

## KATA PENGANTAR

---

Pada era globalisasi, perkembangan teknologi begitu cepat dan kuat. Salah satu dari perkembangan teknologi adalah internet. Internet memberikan kemudahan dalam berbagai hal, salah satu diantaranya adalah memudahkan komunikasi dari jarak yang begitu jauh. Selain itu, internet memberikan kemudahan dalam memberikan akses sebuah informasi terbaru dari segala penjuru dunia.

Perkembangan internet pun memberikan pengaruh pada Dunia Industri yaitu dengan terbentuknya fenomena Industri 4.0, yang mana fenomena tersebut mengkolaborasikan teknologi internet dan teknologi otomatisasi. Salah satu bentuk dari perkembangan yang ada di Industri 4.0 adalah konsep *Internet of Things*, yaitu sebuah konsep yang memberikan kemudahan bagi manusia untuk dapat melakukan aktivitas dari jarak yang jauh dengan sebuah sistem yang terhubung ke internet. Contohnya adalah sebuah rumah yang dapat dikendalikan melalui sebuah *smartphone* atau sebuah *weather station* yang dapat dipantau melalui aplikasi berbasis website.

Dengan perkembangan teknologi yang begitu pesat, maka dibuatlah sebuah Modul Pembelajaran ‘Internet of Things’ yang bertujuan untuk membantu para siswa agar dapat mengetahui dan mengasah kemampuan mereka terkait *skill* yang dibutuhkan untuk kebutuhan Industri 4.0 yang dibuat dengan teknis *based on project*

Bogor, 26 Juni 2021

Penyusun

## DAFTAR ISI

---

|  |     |
|--|-----|
| KATA PENGANTAR .....                                       | III |
| DAFTAR ISI.....  | 1   |
| PENGENALAN INSTALASI ARDUINO IDE DAN NODEMCU LIBRARY ..... | 2   |
| PROJECT I .....  | 24  |
| PROJECT II.....  | 51  |
| PROJECT III.....   | 59  |
| PROJECT IV .....   | 755 |
| PROJECT V.....   | 99  |
| PROJECT VI .....   | 136 |
| DAFTAR PUSTAKA .....                                       | 160 |



## PENGENALAN

### INSTALASI ARDUINO IDE DAN NODEMCU LIBRARY

#### - DESKRIPSI :

Cara melakukan Instalasi Arduino IDE dan Board NodeMCU pada Arduino IDE serta pengenalan Arduino IDE dan juga Board NodeMCU dan membuat Program sederhana pada NodeMCU

#### - BACAAN SINGKAT :

- **Arduino IDE** adalah sebuah Aplikasi lintas *platform* untuk Windows, Linux, dan juga MacOS yang ditulis dalam fungsi-fungsi dari C dan C++ . Aplikasi ini digunakan untuk memogram dan mengunggah program ke Arduino *Board* yang kompatibel dan juga *board-board* lainnya yang kompatibel dengan bantuan *core* dari pihak ke – 3 seperti, NodeMCU

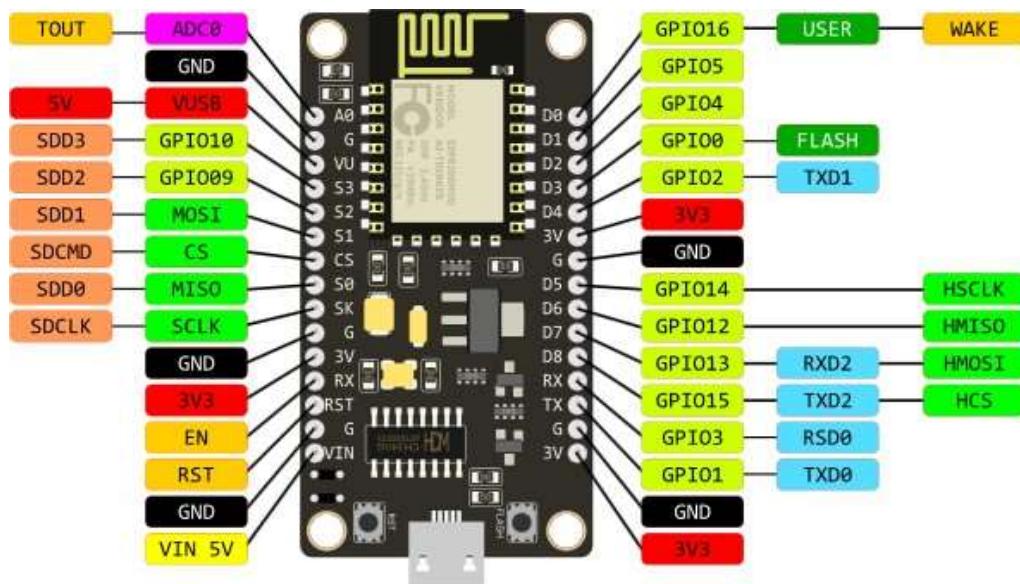
```
sketch_dec07a | Arduino 1.8.3
File Edit Sketch Tools Help
sketch_dec07a
void setup() {
  // put your setup code here, to run once:
}

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

2 Arduino/Genuino Uno on COM3

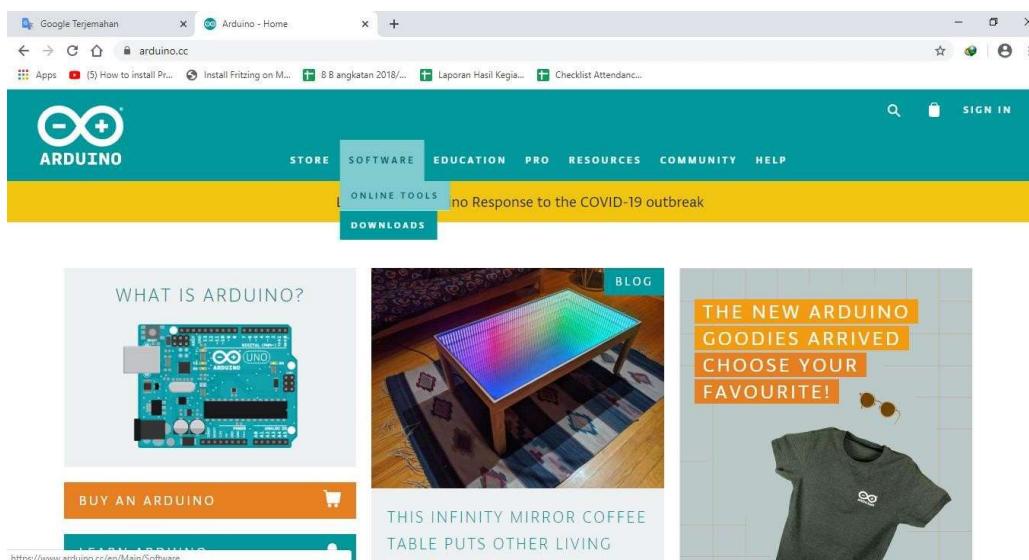


- **NodeMCU** adalah sebuah *board* yang mirip dengan Arduino Uno namun dibuat dengan menggunakan *chip* ESP8266 yang merupakan salah satu *module* Wi-Fi sehingga **NodeMCU** memiliki kelebihan yaitu sudah memiliki integrasi dengan *module* Wi-Fi dan dapat melakukan koneksi dengan Jaringan Wi-Fi



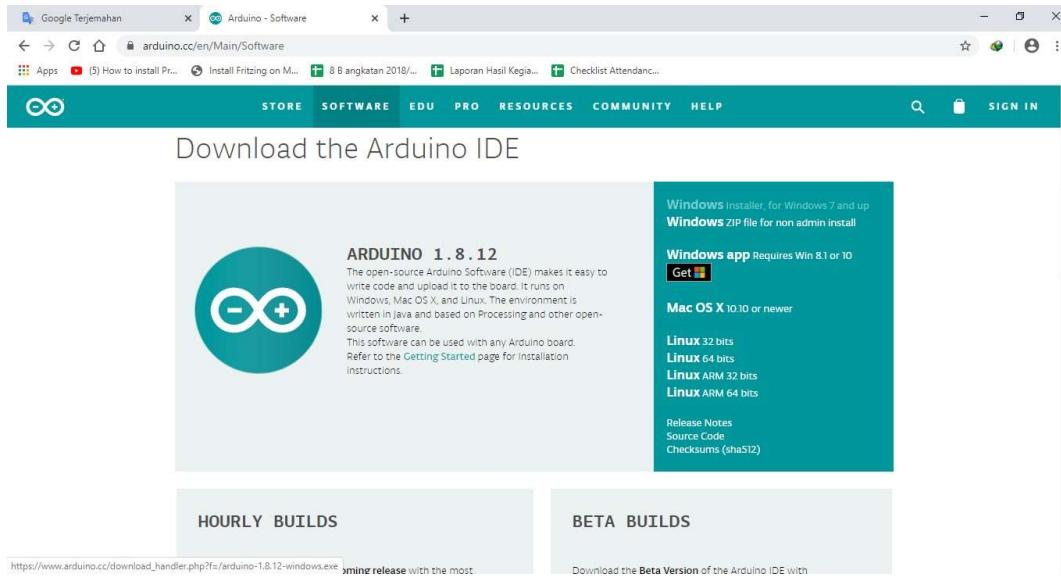
- **Cara Install Arduino IDE (Pengguna Windows)**

1. Pergi menuju halaman Website: <http://arduino.cc>
2. Kemudian klik Menu *Software* dan pilih *Download*

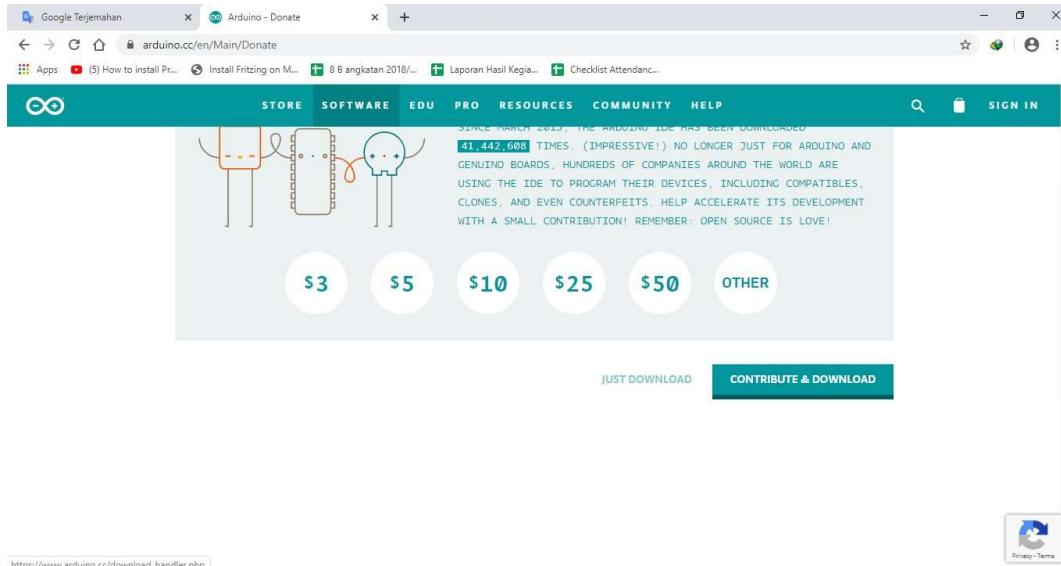




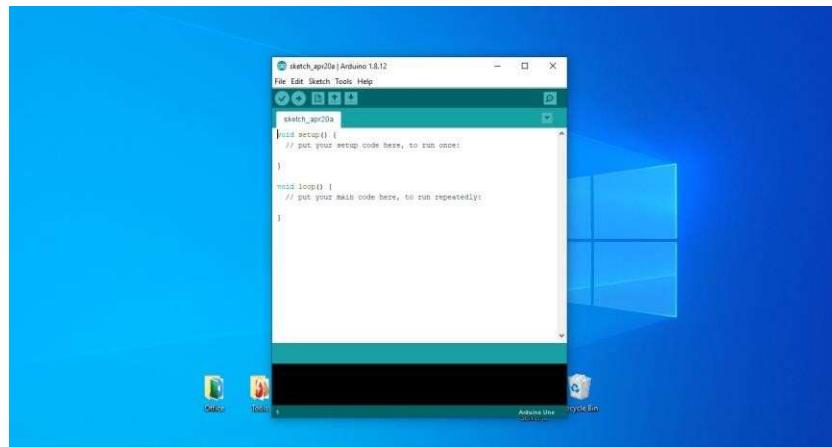
3. Lalu, setelah itu scroll ke bawah dan klik **Download** untuk dengan pilihan **Windows Installer for Windows 7 and Up**



4. Kemudian lewati halaman di bawah dan cukup melanjutkan dengan klik bagian **Just Download**

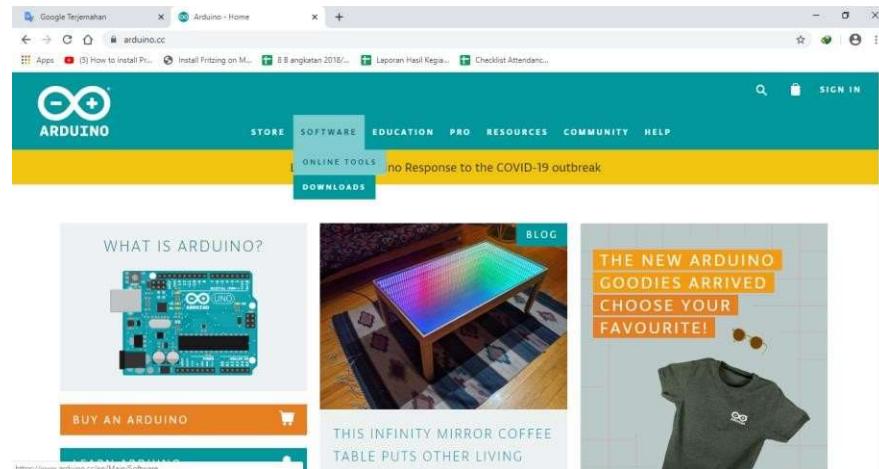


5. Selanjutnya, tunggu hingga *Download* Aplikasi Arduino IDE selesai.
6. Setelah selesai, buka lokasi dimana kalian menyimpan Aplikasi Arduino IDE yang telah di-*download*. Lalu, klik *launcher* Arduino IDE – nya
7. Kemudian, *install* Arduino IDE dengan mengikuti langkah-langkah yang ada. Apabila telah selesai, lalu buka Arduino IDE yang telah berhasil di-*install*

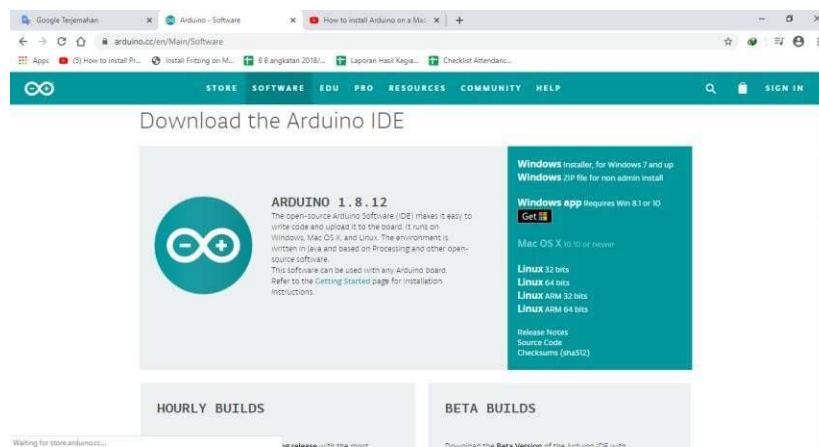


### - Cara Install Arduino IDE (Pengguna MacOS)

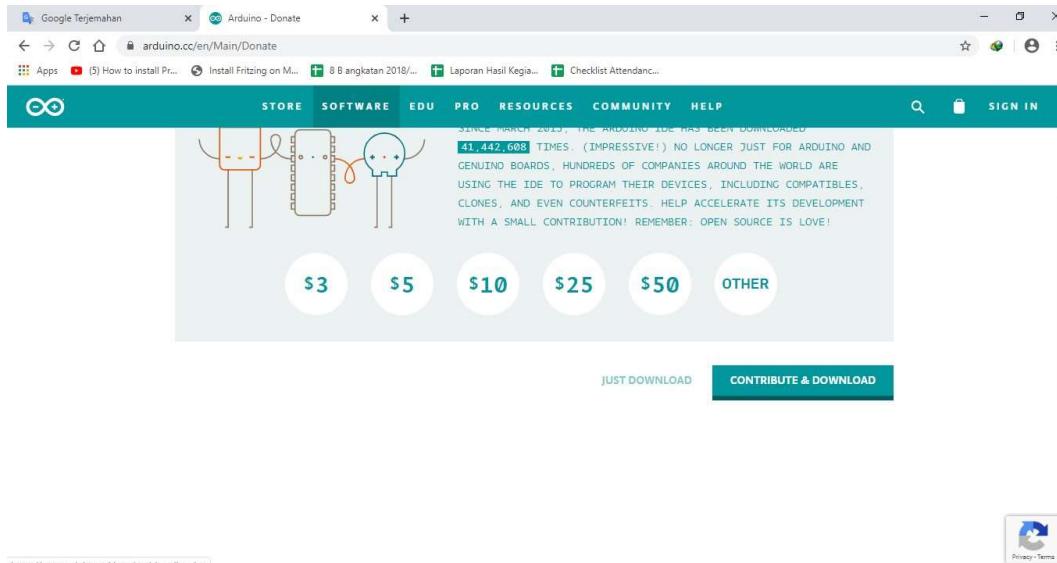
1. Pergi menuju halaman Website: <http://arduino.cc>
2. Kemudian klik Menu **Software** dan pilih **Download**



3. Lalu, setelah itu scroll ke bawah dan klik **Download** untuk dengan pilihan **MacOS X 10.10 or Newer**



4. Kemudian lewati halaman di bawah dan cukup melanjutkan dengan klik bagian **Just Download**



5. Selanjutnya, tunggu hingga *Download* Aplikasi Arduino IDE selesai.
6. Apabila kalian men-**download** dengan menggunakan **Safari Browser**, file akan diekstrak secara otomatis. Namun, apabila menggunakan **Browser lain**, maka kalian perlu melakukan **ekstrak file** karena file berbentuk **.zip**. Setelah **diekstrak**, kalian dapat melakukan **instalasi** seperti biasa dan tunggu hingga instalasi selesai. Lalu, **buka Aplikasi Arduino IDE** yang telah diinstall.

### - Cara Install Board NodeMCU pada Arduino IDE:

1. Pertama, buka Arduino IDE yang telah terinstall



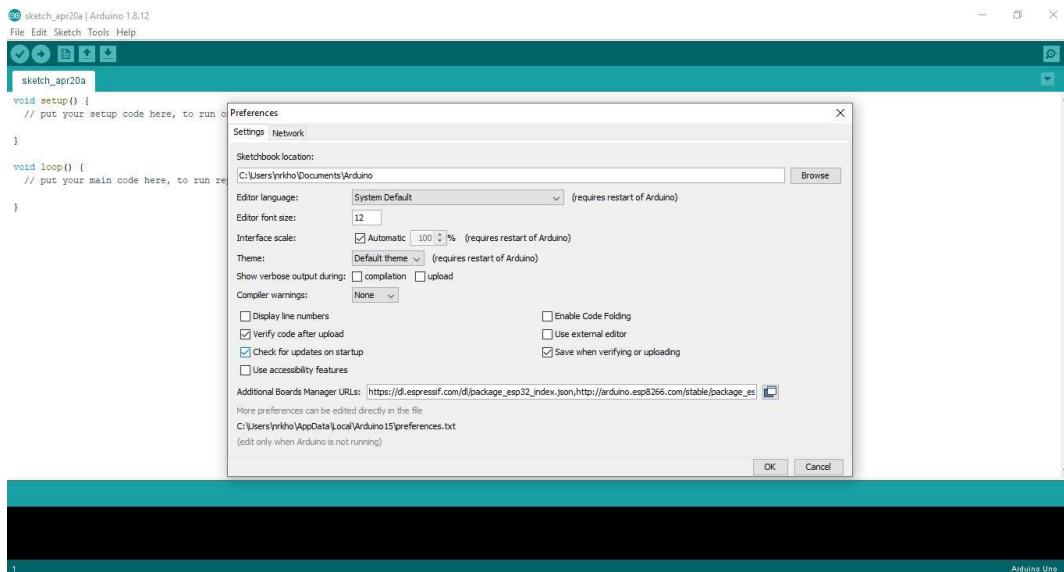
```
sketch_apr20a | Arduino 1.8.12
File Edit Sketch Tools Help
sketch_apr20a
void setup() {
  // put your setup code here, to run once
}

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

### 2. Lalu, klik pada bagian File -> Preferences

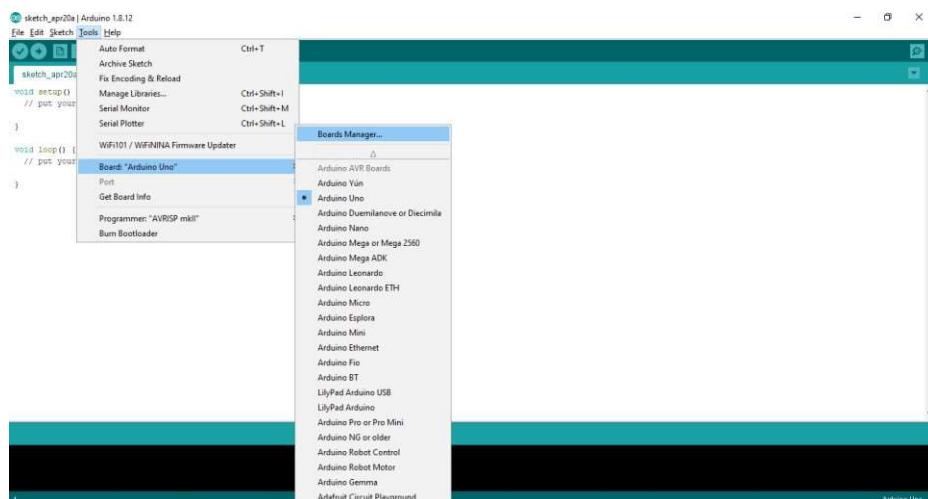


3. Lalu, pada bagian *Additional Board Manager URLs*, isikan dengan Link berikut ini: [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json), [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



#### 4. Lalu, klik **OK**

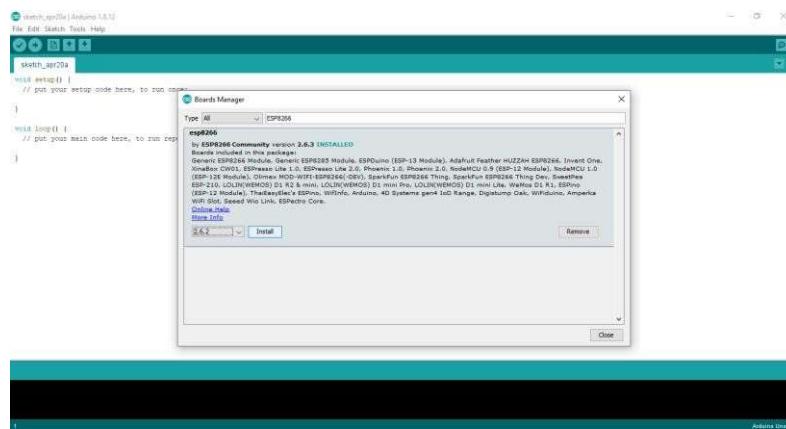
#### 5. Kemudian pilih Menu Tools -> Board -> Board Manager



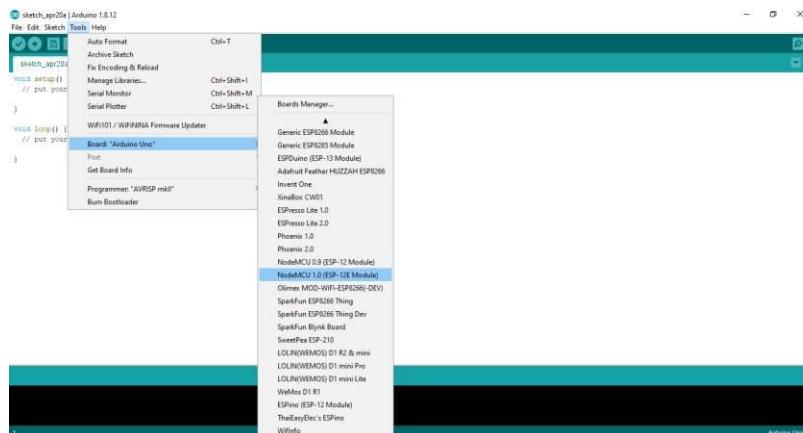
#### 6. Setelah itu, pada bagian Search, kalian ketikkan **ESP8266**. Kemudian pilih **versi terbaru** untuk di-*download*. Setelah itu, klik **Install**



## Pengenalan Instalasi Arduino IDE dan NodeMCU Library



7. Kemudian, tunggu hingga **download** selesai. Apabila download telah selesai kalian dapat melakukan *check* pada bagian **Tools -> Board**. Kemudian scroll hingga kalian menemukan **NodeMCU 1.0 (ESP-12E Module)**

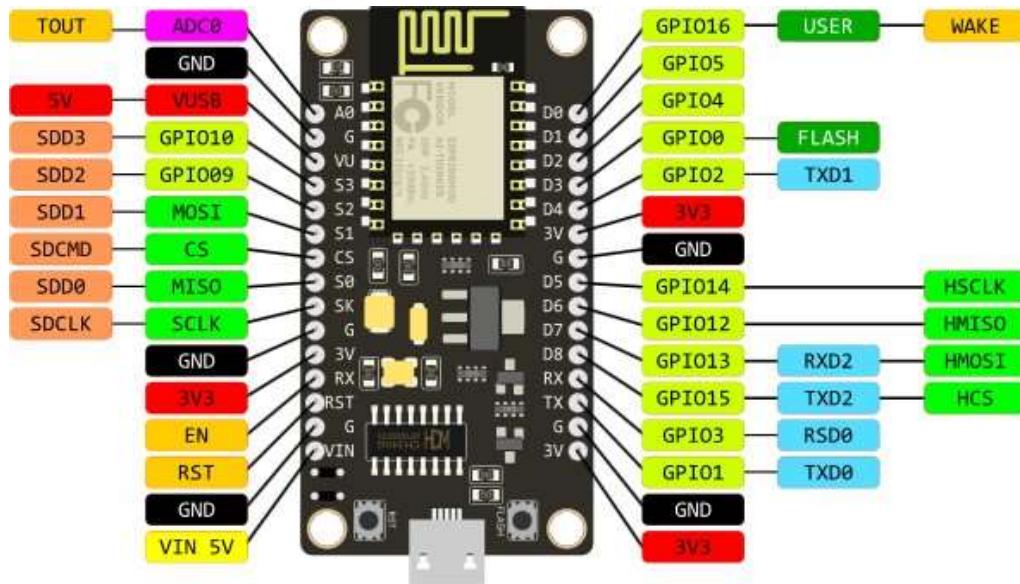




## A. NodeMCU LoLin V3

NodeMCU merupakan *board* atau sebuah mikrokontroller yang menggunakan *chip* ESP8266 dengan tipe ESP-12. Karena sudah terintegrasi dengan ESP8266 yang merupakan salah satu Modul Wi-Fi, maka NodeMCU dapat terhubung dengan Jaringan Wi-Fi untuk digunakan sebagai *Device* yang berperan dalam pengalikasian IoT (*Internet of Things*). Hal tersebutlah yang membedakan antara NodeMCU dengan Arduino Uno, Arduino Mega, dan Arduino Nano. NodeMCU dapat diprogram dengan beberapa cara, salah satunya adalah dengan menggunakan Arduino IDE dengan melakukan instalasi terlebih dahulu Board NodeMCU pada Arduino IDE.

NodeMCU menggunakan Tensilica 32-bit RISC CPU Xtensa LX106 sebagai *microprocessor* yang beroperasi dengan frekuensi antara 80 MHz sampai dengan 160 MHz dan bekerja pada tegangan 3.3 Volt. Namun, *Input Voltage* yang dapat diberikan adalah berkisar 7 – 12 Volt. NodeMCU Lolin V3 memiliki 16 Port Digital, 1 Port Analog, 1 Port untuk Komunikasi UART, SPI, dan I<sup>2</sup>C [1]. Berikut ini adalah gambar dari NodeMCU disertai dengan Detail *Pinout* yang dimiliki:

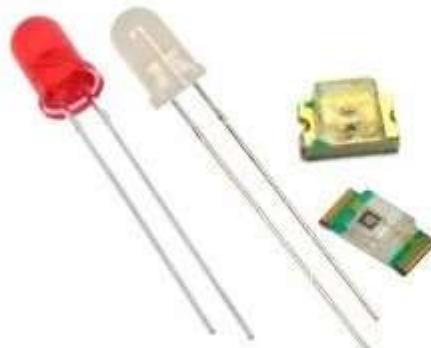
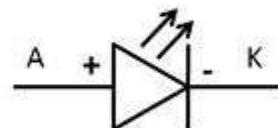


## B. LED

Light Emitting Diode atau sering disebut dengan LED merupakan komponen elektronika yang dapat memancarkan cahaya monokromatik ketika

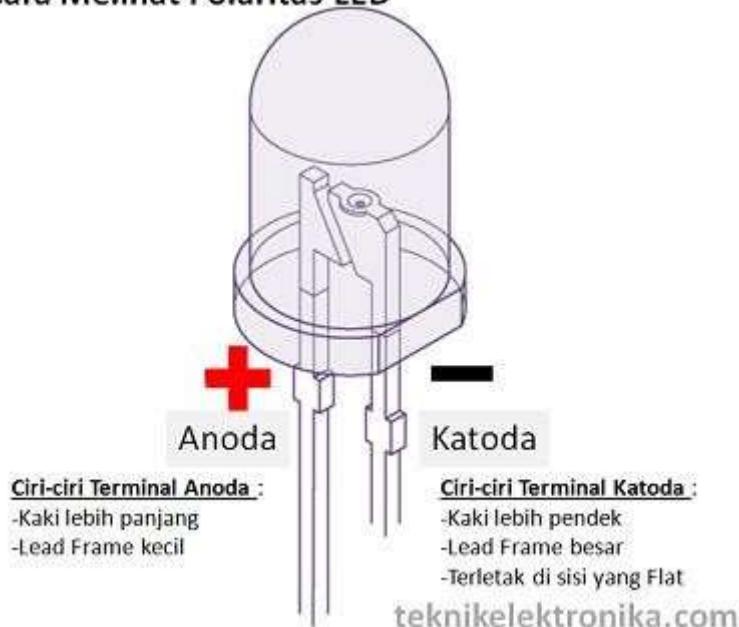


diberi tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor.

**Bentuk LED****Simbol LED**

[teknikelektronika.com](http://teknikelektronika.com)

Cara kerja dari LED adalah dengan mengalirkan tegangan maju dari Anoda ke Katoda. Setiap LED memiliki tegangan maju yang berbeda-beda sehingga diperlukan resistor untuk membatasi arus dan tegangan agar tidak merusak LED yang bersangkutan [2]. Cara mengetahui Polaritas LED dan Tegangan Maju untuk setiap LED dapat dilihat pada gambar dan Tabel di bawah ini

**Cara Melihat Polaritas LED**

| Warna | Tegangan |
|-------|----------|
| Merah | 1.8 V    |



|        |       |
|--------|-------|
| Kuning | 2.2 V |
| Hijau  | 3.5 V |
| biru   | 3.6 V |
| Putih  | 4.0 V |

### C. KOMUNIKASI SERIAL

Komunikasi Serial adalah komunikasi yang digunakan untuk memberikan informasi kepada User untuk berkomunikasi atau mengetahui status informasi dari program yang di-upload pada *board*. Komunikasi Serial yang digunakan oleh *board* adalah tipe CP2102 atau CH340G. Komunikasi Serial yang dilakukan diatur dengan menggunakan *baudrate* atau kecepatan aliran data. Biasanya *baudrate* yang digunakan adalah 9600 bps atau 115200 bps.

