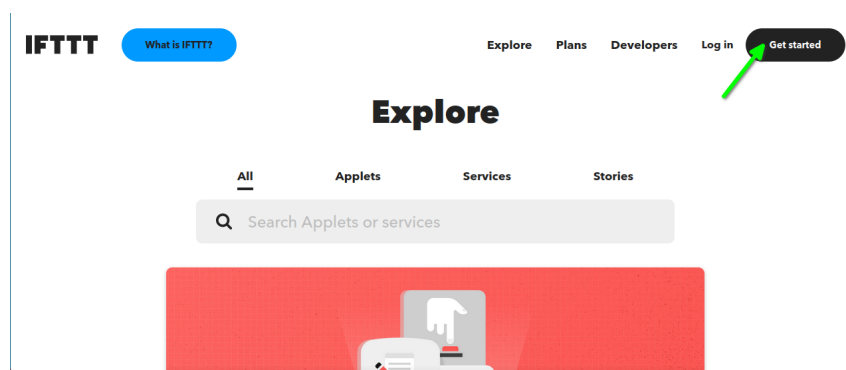
Praktikum 7

7.1 NodeMCU, MicroPython, dan IFTTT

Di bagian ini mahasiswa diajarkan bagaimana menghubungkan perangkat NodeMCU ke layanan global **IF THIS THEN THAT (IFTTT)** sehingga dapat mengirimkan data ke berbagai macam jenis layanan lainnya dari level perangkat hingga *Software-as-a-Service (SaaS)*. Mahasiswa diharapkan untuk membaca, dan memahami **Persiapan Praktikum** yang ada di halaman sebelumnya.

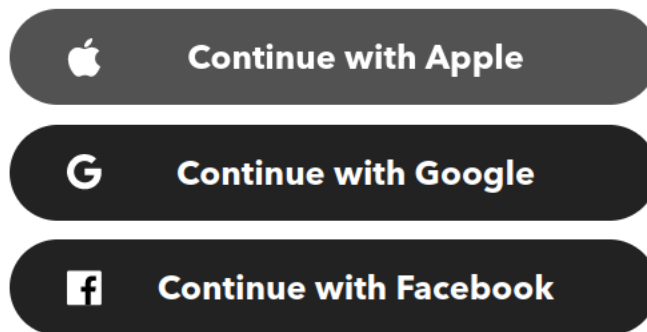
7.2 Tutorial

1. Untuk memulai praktikum ini, mahasiswa diwajibkan melakukan pendaftaran di website IFTTT dengan link sebagai berikut: <https://ifttt.com>. Klik **Get Started**

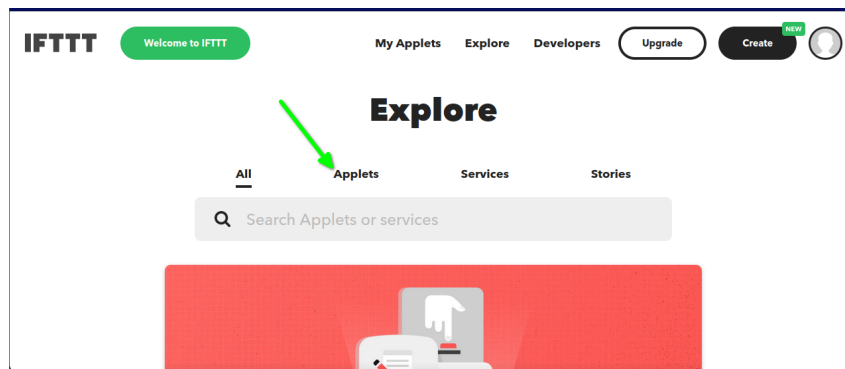


2. Masuk dengan menggunakan salah satu akun yang dipunyai.

Get started with IFTTT



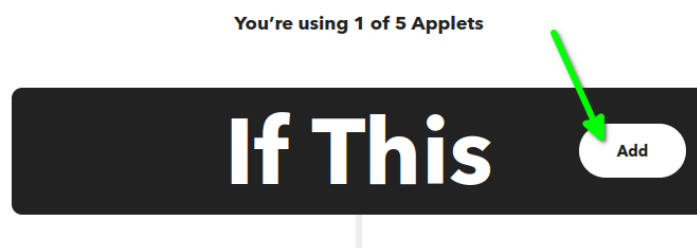
3. Setelah berhasil masuk, maka secara otomatis akan dibawa ke halaman Home lagi. Klik **Applets**



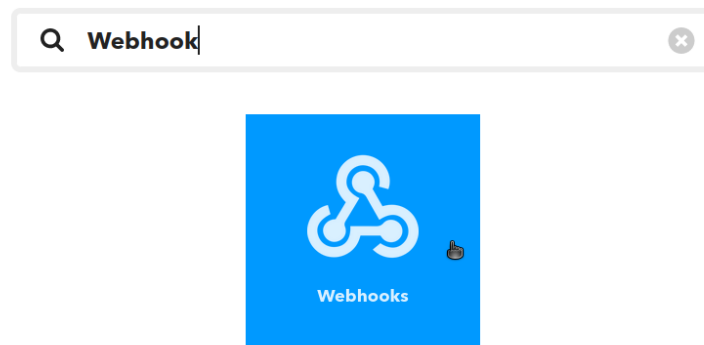
4. Di bagian **Applets** ini terdapat **BANYAK SEKALI** aplikasi-aplikasi otomatisasi yang dapat digunakan untuk berbagai macam hal.
5. Namun untuk praktikum ini, mahasiswa akan membuat **Applet** mereka sendiri untuk menghubungkan perangkat **ESP8266** ke Layanan yang ada
6. Klik **Create** di bagian pojok kanan atas



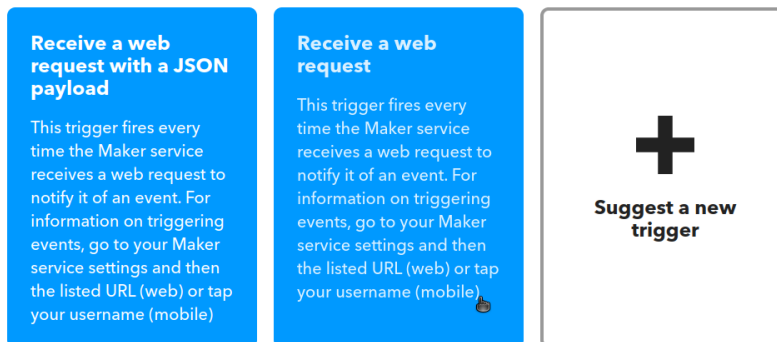
7. Di bagian **If This**, klik tombol **Add**



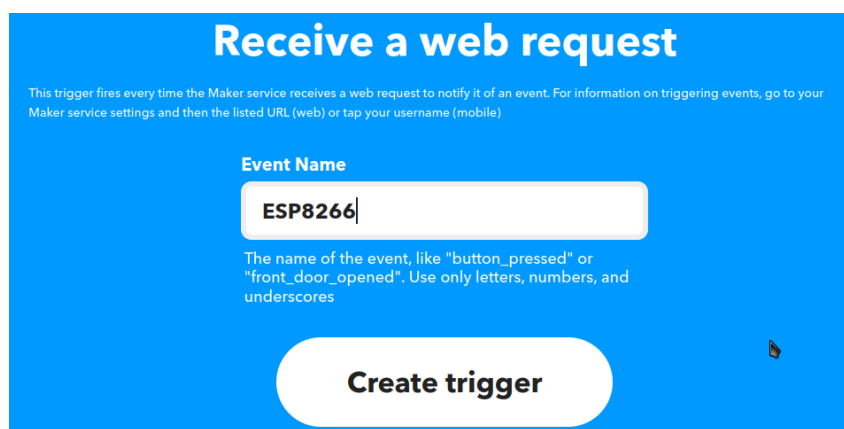
8. Cari **WebHook** dan Klik **Layanan** **Tersebut**