- **KOMPONEN** :

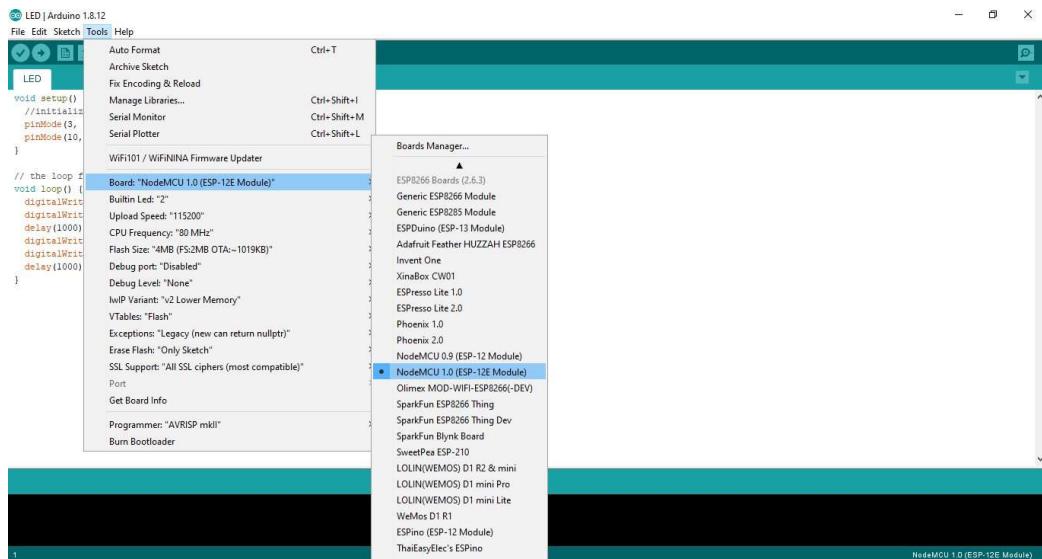
1. NodeMCU LoLIN V3
2. LED
3. Arduino IDE
4. Breadboard
5. Kabel Jumper

- **LANGKAH-LANGKAH**

#### A. BLINK LED

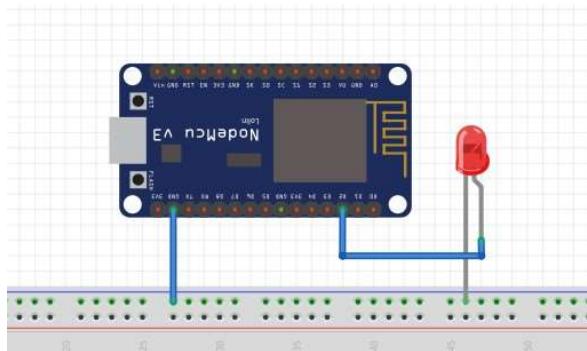
1. Buka **Arduino IDE**, lalu pilih Menu Tools -> Board. Kemudian, ubah Board menjadi **NodeMCU 1.0 (ESP-12E Module)**. Setelah itu, atur konfigurasi sebagai berikut:

- Builtin LED      2
- Upload Speed    115200
- CPU Frequency : 80 MHz
- Flash - Size     : 4MB (FS : 2 MB OTA : 1019KB)
- Sisanya biarkan sesuai dengan *default*



2. Kemudian, hubungkan LED dan Board NodeMCU dengan ketentuan sebagai berikut:

| LED          | NodeMCU |
|--------------|---------|
| Kaki Panjang | D2      |
| Kaki Pendek  | GND     |



3. Kemudian, tuliskan *source code* di bawah ini pada Arduino IDE

```
#define LED D2 //Definisikan Pin D2 NodeMCU sebagai LED

void setup() {
    Serial.begin(115200); //Baudrate Komunikasi Serial Monitor
    pinMode(LED, OUTPUT); //Set Mode untuk LED yaitu Output
}

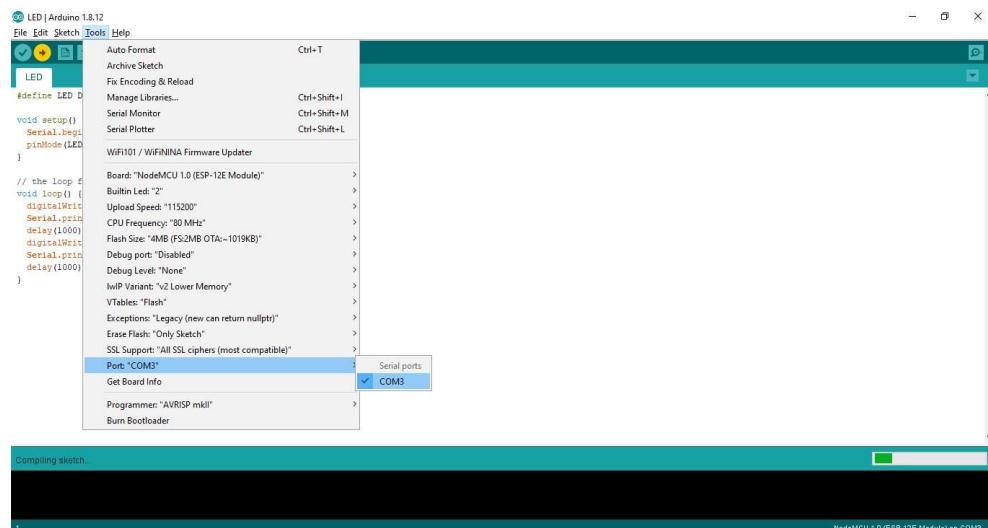
void loop() {
    digitalWrite(LED, LOW); //LED dalam keadaan Mati
    Serial.println("LED Mati"); //Notifikasi pada Serial Monitor
```



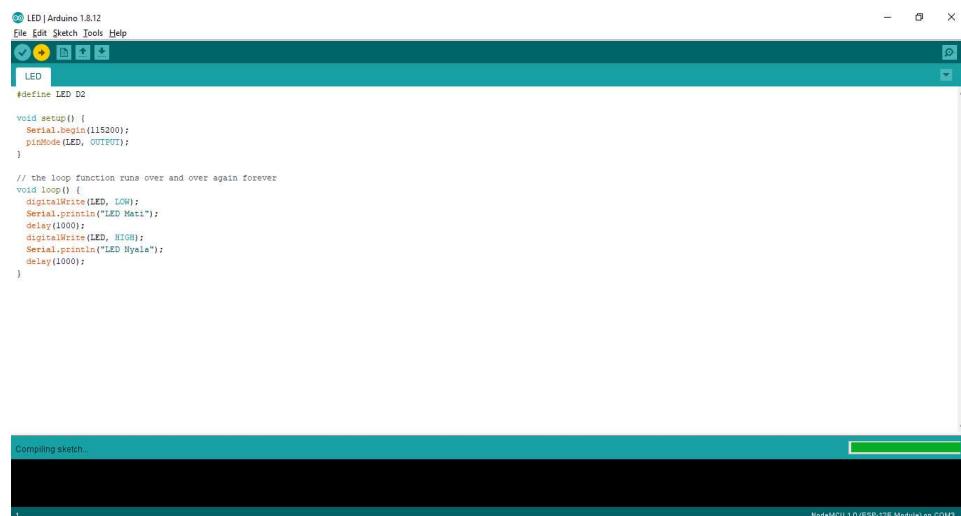
```
delay(1000);      //Delay 1 Detik
digitalWrite(LED, HIGH); //LED dalam keadaan Nyala
Serial.println("LED Nyala"); //Notifikasi pada Serial Monitor
delay(1000);      //Delay 1 Detik
}
```

4. Kemudian, hubungan Board NodeMCU dengan Laptop dengan menggunakan Kabel USB. Selanjutnya, pada pilih Menu **Tools -> Port**.

Pilih **Port** yang terbaca seperti gambar di bawah:



5. Kemudian, klik **Button Upload** yang berbentuk seperti tanda panah untuk melakukan **Compile** dan **Upload Program** ke Board NodeMCU. Setelah itu, amati hasil yang terjadi pada LED di Board NodeMCU

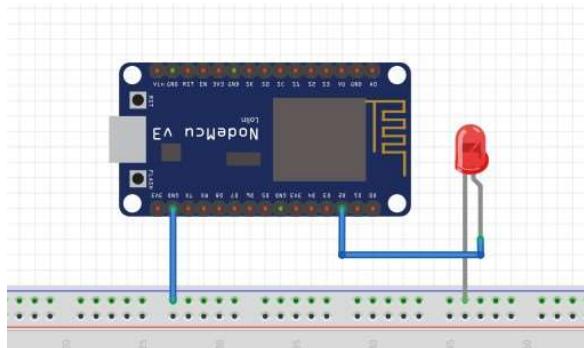




## B. TURN ON / OFF LED via Serial Monitor

1. Hubungkan LED dan Board NodeMCU seperti dengan ketentuan dibawah:

| LED          | NodeMCU |
|--------------|---------|
| Kaki Panjang | D2      |
| Kaki Pendek  | GND     |



2. Buka Arduino IDE, kemudian tuliskan *source code* di bawah ini dengan pada Arduino IDE

```
#define LED D2 //Define Pin D2 NodeMCU sebagai LED

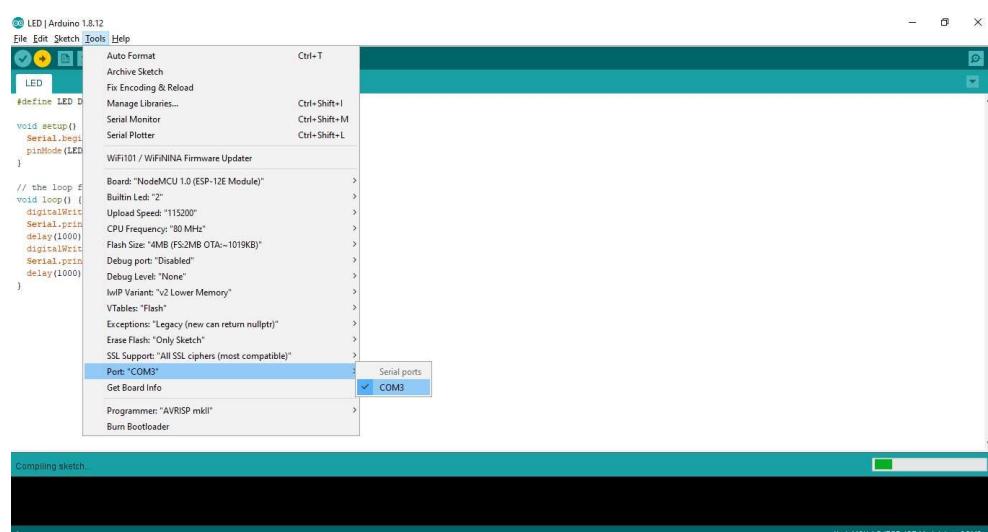
void setup() {
    pinMode(LED, OUTPUT); //Set Mode LED sebagai Output
    Serial.begin(115200); //Komunikasi Serial dengan Baudrate 115200
    Serial.println("Input 1 untuk Nyala dan 2 Untuk Mati");
    while(!Serial);
}

void loop() {
    if(Serial.available()) //Kondisi apabila Serial Monitor aktif
    {
        int Status = Serial.parseInt(); //Mengubah pesan yang masuk
        pada Serial Monitor ke bentuk Integer
        if (Status == 1) //Pengkondisian apabila Nilai yang masuk
        adalah 1 maka LED menyala
        {
            digitalWrite(LED, HIGH);
            Serial.println("LED Nyala");
        }
        else if (Status == 2) //Pengkondisian apabila Nilai yang masuk
        adalah 2 maka LED mati
        {
            digitalWrite(LED, LOW);
            Serial.println("LED Mati");
        }
    }
}
```

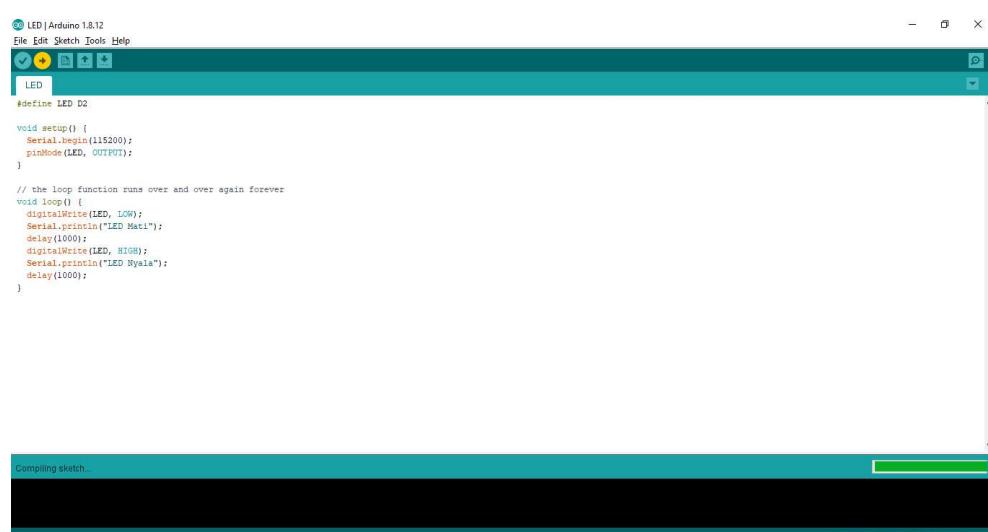


```
{  
    Serial.println("Error");  
}  
}  
}
```

3. Kemudian, hubungan Board NodeMCU dengan Laptop dengan menggunakan Kabel USB. Selanjutnya, pada pilih Menu **Tools -> Port**. Pilih Port yang terbaca seperti gambar di bawah:



4. Kemudian, klik **Button Upload** yang berbentuk seperti tanda panah untuk melakukan **Compile** dan **Upload Program** ke Board NodeMCU.



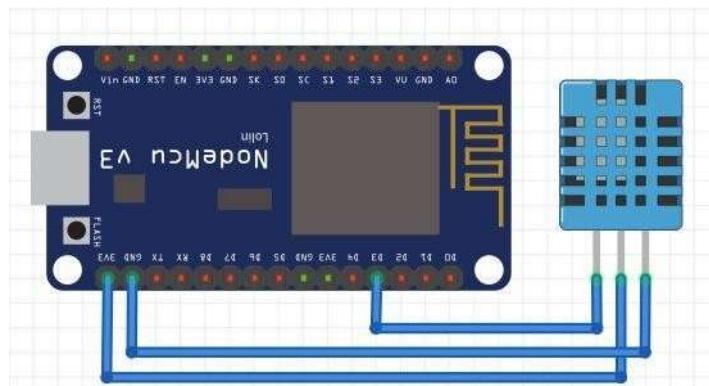
5. Setelah itu, amati hasil yang terjadi pada LED di Board NodeMCU dengan mengetik Angka 1 atau 2 pada Serial Monitor



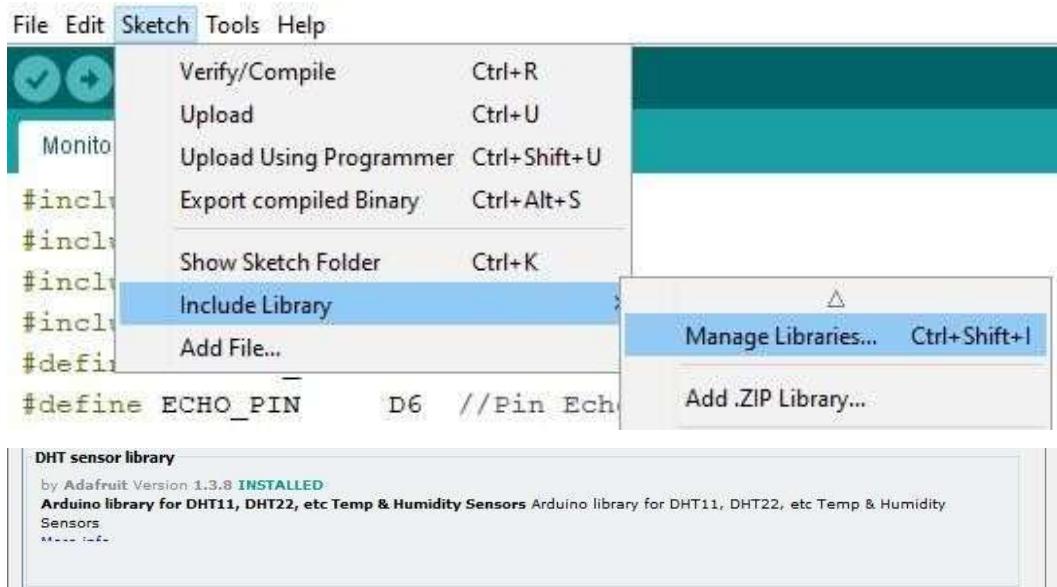
### C. Monitoring Suhu dengan DHT 11

1. Hubungkan DHT11 dan NodeMCU dengan ketentuan seperti berikut:

| Pin DHT11 | Pin NodeMCU |
|-----------|-------------|
| Data      | D3          |
| VCC       | 3.3 V       |
| GND       | GND         |



2. Kemudian buka Arduino IDE dan tambahkan Library untuk DHT dengan menggunakan cara seperti gambar di bawah:



3. Lalu, tuliskan *source code* di bawah pada Arduino IDE:

```
#include "DHT.h" //Library untuk DHT11
#define DHTTYPE DHT11 //Definisi Jenis DHT yang digunakan
#define DHTPIN D3 //Definisi Pin DHT11 pada NodeMCU
```



```
DHT dht(DHTPIN, DHTTYPE); //Aktifkan fungsi DHT pada Library

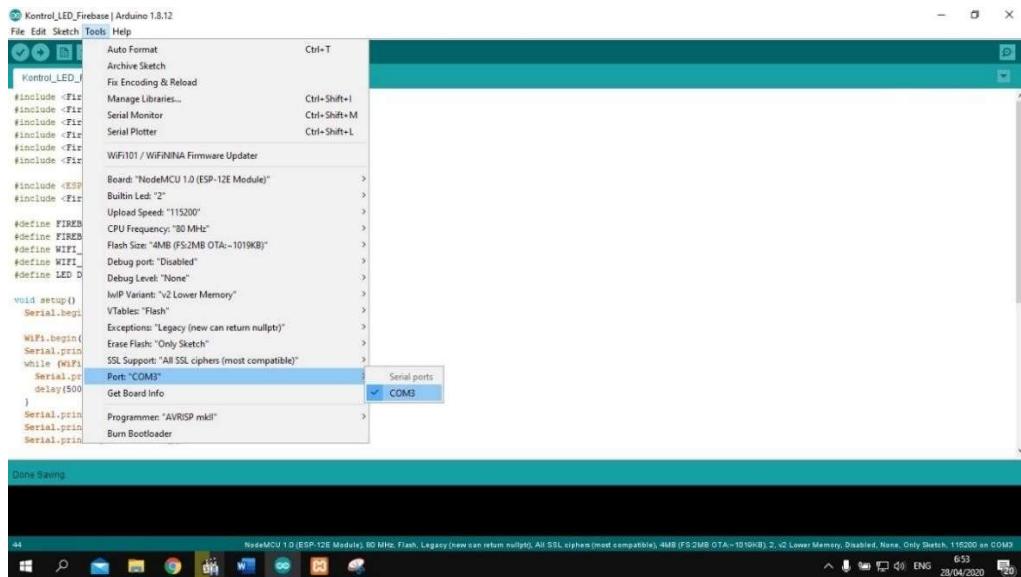
void setup() {
    Serial.begin(115200); //Memulai Komunikasi Serial
    Serial.println("Monitoring Suhu"); //Menampilkan Tulisan pada
    Serial Monitor
    dht.begin(); //Mulai menjalankan fungsi DHT
}

void loop() {
    float h = dht.readHumidity(); //Pembacaan Kelembaban dengan Tipe
    Data Float
    float t = dht.readTemperature(); //Pembacaan Suhu dengan Tipe
    Data Float

    //Kondisi yang muncul ketika Gagal mendapatkan Data dari Sensor
    if (isnan(h) || isnan(t)) {
        Serial.println("Gagal membaca Data dari Sensor");
        return;
    }

    //Mencetak Hasil Pembacaan Kelembaban dan Suhu pada Serial
    Monitor
    Serial.print("Kelembaban: ");
    Serial.print(h);
    Serial.println(" % ");
    Serial.print("Suhu");
    Serial.print(t);
    Serial.println(" °C ");
    Serial.println(" ");
    delay(1000); //Delay 1 Detik
}
```

4. Kemudian hubungkan Board NodeMCU dengan Laptop menggunakan Kabel USB. Selanjutnya, pilih Menu **Tools** -> **Port**. Lalu, pilih Port yang tersedia seperti gambar di bawah:





5. Kemudian, klik **Button Upload** yang berbentuk seperti tanda panah untuk melakukan **Compile** dan **Upload Program** ke Board NodeMCU

```
LED_WebServer | Arduino 1.8.12
File Edit Sketch Tools Help
LED_WebServer
Serial.print("IP: ");
Serial.println(WiFi.localIP());

webpage+= "<center><h1>Kontrol LED via Webserver</h1></center>";
webpage+= "<center><p>LED 1 : </p></center>";
webpage+= "<center><a href=\"/ON\"><button>ON</button></a><br><a href=\"/OFF\"><button>OFF</button></a></center>";

server.on("/", []() {
    server.send(200, "text/html", webpage);
});

server.on("/ON", []() {
    server.send(200, "text/html", webpage);
    digitalWrite(LED, HIGH);
    delay(1000);
});

server.on("/OFF", []() {
    server.send(200, "text/html", webpage);
    digitalWrite(LED, LOW);
    delay(1000);
});

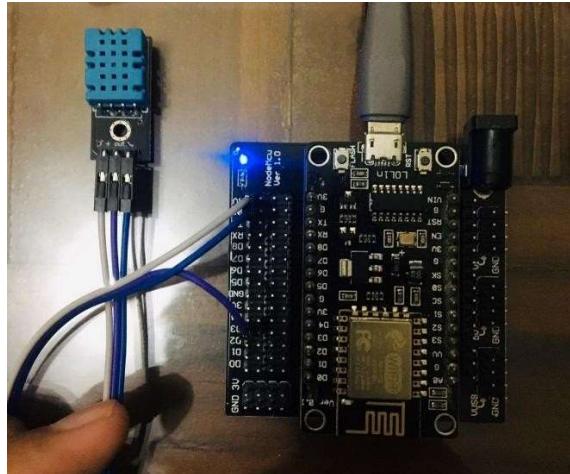
server.begin();
Serial.println("Webserver dijalankan");

}

void loop() {
    server.handleClient();
}
```

6. Setelah program selesai di-upload dan berhasil, maka buka **Serial Monitor** pada bagian kanan atas dan amati hasilnya

- **HASIL** :



```
COM3
|
Kelembaban: 84.00 %
Suhu = 32.20 °C

Kelembaban: 84.00 %
Suhu = 32.20 °C

Kelembaban: 84.00 %
Suhu = 32.20 °C

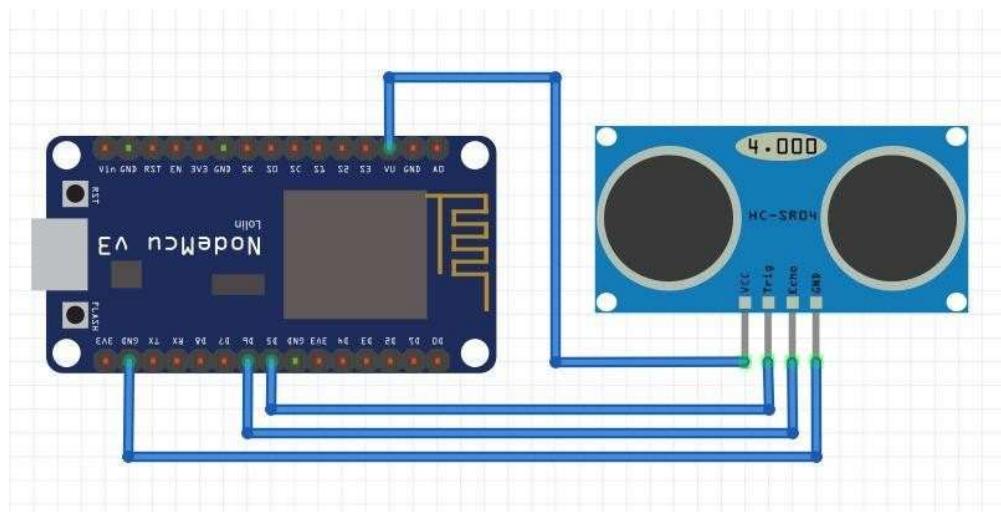
Kelembaban: 84.00 %
Suhu = 32.20 °C
```



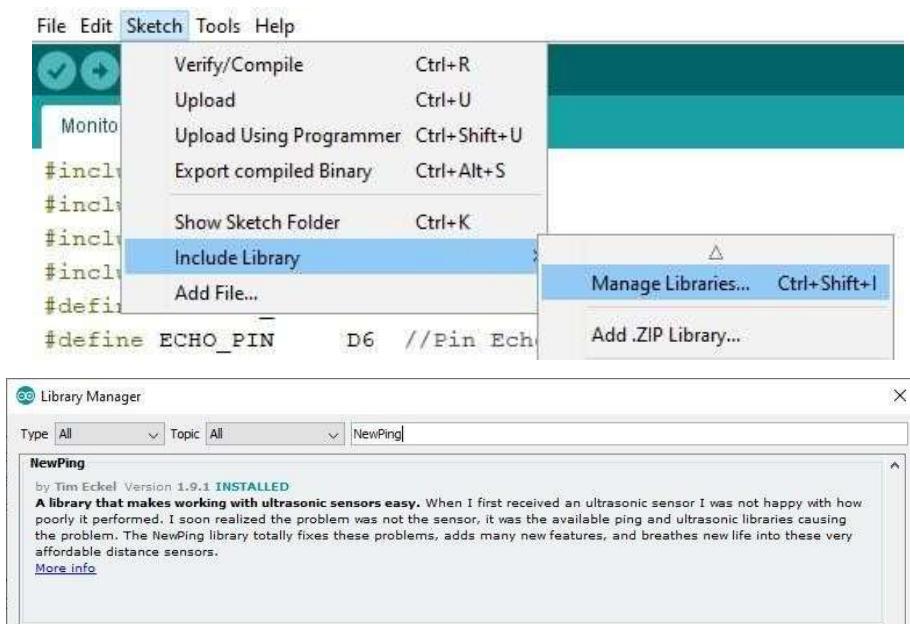
## D. Monitoring Jarak dengan HC-SR04

1. Hubungkan HC-SR04 dan NodeMCU dengan ketentuan seperti berikut:

| Pin HC-SR04 | Pin NodeMCU |
|-------------|-------------|
| Trigger     | D5          |
| Echo        | D6          |
| VCC         | 5 V         |
| GND         | GND         |



2. Kemudian buka Arduino IDE dan tambahkan Library untuk HC-SR04 dengan menggunakan cara seperti gambar di bawah:





### 3. Lalu, tuliskan *source code* di bawah pada Arduino IDE:

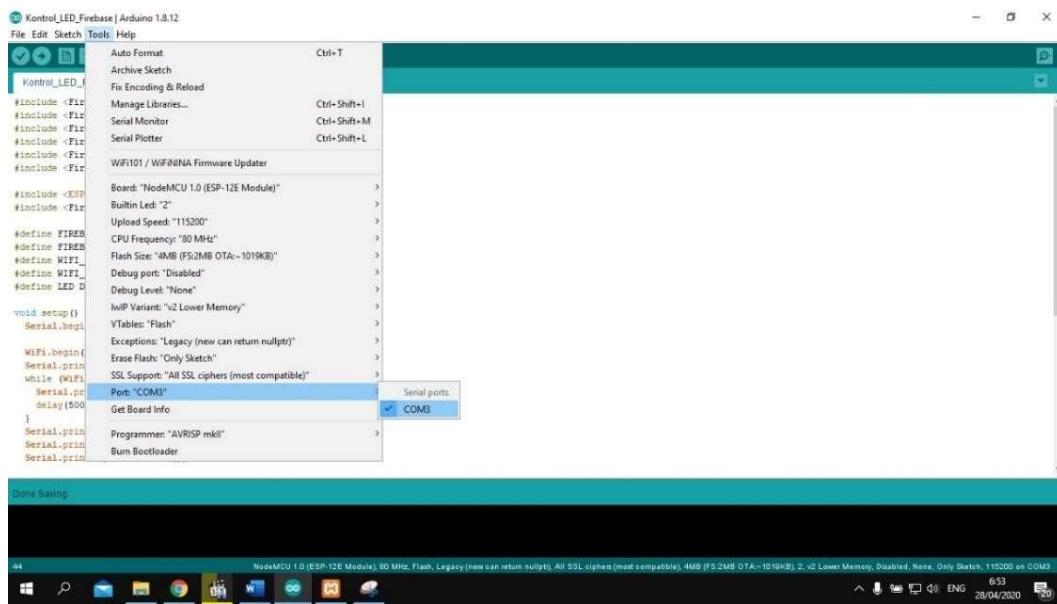
```
#include <NewPing.h> //Library untuk HC-SR04
#define TRIGGER_PIN D5 //Pin Trigger HC-SR04 pada NodeMCU
#define ECHO_PIN      D6 //Pin Echo HC-SR04 pada NodeMCU
#define MAX_DISTANCE 250 //Maksimum Pembacaan Jarak (cm)

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); //Setup Pin
HC-SR04 dan Jarak Pembacaan dengan Fungsi Library

void setup() {
    Serial.begin(115200); //Memulai Komunikasi Serial Monitor
}

void loop() {
    int jarak = sonar.ping_cm(); //Melakukan Pembacaan Jarak dalam
    bentuk cm
    //Mencetak Hasil Pembacaan pada Serial Monitor
    Serial.println("Monitoring Jarak");
    Serial.print("Jarak: ");
    Serial.print(jarak);
    Serial.println(" cm");
    delay(1000); //Delay 1 Detik
}
```

### 4. Kemudian hubungkan Board NodeMCU dengan Laptop menggunakan Kabel USB. Selanjutnya, pilih Menu **Tools -> Port**. Lalu, pilih Port yang tersedia seperti gambar di bawah:



### 5. Kemudian, klik Button **Upload** yang berbentuk seperti tanda panah untuk melakukan **Compile** dan **Upload Program** ke Board NodeMCU



```
LED_WebServer | Arduino 1.8.12
File Edit Sketch Tools Help
LED_WebServer $ 
Serial.print("IP: ");
Serial.println(WiFi.localIP());

webpage+= "<center><h1>Kontrol LED via Webserver</h1></center>";
webpage+= "<center><p>LED 1 : </p></center>";
webpage+= "<center><a href=\"ON\"><button>ON</button><br><a href=\"OFF\"><button>OFF</button></a></center>";

server.on("/", []() {
  server.send(200, "text/html", webpage);
});

server.on("/ON", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(LED, HIGH);
  delay(1000);
});

server.on("/OFF", []() {
  server.send(200, "text/html", webpage);
  digitalWrite(LED, LOW);
  delay(1000);
});

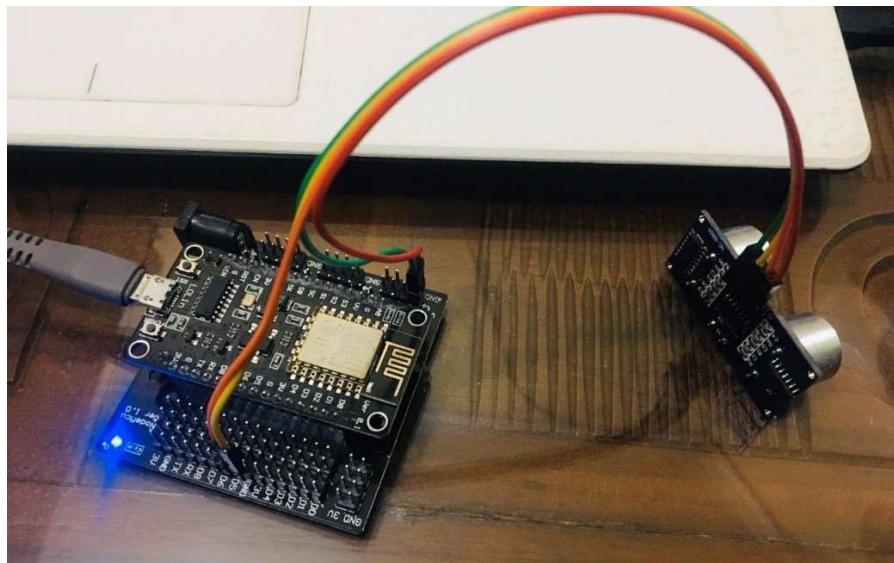
server.begin();
Serial.println("Webserver dijalankan");

void loop() {
  server.handleClient();
}

20
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Legacy (new can return nullptr), All SSL ciphers (most compatible), 4MB (FS:2MB OTA=1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3
```

6. Setelah program selesai di-*upload* dan **berhasil**, maka buka **Serial Monitor** pada bagian kanan atas dan amati hasilnya

- **HASIL** :



```
COM3
Monitoring Jarak
Jarak: 19 cm

Monitoring Jarak
Jarak: 17 cm

Monitoring Jarak
Jarak: 14 cm

Monitoring Jarak
Jarak: 16 cm
```



# PROJECT I

## SISTEM **MONITORING DAN CONTROLLING TANGKI AIR**

### MENGGUNAKAN NODEMCU DENGAN APLIKASI BERBASIS WEBSITE

---

- **DESKRIPSI** :

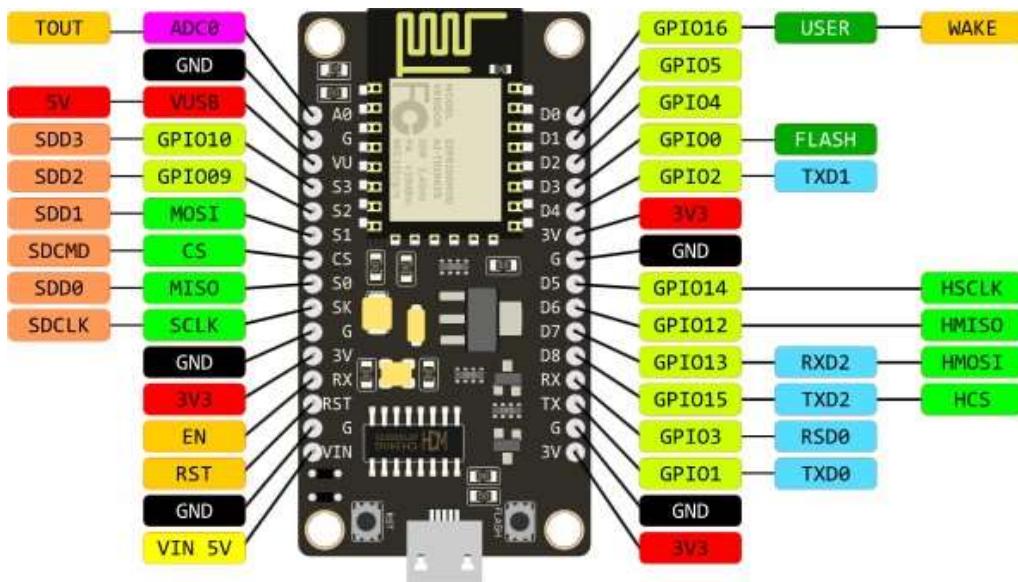
Sistem dapat melakukan *monitoring* dan *controlling* terhadap tangki air dan dapat dipantau dan dikontrol dengan menggunakan Aplikasi berbasis Website serta men-*download record* data pada Sistem.

- **BACAAN SINGKAT** :

**A. NodeMCU V3 LoLin**

NodeMCU merupakan *board* atau sebuah mikrokontroller yang menggunakan *chip* ESP8266 dengan tipe ESP-12. Karena sudah terintegrasi dengan ESP8266 yang merupakan salah satu Modul Wi-Fi, maka NodeMCU dapat terhubung dengan Jaringan Wi-Fi untuk digunakan sebagai *Device* yang berperan dalam pengalokasian IoT (*Internet of Things*). Hal tersebutlah yang membedakan antara NodeMCU dengan Arduino Uno, Arduino Mega, dan Arduino Nano. NodeMCU dapat diprogram dengan beberapa cara, salah satunya adalah dengan menggunakan Arduino IDE dengan melakukan instalasi terlebih dahulu Board NodeMCU pada Arduino IDE.

NodeMCU menggunakan Tensilica 32-bit RISC CPU Xtensa LX106 sebagai *microprocessor* yang beroperasi dengan frekuensi antara 80 MHz sampai dengan 160 MHz dan bekerja pada tegangan 3.3 Volt. Namun, *Input Voltage* yang dapat diberikan adalah berkisar 7 – 12 Volt. NodeMCU Lolin V3 memiliki 16 Port Digital, 1 Port Analog, 1 Port untuk Komunikasi UART, SPI, dan I<sup>2</sup>C. Berikut ini adalah gambar dari NodeMCU disertai dengan Detail *Pinout* yang dimiliki:

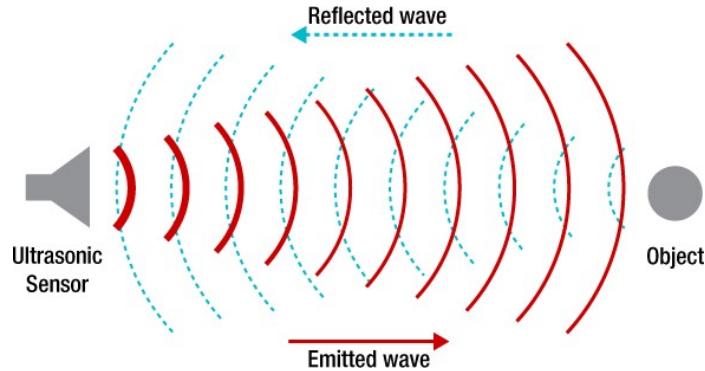


## B. Sensor HC-SR-04

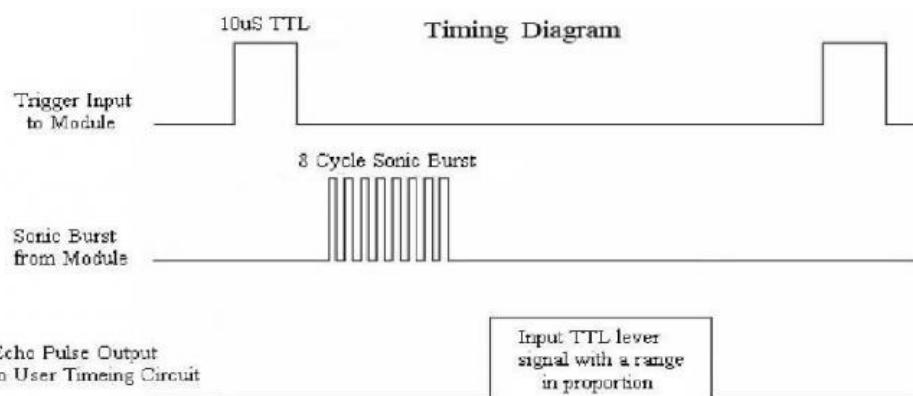


HC-SR04 adalah sensor non-kontak pengukur jarak yang menggunakan ultrasonik. Prinsip kerja sensor ini mirip dengan sistem ekolokasi pada kelelawan, dimana terdapat pemancar mengirimkan seberkas gelombang ultrasonik yang memiliki kecepatan  $v$ , lalu diukur waktu yang dibutuhkan hingga datangnya pantulan dari obyek ( $t$ ). Lamanya waktu ini sebanding dengan dua kali jarak sensor dengan obyek ( $R$ ), sehingga jarak sensor dengan obyek dapat ditentukan dalam persamaan:

$$R = \frac{v * t}{2} \text{ dengan } v = 340 \text{ m/s}$$



Sensor ini hanya memerlukan 2 pin I/O untuk berkomunikasi dengan mikrokontroler, yaitu **TRIGGER** dan **ECHO**. Untuk mengaktifkannya, mikrokontroler mengirimkan pulse positif melalui pin **TRIGGER** minimal 10 us, selanjutnya sensor akan mengirimkan 8 sinyal berfrekuensi 40kHz dan mendeteksi adakah sinyal yang kembali ke sensor karena dipantulkan oleh suatu objek. Jika terdapat sinyal yang kembali ke sensor maka akan terbaca oleh receiver dari sensor. Rentang waktu dari sinyal yang dikirim hingga diterima akan berbanding lurus dengan jarak dari objek yang memantulkan sinyal tersebut.



#### - Spesifikasi HC-SR04:

|                  |               |
|------------------|---------------|
| Tegangan Input   | 5 Volt        |
| Frekuensi        | 40 kHz        |
| Range Pembacaan  | 2 cm – 400 cm |
| Akurasi          | 3 mm          |
| Sudut Pengukuran | 15°           |
| Dimensi          | 45 20 * 15 mm |



### C. Relay

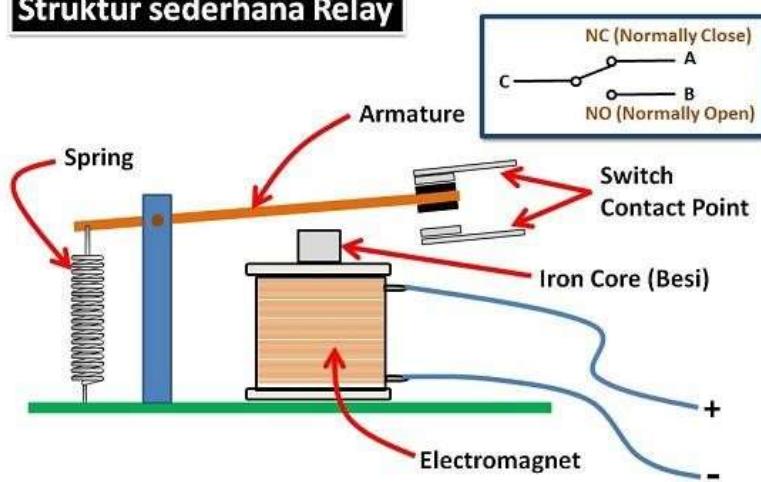


Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen Electromechanical (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat Kontak Saklar/Switch). Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan Relay yang menggunakan Elektromagnet 5V dan 50 mA mampu menggerakan Armature Relay (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A.

Pada dasarnya, Relay terdiri dari 4 komponen dasar yaitu :

1. Electromagnet (Coil)
2. Armature
3. Switch Contact Point (Saklar)
4. Spring

**Struktur sederhana Relay**





Kontak Poin (Contact Point) Relay terdiri dari 2 jenis yaitu :

- Normally Close (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi CLOSE (tertutup)
- Normally Open (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi OPEN (terbuka)

Berdasarkan gambar diatas, sebuah Besi (Iron Core) yang dililit oleh sebuah kumparan Coil yang berfungsi untuk mengendalikan Besi tersebut. Apabila Kumparan Coil diberikan arus listrik, maka akan timbul gaya Elektromagnet yang kemudian menarik Armature untuk berpindah dari Posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi Saklar yang dapat menghantarkan arus listrik di posisi barunya (NO). Posisi dimana Armature tersebut berada sebelumnya (NC) akan menjadi OPEN atau tidak terhubung. Pada saat tidak dialiri arus listrik, Armature akan kembali lagi ke posisi Awal (NC). Coil yang digunakan oleh Relay untuk menarik Contact Poin ke Posisi Close pada umumnya hanya membutuhkan arus listrik yang relatif kecil.

#### D. Database

Database merupakan sekumpulan data yang tersusun dan tersimpan rapi dalam computer, dan dapat diolah maupun dimanipulasi dengan menggunakan *software* atau perangkat lunak untuk dijadikan sebagai informasi. Database adalah kumpulan informasi atau data yang tersimpan secara sistematis sehingga temu kembali informasinya menjadi mudah dan cepat

Terdapat 12 tipe database, antara lain Operational database, Analytical database, Data warehouse, Distributed database, End-user database, External database, Hypermedia databases on the web, Navigational database, In-memory databases, Document-oriented databases, Real-time databases, dan Relational Database. Ada beberapa contoh dari *software Database* diantaranya adalah MySQL, PostgreSQL, Microsoft SQL Server, SQLite, Oracle, dan MongoDB.



## E. Aplikasi berbasis Web



Aplikasi berbasis Web adalah sebuah Aplikasi yang dapat diakses menggunakan *Web Brower* melalui jaringan internet maupun intranet baik menggunakan PC / Laptop dan juga *smartphone*.

Aplikasi berbasis Web biasanya dibangun dengan menggunakan HTML, CSS, Javascript untuk bagian *front-end* dan juga dapat menggunakan bahasa pemrograman PHP, Javascript, dan juga Python untuk bagian *back-end*. Namun, biasanya mayoritas Programmer Web menggunakan PHP untuk bagian *back-end*.

## F. XAMPP



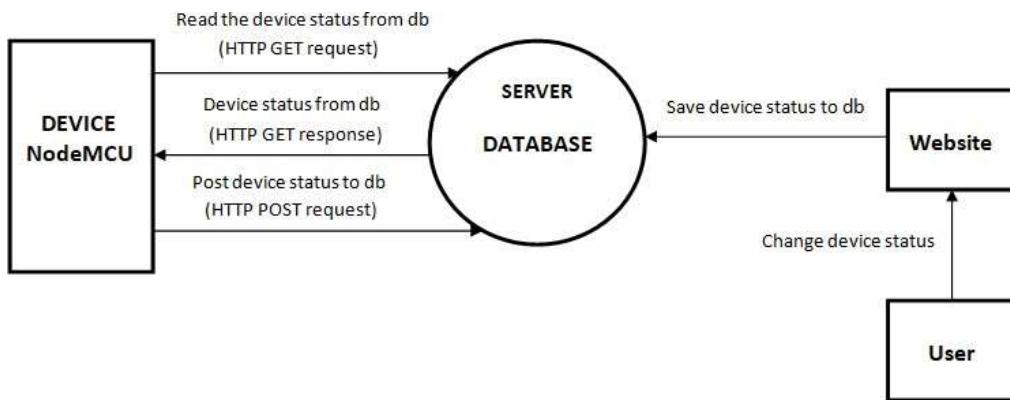
XAMPP adalah sebuah paket perangkat lunak (software) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf "X" yang terdapat pada awal kata berasal dari istilah cross platform sebagai simbol bahwa aplikasi ini bisa



dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris.

XAMPP berfungsi sebagai server lokal untuk mengampu berbagai jenis data website yang sedang dalam proses pengembangan. Dalam prakteknya, XAMPP bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam website kepada orang lain tanpa harus terkoneksi dengan internet, atau istilahnya website offline. XAMPP bekerja secara offline layaknya web hosting biasa namun tidak bisa diakses oleh banyak orang. Maka dari itu, XAMPP biasanya banyak digunakan oleh para mahasiswa maupun pelajar untuk melihat hasil desain website sebelum akhirnya dibuat online menggunakan web hosting yang biasa dijual dipasaran.

## G. Cara Kerja



Cara kerja dari Aplikasi ini adalah User mengirimkan perintah melalui Aplikasi berbasis Website yang kemudian perintah tersebut disimpan di dalam *database* yang dibuat. Lalu, perubahan yang terjadi pada *database* diproses oleh NodeMCU dengan menggunakan konsep HTTP (*request-response*) untuk menerima perubahan yang terjadi pada *database* untuk nantinya di-*output*-kan pada perangkat atau komponen yang terhubung sesuai dengan pemrograman yang diatur pada NodeMCU, seperti LED atau Aktuator.



- **KOMPONEN** :

1. NodeMCU LoLIN V3
2. Sensor HC-SR04
3. Relay 2 Channel
4. Arduino IDE
5. Breadboard
6. Kabel Jumper dan Kabel Data USB
7. XAMPP
8. Visual Studio Code
9. Aplikasi berbasis Website dengan Bahasa Pemrograman PHP Native

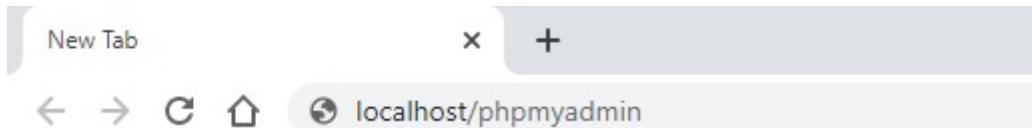
- **LANGKAH-LANGKAH**

**A. Konfigurasi Database**

1. Pastikan Laptop sudah terinstall XAMPP dan buka XAMPP
2. Kemudian Aktifkan Apache dan juga MySQL



3. Lalu, buka browser kalian dan ketikkan pada URL Bar:  
**localhost/phpMyAdmin**



4. Lalu, buatlah sebuah **database** dengan nama **tangki\_air**





5. Kemudian buatlah **Tabel** pada **database LED** dengan nama **tabel\_kontrol** dengan jumlah kolom : **2 Kolom** dan **tabel\_monitoring** dengan jumlah kolom : **3 kolom**

The screenshot shows the MySQL Workbench interface. At the top, there are tabs: Structure, SQL, Search, Query, Export, and Import. Below the tabs, a message says "No tables found in database." In the main area, there are two separate "Create table" dialog boxes. The first dialog is for the table "tabel\_kontrol" and specifies "Number of columns: 2". The second dialog is for the table "tabel\_monitoring" and specifies "Number of columns: 3".

6. Kemudian isi attribut / kolom dari **table\_kontrol** dan **tabel\_monitoing** dengan ketentuan seperti gambar di bawah

The screenshot shows the MySQL Workbench interface for managing table structures. The top part shows the structure for the table "tabel\_kontrol". It has two columns: "id" (Type: INT, Primary Key) and "CH\_1" (Type: INT). The bottom part shows the structure for the table "tabel\_monitoring". It has three columns: "id" (Type: INT, Primary Key), "waktu" (Type: TIMESTAMP, with CURRENT\_TIME as the default value), and "ketinggian" (Type: VARCHAR, with a length of 20).

7. Kemudian, *insert* data awal untuk **tabel\_kontrol** seperti gambar di bawah:



| Column | Type    | Function | Null | Value |
|--------|---------|----------|------|-------|
| id     | int(11) |          |      |       |
| CH_1   | int(11) |          |      | 0     |

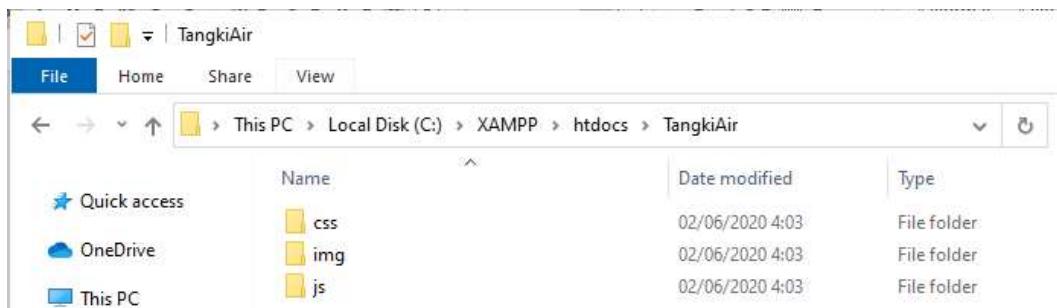
8. Konfigurasi pada *database* selesai dilakukan.

## B. Konfigurasi Aplikasi Berbasis Website

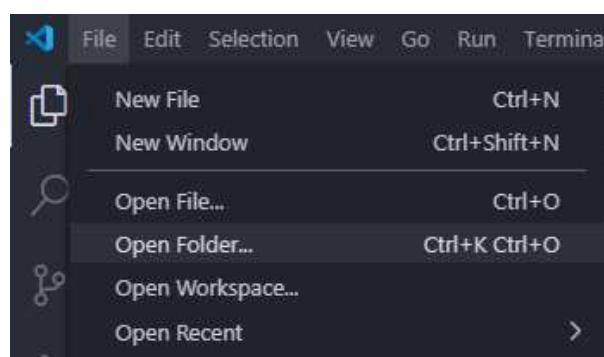
1. Pastikan Laptop sudah terinstall XAMPP dan buka XAMPP
2. Jalankan Apache dan juga MySQL

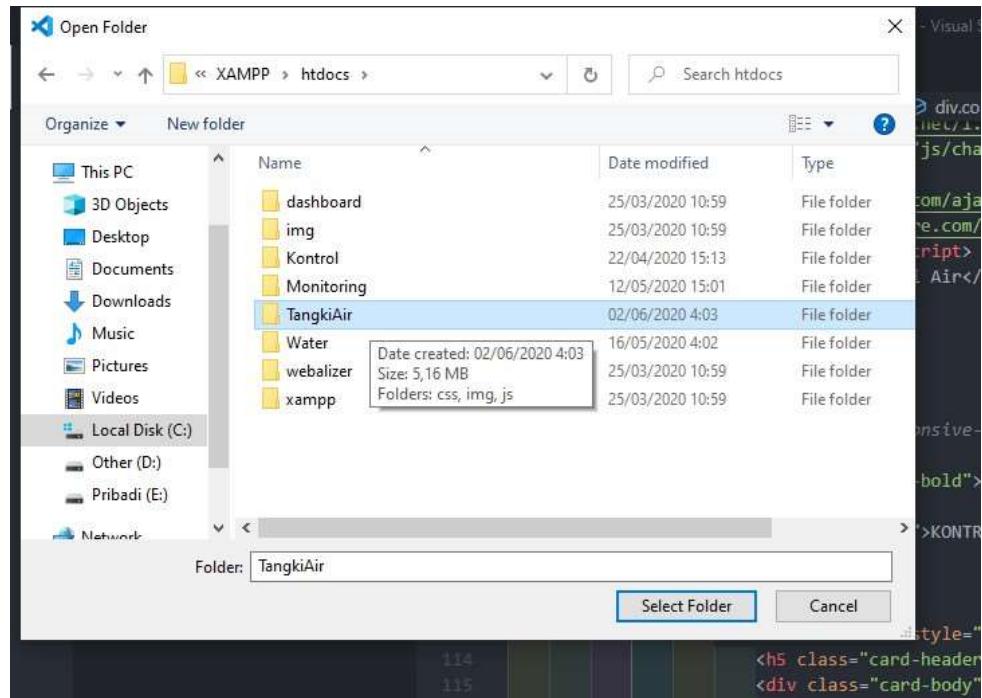


3. Lalu, pindahkan Folder **TangkiAir** yang berisikan beberapa *file* ke **Directory C -> XAMPP -> htdocs** atau lokasi dimana kalian menyimpan instalasi XAMPP

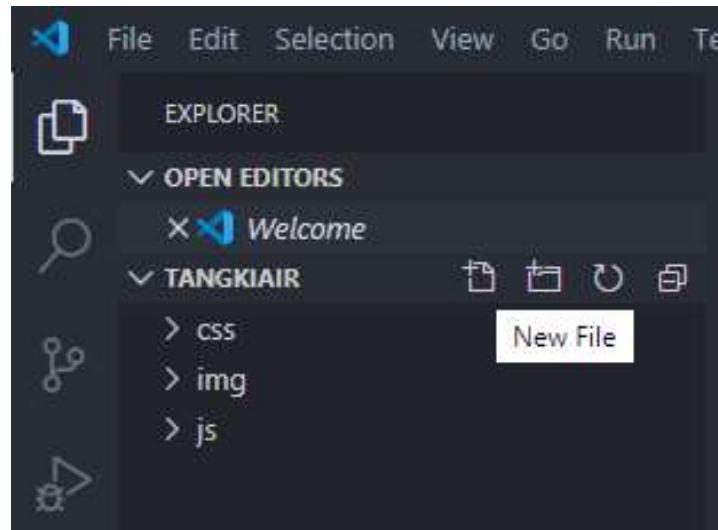


4. Lalu, selanjutnya kalian buka Aplikasi **Visual Studio Code** dan bukalah Folder **TangkiAir** pada Aplikasi tersebut dengan klik **File -> Open Folder** dan Cari Folder **TangkiAir** pada **htdocs** di **XAMPP** kalian





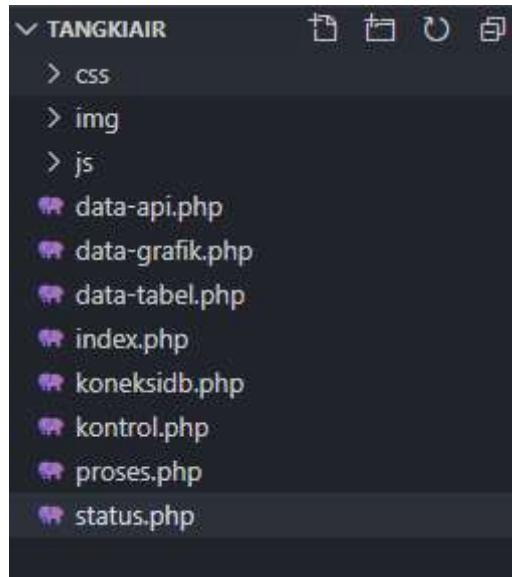
5. Kemudian buat beberapa File dengan menggunakan *shortcut New File* pada Folder TangkiAir seperti gambar dibawah



6. Buatlah beberapa File baru dengan nama-nama berikut **pada Folder TangkiAir:**
  - A. data-api.php
  - B. data-grafik.php
  - C. data-tabel.php
  - D. index.php



- E. koneksidb.php
- F. kontrol.php
- G. proses.php
- H. status.php



7. Kemudian *copy* dan *paste source code* berikut pada masing-masing File:

A. koneksidb.php

```
<?php

$server      = "localhost"; //lokasi server
$user        = "root"; //user
$password    = ""; //password
$database    = "tangki_air"; //Nama Database di phpMyAdmin

//Koneksi ke Database
$koneksi       = mysqli_connect($server, $user, $password,
$database);

function query($query) {
    global $koneksi;
    $result = mysqli_query($koneksi, $query );
    $box = [];
    while( $sql = mysqli_fetch_assoc($result) ){
        $box[] = $sql;
    }
    return $box;
}
?>
```



## B. data-api.php

```
<?php
require "koneksi.php"; //Import file koneksi.php
$nilai = $_GET["nilai"]; //Mendapatkan Nilai Jarak dari Sensor

// Kondisi untuk mengaktifkan Relay apabila jarak lebih dari 100cm
// dan mematikan Relay apabila jarak kurang dari 30 secara otomatis
if ($nilai >= 100) {
    $sql1 = "UPDATE tabel_kontrol SET CH_1 = '0'";
} else if ($nilai <= 30) {
    $sql1 = "UPDATE tabel_kontrol SET CH_1 = '1'";
}

//Menyimpan hasil pembacaan jarak ke database
$koneksi->query($sql1);
$sql      = "INSERT INTO tabel_monitoring(nilai) VALUES ('$nilai')";
$conneksi->query($sql);
$response = query("SELECT * FROM tabel_monitoring ORDER BY id DESC") [0];

//convert data menjadi JSON
$result = json_encode($response);
echo $result;
?>
```

## C. kontrol.php

```
<?php
//Import File koneksi.php
require "koneksi.php";

//Mengambil data pada table_monitoring
$data1 = query("SELECT * FROM tabel_monitoring ORDER BY id DESC") [0];
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title></title>
</head>

<body>
    <div class="row mt-5">
        <div class="col">
            <div class="card" style="max-width:25rem; height:165px">
                <h5 class="card-header bg-dark text-white font-weight-bold">Ketinggian (cm)</h5>
                <div class="card-body">
```



```
                <h3      class="font-weight-bold"      mt-3"><?=
$data1["nilai"]; ?></h3>
                </div>
            </div>
        <div class="col">
            <div class="card" style="max-width:25rem; height:
165px">
                <h5 class="card-header bg-dark text-white font-
weight-bold">Status Tangki</h5>
                <div class="card-body">
                    <?php if ($data1["nilai"] >= "100") { ?>
                    <h3 class="font-weight-bold" mt-3 text-
danger">HABIS</h3>
                    <?php } else { ?>
                    <h3 class="font-weight-bold" mt-3 text-
success">AMAN</h3>
                    <?php } ?>
                </div>
            </div>
        </div>
    </body>
</html>
```

#### D. proses.php

```
<?php

//Import File koneksidb.php
require "koneksidb.php";

//Mengambil data pada Tabel_Kontrol
$data = query("SELECT * FROM tabel_kontrol")[0];

//Melakukan Update Status pada Tabel_Monitoring
if (isset($_GET['channel']) && isset($_GET['state'])) {
    $channel = $_GET['channel'];
    $state   = $_GET['state'];
    if ($channel == 'CH_1') {
        $sql = "UPDATE tabel_kontrol SET CH_1 = '$state'";
    } else if ($channel == 'CH_2') {
        $sql = "UPDATE tabel_kontrol SET CH_2 = '$state'";
    }
    $koneksi->query($sql);
    header('Location:index.php');
}

//Output hasil dalam bentuk JSON
$result = json_encode($data);
echo $result;
?>
```



## E. status.php

```
<?php
//Import file koneksidb.php
require "koneksidb.php";

//Mengambil data pada Tabel_Kontrol dalam bentuk Array
$data = query("SELECT * FROM tabel_kontrol")[0];
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title></title>
</head>

<body>
    <div class="card" style="max-width:25rem; height: 165px">
        <h5 class="card-header bg-dark text-white font-weight-
bold">Status Pompa</h5>
        <div class="card-body">
            <?php if ($data["CH_1"] == 1) { ?>
                <h3 class="font-weight-bold" style="color: red; margin-top: 3px; margin-bottom: 0px">MATI</h3>
            <?php } else { ?>
                <h3 class="font-weight-bold" style="color: green; margin-top: 3px; margin-bottom: 0px">NYALA</h3>
            <?php } ?>
        </div>
    </div>
</body>
</html>
```