9. Di bagian ini konfigurasi **WebHook** untuk menerima Web Request dengan klik **Receive a web request** biasa.



10. Berikutnya adalah memberikan nama **Trigger**. Sebagai contoh adalah **ESP8266**. Klik **Create Trigger**

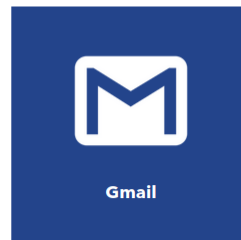


11. Berikutnya adalah menambah aksi yang harus dilakukan oleh IFTTT ketika menerima data dari perangkat. Klik **Add**

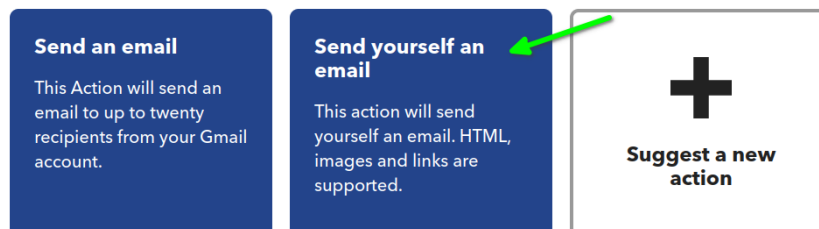


12. Dari sini mahasiswa dapat memilih satu dari banyak layanan yang ada. Namun untuk praktikum ini, layanan yang digunakan adalah **GMAIL**

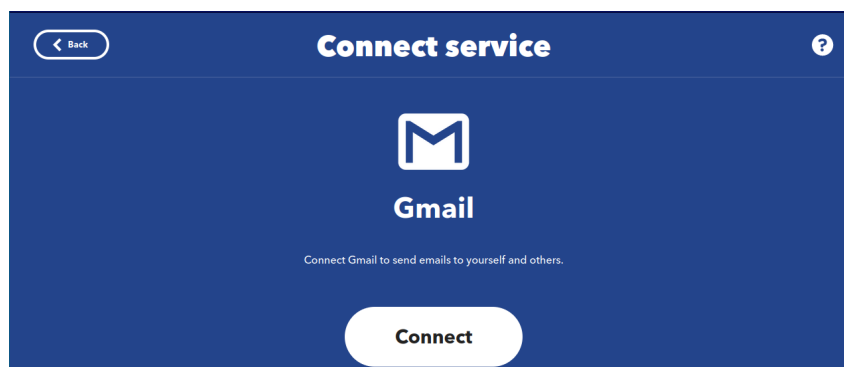
Choose a service



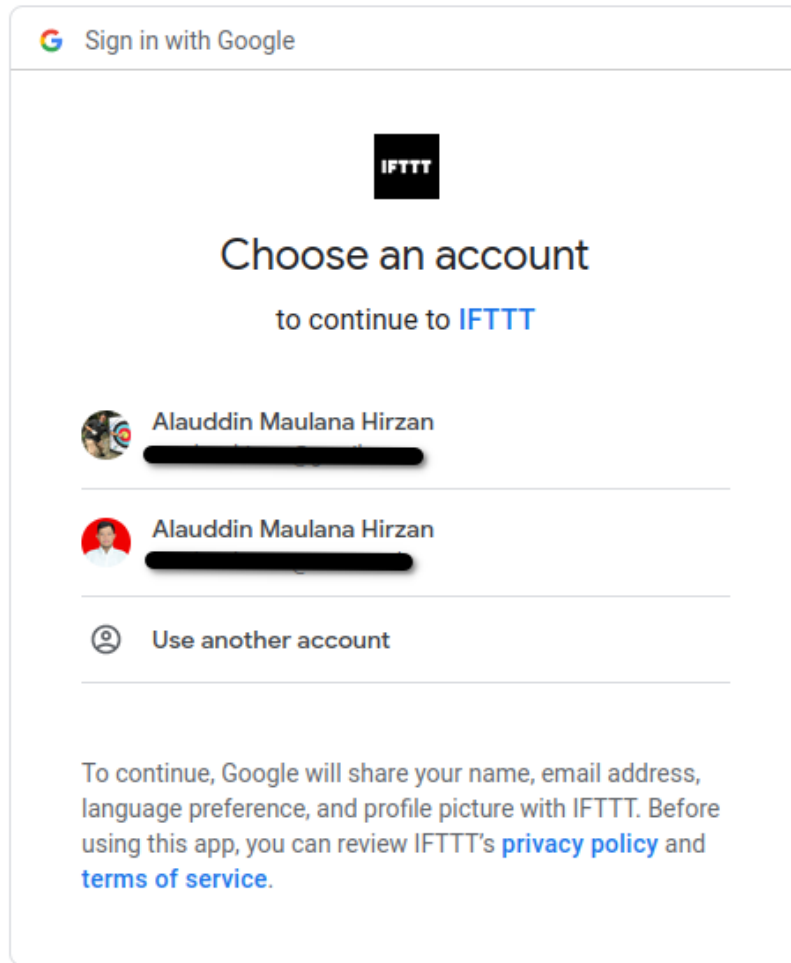
13. Pilih opsi "**Send yourself an e-mail**" untuk mengirim ke diri sendiri.