## F. data-grafik.php

```
<?php
//Import file koneksidb.php
require "koneksidb.php";

//Mengambil data waktu dan nilai pada Tabel_monitoring
$data3 = mysqli_query($koneksi, "SELECT TIME(waktu) as pukul, nilai
FROM (SELECT * FROM tabel_monitoring ORDER BY id DESC LIMIT 6) as
Test ORDER BY id ASC LIMIT 6");

//Membuat variable data array
$data_tanggal = array();
$data_jarak = array();

//Memasukkan data pada array
while ($data4 = mysqli_fetch_array($data3)) {
```



```
$data_tanggal[] = $data4['pukul']; // Memasukan tanggal ke dalam array
$data_jarak[] = $data4['nilai']; // Memasukan total ke dalam array
}
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title></title>
</head>

<body>
    <div class="row mt-2">
        <div class="col" style="width: 700px;">
            <hr>
            <h5 class="font-weight-bold">GRAFIK KETINGGIAN AIR</h5>
            <hr>
            <canvas id="myChart"></canvas>
        </div>
    </div>
    <script>
        var linechart = document.getElementById('myChart');
        var myChart = new Chart(linechart, {
            type: 'line',
            data: {
                labels: <?php echo json_encode($data_tanggal); ?>,
// Merubah data tanggal menjadi format JSON
                datasets: [
                    {
                        label: ' Data Ketinggian',
                        data: <?php echo json_encode($data_jarak); ?>,
                        borderColor: 'rgba(255,99,132,1)',
                        backgroundColor: 'transparent',
                        borderWidth: 2
                    }
                ],
                options: {
                    responsive: true,
                    scales: {
                        xAxes: [
                            ticks: {
                                display: true,
                                autoSkip: true,
                                maxTicksLimit: 6
                            }
                        ]
                    },
                    tooltips: {
                        mode: 'index'
                    }
                }
            }
        });
    </script>
</body>
```



```
        }
    });
</script>
</body>
</html>
```

#### G. data-tabel.php

```
<?php
//Import file koneksidb.php
require "koneksidb.php";
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title></title>
</head>

<body>
    <div class="row mt-4">
        <div class="col">
            <hr>
            <h5 class="font-weight-bold">TABEL KETINGGIAN AIR</h5>
            <hr>
            <table class="table text-center" id="myTable">
                <thead>
                    <tr>
                        <th>Tanggal</th>
                        <th>Waktu</th>
                        <th>Nilai</th>
                        <th>Status</th>
                    </tr>
                </thead>
                <tbody>
//Mengambil data dari Tabel_Monitoring dan menampilkan pada Tabel
                    <?php
                        $data2 = mysqli_query($koneksi, "SELECT
DATE(waktu) AS tanggal, TIME(waktu) AS pukul, nilai FROM
tabel_monitoring ORDER BY waktu DESC LIMIT 10");
                        while ($row = mysqli_fetch_array($data2)) {
                            if ($row["nilai"] >= "100") {
?
                            <tr>
                                <td class="table-danger font-weight-
bold"><?php echo $row['tanggal'] ?></td>
                                <td class="table-danger font-weight-
bold"><?php echo $row['pukul'] ?></td>
                                <td class="table-danger font-weight-
bold"><?php echo $row['nilai'] ?></td>
                            
```



```
<td      class="table-danger"      font-weight-
bold"><?php echo "HABIS" ?></td>
</tr>
<?php } else { ?>
<tr>
    <td      class="table-success"   font-weight-
bold"><?php echo $row['tanggal'] ?></td>
        <td      class="table-success"   font-weight-
bold"><?php echo $row['pukul'] ?></td>
            <td      class="table-success"   font-weight-
bold"><?php echo $row['nilai'] ?></td>
                <td      class="table-success"   font-weight-
bold"><?php echo "AMAN" ?></td>
                    </tr>
                    <?php }
                }
                ?>
            </tbody>
        </table>
    </div>
</div>
<script>
$(document).ready(function() {
    $('#myTable').DataTable({
        "searching": false,
        "ordering": false,
        "lengthChange": false,
        "scrollY": '50vh',
        "scrollCollapse": true,
        "paging": false,
        "info": false,
        "order": [
            [0, "desc"]
        ]
    });
});
</script>
</body>

</html>
```

## H. index.php

```
<?php
//Import file koneksidb.php
require "koneksidb.php";

//Untuk Download Report data ke CSV dari Database
$start_date_error = '';
$end_date_error = '';

if (isset($_POST["export"])) {
    if (empty($_POST["start_date"])) {
        $start_date_error = '<div  class="alert  alert-danger"
role="alert">
```



```
<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span>Masukkan Tanggal Awal!</span></button>
} else if (empty($_POST["end_date"])) {
    $end_date_error = '<div class="alert alert-danger" role="alert">
        <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span>Masukkan Tanggal Akhir!</span></button>
    } else {
        $file_name = 'Data.csv';
        header("Content-Description: File Transfer");
        header("Content-Disposition: attachment; filename=$file_name");
        header("Content-Type: application/csv;");

        $file = fopen('php://output', 'w');

        $header = array("tanggal", "waktu", "nilai");
        fputcsv($file, $header);

        $query =
            "SELECT DATE(waktu) AS tanggal, TIME(waktu) AS pukul, nilai FROM
            tabel_monitoring
            WHERE DATE(waktu) >= '" . $_POST["start_date"] . "'"
            AND DATE(waktu) <= '" . $_POST["end_date"] . "'"
            ORDER BY waktu DESC
        ";
        $statement = $connect->prepare($query);
        $statement->execute();
        $result = $statement->fetchAll();
        foreach ($result as $row) {
            $data = array();
            $data[] = $row["tanggal"];
            $data[] = $row["pukul"];
            $data[] = $row["nilai"];
            fputcsv($file, $data);
        }
        fclose($file);
        exit;
    }
}
?>

<!doctype html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
```



```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
<!-- Font Awesome -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css" />
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abTE1Pi6jizo" crossorigin="anonymous">
</script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAIfFrlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZObtgt" crossorigin="anonymous">
</script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k" crossorigin="anonymous">
</script>

<script src="http://code.jquery.com/jquery-3.3.1.min.js" integrity="sha256-" FgpCb/KJQ1LNfOu91ta32o/NMZxltwRo8QtmkMRdAu8=" crossorigin="anonymous"></script>
<link href="https://cdn.datatables.net/1.10.19/css/jquery.dataTables.min.css" rel="stylesheet" />
<script src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js"></script>
<script type="text/javascript" src="js/chart.bundle.min.js"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.6.4/css/bootstrap-datepicker.css" />
<script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.6.4/js/bootstrap-datepicker.js"></script>
```



```
<script src="js/datepickers.js"></script>
<title>Monitoring dan Kontrol Tangki Air</title>

</head>

<body class="bg-light">
    <center>
        <!--       -->
        <div class="container">
            <h2 class="mt-3 font-weight-bold">PANEL KONTROL DAN
MONITORING TANGKI AIR</h2>
            <hr>
            <h5 class="font-weight-bold">KONTROL DAN STATUS</h5>
            <hr>
            <div class="row mt-4">
                <div class="col ">
                    <div class="card" style="max-width:25rem;
height: 165px">
                        <h5 class="card-header bg-dark text-white
font-weight-bold">Kontrol Pompa</h5>
                        <div class="card-body">
                            <a
                            href="proses.php?channel=CH_1&state=0" class="btn btn-success btn-
lg mt-2">ON</a>
                            <a
                            href="proses.php?channel=CH_1&state=1" class="btn btn-danger btn-
lg mt-2">OFF</a>
                        </div>
                    </div>
                    <div class="col status-pompa">
                    </div>
                </div>
                <div class="load-monitor"></div>
                <div class="row mt-3">
                    <div class="col">
                        <hr>
                        <h5 class="font-weight-bold">EXPORT DATA KE
CSV</h5>
                        <hr>
                        <?php echo $start_date_error; ?>
                        <?php echo $end_date_error; ?>
                        <form method="post">
                            <div class="row input-daterange">
                                <div class="col">
                                    <input type="text"
name="start_date" class="form-control" placeholder="Tanggal Awal"
                                    readonly />
                                </div>
                                <div class="col">
                                    <input type="text" name="end_date"
class="form-control" placeholder="Tanggal Akhir"
                                    readonly />
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </center>
</body>
```



```
        </div>
        <div class="col mt-3">
            <input type="submit" name="export" value="Export Data" class="btn btn-info" />
        </div>
    </form>
</div>
<div class='load-grafik'></div>
<div class='load-tabel'></div>

</div>

<footer class="py-3">
    <div class="container">
        <p class="m-0 text-center">Copyright © 2020
SMP IDN Boarding School</p>
    </div>
</footer>
</center>

</div>

<script>
$(document).ready(function() {
    window.setTimeout(function() {
        $(".alert").fadeTo(500, 0).slideUp(500, function() {
            $(this).remove();
        });
    }, 4000);
    $('.input-daterange').datepicker({
        todayBtn: 'linked',
        format: "yyyy-mm-dd",
        autoclose: true
    });
    setInterval(function() {
        $('.status-pompa').load("status.php")
    }, 3000);
    setInterval(function() {
        $('.load-monitor').load("kontrol.php")
    }, 3000);
    setInterval(function() {
        $('.load-grafik').load("data-grafik.php")
    }, 3000);
    setInterval(function() {
        $('.load-tabel').load("data-tabel.php")
    }, 3000);
});
</script>

<!-- Sweet Alert -->
<script src="js/sweetalert2.all.min.js"></script>
</body>
</html>
```



8. Jika semua sudah dibuat, maka buka browser kalian dan ketikkan pada url bar: **localhost/tangkiair/index.php** dan tampilan akan seperti berikut:

**PANEL KONTROL DAN MONITORING TANGKI AIR**

KONTROL DAN STATUS

**Kontrol Pompa**  
  
**ON** **OFF**

**Status Pompa**  
  
**NYALA**

**Ketinggian (cm)**  
  
**13**

**Status Tangki**  
  
**AMAN**

EXPORT DATA KE CSV

**Export Data**

GRAFIK KETINGGIAN AIR

Data Ketinggian

TABEL KETINGGIAN AIR

| Tanggal    | Waktu    | Nilai | Status |
|------------|----------|-------|--------|
| 2020-06-01 | 08:47:09 | 13    | AMAN   |
| 2020-06-01 | 08:47:06 | 13    | AMAN   |
| 2020-06-01 | 08:47:03 | 13    | AMAN   |
| 2020-06-01 | 08:47:00 | 13    | AMAN   |
| 2020-06-01 | 08:46:57 | 13    | AMAN   |
| 2020-06-01 | 08:46:53 | 13    | AMAN   |
| 2020-06-01 | 08:46:50 | 13    | AMAN   |
| 2020-06-01 | 08:46:47 | 13    | AMAN   |
| 2020-06-01 | 08:46:44 | 13    | AMAN   |

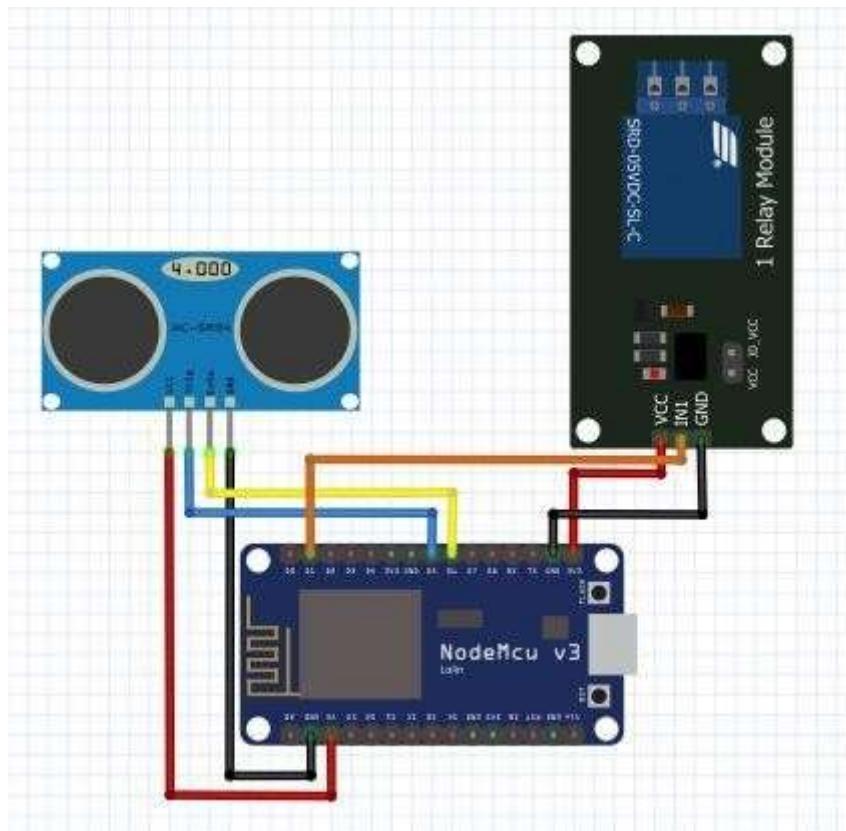
Copyright © 2020 SMP IDN Boarding School



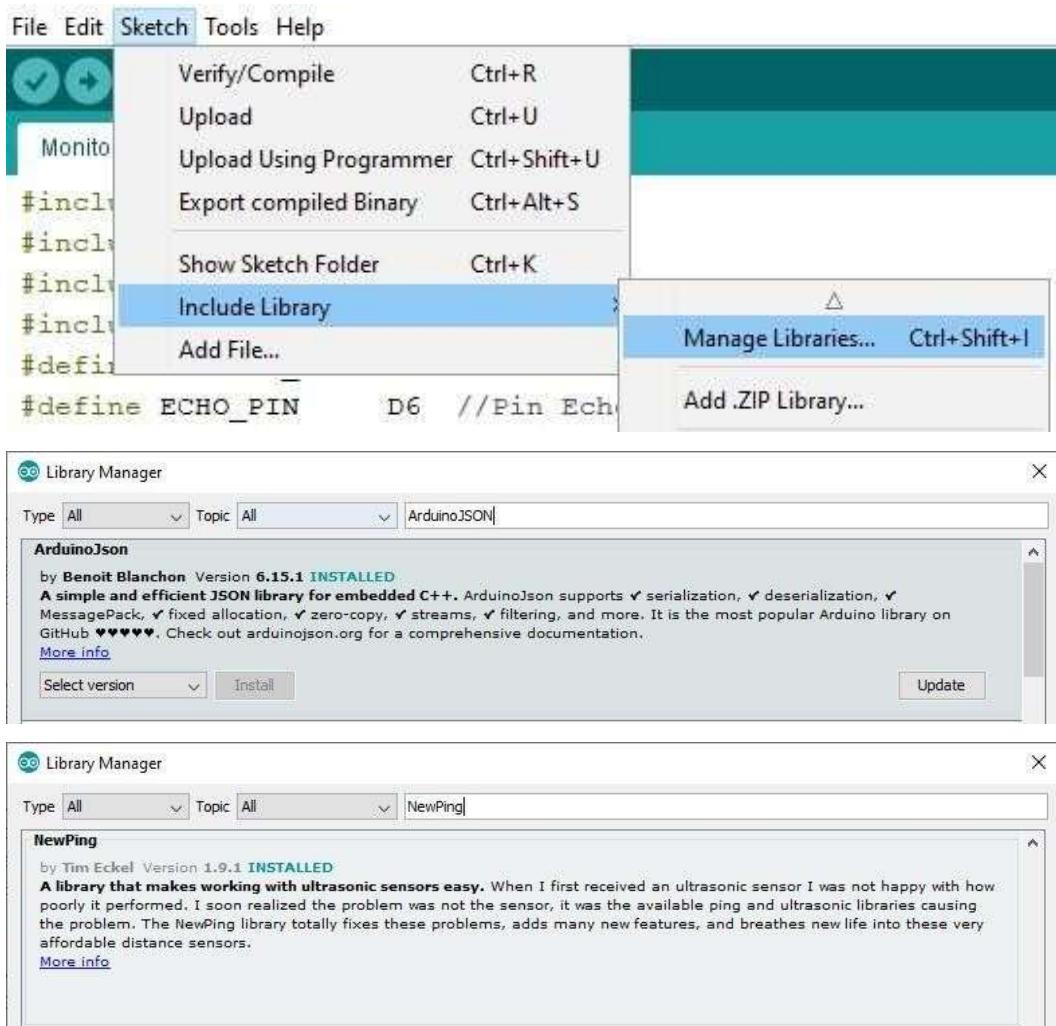
### C. Pemrograman pada NodeMCU

1. Hubungkan HC-SR04 dan Relay pada Board NodeMCU seperti ketentuan dibawah:

| Pin HC-SR04 | Pin NodeMCU |
|-------------|-------------|
| Trigger     | D5          |
| Echo        | D6          |
| VCC HC-SR04 | 5 V         |
| GND HC-SR04 | GND         |
| Pin Relay   | Pin NodeMCU |
| Pin IN1     | D1          |
| VCC Relay   | 3.3V        |
| GND Relay   | GND         |



2. Buka Arduino IDE, kemudian install Library yang diperlukan yaitu **NewPing** dan **ArduinoJSON** versi terbaru seperti gambar di bawah:



### 3. Kemudian, copy dan paste source code di bawah ke Arduino IDE

```
#include <ESP8266WiFi.h> //Library ESP terhubung ke WI-Fi
#include <ESP8266HTTPClient.h> //Library ESP HTTP Client
#include <ArduinoJson.h> //Library ArduinoJSON
#include <NewPing.h> //Library untuk HC-SR04
#define TRIGGER_PIN D5 //Pin Trigger HC-SR04 pada NodeMCU
#define ECHO_PIN D6 //Pin Echo HC-SR04 pada NodeMCU
#define MAX_DISTANCE 250 //Maksimum Pembacaan Jarak (cm)
#define pinCH_1 D1 //Pin untuk Relay

const char* ssid = "KILLUA"; //masukkan ssid
const char* password = "nada124@"; //masukkan password

int state;
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); //Setup Pin HC-SR04 dan Jarak Pembacaan dengan Fungsi Library
void setup () {
    Serial.begin(115200); //Komunikasi Serial
    pinMode(pinCH_1, OUTPUT); //Set Relay sebagai Output
    WiFi.begin(ssid, password); //Terhubung ke Wi-Fi
```



```
//Perulangan hingga NodeMCU terhubung ke Wi-Fi
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Menghubungkan");
}

//Kondisi yang keluar ketika NodeMCU mendapat respon dari Wi-Fi
if(WiFi.status() == WL_CONNECTED) {
    Serial.println("Terhubung");
}
else{
    Serial.println("Gagal Terhubung");
}
}

void loop() {
    int jarak = sonar.ping_cm(); //Melakukan Pembacaan Jarak dalam bentuk cm

    //Kondisi yang berjalan ketika Wi-Fi terhubung
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http; //Method http dengan menggunakan HTTPClient
        HTTPClient http1; //Method http1 dengan menggunakan HTTPClient
        http.begin("http://192.168.1.3/water/data-api.php?nilai=" +
String(jarak)); //Mengirim data pembacaan jarak ke Database
        http1.begin("http://192.168.1.3/water/proses.php");
        //Mengambil data kontrol dari database
        int httpCode = http.GET(); //Mendapatkan Pesan
        int httpCode1 = http1.GET(); //Mendapatkan Pesan

        //Kondisi yang berjalan ketika pesan ada dan didapatkan
        if (httpCode || httpCode1 > 0) {

            //Membuat Variabel JSON dan Mengambil Data dari Pesan yang masuk
            char json[500];
            String payload = http.getString();
            payload.toCharArray(json, 500);

            //Membuat Variabel JSON dan Mengambil Data dari Pesan yang masuk

            char json1[500];
            String payload1 = http1.getString();
            payload1.toCharArray(json1, 500);

            //StaticJsonDocument<200> doc;
            DynamicJsonDocument doc(JSON_OBJECT_SIZE(5));
            DynamicJsonDocument doc1(JSON_OBJECT_SIZE(5));

            // Deserialize the JSON document
            deserializeJson(doc, json);
            deserializeJson(doc1, json1);

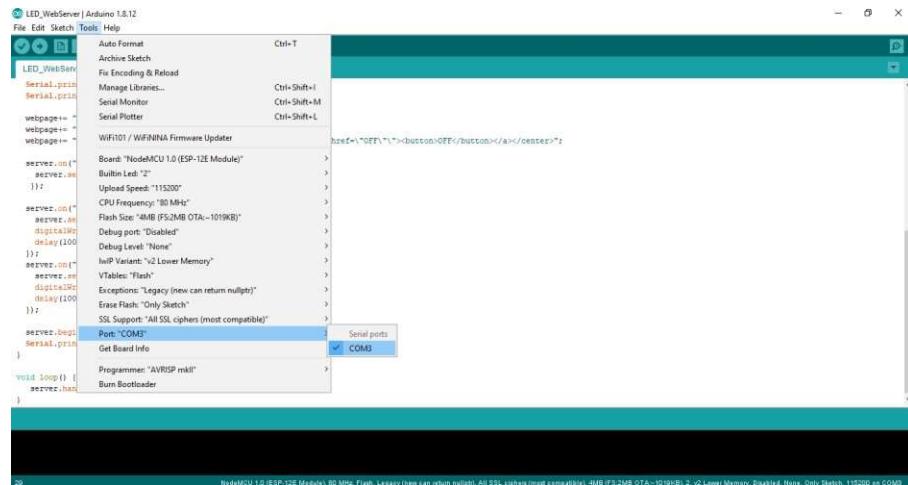
            //Mengambil Nilai dari Data Pesan yang diubah ke JSON
            String Jarak1 = doc["nilai"];
            int id = doc1["id"];
            int CH_1 = doc1["CH_1"];
```



```
//Menampilkan data pada Serial Monitor
Serial.println("Monitoring Ketinggian Air");
Serial.print("Jarak : ");
Serial.println(Jarak1);
Serial.print("ID = ");
Serial.println(id);
Serial.print("Channel 1 = ");
Serial.println(CH_1);
Serial.println(" ");

//Kondisi untuk menyalakan relay berdasarkan nilai dari pembacaan
//sensor dan kontrol
if(30 < Jarak1.toInt() < 100 && CH_1 == 0){
    digitalWrite(pinCH_1, LOW);
    state = digitalRead(pinCH_1);
}
else if(30 < Jarak1.toInt() < 100 && CH_1 == 1){
    digitalWrite(pinCH_1, HIGH);
    state = digitalRead(pinCH_1);
}
if(CH_1 == 0){
    digitalWrite(pinCH_1, LOW);
}
else{
    digitalWrite(pinCH_1, HIGH);
}
http.end();
http1.end();
}
delay(3000);
}
```

4. Kemudian hubungkan Board NodeMCU dengan Laptop menggunakan Kabel USB. Selanjutnya, pilih Menu **Tools -> Port**. Lalu, pilih Port yang tersedia seperti gambar di bawah:





5. Kemudian, klik **Button Upload** yang berbentuk seperti tanda panah untuk melakukan **Compile** dan **Upload Program** ke Board NodeMCU.



6. Setelah program selesai di-*upload* dan berhasil, maka buka **Serial Monitor** dan lihat hasil yang ditampilkan pada **Serial Monitor**:



7. Hasil yang ditampilkan pada Serial Monitor adalah sebagai berikut:

```
Monitoring Ketinggian Air
Jarak : 13
ID = 1
Channel 1 = 1

Monitoring Ketinggian Air
Jarak : 13
ID = 1
Channel 1 = 1
```



## PROJECT II

### NOTIFIKASI SMART WATER TANK LEVEL DENGAN TELEGRAM

#### - DESKRIPSI :

Project ini merupakan *project* lanjutan dari *project* sebelumnya, yaitu Sistem *Monitoring* dan *Controlling* Tangkir Air dengan Aplikasi berbasis Website. Project ini akan mengirimkan informasi singkat dari Tangkir Air ke *Smartphone User / Admin* dengan menggunakan Aplikasi Telegram.

#### - BACAAN SINGKAT :

##### A. *Internet of Things*

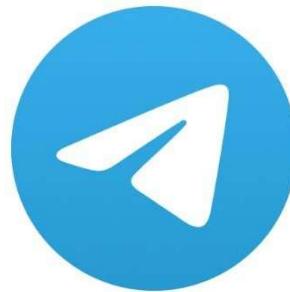
*Internet of Things (IoT)* merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari koneksi internet yang tersambung secara terus-menerus [3]. *Internet of Things* digunakan untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara *independent*. Salah satu contohnya adalah *smarthome* yang dapat memantau dan mengontrol kondisi rumah melalui *smartphone* dengan bantuan koneksi internet.

Pada dasarnya, perangkat *Internet of Things* terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komunikasi dan *server* sebagai pengumpul informasi yang diterima dari *sensor* dan dianalisa. Gambar di bawah menunjukkan gambaran cara kerja *Internet of Things* yang terdiri dari perangkat fisik, jaringan internet, dan *cloud database / server*.



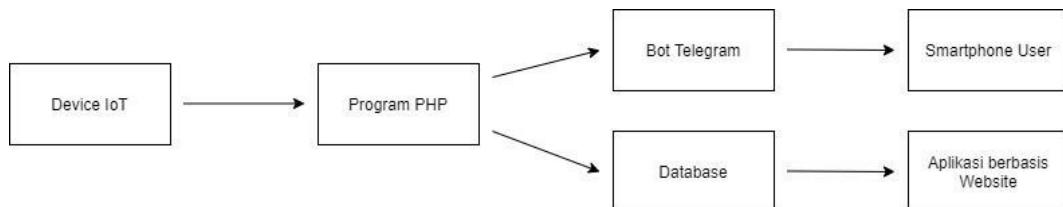


## B. Telegram



Telegram adalah sebuah aplikasi layanan pengirim pesan yang memiliki berbagai fitur unik di dalamnya. Salah satu contohnya adalah Fitur Bot dimana *User* dapat membuat sebuah Bot untuk melakukan berbagai fungsi atau tujuan, salah satunya adalah untuk Aplikasi *Internet of Things*. Bot pada Telegram dapat menjadi sumber informasi dari *Device IoT* bagi *User* serta memberikan kemudahan bagi *User* untuk melakukan *monitoring* dan *controlling* terhadap *Device IoT*.

## C. Cara Kerja



Cara kerja dari *Project* ini adalah *Device IoT* yang terdiri dari *Microcontroller*, *Sensor* dan juga *Actuator* mengirimkan informasi ke *User* melalui *Program PHP* yang sudah dibuat untuk mengirimkan informasi singkat ke Bot Telegram dan juga *Database* sebelum dikirimkan ke *User*, baik melalui *Smartphone* atau Aplikasi berbasis Website.

### - KOMPONEN :

1. XAMPP
2. Visual Studio Code
3. Aplikasi Telegram



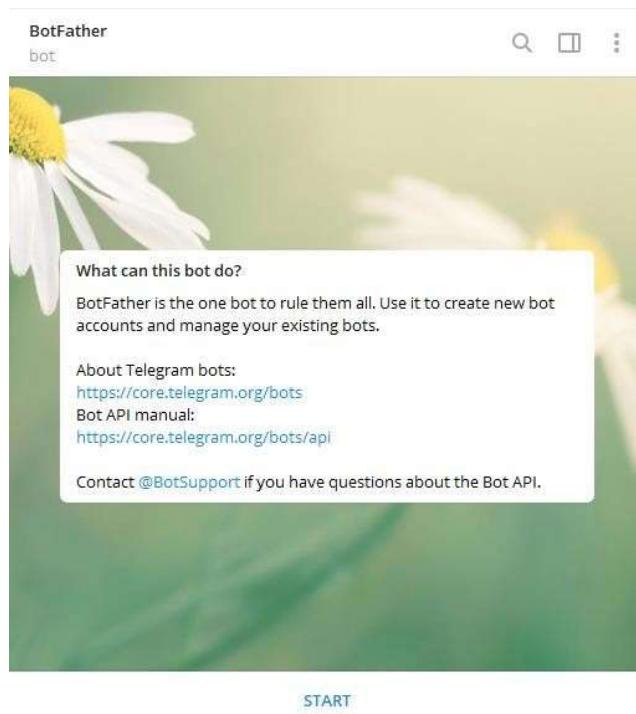
## - LANGKAH-LANGKAH

### A. Konfigurasi *Bot Telegram*

1. Pastikan *Smartphone* kalian sudah terinstall *Telegram* dan kalian memiliki Akun *Telegram*
2. Buka Aplikasi *Telegram*, lalu ketikkan **BotFather** di kolom Pencarian



3. Setelah itu, buka **BotFather** dan klik Start atau /start pada Kolom Chat.



4. Kemudian, ketik /newbot dan berikan **nama bot & username bot**.  
Jika sudah, maka akan muncul pesan yang menyertakan Token dari Bot *Telegram* yang telah kita buat seperti gambar di bawah



The screenshot shows a Telegram chat window with the title "BotFather" and subtitle "bot". The message history is as follows:

- /newbot 13:03 ✓
- Alright, a new bot. How are we going to call it? Please choose a name for your bot. 13:03
- Tangki Air 13:03 ✓
- Good. Now let's choose a username for your bot. It must end in 'bot'. Like this, for example: TetrisBot or tetris\_bot. 13:03
- TangkirAir2\_Bot 13:03 ✓
- Done! Congratulations on your new bot. You will find it at [t.me/TangkirAir2\\_Bot](https://t.me/TangkirAir2_Bot). You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.
- Token Bot
- Use this token to access the HTTP API:  
1054478438:AAEvWZ7F55\_ZmpLdEpL [REDACTED]
- Keep your token secure and store it safely, it can be used by anyone to control your bot.
- For a description of the Bot API, see this page:  
<https://core.telegram.org/bots/api> 13:03

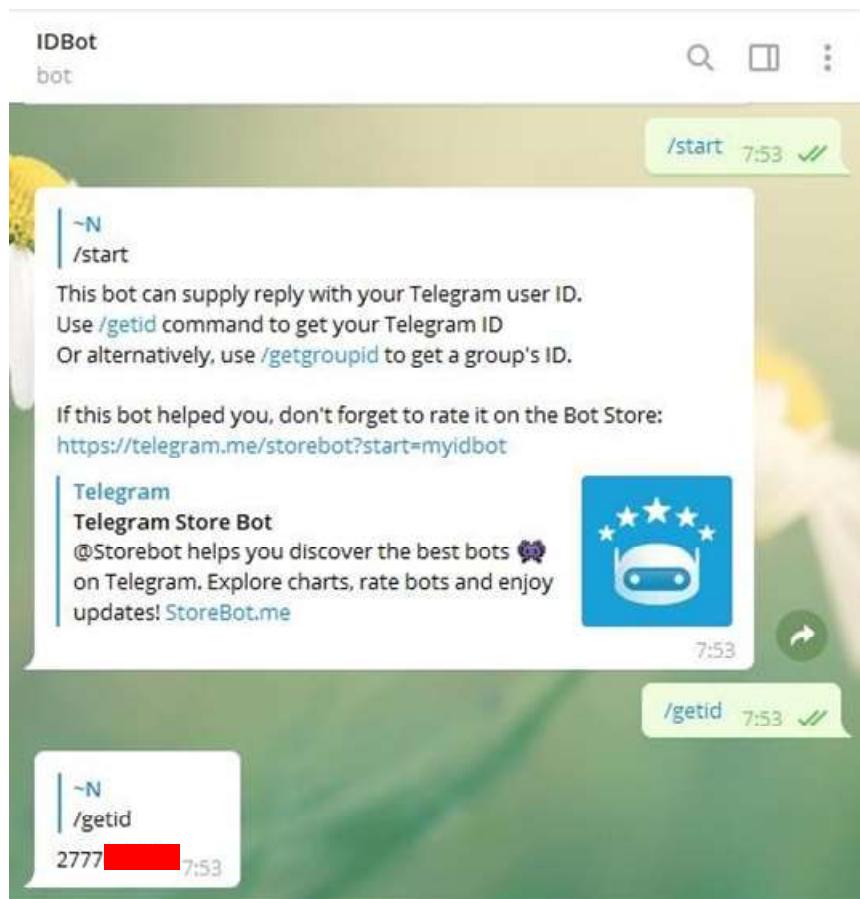
At the bottom, there is a message input field with placeholder "Write a message..." and icons for a paperclip, a search bar, a smiley face, and a microphone.

5. Kemudian, cari ID Telegram kalian dengan menggunakan **IDBot**

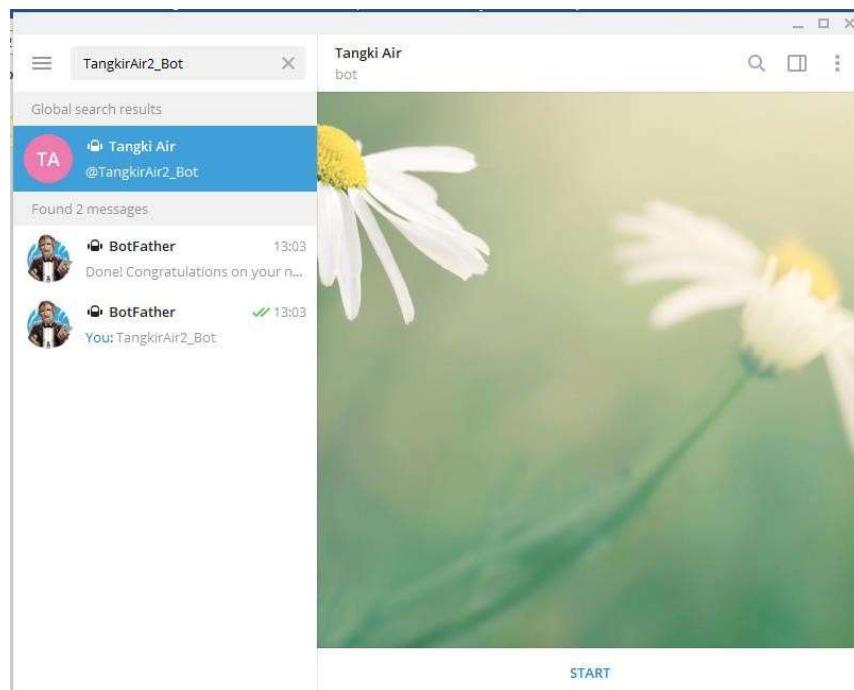
The screenshot shows a Telegram chat window with the title "IDBot". The message history is as follows:

- IDBOT
- IDBot 13:07  
This bot can supply reply wit... ➡

6. Lalu, klik Start dan ketikkan /getid seperti gambar di bawah



7. Selanjutnya, kita aktifkan bot yang sudah kita buat dengan mencari **Username Bot** yang kita buat dan kemudian ketik **/start**



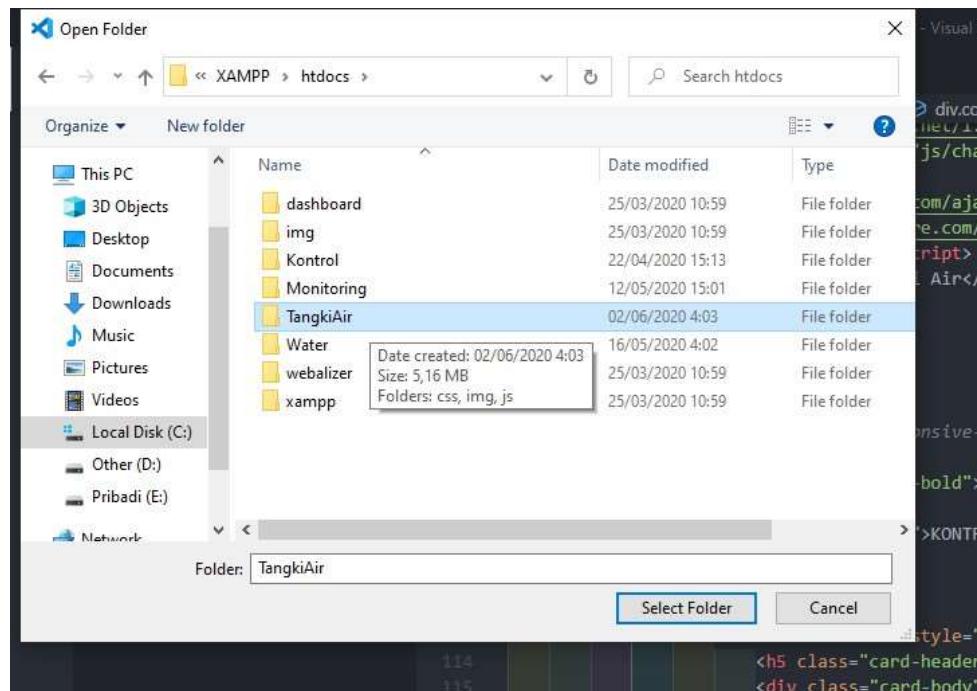
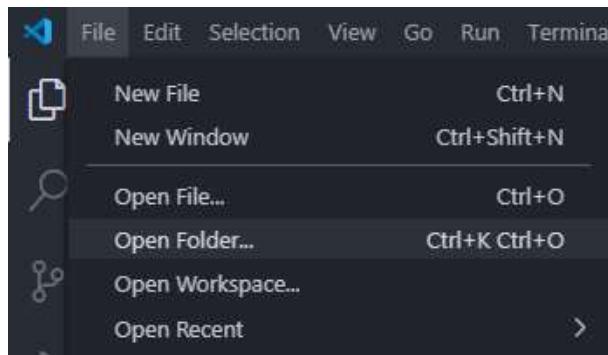


## B. Konfigurasi Bot Telegram pada Program PHP

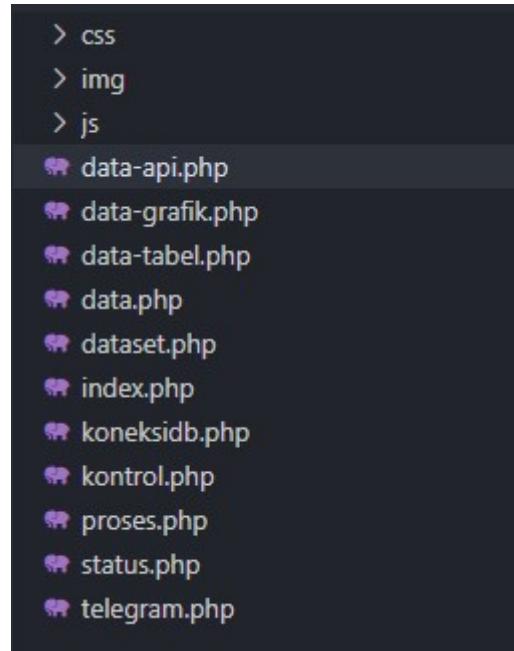
1. Pastikan Laptop sudah terinstall XAMPP dan buka XAMPP
2. Jalankan Apache dan juga MySQL



3. Lalu, buka Folder Aplikasi **TangkiAir** dari *Project* sebelumnya dengan menggunakan **Visual Studio Code**



4. Kemudian buka file **data-api.php** dari *List File* yang ada dalam Folder TangkiAir



##### 5. Selanjutnya *copy* dan *paste source code* ini pada file **data-api.php**

```
<?php
require "koneksidb.php";
$nilai = $_GET["nilai"];

define('BOT_TOKEN', '1235911992:AAGJC-9xPplpkp2lXcN3bodBG1U-DfCC4iI'); //Token Bot Telegram
define('CHAT_ID', '277760250'); //ID Telegram Kalian

//Fungsi untuk mengirimkan notifikasi ke Telegram User
function kirimTelegram($pesan)
{
    $pesan = json_encode($pesan);
    $API = "https://api.telegram.org/bot" . BOT_TOKEN . "/sendmessage?chat_id=" . CHAT_ID . "&text=$pesan";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'POST');
    curl_setopt($ch, CURLOPT_URL, $API);
    $result = curl_exec($ch);
    curl_close($ch);
    return $result;
}

//Pengkondisionan untuk mengubah status Relay dan Menjalankan Fungsi Notifikasi ke Telegram
if ($nilai >= 100) {
    $sql1 = "UPDATE tabel_kontrol SET CH_1 = '0'";
    $koneksi->query($sql1);
    $a = "Status Tangki Air Kosong ";
    $a .= "dan Pompa Menyala";
    kirimTelegram($a);
} else if ($nilai <= 30) {
```



```
$sql1 = "UPDATE tabel_kontrol SET CH_1 = '1'";
$koneksi->query($sql1);
$a = "Status Tangki Penuh ";
$a .= "dan Pompa Mati";
kirimTelegram($a);
}

$sql      = "INSERT INTO tabel_monitoring(nilai) VALUES
('{$nilai}')";
$koneksi->query($sql);
$response = query("SELECT * FROM tabel_monitoring ORDER BY id
DESC") [0];

$result = json_encode($response);
echo $result;
```

6. Kemudian **save** dengan mengklik Ctrl + S pada *keyboard* kalian
7. Kemudian, coba kalian aktifkan *Device IoT* dari *Project* sebelumnya. Apabila pembacaan jarak > 100 atau jarak < 30, maka akan mendapatkan notifikasi pada Telegram seperti gambar di bawah





## PROJECT III

### MEMBUAT SMARTHOME DENGAN APLIKASI BLYNK

- **DESKRIPSI** :

Project ini bertujuan untuk membuat sebuah *smarthome* sederhana yang dikendalikan melalui *smartphone* dengan menggunakan Aplikasi Blynk dan juga *Board NodeMCU* yang saling terhubung.

- **BACAAN SINGKAT** :

**A. Internet of Things**

*Internet of Things (IoT)* merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari koneksi internet yang tersambung secara terus-menerus [4]. *Internet of Things* digunakan untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara *independent*. Salah satu contohnya adalah *smarthome* yang dapat memantau dan mengontrol kondisi rumah melalui *smartphone* dengan bantuan koneksi internet.

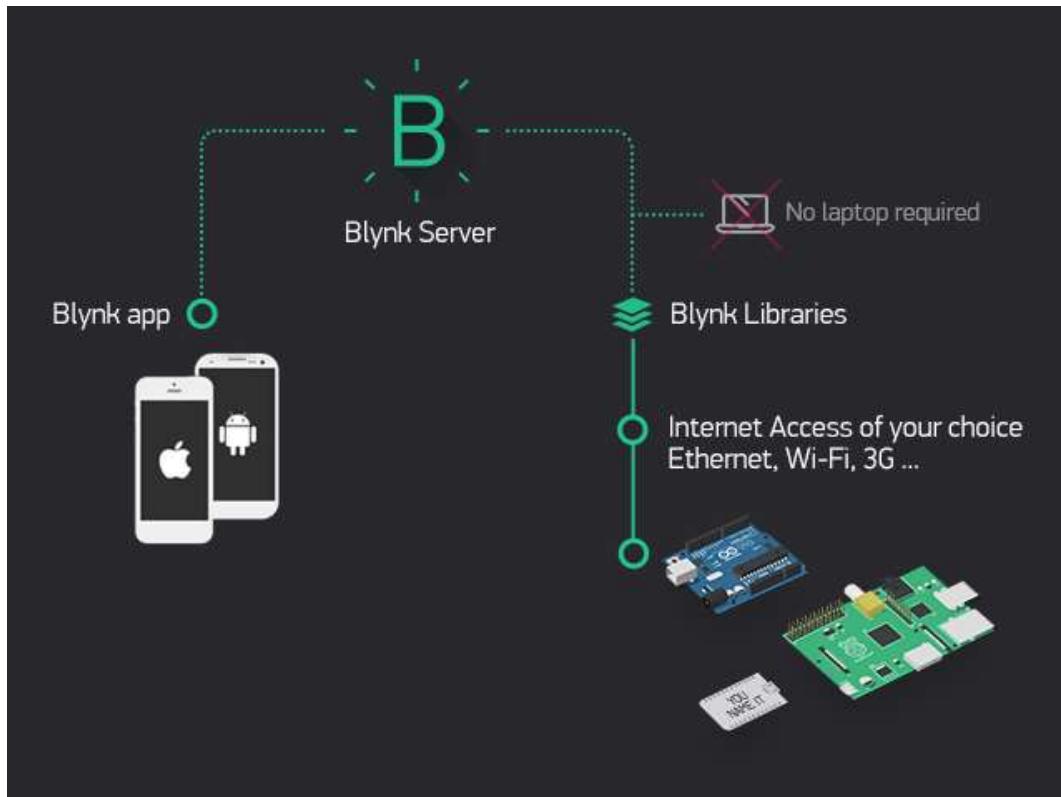
Pada dasarnya, perangkat *Internet of Things* terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komunikasi dan *server* sebagai pengumpul informasi yang diterima dari *sensor* dan dianalisa. Gambar di bawah menunjukkan gambaran cara kerja *Internet of Things* yang terdiri dari perangkat fisik, jaringan internet, dan *cloud database / server*.





## B. Blynk

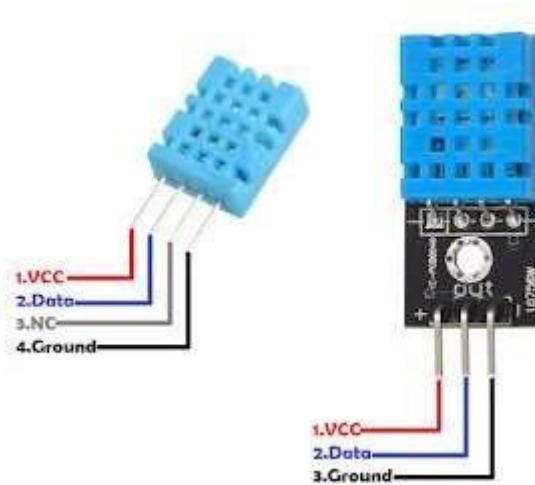
Blynk adalah sebuah konsep yang didesain untuk keperluan *Internet of Things*. Blynk dapat melakukan kontrol dan juga monitoring terhadap suatu sistem melalui Aplikasi Mobile. Blynk memiliki 3 komponen utama, yaitu diantara Blynk App, Blynk Server, dan Blynk Library. Berikut cara kerja dari Blynk [5]:



Berdasarkan 3 komponen utama di atas, Blynk berjalan dengan melakukan pemrograman terlebih dahulu pada perangkat keras seperti NodeMCU atau Raspberry Pi yang di dalam programnya dimasukkan Blynk Library untuk dapat terhubung ke Aplikasi dan juga Server. Lalu, pada Aplikasi diperlukan pengaturan Widget atau Komponen yang ingin digunakan untuk melakukan monitoring atau kontrol terhadap sistem.



### C. DHT11



DHT11 adalah salah satu sensor yang dapat mengukur dua parameter lingkungan sekaligus, yakni suhu dan kelembaban udara (*humidity*). Dalam sensor ini terdapat sebuah thermistor tipe NTC (*Negative Temperature Coefficient*) untuk mengukur suhu, sebuah sensor kelembaban tipe resistif dan sebuah mikrokontroller 8-bit yang mengolah kedua sensor tersebut dan mengirim hasilnya ke pin output dengan format *single-wire bi-directional* (kabel tunggal dua arah) [6].

NTC memiliki karakteristik dimana nilai resistansinya berbanding terbalik dengan kenaikan suhu. Artinya, semakin tinggi suhu ruangan maka nilai resistansi NTC akan semakin kecil. Sebaliknya nilai resistansi akan meningkat ketika suhu disekitar sensor menurun

- **Spesifikasi dari DHT11:**

|                                   |   |
|-----------------------------------|---|
| Tegangan Input                    | 3.3 – 5.5 V                                 |
| Output                            | Sinyal Digital                              |
| Range Pengukuran Suhu             | 0-50 °C dengan Akurasi $\pm 2$ °C           |
| Range Pengukuran Kelembaban Udara | 20-90% RH dengan Akurasi $\pm 5\%$ RH error |
| Periode Sampling                  | 2 Detik                                     |
| Resolusi Pengukuran               | 16 Bit                                      |



#### D. MQ-9 Gas Sensor



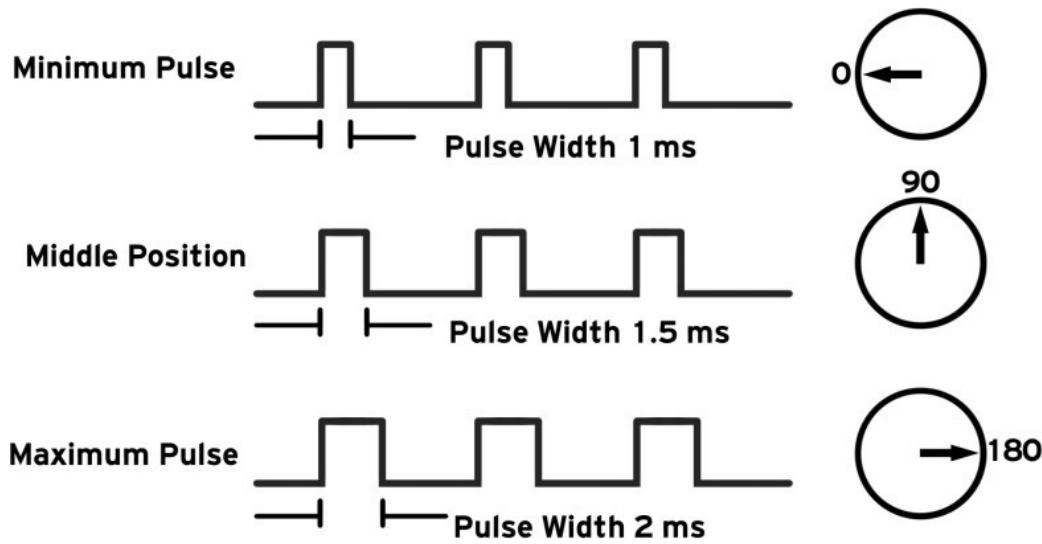
MQ-9 Gas Sensor adalah salah satu keluarga dari Sensor MQ yang memiliki kegunaan untuk mendeteksi kandungan CO, Metana, dan juga LPG atau Gas mudah terbakar. Sensor ini memiliki 4 Pin, diantaranya Pin Analog Output, Digital Output, VCC, dan juga Gnd. Sensor ini bekerja optimal pada tegangan 5V dan memerlukan kalibrasi untuk memulai perhitungannya.

#### E. Motor Servo



Motor Servo adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem closed feedback yang terintegrasi dalam motor tersebut. Pada motor servo posisi putaran sumbu (axis) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo.

Operasional motor servo dikendalikan oleh sebuah pulsa selebar  $\pm 20$  ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum. Apabila motor servo diberikan pulsa dengan besar 1.5 ms mencapai gerakan  $90^\circ$ , maka bila kita berikan pulsa kurang dari 1.5 ms maka posisi mendekati  $0^\circ$  dan bila kita berikan pulsa lebih dari 1.5 ms maka posisi mendekati  $180^\circ$ .



Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi Ton duty cycle 1.5 ms, maka rotor dari motor akan berhenti tepat di tengah-tengah

Pada saat Ton duty cycle dari sinyal yang diberikan kurang dari 1.5 ms, maka rotor akan berputar ke berlawanan arah jarum jam (Counter Clock wise, CCW) dengan membentuk sudut yang besarnya linier terhadap besarnya Ton duty cycle, dan akan bertahan diposisi tersebut. Dan sebaliknya, jika Ton duty cycle dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (Clock Wise, CW) dengan membentuk sudut yang linier pula terhadap besarnya Ton duty cycle, dan bertahan diposisi tersebut [7].

- **KOMPONEN :**

1. Board NodeMCU V3
2. Kabel Jumper
3. Breadboard / Expansion Board
4. Kabel USB / Data
5. DHT 11
6. MQ-9 Gas Sensor
7. Relay 2 Channel
8. Motor Servo

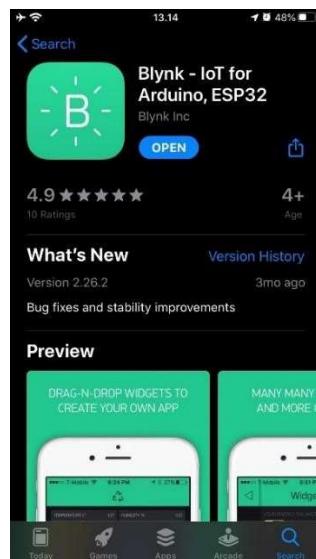


9. Smartphone
10. Aplikasi Blynk
11. Arduino IDE

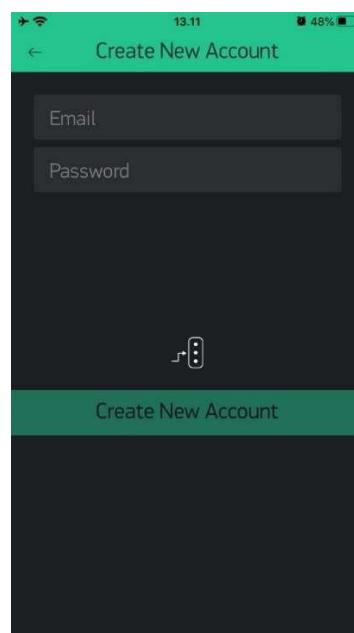
- **LANGKAH-LANGKAH**

**A. Konfigurasi Blynk**

1. Download Aplikasi Blynk melalui AppStore atau Play Store

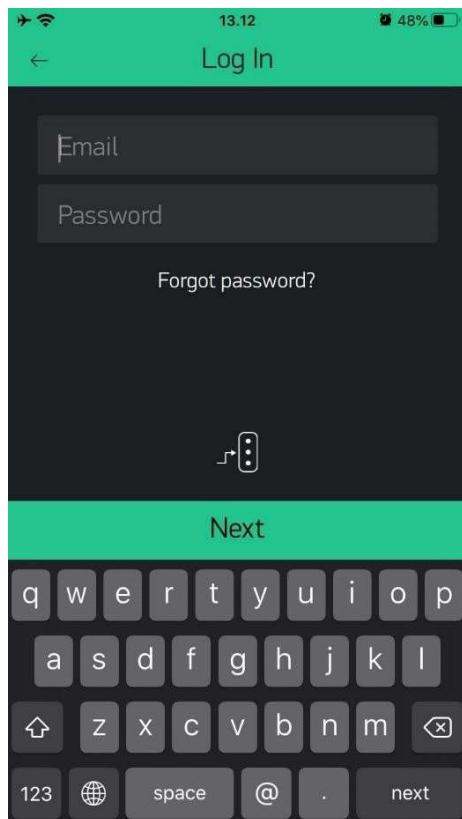


2. Setelah selesai di-*download*, kemudian buka Aplikasi Blynk
3. Apabila belum memiliki Akun, kalian dapat mendaftarkannya

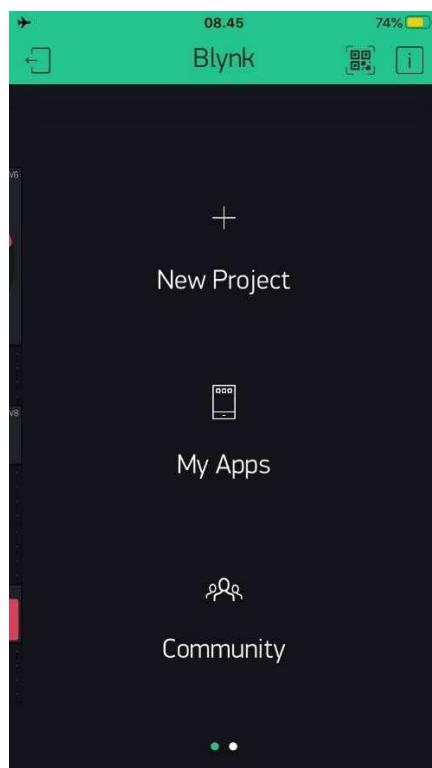




4. Setelah daftar, login dengan akun yang telah kalian daftarkan



5. Kemudian buatlah sebuah Project baru

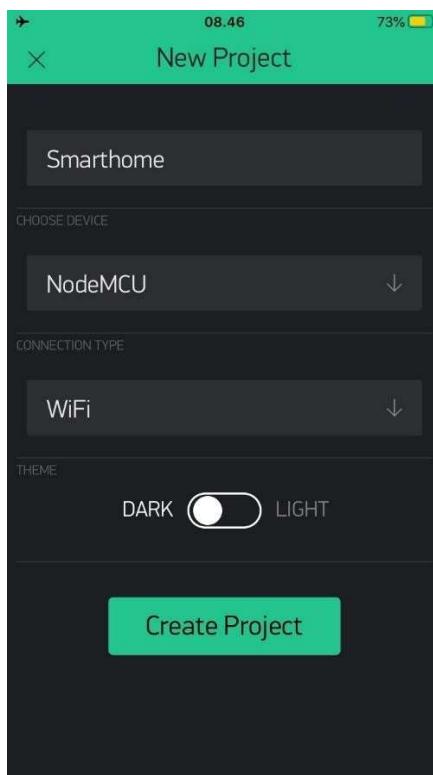




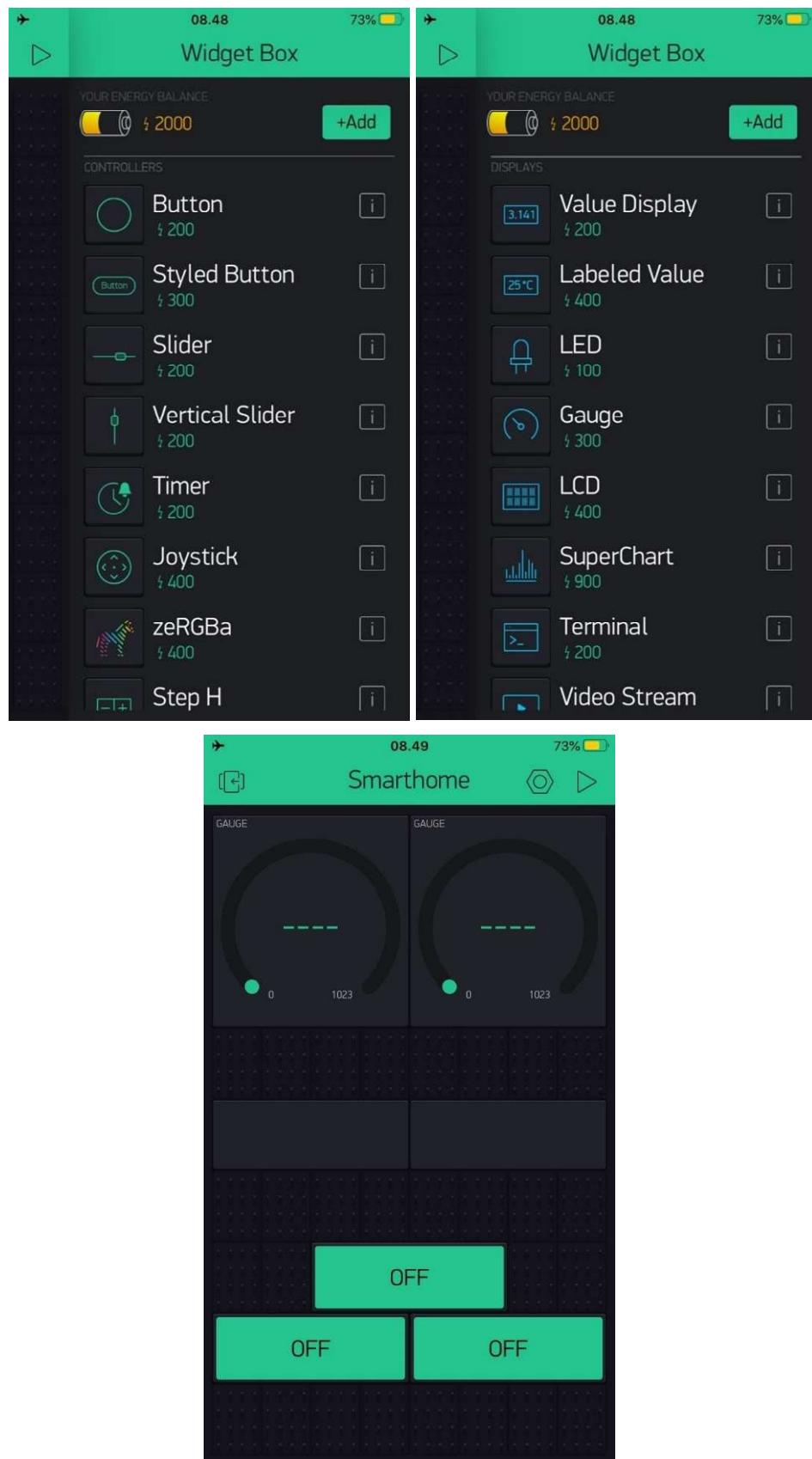
6. Lalu, isikan detail Project sesuai dengan ketentuan dibawah:

- Nama Project : Smarthome
- Device : NodeMCU
- Connection Type : WiFi

Lalu, klik Create Project

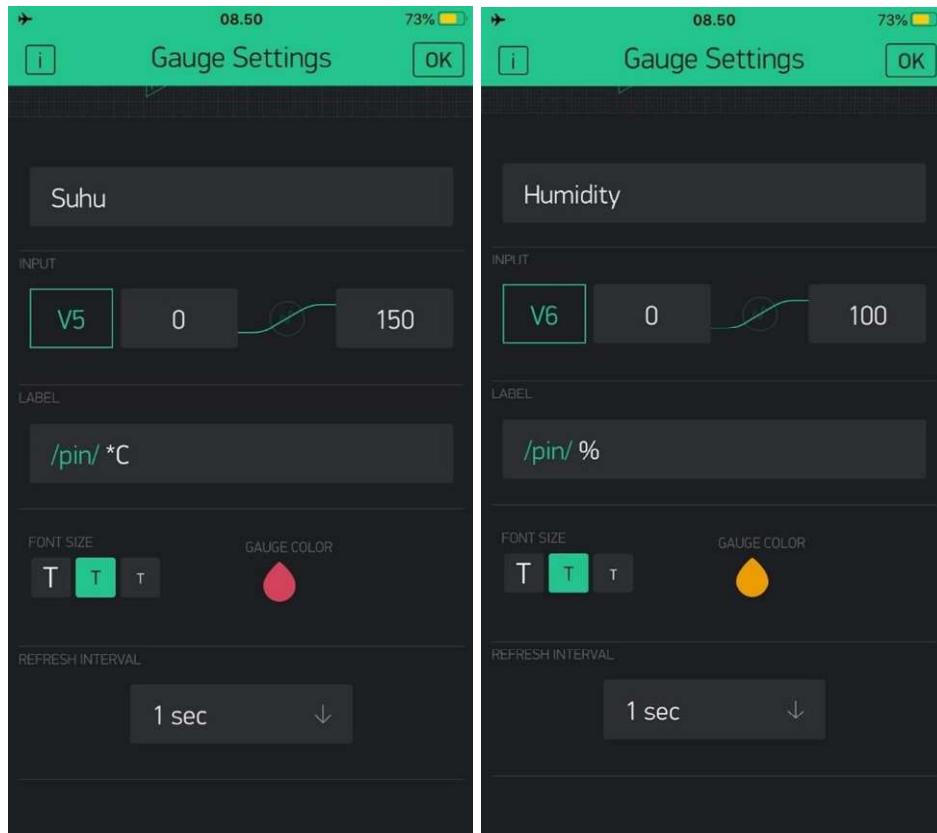


7. Kemudian, silahkan Check Email kalian yang digunakan untuk mendaftar akun Blynk untuk mendapatkan Token yang akan digunakan dalam Program Arduino.
8. Kemudian, pada Halaman Utama Project Tambahkan Widget 3 Styled Button, 2 Gauge, dan 2 Value Display lalu bentuk UI seperti gambar di bawah:

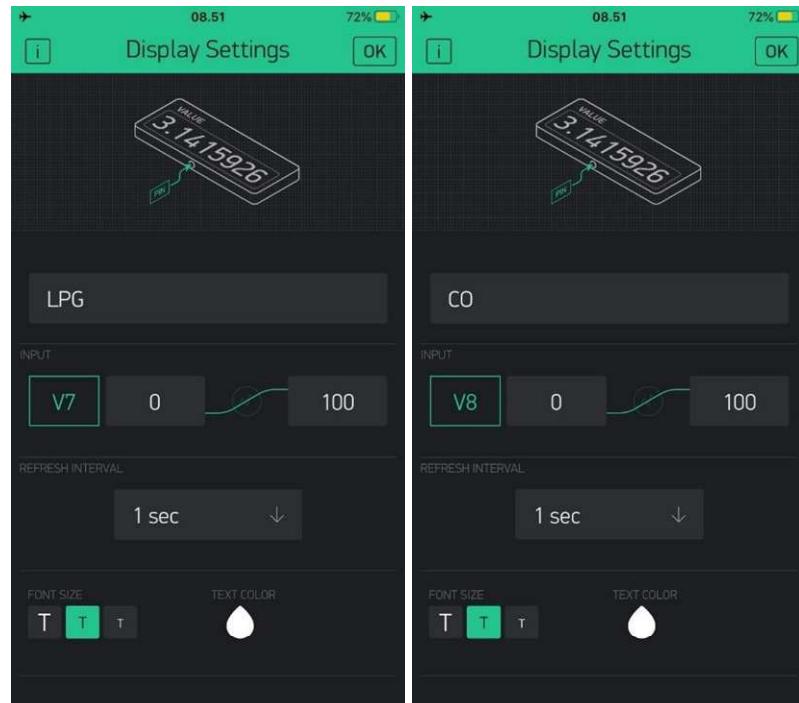




9. Lalu, atur 2 Widget Gauge dengan ketentuan seperti gambar di bawah:

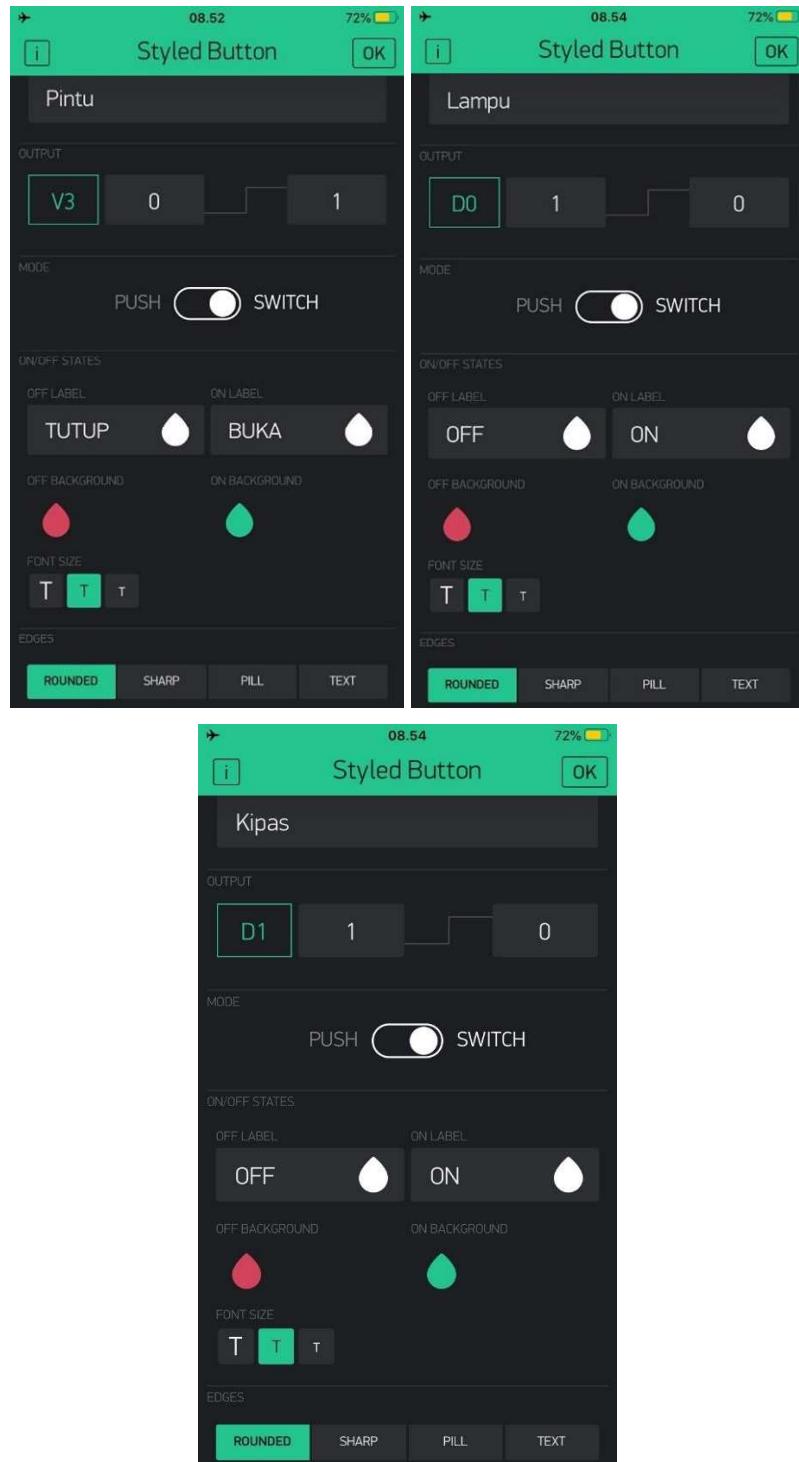


10. Selanjutnya, atur 2 Widget Value Display dengan ketentuan seperti gambar di bawah:





11. Lalu, atur 3 Widget Styled Button dengan ketentuan seperti gambar di bawah:



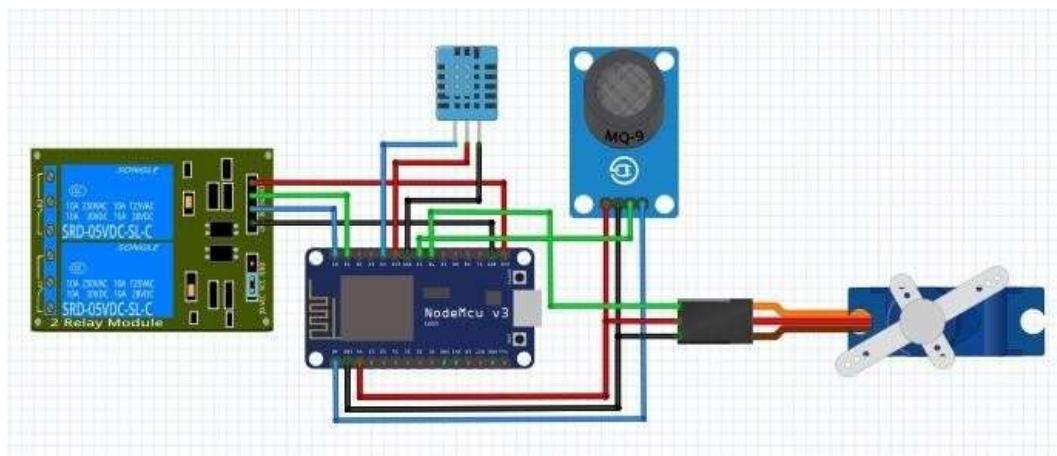
12. Akhirnya, UI pada Aplikasi Blynk selesai dibuat dan lanjut ke Pemrograman di Arduino IDE untuk Board NodeMCU

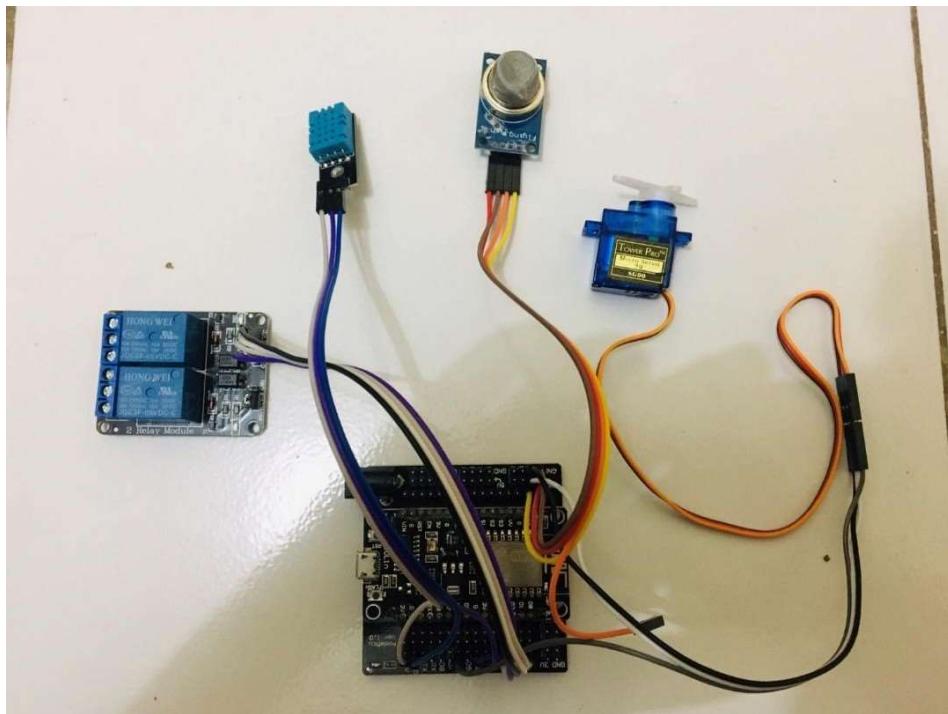


## B. Pemrograman pada NodeMCU

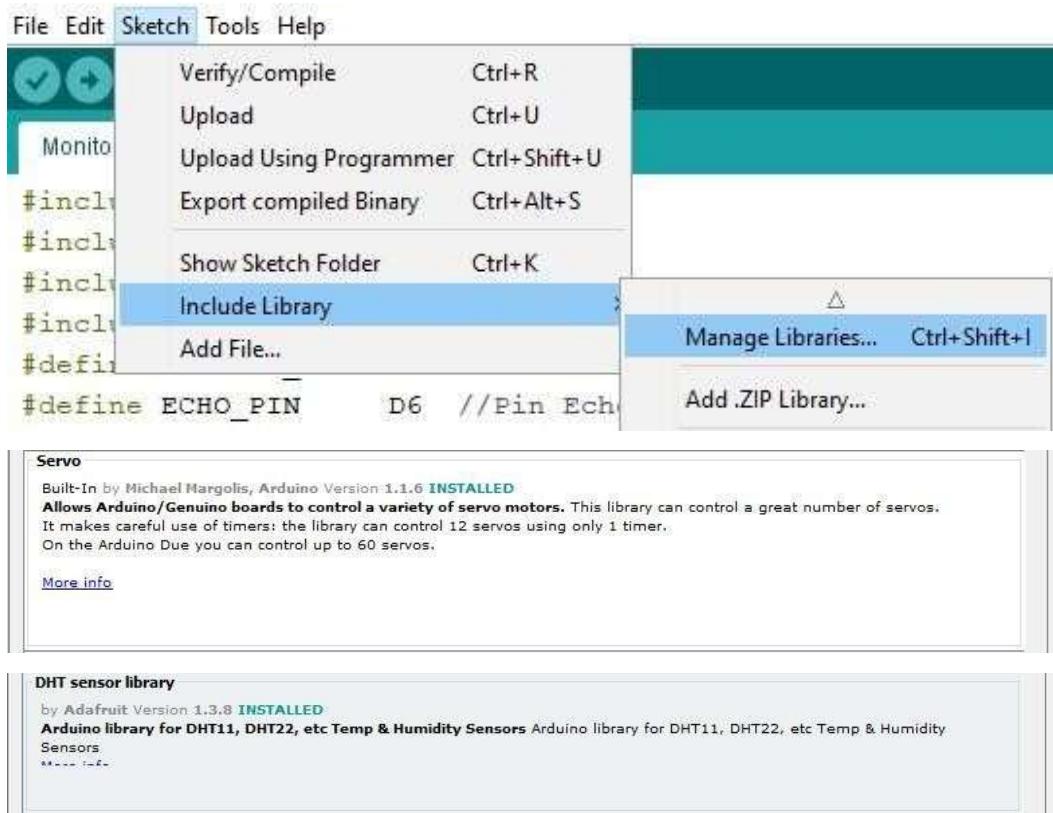
1. Hubungkan DHT11, MQ-9, Motor Servo, dan Relay pada Board NodeMCU seperti ketentuan dibawah:

| Pin MQ-9        | Pin NodeMCU |
|-----------------|-------------|
| A0              | A0          |
| D0              | D6          |
| VCC             | V USB       |
| GND             | GND         |
| Pin DHT11       | Pin NodeMCU |
| Pin Data        | D4          |
| VCC DHT11       | 3.3V        |
| GND DHT11       | GND         |
| Pin Motor Servo | Pin NodeMCU |
| Pin Data        | D5          |
| VCC Motor Servo | V USB       |
| GND Motor Servo | GND         |
| Pin Relay       | Pin NodeMCU |
| Pin IN1         | D0          |
| Pin IN2         | D1          |
| VCC Relay       | 3.3V        |
| GND Relay       | GND         |





2. Buka Arduino IDE, kemudian install Library yang diperlukan yaitu **DHT**, **Servo**, dan **MQUnifiedSensor** versi terbaru seperti gambar di bawah:





### 3. Kemudian, copy dan paste source code di bawah ke Arduino IDE

```
#define BLYNK_PRINT Serial //Untuk Komunikasi Serial Blynk
#include <ESP8266WiFi.h> //Untuk Library NodeMCU WiFi
#include <BlynkSimpleEsp8266.h> //Untuk Konek Blynk dan NodeMCU
#include <DHT.h> //Library DHT
#include <MQUnifiedsensor.h> //Library MQ
#include <Servo.h> //Library Servo

char auth[] = "Token Blynk Kalian"; //Token Blynk
char ssid[] = "Wi-Fi Kalian"; //SSID Wi-Fi
char pass[] = "Pass Wi-Fi kalian"; //Password Wi-Fi

#define DHTPIN D4 //Pin DHT di NodeMCU
#define DHTTYPE DHT11 //Jenis DHT yang digunakan
#define Board ("ESP8266")
#define Pin (A0) //Pin MQ-9 di NodeMCU
#define Type ("MQ-9") //MQ9
#define Voltage_Resolution (5) //Tegangan yang digunakan
#define ADC_Bit_Resolution (10) //Nilai Bit ADC NodeMCU
#define RatioMQ9CleanAir (9.6) //Parameter
#define Lampu D0 //Relay 1
#define Kipas D1 //Relay 2

DHT dht(DHTPIN, DHTTYPE); //Identifikasi DHT dan aktifkan fungsi DHT
Servo servo; //Mengaktifkan fungsi Servo
MQUnifiedsensor MQ9(Board, Voltage_Resolution, ADC_Bit_Resolution, Pin, Type); //Identifikasi MQ dan Aktifkan fungsi MQ
BlynkTimer timer; //Blynk Timer

//Untuk melakukan pembacaan sensor
void sendSensor()
{
//Untuk melakukan pembacaan nilai LPG dan CO
    MQ9.update();
    MQ9.setA(1000.5); MQ9.setB(-2.186);
    float LPG = MQ9.readSensor();
    MQ9.setA(599.65); MQ9.setB(-2.244);
    float CO = MQ9.readSensor();

//Untuk melakukan pembacaan nilai Suhu dan Humidity
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }
}
```



```
}

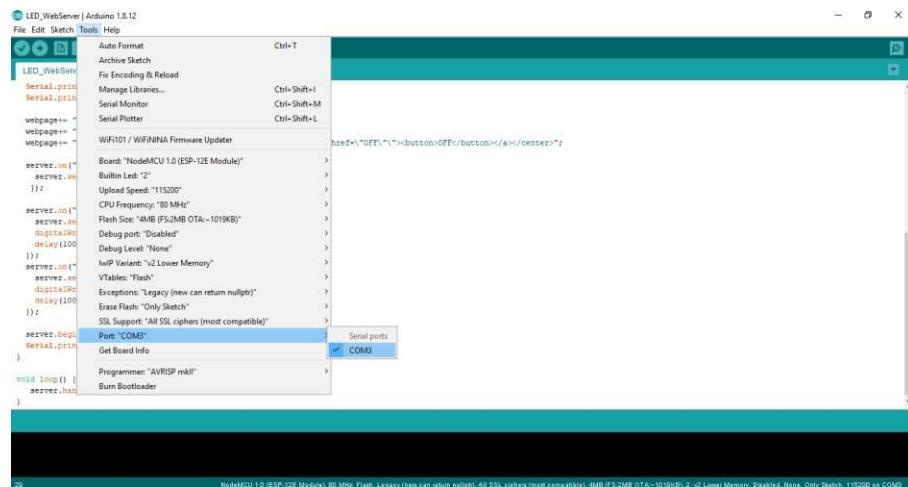
//Mengirimkan nilai ke Blynk
Blynk.virtualWrite(V5, t);
Blynk.virtualWrite(V6, h);
Blynk.virtualWrite(V7, LPG);
Blynk.virtualWrite(V8, CO);
}

//Untuk fungsi menggerakkan Servo
BLYNK_WRITE(V3)
{
//Kondisi apabila nilai yang diterima 1, maka servo bergerak ke
180 derajat, dan apabila nilai yang diterima 0, maka servo
bergerak ke 0 derajat
    int pinValue = param.asInt();
    if (pinValue == 1) {
        servo.write(180);
    }
    if (pinValue == 0) {
        servo.write(0);
    }
}
void setup()
{
    Serial.begin(115200); //Komunikasi Serial Baudrate 115200
    pinMode(Lampu, OUTPUT); //Set Relay 1 jadi Output
    pinMode(Kipas, OUTPUT); //Set Relay 2 jadi Output
    digitalWrite(Lampu, HIGH); //Set nilai Awal Relay 1
    digitalWrite(Kipas, HIGH); //Set nilai Awal relay 2
    Blynk.begin(auth, ssid, pass); //Komunikasi ke Wifi dan Blynk
    servo.attach(12); //Pin Servo di NodeMCU
    dht.begin(); //Memulai aktif DHT
    MQ9.setRegressionMethod(1); //Mengatur Metode Kalibrasi MQ
    MQ9.init(); //Memulai aktif MQ
//Kalkulasi untuk Kalibrasi MQ-9
    float calcR0 = 0;
    for(int i = 1; i<=10; i++)
    {
        MQ9.update();
        calcR0 += MQ9.calibrate(RatioMQ9CleanAir);
        Serial.print(".");
    }
    MQ9.setR0(calcR0/10);
//Timer untuk interval pengiriman hasil pembacaan sensor ke Blynk
    timer.setInterval(1000L, sendSensor);
}

void loop()
{
    Blynk.run(); //Menjalankan Blynk
    timer.run(); //Mengaktifkan fungsi Timer
}
```



4. Kemudian hubungkan Board NodeMCU dengan Laptop menggunakan Kabel USB. Selanjutnya, pilih Menu **Tools -> Port**. Lalu, pilih Port yang tersedia seperti gambar di bawah:



5. Kemudian, klik **Button Upload** yang berbentuk seperti tanda panah untuk melakukan **Compile** dan **Upload Program** ke Board NodeMCU.



6. Setelah program selesai di-upload dan berhasil, maka buka **Serial Monitor** dan lihat hasil yang ditampilkan pada **Serial Monitor**:



7. Hasil yang ditampilkan pada Serial Monitor adalah sebagai berikut apabila berhasil terhubung ke Server Blynk

```
[2563] Connected to WiFi
[2564] IP: 192.168.1.8
[2564]
/ \ / \ / \ / \ / \ / \
/ = / / / / \ \ \ \ \ 
/ \ / \ \ \ / \ / \ / \ \ \
/ \ / \ v0.6.1 on NodeMCU

[2570] Connecting to blynk-cloud.com:80
[2633] Ready (ping: 24ms).
.....
< >
 Autoscroll  Show timestamp
Newline 115200 baud Clear output
```



## PROJECT IV

### MEMBUAT SMART GREEN HOUSE DENGAN MENGGUNAKAN NODEMCU DAN THINGER IO

#### - DESKRIPSI :

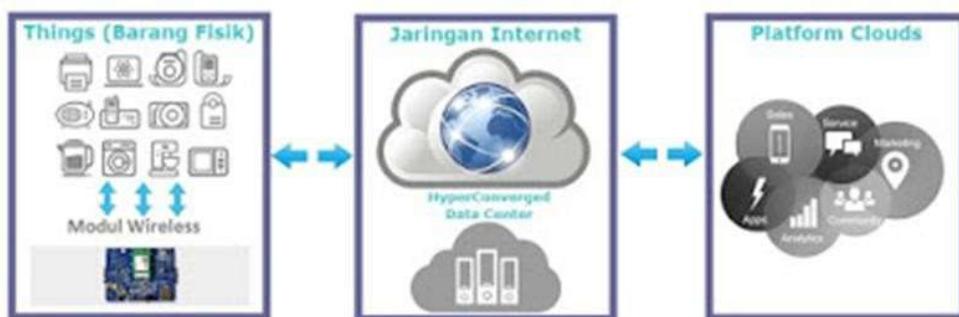
Project ini bertujuan untuk membuat sebuah *Smart Green House* yang dikendalikan dan dimonitoring melalui Platform Thinger IO dan juga *Board NodeMCU* yang saling terhubung.

#### - BACAAN SINGKAT :

##### A. *Internet of Things*

*Internet of Things (IoT)* merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari koneksi internet yang tersambung secara terus-menerus. *Internet of Things* digunakan untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara *independent*. Salah satu contohnya adalah *smarthome* yang dapat memantau dan mengontrol kondisi rumah melalui *smartphone* dengan bantuan koneksi internet.

Pada dasarnya, perangkat *Internet of Things* terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komunikasi dan *server* sebagai pengumpul informasi yang diterima dari *sensor* dan dianalisa. Gambar di bawah menunjukkan gambaran cara kerja *Internet of Things* yang terdiri dari perangkat fisik, jaringan internet, dan *cloud database / server*.





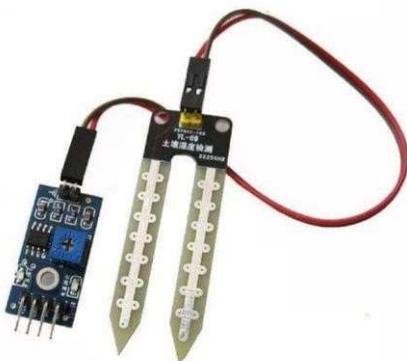
## B. Thinger IO



Thinger IO adalah sebuah *platform* yang bersifat *open source* yang memiliki fungsi untuk keperluan IoT. Thinger IO menyediakan layanan yang sudah siap untuk digunakan dengan UI yang mudah dimengerti dan mudah dalam manajemen.

Thinger IO pun memberikan layanan *cloud* untuk menyimpan data yang berasal dari IoT Device agar dapat disimpan dan dapat ditampilkan data tersebut melalui UI yang sudah dibuat.

## C. Sensor Soil Moisture YL-69



Sensor YL-69 adalah salah satu sensor yang dapat digunakan untuk mengukur kelembaban tanah. Karakteristik dari sensor ini adalah dengan mencari hubungan tegangan dan Panjang sensor yang ditancapkan terhadap nilai resistivitas. Parameter dari penggunaan Sensor ini adalah apabila tegangan rendah, maka tanah dianggap kering dan apabila tegangan tinggi, maka tanah dianggap basah. Sensor ini terdiri dari Modul dan juga Probe yang saling terhubung 1 sama lain. Sensor ini juga bekerja pada tegangan 3.3V – 5V, dan memiliki keluaran nilai Analog.



## D. Cara Kerja



Cara kerja dari sistem ini adalah dengan mengambil data dari sensor yang digunakan pada *IoT Device* kemudian dikirimkan ke *cloud* atau Thinger IO *Service* melalui koneksi internet dengan menggunakan topik yang sudah dipasang pada program. Lalu, data yang diterima pada Thinger IO dapat di-*output*-kan melalui *Dashboard* yang sudah dibuat pada Thinger IO.

Untuk bagian kontrol juga dilakukan dengan melakukan pengiriman perintah melalui Thinger IO dengan *Dashboard* yang sudah dibuat ke *IoT Device* melalui pengiriman nilai 0 atau 1 dengan topik yang sudah diprogram pada *IoT Device*.

### - KOMPONEN :

1. Board NodeMCU V3
2. Kabel Jumper
3. Breadboard / Expansion Board
4. Kabel USB / Data
5. DHT 11
6. HCSR-04
7. Relay 2 Channel
8. LED
9. Soil Moisture YL-69
10. Thinger IO
11. Web Browser
12. Arduino IDE



- LANGKAH-LANGKAH

A. Konfigurasi Awal pada Thinger IO

1. Buka halaman Thinger IO pada Browser : <https://thinger.io/>
2. Setelah itu, pilih **Sign Up** apabila kalian belum memiliki akun, lalu pilih **Create An Account**

The screenshot shows the Thinger.io login interface. At the top, there are navigation links: BLOG, DEVELOPERS (with a dropdown arrow), PRICING (highlighted in blue), CONTACT US, SHOP, and CART. To the right are social media icons for Facebook, Twitter, YouTube, and Instagram. Below the header is the Thinger.io logo, which includes a blue icon of three cubes stacked in a triangle and the text "thinger.io platform". A sub-header reads "Sign in to manage your cloud". The main form has two input fields: "Username or Email" and "Password", both with placeholder text. There is a checkbox labeled "Remember me on this computer" and a large blue "Log in" button. Below the log-in area are links for "Forgot password?" and "Do not have an account?". The "Create an account" link is located within a white rectangular box.



3. Lalu masukkan data diri pada kolom yang sudah disediakan dan klik Sign Up

The screenshot shows the Thinger.io sign-up interface. At the top, there's a logo with three blue cubes and the text "thinger.io platform". Below it, a button says "Sign up to start connecting your things". The form has four input fields:

- First name: nrkholid
- Email: nrkholid1@gmail.com
- Password (obscured)
- Confirm Password (obscured)

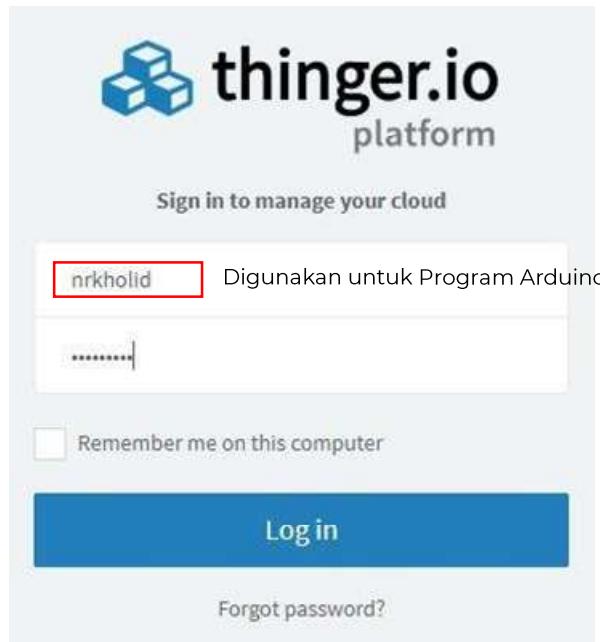
Below the form is a checkbox labeled "Agree the terms and privacy policy". Underneath that is a reCAPTCHA box with a green checkmark and the text "I'm not a robot". To the right of the reCAPTCHA is the reCAPTCHA logo and the text "reCAPTCHA Privacy - Terms". At the bottom is a large blue "Sign up" button.