14. Jika belum pernah menghubungkan e-mail ke IFTTT, maka IFTTT akan meminta user untuk menghubungkan ke email mereka. Klik **Connect**

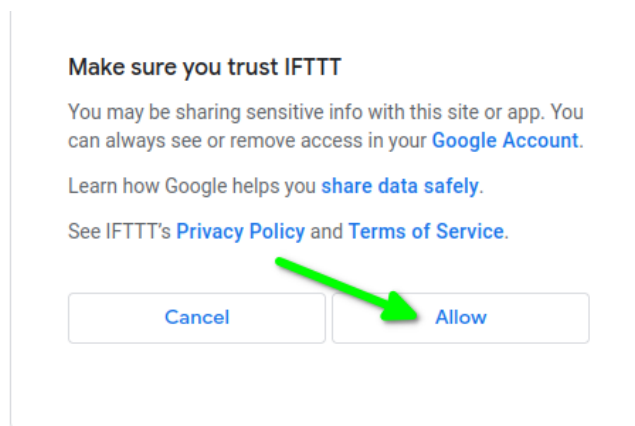


15. Pilih email target

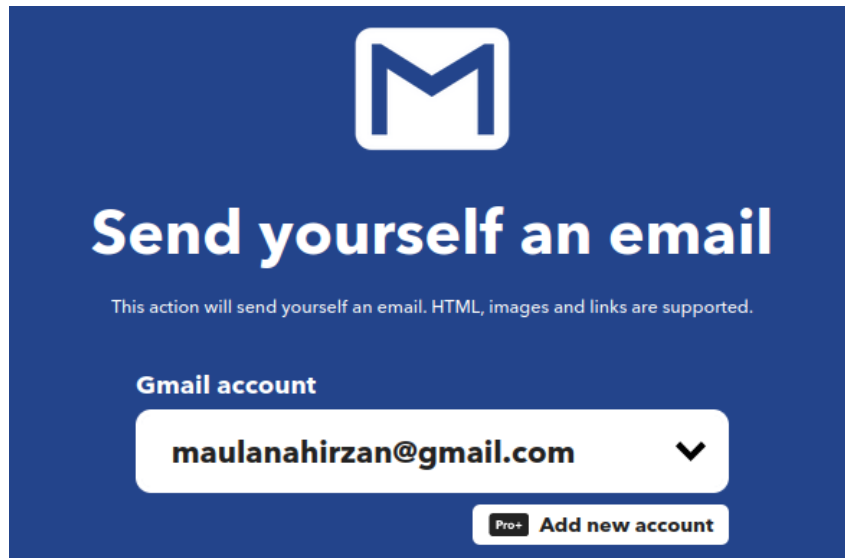


}

16. Izinkan IFTTT untuk mengakses email



17. Jika sudah terhubung dengan benar, maka email akan muncul di sana

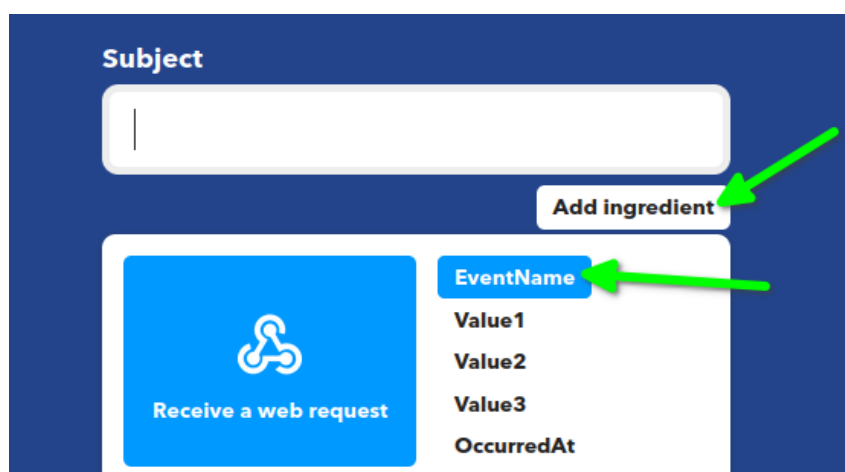


18. Berikutnya adalah membuat Judul dan Isi Pesan E-mail. Scroll down dan kosongkan **Subject**