4. Setelah daftar, buka Email kalian untuk mengaktifkan dan verifikasi Akun kalian

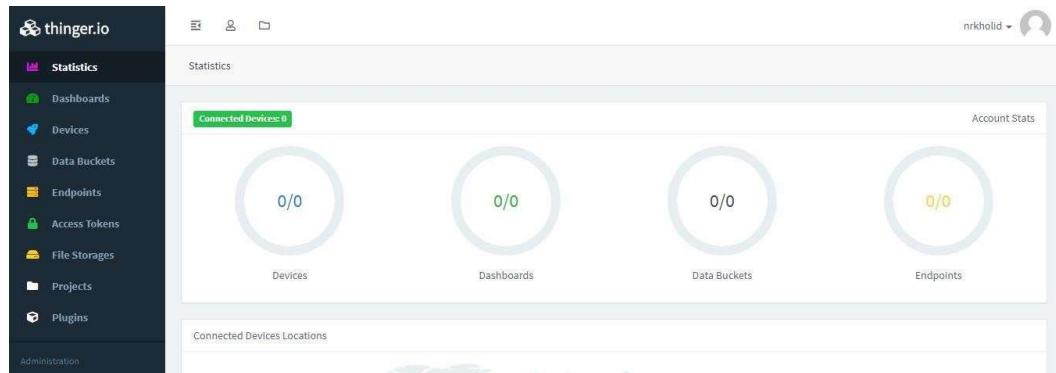




5. Lalu lakukan login Thinger IO dengan akun yang sudah kalian buat



6. Selanjutnya, kalian akan tiba di halaman *Dashboard*



7. Kemudian, pada halaman *Dashboard* tersebut di sebalah kiri ada menu *Devices*, lalu kalian pilih Menu tersebut





8. Lalu kemudian kalian pilih Add Device

| Device             | Type | Description | Last Connection |
|--------------------|------|-------------|-----------------|
| No devices to show |      |             |                 |

9. Kemudian kolom isian yang ada pada halaman tersebut dengan ketentuan seperti gambar di bawah:

Device Details

|                    |  |
|--------------------|--|
| Device Type        | Generic Thinger Device (WiFi, Ethernet, GSM)           |
| Device Id          | MCUNR  |
| Device description | NodeMCU  |
| Device credentials | Enter device credentials<br>Generate Random Credential |

Add Device

10. Pada bagian **Device Credentials**, kalian klik **Generate Random Credential**. Lalu, akan muncul kode unik seperti gambar di bawah dan selanjutnya kalian klik Add Device

Device Details

|                    |  |
|--------------------|--|
| Device Type        | Generic Thinger Device (WiFi, Ethernet, GSM)         |
| Device Id          | MCUNR  |
| Device description | NodeMCU  |
| Device credentials | QIA+%Y3GiXKJ<br>Digunakan untuk Program pada Arduino |

Generate Random Credential

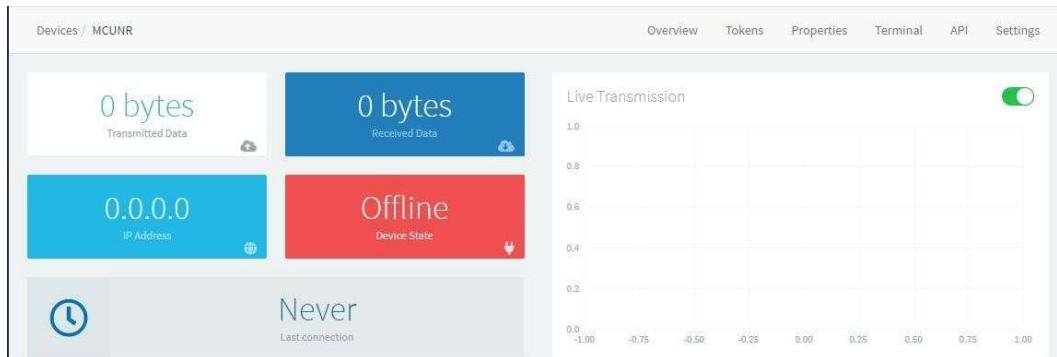
Add Device



11. Selanjutnya, Device pun selesai dibuat dan pada Halaman Device akan muncul List Device yang sudah dibuat seperti gambar di bawah

| Device List                    |              |                 |                 |
|--------------------------------|--------------|-----------------|-----------------|
|                                | + Add Device | Refresh Devices |                 |
| Device                         | Type         | Description     | Last Connection |
| <input type="checkbox"/> MCUNR | Generic      | NodeMCU         | Never           |
| Showing 1 device.              |              |                 |                 |

12. Selanjutnya kalian klik Device yang telah kalian buat, lalu akan muncul tampilan seperti berikut:



13. Pada bagian tersebut ditampilkan bahwa status *Device* adalah offline karena kita belum melakukan program pada *Board NodeMCU* agar terhubung ke Thinger IO

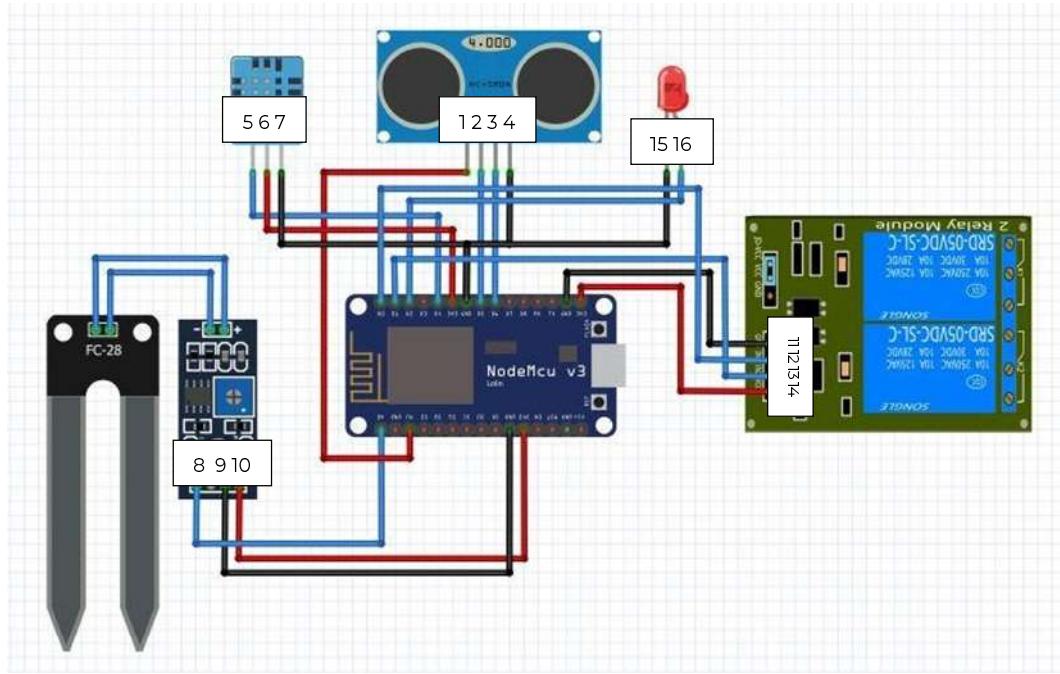
## B. Konfigurasi pada Board NodeMCU dan Arduino IDE

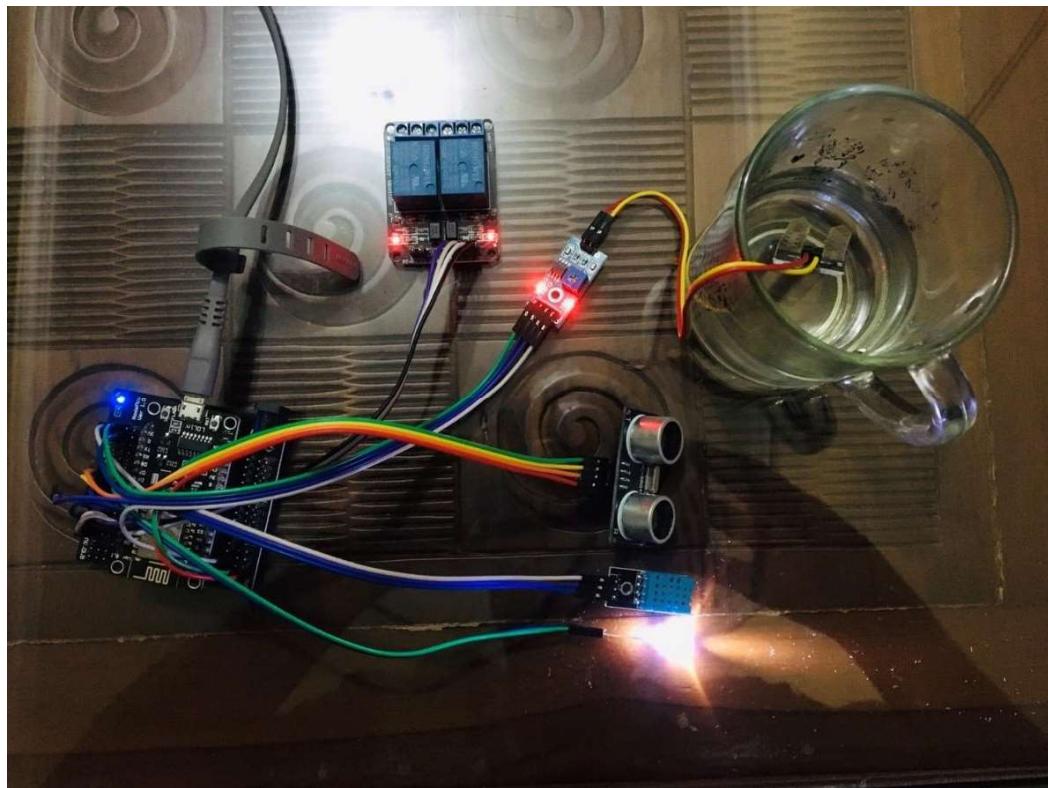
1. Hubungkan DHT11, HCSR-04, Soil Moisture YL-69, LED dan Relay pada Board NodeMCU seperti ketentuan dibawah:

| No | Pin HC-SR04 | Pin NodeMCU |
|----|-------------|-------------|
| 1  | VCC         | V USB       |
| 2  | Trigger     | D5          |
| 3  | Echo        | D6          |
| 4  | GND         | GND         |
| No | Pin DHT11   | Pin NodeMCU |
| 5  | Pin Data    | D4          |

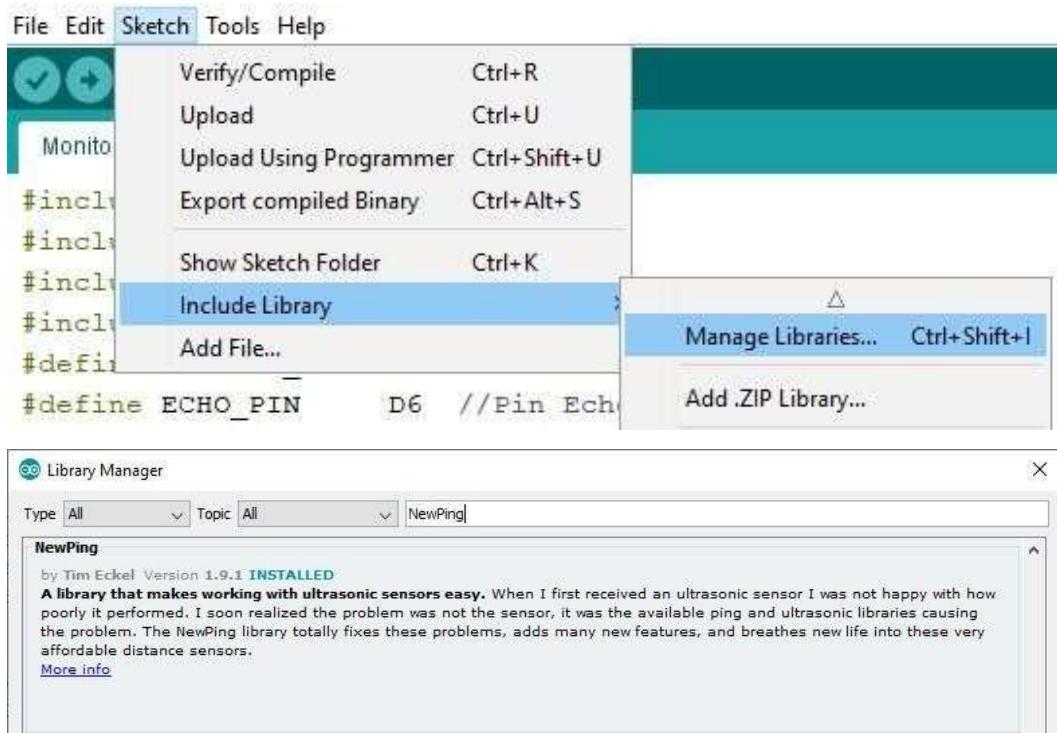


|           |                                |                    |
|-----------|--------------------------------|--------------------|
| 6         | VCC DHT11                      | 3.3V               |
| 7         | GND DHT11                      | GND                |
| <b>No</b> | <b>Pin Soil Moisture YL-69</b> | <b>Pin NodeMCU</b> |
| 8         | Pin Data                       | A0                 |
| 9         | GND YL-69                      | Gnd                |
| 10        | VCC YL-69                      | 3.3V               |
| <b>No</b> | <b>Pin Relay</b>               | <b>Pin NodeMCU</b> |
| 11        | GND Relay                      | Gnd                |
| 12        | Pin IN1                        | D0                 |
| 13        | Pin IN2                        | D1                 |
| 14        | VCC Relay                      | 3.3V               |
| <b>No</b> | <b>Pin LED</b>                 | <b>Pin NodeMCU</b> |
| 15        | Kaki Pendek                    | Gnd                |
| 16        | Kaki Panjang                   | D2                 |





2. Buka Arduino IDE, kemudian install Library yang diperlukan yaitu **DHT**, **NewPing**, dan **Thinger IO** versi terbaru seperti gambar di bawah:





### 3. Kemudian, copy dan paste source code di bawah ke Arduino IDE

```
#include <ESP8266WiFi.h> //Library untuk koneksi ke Wi-Fi
#include <ThingerESP8266.h> //Library untuk terhubung ke Thinger IO
#include <NewPing.h> //Library untuk HC-SR04
#include <DHT.h> //Library untuk DHT

#define USERNAME "nrkholid" //Username Thinger IO
#define DEVICE_ID "MCUNR" //Device ID Thinger IO
#define DEVICE_CREDENTIAL "QlA+%Y3GiXKJ" //Device Credential Thinger IO

#define SSID "KILLUA" //SSID Wi-Fi
#define SSID_PASSWORD "nada124@" //Password Wi-Fi

#define TRIGGER_PIN D5 //Pin Trigger HC-SR04 pada NodeMCU
#define ECHO_PIN D6 //Pin Echo HC-SR04 pada NodeMCU
#define MAX_DISTANCE 250 //Maksimum Pembacaan Jarak (cm)

#define DHTPIN D4 //Pin DHT pada NodeMCU
#define DHTTYPE DHT11 //Jenis DHT yang digunakan
#define pinYL A0 //pin Analog YL-69
#define WaterPump D0 //Pin Relay 1
#define WaterPump2 D1 //Pin Relay 2
#define Lampu D2 //Pin LED

ThingerESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
//Aktifkan Fungsi Thinger IO
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); //Aktifkan Fungsi HCSR-04
DHT dht(DHTPIN, DHTTYPE); //Aktifkan Fungsi DHT

int outputValue; //Variabel untuk Output YL-69

void setup() {
    pinMode(Lampu, OUTPUT); //Set LED sebagai Output
```



```
pinMode(WaterPump, OUTPUT); //Set Relay 1 sebagai Output
pinMode(WaterPump2, OUTPUT); //Set Relay 2 sebagai Output
digitalWrite(Lampu, LOW); //Set nilai LED = 0
digitalWrite(WaterPump, HIGH); //Set nilai Relay 1 = 1
digitalWrite(WaterPump2, HIGH); //Set nilai Relay 2 = 1
dht.begin(); //Memulai fungsi DHT
thing.add_wifi(SSID, SSID_PASSWORD); //Mencoba terhubung ke WiFi dan Thinger IO

//Fungsi untuk Kontrol Lampu dengan Topik Lampu
thing["Lampu"] << [] (pson& in) {
    if(in.is_empty()){
        in = (bool) digitalRead(Lampu);
    }
    else{
        digitalWrite(Lampu, in ? HIGH : LOW);
    }
};

//Fungsi untuk Kontrol Relay 1 dengan Topik Pompa Air
thing["Pompa Air"] << [] (pson& in) {
    if(in.is_empty()){
        in = (bool) digitalRead(WaterPump);
    }
    else{
        digitalWrite(WaterPump, in ? HIGH : LOW);
    }
};

//Fungsi untuk Kontrol Relay 2 dengan Topik Watering
thing["Watering"] << [] (pson& in) {
    if(in.is_empty()){
        in = (bool) digitalRead(WaterPump2);
    }
    else{
        digitalWrite(WaterPump2, in ? HIGH : LOW);
    }
};

//Mengirim Pembacaan Sensor DHT 11 ke Thinger IO
thing["DHT11"] >> [] (pson& out) {
    out["Humidity"] = dht.readHumidity();
    out["Suhu"] = dht.readTemperature();
};

//Mengirim Pembacaan Sensor HCSR-04 ke Thinger IO
thing["HCSR04"] >> [] (pson& out) {
    out["Jarak"] = sonar.ping_cm();
};

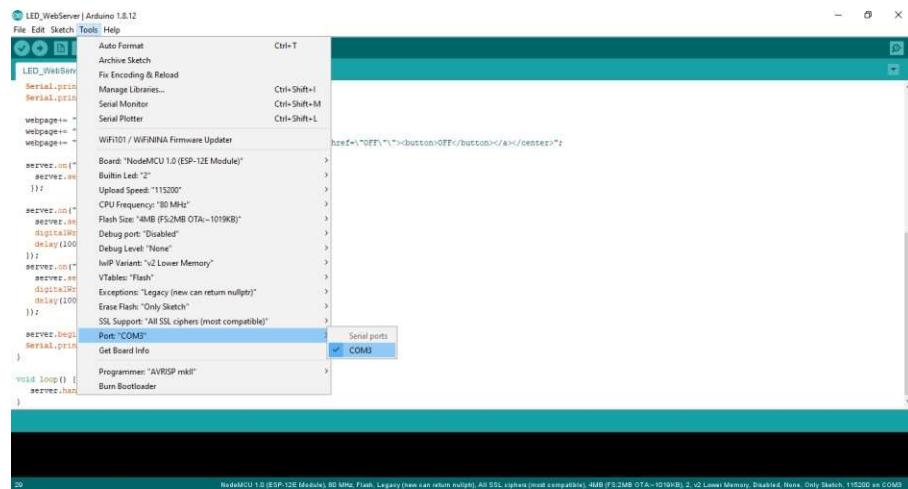
//Mengirim Pembacaan Sensor YL-69 ke Thinger IO
thing["Soil_Moisture"] >> [] (pson& out) {
    outputValue = analogRead(pinYL);
    outputValue = map(outputValue, 1024, 250, 0, 100);
    out["Moisture"] = outputValue;
};
```



```
}
```

```
void loop() {
    //Menjalankan fungsi Thinger IO
    thing.handle();
}
```

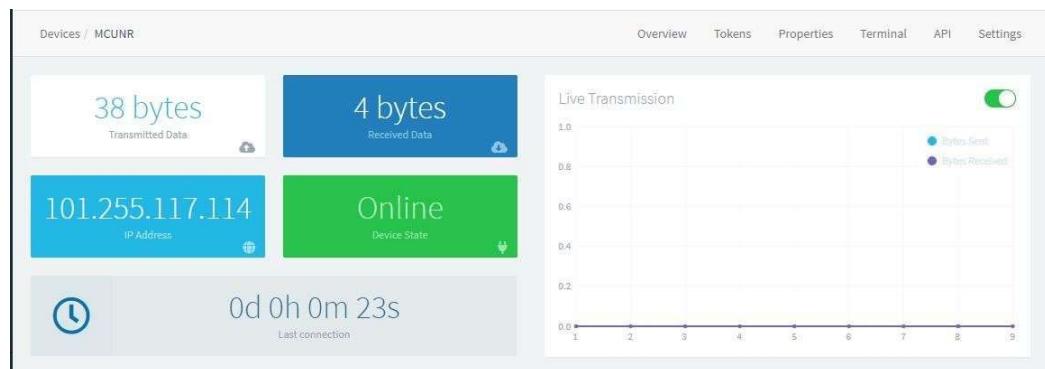
4. Kemudian hubungkan Board NodeMCU dengan Laptop menggunakan Kabel USB. Selanjutnya, pilih Menu **Tools -> Port**. Lalu, pilih Port yang tersedia seperti gambar di bawah:



5. Kemudian, klik **Button Upload** yang berbentuk seperti tanda panah untuk melakukan **Compile** dan **Upload Program** ke Board NodeMCU.



6. Setelah program selesai di-upload, buka kembali halaman Device pada Thinger IO dan kemudian lihat status dari Device kalian. Apabila sudah berubah menjadi Online, maka Device sudah terhubung ke Thinger IO





7. Selanjutnya kalian pilih Menu API pada Halaman Device tersebut

Overview    Tokens    Properties    Terminal    API    Settings

8. Kemudian akan muncul Topik yg tadi sudah kalian program pada Board NodeMCU yang berfungsi untuk menerima data dari sensor dan melakukan kontrol terhadap LED dan Relay

The screenshot shows the 'Devices / MCUNR / Api' section of the Thinger IO interface. At the top, there are tabs for Overview, Tokens, Properties, Terminal, API, and Settings. Below the tabs, a list of device topics is displayed in a table format:

| Topic                   | Type |
|-------------------------|------|
| Lampu - Private         |      |
| Pompa Air - Private     |      |
| Watering - Private      |      |
| DHT11 - Private         |      |
| HCSR04 - Private        |      |
| Soil_Moisture - Private |      |

### C. Konfigurasi Dashboard pada Thinger IO

1. Setelah Device kalian berhasil terhubung ke Thinger IO. Langkah selanjutnya adalah Pembuatan Dashboard
2. Pada Menu sebelah kiri terdapat Menu Dashboard. Lalu, kalian klik dan pilih Menu tersebut



3. Kemudian, kalian klik dan pilih Add Dashboard

The screenshot shows the 'Dashboard List' section of the Thinger IO interface. It features a header 'Dashboard List' with a refresh icon. Below the header are two buttons: a green 'Add Dashboard' button with a plus sign icon, and a blue 'Refresh Dashboards' button with a circular arrow icon.



4. Selanjutnya, isi kolom isian pada Add Dashboard dengan ketentuan seperti gambar di bawah dan kemudian klik Add Dashboard

|                       |                   |
|-----------------------|-------------------|
| Dashboard id          | SmartGreenHouse   |
| Dashboard name        | Smart Green House |
| Dashboard description | Green House       |

Add Dashboard

5. Selanjutnya, kembali ke Halaman Utama Dashboard dan pada List Dashboard sudah terdapat Dashboard yang sudah kita buat seperti gambar di bawah

| Dashboard List                               |            |   |
|--|------------|---|
| <input type="button" value="Add Dashboard"/> |            | <input type="button" value="Refresh Dashboards"/> |
| Dashboard                                    | Name       | Description                                       |
| <input type="checkbox"/>                     | GreenHouse | Green House using NodeMCU                         |
| Showing 1 dashboard                          |            |   |

6. Lalu, selanjutnya klik Dashboard ‘Green House’ yang sudah dibuat dan kemudian klik bagian switch di sebalah pojok kanan menjadi warna hijau agar dapat dilakukan editing seperti gambar di bawah





7. Kemudian, pilih Add Widget dan buatlah sebuah Widget dengan ketentuan seperti gambar di bawah:

### Widget Settings

Widget      On/Off State      Display Options

Title: Lampu  
Subtitle: Lampu Green House  
Background:    
Type: On/Off State

Save      Cancel

### Widget Settings

Widget      On/Off State      Display Options

Target Value: Device Resource  
Select Device: MCUNR  
Select Resource: Lampu

Save      Cancel

### Widget Settings

Widget      On/Off State      Display Options

Switch Style: Switch

Save      Cancel



91. Lalu, Tambahkan kembali Widget ke – 2 dengan ketentuan seperti gambar di bawah:

Widget Settings

Widget      On/Off State      Display Options

Title ⓘ Pompa Air

Subtitle ⓘ Pompa dari Mesin

Background ⓘ  #ffffff +

Type ⓘ On/Off State

Save Cancel

Widget Settings

Widget      On/Off State      Display Options

Target Value ⓘ Device Resource

ⓘ Select Device MCUNR

ⓘ Select Resource Pompa Air

Save Cancel

Widget Settings

Widget      On/Off State      Display Options

Switch Style ⓘ Switch

Save Cancel



92. Lalu, Tambahkan kembali Widget ke – 3 dengan ketentuan seperti gambar di bawah:

### Widget Settings

Widget      On/Off State      Display Options

Title  Pompa Air

Subtitle  Pompa untuk Penyiraman . ON = Mati , OFF = Nyala

Background  #ffffff

Type

### Widget Settings

Widget      On/Off State      Display Options

Target Value  Device Resource

Select Device  
MCUNR

Select Resource  
Watering

### Widget Settings

Widget      On/Off State      Display Options

Switch Style  Switch



10. Lalu, Tambahkan kembali Widget ke – 4 dengan ketentuan seperti gambar di bawah:

### Widget Settings

Widget      Donut Chart      Display Options

Title ⓘ      Suhu

Subtitle ⓘ      Suhu Green House

Background ⓘ      #d6d6d6      +

Type ⓘ      Donut Chart

**Save**      **Cancel**

### Widget Settings

Widget      **Donut Chart**      Display Options

Widget Value ⓘ      From Device Resource

① Select Device  
MCUNR

① Select Resource  
DHT11

① Select Value  
Suhu

① Refresh Mode  
Sampling Interval      1 seconds

**Save**      **Cancel**



## Widget Settings

Widget    Donut Chart    Display Options

Units  °C

Min Value  -50

Max Value  150

Donut Color  #000000

Save Cancel

11. Lalu, Tambahkan kembali Widget ke – 5 dengan ketentuan seperti gambar di bawah:

## Widget Settings

Widget    Donut Chart    Display Options

Title  Humidity

Subtitle  Humidity Green House

Background  #ffffff +

Type  Donut Chart

Save Cancel



## Widget Settings

Widget      Donut Chart      Display Options

**Widget Value** From Device Resource

Select Device  
MCUNR

Select Resource  
DHT11

Select Value  
Humidity

Refresh Mode  
Sampling Interval 1 seconds

Save Cancel

## Widget Settings

Widget      Donut Chart      Display Options

**Units** %

**Min Value** 0

**Max Value** 100

**Donut Color** #666666

Save Cancel



12. Lalu, Tambahkan kembali Widget ke – 6 dengan ketentuan seperti gambar di bawah:

### Widget Settings

Widget      Time Series Chart      Display Options

Title ⓘ Chart DHT 11

Subtitle ⓘ Chart Suhu dan Humidity Green House

Background ⓘ  #ffffff +

Type ⓘ Time Series Chart ▼

Save Cancel

### Widget Settings

Widget      Time Series Chart      Display Options

Chart Input ⓘ From Device Resource ▼

    Select Device  
    MCUNR ▼

    Select Resource  
    DHT11 ▼

    Select Fields to Plot  
    Humidity X Suhu X

    Refresh Mode  
    Sampling Interval ▼ 5 seconds ▼

    Time Period ⓘ 60 seconds ▼

Save Cancel



## Widget Settings

Widget   Time Series Chart   Display Options

Options  Axis  Legend  Fill Splines  Multiple Axes

Serie Color  #2377af Humidity

Serie Color  #c93131 Suhu

Save Cancel

13. Lalu, Tambahkan kembali Widget ke – 7 dengan ketentuan seperti gambar di bawah:

## Widget Settings

Widget   Gauge   Display Options

Title  Ketinggian Tangki Air

Subtitle  Widget Subtitle

Background  #ffffff +

Type

Save Cancel

## Widget Settings

Widget   Gauge   Display Options

Widget Value

Select Device MCUNR

Select Resource HCSR04

Select Value Jarak

Refresh Mode Sampling Interval  seconds ▼

Save Cancel



## Widget Settings

Widget      Gauge      Display Options

Units

Min Value

Max Value

Text Color  #666666

Gradient

Save      Cancel

14. Lalu, Tambahkan kembali Widget ke – 8 dengan ketentuan seperti gambar di bawah:

## Widget Settings

Widget      Progressbar      Display Options

Title

Subtitle

Background  #ffffff

Type

Save      Cancel



## Widget Settings

Widget      Progressbar      Display Options

Widget Value ⓘ      From Device Resource

Select Device      MCUNR

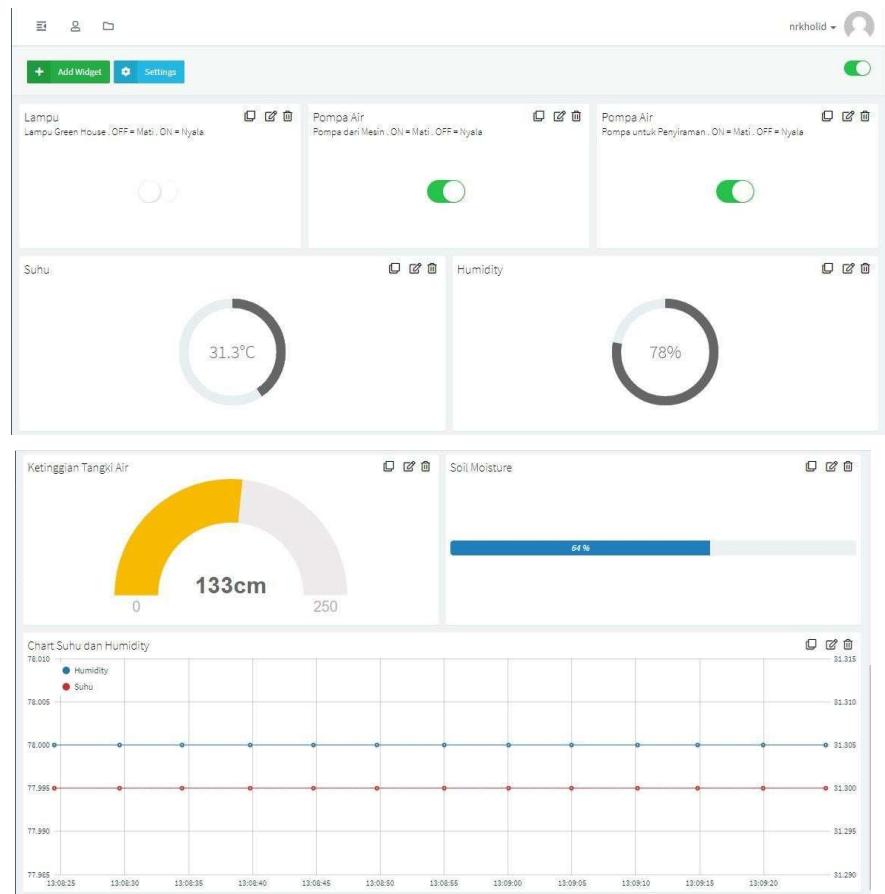
Select Resource      Soil\_Moisture

Select Value      Moisture

Refresh Mode:      Sampling Interval: 1 seconds

Save      Cancel

15. Setelah itu, atur Setiap Widget yang telah dibuat hingga memiliki tampilan seperti gambar di bawah:





## PROJECT V

### MEMBUAT SMART FARM DENGAN MENGGUNAKAN NODEMCU DAN UBIDOTS

#### - DESKRIPSI :

Project ini bertujuan untuk membuat sebuah *Smart Farm* untuk Kandang Sapi atau Ayam yang dapat dikendalikan dan dimonitoring melalui Platform Ubidots dan juga *Board NodeMCU* yang saling terhubung.

#### - BACAAN SINGKAT :

##### A. *Internet of Things*

*Internet of Things (IoT)* merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari koneksi internet yang tersambung secara terus-menerus. *Internet of Things* digunakan untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara *independent*. Salah satu contohnya adalah *smarthome* yang dapat memantau dan mengontrol kondisi rumah melalui *smartphone* dengan bantuan koneksi internet.

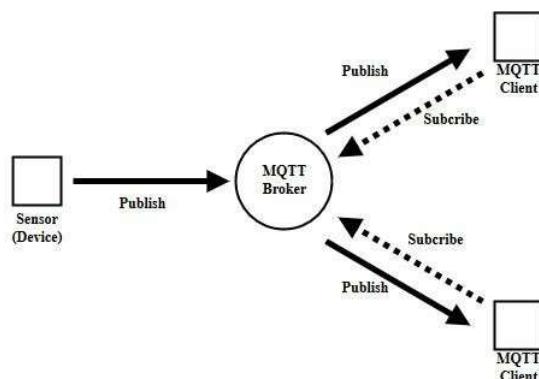
Pada dasarnya, perangkat *Internet of Things* terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komunikasi dan *server* sebagai pengumpul informasi yang diterima dari *sensor* dan dianalisa. Gambar di bawah menunjukkan gambaran cara kerja *Internet of Things* yang terdiri dari perangkat fisik, jaringan internet, dan *cloud database / server*.





### B. Message Queuing Telemetry Transport

Protokol MQTT (*Message Queuing Telemetry Transport*) merupakan protokol pesan yang sangat sederhana dan juga ringan. Protokol MQTT menggunakan arsitektur *publish* dan *subscribe* yang dirancang secara terbuka dan mudah untuk diimplementasikan. Selain itu, MQTT mampu menangani ribuan pengguna jarak jauh dengan hanya menggunakan satu *server*. MQTT menggunakan *bandwidth* jaringan yang sedikit, selain itu juga menggunakan sumber daya perangkat yang sedikit pula. Pada protokol ini berbasis M2M (*Machine to Machine*) yang artinya dapat berkomunikasi antar mesin atau perangkat, bisa dua perangkat atau lebih [8]. Arsitektur dari protokol MQTT dapat dilihat pada Gambar 2.2.



Gambar 2.2 Arsitektur Protokol MQTT

Pada gambar di atas, terlihat terdiri dari 2 komponen utama, yaitu *MQTT client* dan *MQTT broker*. *MQTT client* merupakan protokol yang berada pada perangkat, seperti halnya Arduino. *MQTT broker* berfungsi untuk menangani *publish* dan juga *subscribe* data.