19. Klik **Add Ingredient** untuk memasukkan data yang berasal dari perangkat seperti **Nama Event, Waktu, dan Data**

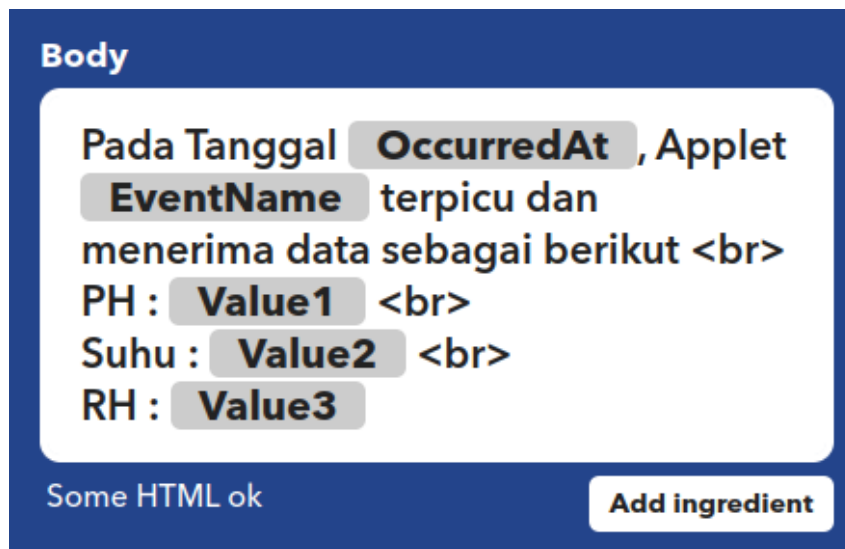
20. Klik **Add Ingredient** dan masukkan **EventName**



21. Setelah itu variabel akan dimasukkan ke dalam **Subject**. Tambahkan kata-kata bebas. Contoh Final

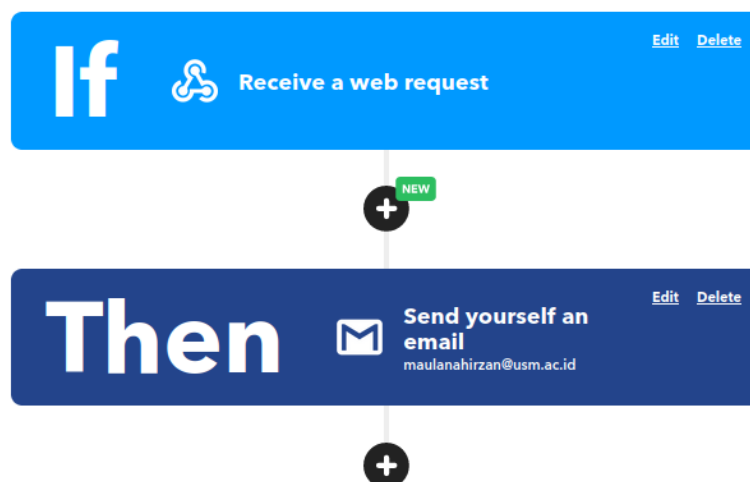


22. Di bagian **Body**, isi pesan bebas. Contoh



23. Klik **Button** yang ada di bawah untuk menyelesaikan tahapan ini.

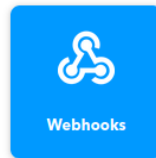
24. Applet sudah siap. Klik lagi tombol yang ada di bawah untuk menyelesaikan.



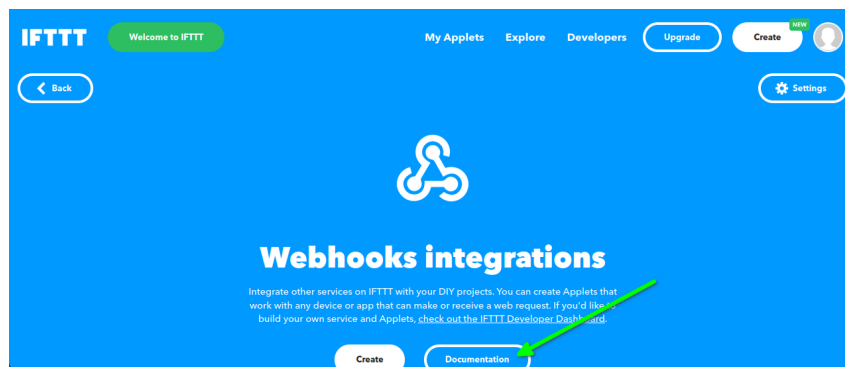
25. Berikutnya adalah mencari **LINK API WebHooks**

26. Kembali ke **Home** lalu klik **Service**, atau klik link <https://ifttt.com/explore/services>

27. Cari **WebHook**, lalu klik **Service** itu



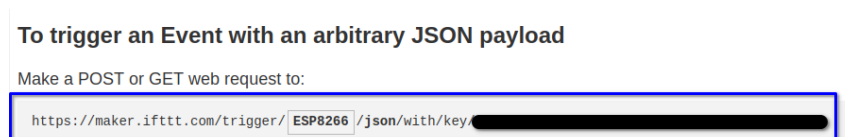
28. untuk mendapatkan link API, klik **Documentation** di halaman **WebHook**



29. Masukkan **Event** yang ditunjuk oleh panah dengan **EVENT** yang sudah dibuuh sebelumnya. Contoh **ESP8266**



30. Untuk memperoleh link, kopi **URL** yang ditunjuk oleh kotak biru. Simpan **URL** tersebut !!!



31. Berikutnya adalah membuat kode dengan **MicroPython**

(a) Kode Import

Potongan Kode

```
from machine import Pin
import time
import network
import urandom as random
import urequests as requests
```

- (b) Kode Variabel Global. Isi <Link API> dengan URL sebelumnya

Potongan Kode

```
# Variabel Wi-Fi
SSID = "<SSID>"
Password = "<Password>"
reset = True

# Variabel Mesin
p0 = Pin(2, Pin.OUT)
blinker = 0

# WebHook URL
api_url = "<Link API>"
```

- (c) Masukkan Fungsi LED Blink

Potongan Kode

```
# Blink Pendek 50ms
def blinkPendek():
    p0.value(0)
    time.sleep_ms(100)
    p0.value(1)
    time.sleep_ms(100)

# Blink Normal 1s
def blinkNormal():
    p0.value(0)
    time.sleep(1)
    p0.value(1)
    time.sleep(1)
```

- (d) Masukkan Fungsi Wi-Fi

Potongan Kode

```
# Koneksi Wi-Fi
def connectWLAN():
    wlan0 = network.WLAN(network.STA_IF)
    if reset:
        wlan0.active(True)

        # Masukkan SSID dan Password
        wlan0.connect(SSID, Password)

        while not wlan0.isconnected():
            wlan0.active(True)
            pass

    status = wlan0.isconnected()
    ip_addr = wlan0.ifconfig()
    return status, ip_addr
```

(e) Masukkan kode **Generator Data**

Potongan Kode

```
# Data Generator
def genData():
    ph = random.getrandbits(12)%14
    suhu = random.getrandbits(12)%38
    rh = random.getrandbits(12)%100
    return ph, suhu, rh
```

(f) Terakhir adalah kode **main**

Potongan Kode

```
# Main Function
def main():
    print(">> Memulai ESP826 ... ")
    print(">> Menghubungkan ke {}".format(SSID))
    status,ip_addr = connectWLAN()

    if(status == True):
        print("==>> Terhubung")

        print("")
        print(">> ESP8266 Mengambil Data Sensor ... ")
        ph,suhu,rh = genData() # Seolah-olah Ambil Data Sensor
        print("==>> Sukses")

        print("")
        print(">> ESP8266 Mengirim Data ke GMAIL via IFTTT WebHook")
        print("==>> PH : {}, Suhu : {}, dan RH : {}".format(ph,suhu,rh))
        data = {"value1":ph,"value2":suhu,"value3":rh}
        resp = requests.post(api_url,json=data)
        if(resp.status_code==200):
            print("==>> Sukses")
            blinkPendek()
        else:
            print("==>> Gagal")
            blinkPanjang()
        resp.close()
    else:
        print("==>> Gagal Terhubung")

main()
```

32. Jalankan kode, tunggu sampai berhasil mengirim

```
>>> %Run -c $EDITOR_CONTENT
>> Memulai ESP826 ...
>> Menghubungkan ke MikroTik-Net
==>> Terhubung

>> ESP8266 Mengambil Data Sensor ...
==>> Sukses

>> ESP8266 Mengirim Data ke GMAIL via IFTTT WebHook
==>> PH : 5, Suhu : 29, dan RH : 97
==>> Sukses

>>> |
```

33. Cek E-Mail

ESP8266 Aktif Saat November 9, 2022 at 11:10AM ⌵ Kotak Masuk ×



maulanahirzan@usm.ac.id

11.10 (0)

kepada saya ▾

Pada Tanggal November 9, 2022 at 11:10AM, Applet ESP8266 terpicu dan menerima data sebagai berikut

PH : 5

Suhu : 29

RH : 97

34. Praktikum 7 Selesai

Bab 8

Praktikum 8

8.1 NodeMCU, MicroPython, dan Tugas Praktikum

Di bagian ini mahasiswa diwajibkan membuat sebuah proyek dengan perangkat yang dipunyai dengan metode-metode yang sudah diajarkan di pertemuan-pertemuan sebelumnya. Untuk mendapatkan nilai Praktikum mahasiswa wajib memenuhi kriteria yang diberikan oleh dosen.

8.2 Tugas Praktikum

Berikut ini adalah **kriteria** yang dinilai dari **Tugas Praktikum**:

1. Tugas bersifat **Individual** meski dikerjakan secara tim
2. Kelompok minimal **3 orang**
3. Anggota kelompok wajib masing-masing membuat satu proyek IoT. Sehingga **1 Kelompok = 3 Orang = 3 Proyek IoT, dst**
4. Platform yang bisa digunakan:
 - Blynk
 - Telegram Bot
 - IFTTT
 - ThingSpeak
 - Platform Lain (KaaIoT, Cayenne, dll)
5. Apa yang harus dibuat
 - (a) Dokumentasi Perangkat → **PDF**
 - Daftar Perangkat Hardware
 - Daftar Perangkat Software
 - Rangkaian Perangkat
 - Topologi Konektivitas

- Kode-kode Perangkat

6. Kode Pembantu

- Kode LED

Potongan Kode

```
# Blink Pendek 50ms
def blinkPendek():
    p0.value(0)
    time.sleep_ms(100)
    p0.value(1)
    time.sleep_ms(100)

# Blink Normal 1s
def blinkNormal():
    p0.value(0)
    time.sleep(1)
    p0.value(1)
    time.sleep(1)
```

- Kode Data Generator

Potongan Kode

```
# Data Generator
def genData():
    ph = random.getrandbits(12)%14
    suhu = random.getrandbits(12)%38
    rh = random.getrandbits(12)%100
    return ph,suhu,rh
```

7. Projek harus dikumpulkan sesuai dengan arahan yang diberikan dosen

8. Praktikum 8 Selesai