### C. Ubidots



Ubidots adalah salah satu Platform IoT yang memberikan kemudahan dalam membuat sebuah layanan yang menggunakan konsep IoT. Berbagai fitur yang dimiliki oleh Ubidots memberikan kemudahan bagi pengguna untuk bisa membuat sebuah layanan IoT salah satunya adalah *Cloud Storage* yang memberikan



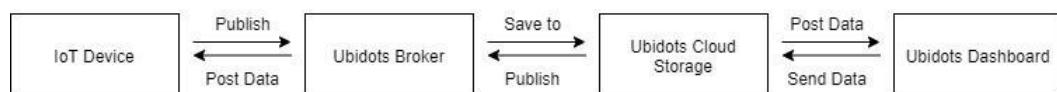
kemudahan dalam penyimpanan data IoT dan juga *Dashboard* yang memberikan kemudahan dalam membuat sebuah UI untuk *monitoring* atau *controlling* sebuah *IoT Device*.

#### D. MQ-135



Sensor MQ-135 adalah salah satu keluarga dari Sensor MQ yang memiliki kegunaan untuk mendeteksi kandungan CO, CO<sub>2</sub>, NH<sub>4</sub>, Alkohol, dan Bensol. Sensor ini memiliki 4 Pin, diantaranya Pin Analog Output, Digital Output, VCC, dan juga Gnd. Sensor ini bekerja optimal pada tegangan 5V dan memerlukan kalibrasi untuk memulai perhitungannya.

#### E. Cara Kerja



Cara kerja dari sistem ini adalah *IoT Device* mengambil Data melalui sensor yang digunakan pada sistem, kemudian Data tersebut dikirimkan ke MQTT *Broker* yang pada sistem ini menggunakan Ubidots Broker. Selanjutnya, Data tersebut akan disimpan pada *Ubidots Cloud Storage* dan kemudian Data tersebut akan ditampilkan pada Dashboard yang telah dibuat pada Web Ubidots.

- KOMPONEN :

1. Board NodeMCU V3
2. Kabel Jumper
3. Breadboard / Expansion Board



4. Kabel USB / Data
5. DHT 11
6. HCSR-04
7. Relay 2 Channel
8. LED
9. Motor Servo
10. Sensor MQ-135
11. Ubidots IoT Platform
12. Web Browser
13. Arduino IDE

- **LANGKAH-LANGKAH**

**A. Konfigurasi Device pada Ubidots**

1. Buka halaman **Ubidots** pada Browser : <https://ubidots.com/>
2. Setelah itu, pilih **Sign Up** apabila kalian belum memiliki akun

3. Lalu pilih **For Educational or Personal Use**

Whether you're prototyping or going to market, Ubidots is designed to accelerate your IoT initiatives.

**For Educational or Personal Use**

Join 60,000+ students, makers and professors using our FREE Ubidots STEM platform to prototype, learn, or teach IoT.

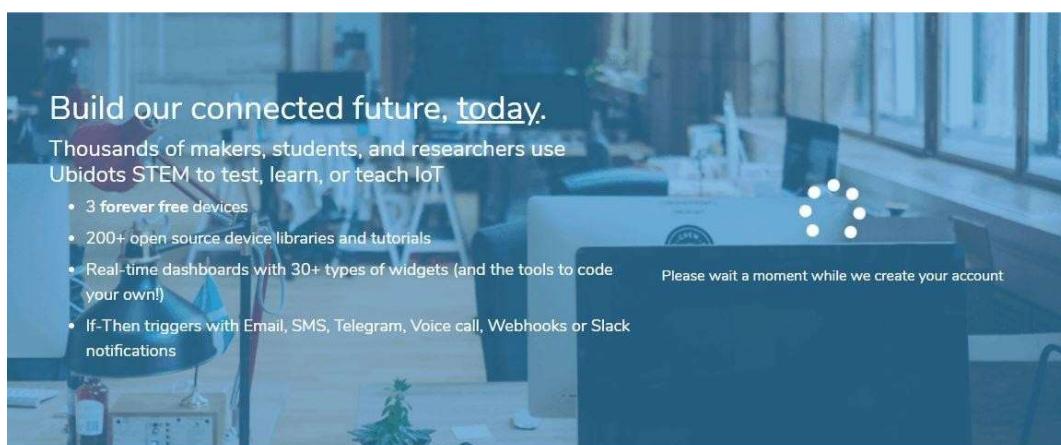
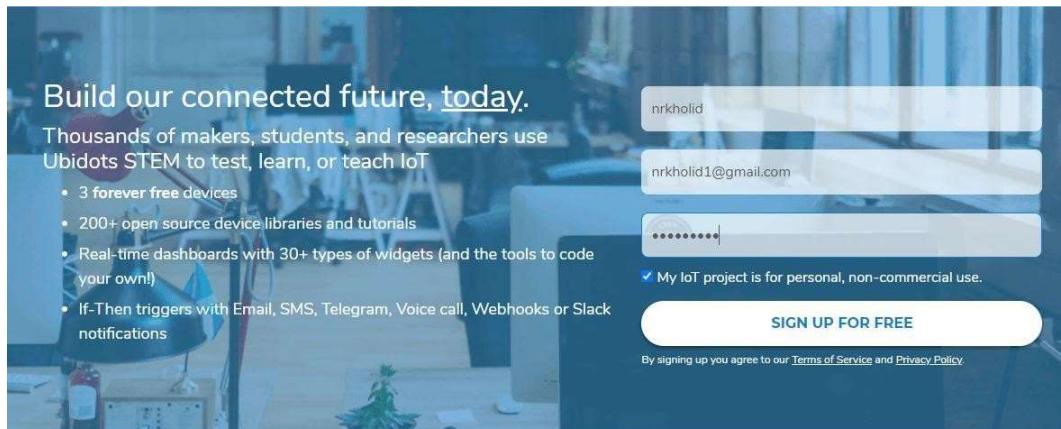
**For Business**

Join 1,000+ System Integrators, OEMs and IoT Entrepreneurs building connected products and services with Ubidots.

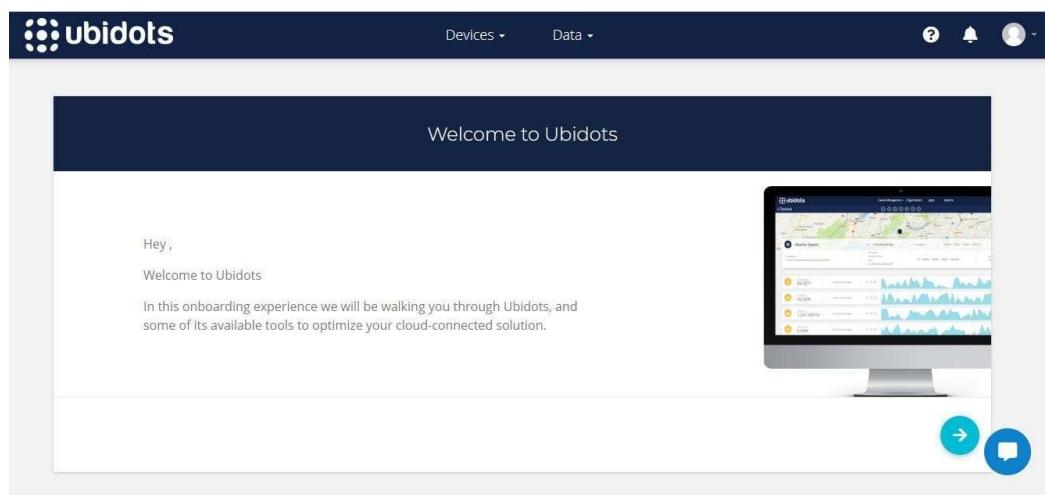
[TAKE ME TO UBIDOTS STEM](#)      [CONTINUE →](#)



- Setelah isi kolom **Username**, **Email**, dan juga **Password** untuk Registrasi seperti gambar di bawah dan klik **Sign Up for Free** dan tunggu beberapa saat



- Kemudian kalian akan tiba di halaman awal dan kalian dapat meng-klik tombol **Next** di pojok kiri bawah seperti gambar di bawah





6. Selanjutnya, kalian akan tiba di halaman **Dashboard** dan kemudian klik Tab **Devices** dan pilih **Device**

7. Lalu kalian akan tiba di **Halaman Device**, kemudian buatlah sebuah **Device** dengan klik Tombol (+) di pojok kanan atas seperti gambar di bawah

8. Lalu pilih **Blank Device**



9. Dan selanjutnya isi **Name Device** seperti gambar di bawah dan klik Tombol Centang di bagian sudut pojok kanan bawah

Device name  
Smart Farm

Device label  
smart-farm

10. Dan **Device** pun berhasil dibuat seperti gambar di bawah. Kemudian, klik **Device** yang telah kalian buat

| NAME       | LAST ACTIVITY    | CREATED AT                 | ACTIONS |
|------------|------------------|----------------------------|---------|
| Smart Farm | No last activity | 2020-06-23 12:53:23 +07:00 |         |

11. Selanjutnya, isi bagian **Deskripsi** pada **Device** seperti gambar di bawah dan **Ingatlah API Label dan Token** dari **Device** yang dibuat karena akan digunakan pada Program di Arduino IDE



The screenshot shows a device configuration page on the Ubidots platform. The device name is "Smart Farm". The "Description" field contains "Smart Cow / Chicken Farm". The "API Label" field is set to "smart-farm", which is highlighted with a red box. The "ID" field shows the unique identifier "5ef198d31d84722b9db27173". The "Token" field displays a long string of characters starting with "BBFF-tCGUs5jrHw1pllk...", also highlighted with a red box. The "Tags" section has a placeholder "Add new tag". The "Last activity" section indicates "No last activity".

12. Selanjutnya tambahkan **8 Variabel RAW** dengan mengikuti cara seperti gambar di bawah





The screenshot shows the Ubidots platform's variable creation interface. At the top right, there are two options: "Raw" with a bar chart icon and "Synthetic" with a summation symbol icon. Below these is a large button labeled "Add Variable". The main area contains a grid of nine variable cards, each with a yellow header and a white body. Each card has a cloud icon and the text "... New Variable". Below this, it says "Last activity: No last activity". To the right of the grid is another "Add Variable" button.

13. Selanjutnya, klik **setiap Variabel** yang telah dibuat dan isilah **Nama, Deskripsi, API Label, Allowed Range, dan Unit** dari **setiap Variabel** dengan mengikuti cara seperti gambar di bawah



**LED**

Description: Lampu

API Label: led

ID: 5ef199891d84722b9b02d756

Allowed range: From: 0 to: 1

Unit: Add unit

Tags: Add new tag

Last activity: No last activity

**Pompa**

Description: Pompa Tangki

API Label: relay1

ID: 5ef199881d84722b1e974b58

Allowed range: From: 0 to: 1

Unit: Add unit

Tags: Add new tag

Last activity: No last activity

**Water**

Description: Minum untuk Hewan

API Label: relay2

ID: 5ef199881d84722cc2a80e2e

Allowed range: From: 0 to: 1

Unit: Add unit

Tags: Add new tag

Last activity: No last activity

**Pakan**

Description: Pakan Hewan

API Label: servo

ID: 5ef1996e1d84722cc2a80e2c

Allowed range: From: 0 to: 1

Unit: Add unit

Tags: Add new tag

Last activity: No last activity



|  |   |
|--|---|
| <b>Amonia</b>                          | <b>Ketinggian Air</b>                   |
| Description<br>Gas Amonia pada Kandang | Description<br>Ketinggian di Tangki Air |
| API Label<br>amonia                    | API Label<br>ketinggian                 |
| ID<br>5ef1997e1d84722b9db27178         | ID<br>5ef199871d84722cc2a80e2d          |
| Allowed range<br>From: 0 to: 100       | Allowed range<br>From: 0 to: 250        |
| Unit<br>PPM                            | Unit<br>cm                              |
| Tags<br>Add new tag                    | Tags<br>Add new tag                     |
| Last activity<br>No last activity      | Last activity<br>No last activity       |

|                                   |  |
|-----------------------------------|--|
| <b>Suhu</b>                       | <b>Kelembaban</b>                      |
| Description<br>Suhu pada Kandang  | Description<br>Kelembaban pada Kandang |
| API Label<br>suhu                 | API Label<br>kelembaban                |
| ID<br>5ef199801d84722c79eb0567    | ID<br>5ef199841d84722c233a81bb         |
| Allowed range<br>From: 5 to: 120  | Allowed range<br>From: 0 to: 100       |
| Unit<br>°C                        | Unit<br>%                              |
| Tags<br>Add new tag               | Tags<br>Add new tag                    |
| Last activity<br>No last activity | Last activity<br>No last activity      |



14. Setelah selesai, maka hasilnya akan seperti gambar di bawah:

The screenshot shows a grid of nine device cards in the Ubidots Device dashboard. Each card has a yellow header and a white body. The cards are arranged in three rows of three. The first row contains '... PPM Amonia', '... % Kelembaban', and '... cm Ketinggian Air'. The second row contains '... LED', '... Pakan', and '... Pompa'. The third row contains '... °C Suhu' and '... Water'. Below each card, it says 'Last activity: No last activity'. To the right of the grid is a dashed box containing a '+' button and the text 'Add Variable'.

## B. Konfigurasi Dashboard pada Ubidots

1. Setelah selesai dengan Konfigurasi pada Device, selanjutnya, klik Tab Data lalu pilih Dashboard.

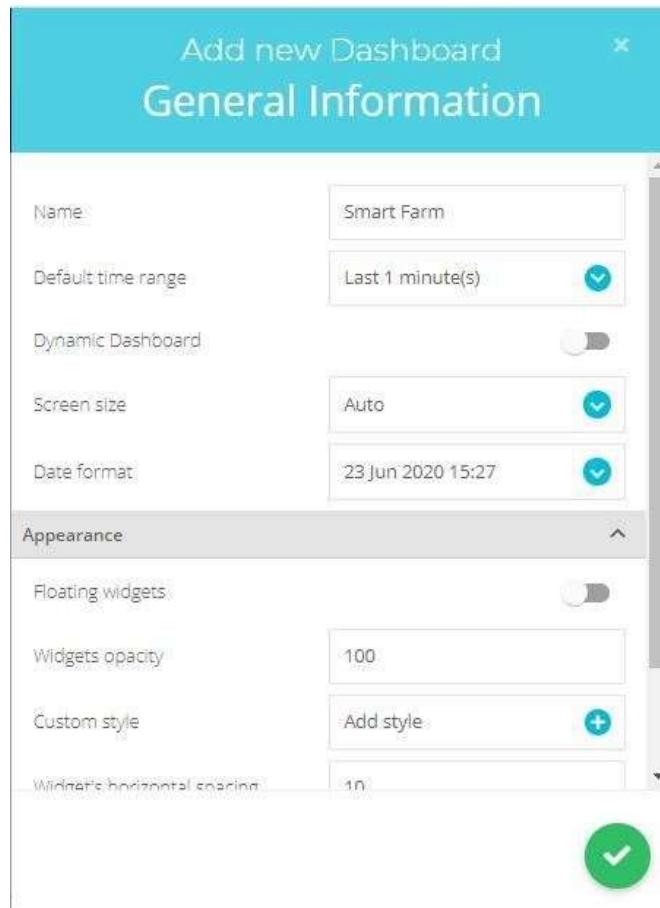
The screenshot shows the Ubidots main dashboard. At the top, there is a navigation bar with tabs for 'Devices', 'Data', and 'Dashboards'. The 'Dashboards' tab is currently selected, indicated by a dropdown menu showing 'Dashboards', 'Events', and 'Analytics (BETA)'. Below the navigation bar is a world map. On the right side, there is a 'SET LOCATION' button and a 'SET' icon. The main area of the dashboard displays a message: 'No Dashboards created yet' and 'Create Dashboards to visualize your data in realtime'.

2. Lalu, kalian akan tiba di Halaman Dashboard seperti gambar di bawah

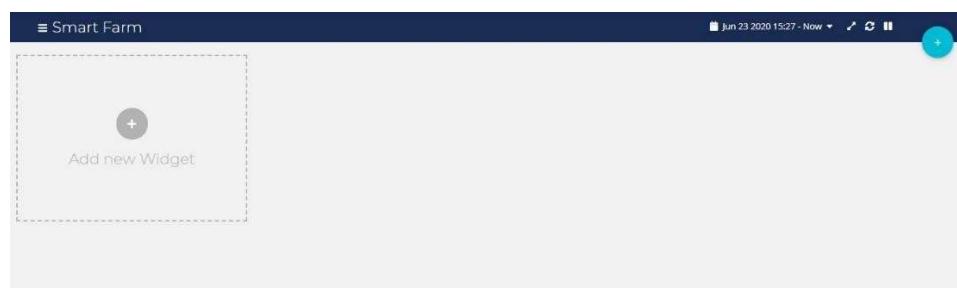
The screenshot shows the Ubidots dashboard. The top navigation bar is visible with the 'Dashboards' tab selected. The main area displays the message 'No Dashboards created yet' and 'Create Dashboards to visualize your data in realtime'. At the bottom of the main area is a blue button labeled 'Add new Dashboard'.



3. Kemudian, klik **Add New Dashboard** untuk membuat **Dashboard baru**
4. Setelah itu, isi kolom isian pada Pembuatan **Dashboard** dengan mengikuti gambar di bawah dan klik **Tombol Centang berwarna Hijau** apabila telah selesai



5. Setelah berhasil dibuat, maka tampilannya akan seperti gambar di bawah



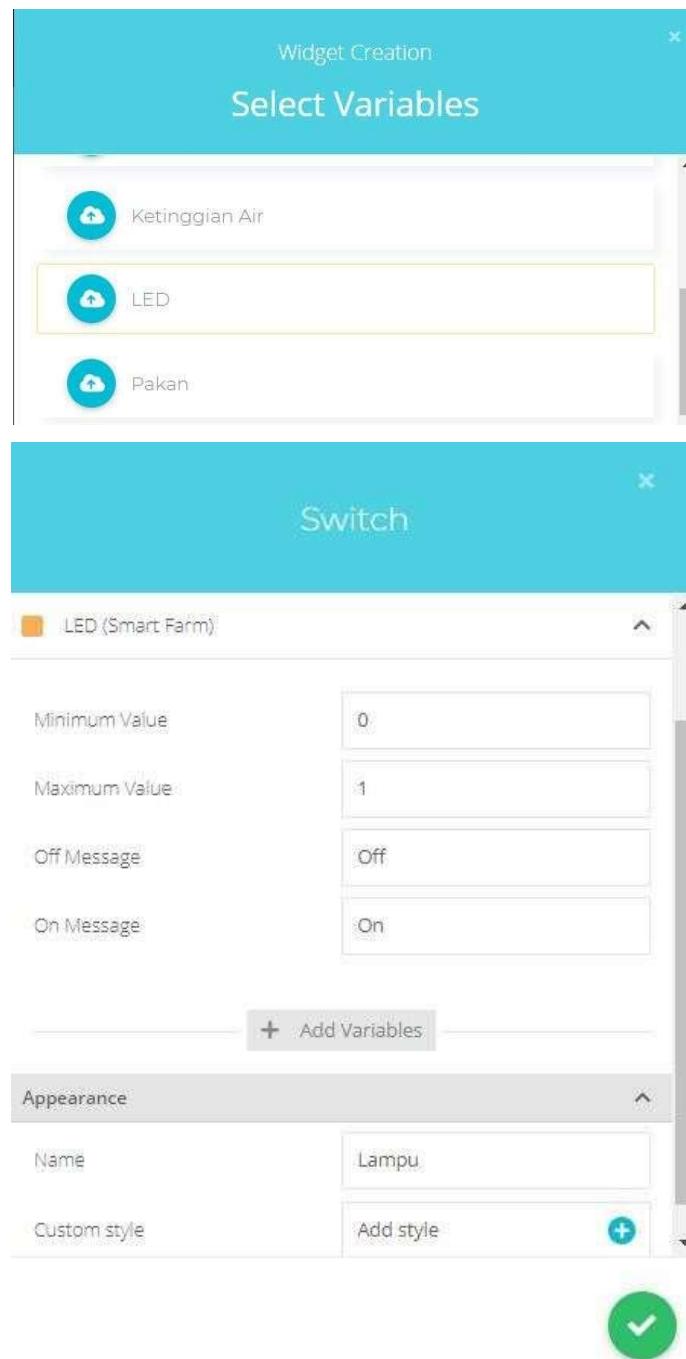
6. Kemudian tambahkan **Widget** yang dibutuhkan dengan klik **Add New Widget** atau **Tombol (+)** pada bagian pojok sudut kanan atas



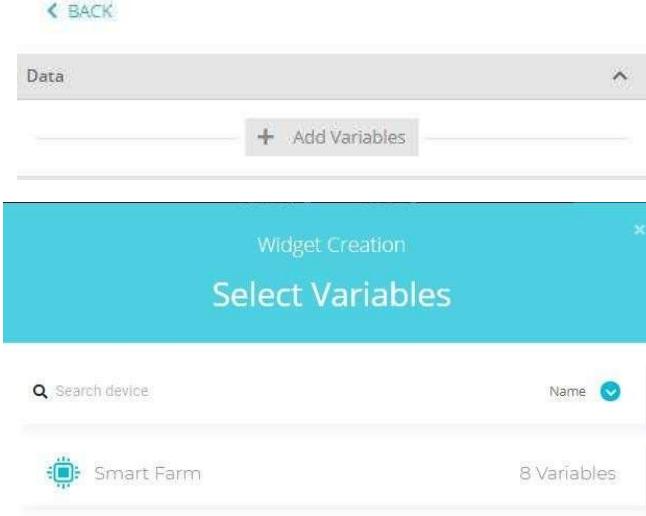
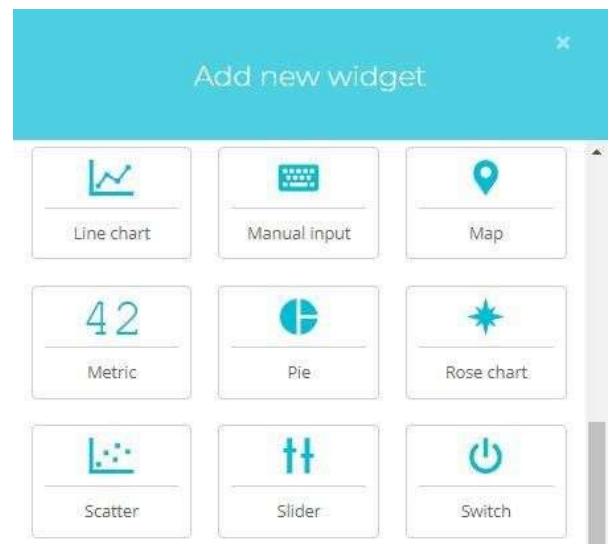
7. Lalu, pilih **Widget Switch** kemudian pilih **Add Variables** dan pilih **Smart Farm** kemudian pilih **Variabel LED** dan ikuti ketentuannya seperti pada gambar dan kemudian klik **Tombol Centang Hijau** ketika sudah selesai

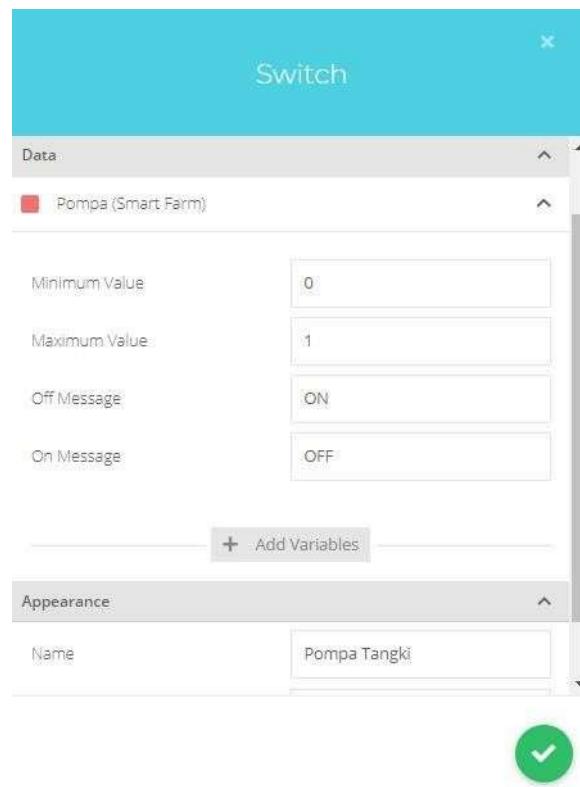
The image consists of three vertically stacked screenshots from the Ubidots platform:

- Screenshot 1:** A modal window titled "Add new widget". It displays nine different widget options in a 3x3 grid:
  - Line chart
  - Manual input
  - Map
  - Metric
  - Pie
  - Rose chart
  - Scatter
  - Slider
  - Switch
- Screenshot 2:** A modal window titled "Switch". This is the selected widget type.
- Screenshot 3:** A modal window titled "Widget Creation" with the sub-section "Select Variables". It includes a search bar labeled "Search device", a dropdown menu set to "Name", and a list item "Smart Farm" with "8 Variables" next to it.

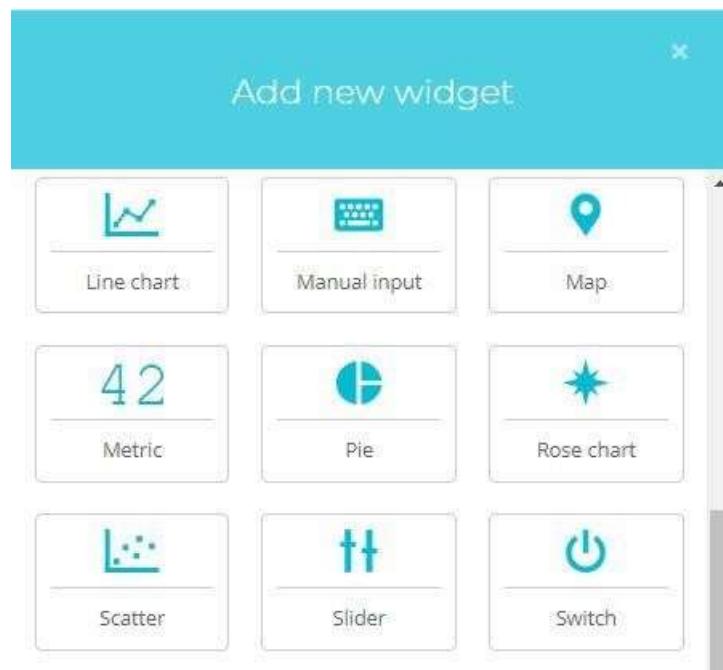


8. Lalu, pilih **Widget Switch** ke - 2 kemudian pilih **Add Variables** dan pilih **Smart Farm** kemudian pilih **Variabel Pompa** dan ikuti ketentuannya seperti pada gambar dan kemudian klik **Tombol Centang Hijau** ketika sudah selesai





9. Lalu, pilih **Widget Switch** ke - 3 kemudian pilih **Add Variables** dan pilih **Smart Farm** kemudian pilih **Variabel Water** dan ikuti ketentuannya seperti pada gambar dan kemudian klik **Tombol Centang Hijau** ketika sudah selesai





The screenshot shows the Ubidots Platform interface for creating a 'Widget Creation'.

**Widget Creation - Select Variables:**

- Device: Smart Farm
- Variables: 8 Variables

**Widget Creation - Select Variables (Modal):**

- Pompa
- Suhu
- Water (Selected)

**Switch Widget Configuration:**

- Variable: Water (Smart Farm)
- Minimum Value: 0
- Maximum Value: 1
- Off Message: ON
- On Message: OFF

**Appearance:**

- Name: Watering
- Custom style: Add style

A green checkmark icon is visible in the bottom right corner of the configuration area.

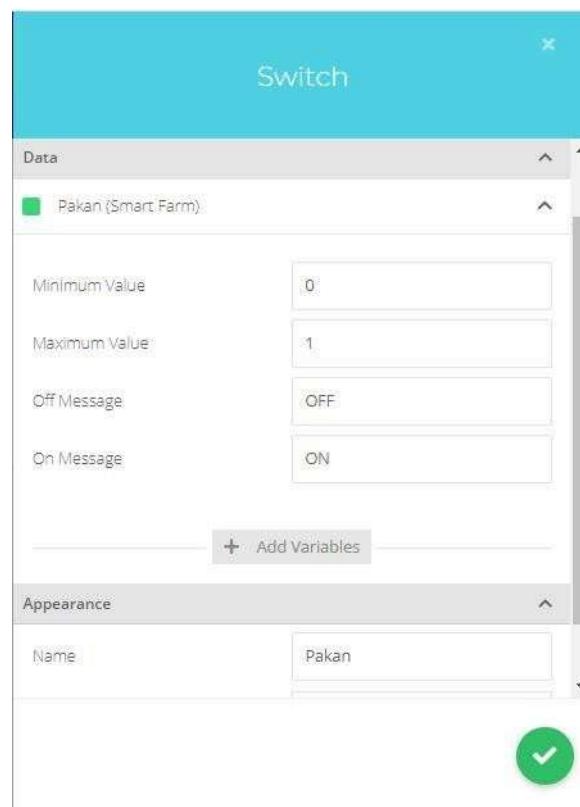


10. Lalu, pilih **Widget Switch ke -4** kemudian pilih **Add Variables** dan pilih **Smart Farm** kemudian pilih **Variabel Pakan** dan ikuti ketentuannya seperti pada gambar dan kemudian klik **Tombol Centang Hijau** ketika sudah selesai

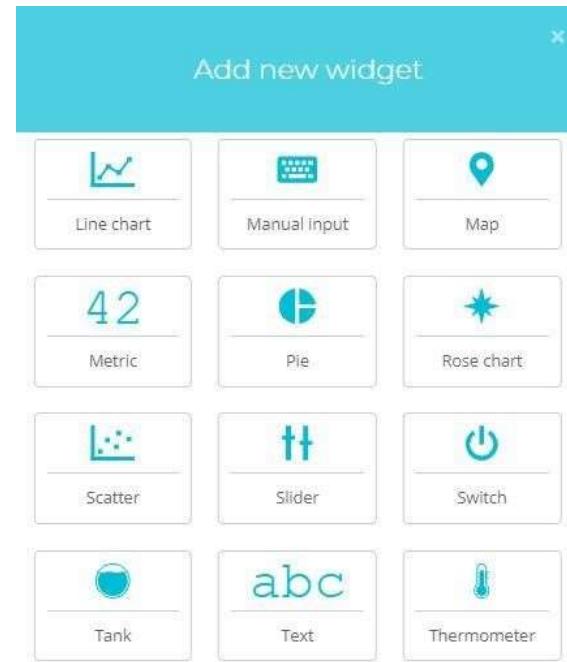
The first screenshot shows the "Add new widget" dialog with a "Switch" icon selected from a grid of nine options: Line chart, Manual input, Map, Metric, Pie, Rose chart, Scatter, Slider, and Switch.

The second screenshot shows the "Widget Creation" step for a "Switch" widget. It displays a list of variables under the heading "Select Variables". The variable "Pakan" is highlighted with a yellow border, indicating it has been selected. Other variables listed are Ketinggian Air, LED, and Pompa.

The third screenshot shows the "Widget Creation" step again, with the "Select Variables" dialog open. The variable "Pakan" is now highlighted with a yellow border, indicating it has been selected for the switch widget.



11. Lalu, pilih **Widget Thermometer** kemudian pilih **Add Variables** dan pilih **Smart Farm** kemudian pilih **Variabel Suhu** dan ikuti ketentuannya seperti pada gambar dan kemudian klik **Tombol Centang Hijau** ketika sudah selesai





### Thermometer

BACK

Data

Add Variables

### Widget Creation

#### Select Variables

Search device

Name: Smart Farm

8 Variables

Pompa

Suhu

Water

### Widget Creation

#### Select Variables

Pompa

Suhu

Water

✓

### Thermometer

Suhu (Smart Farm)

Aggregation method: Last value

Span: All time

Appearance

Name: Suhu

Font family: Open Sans

Decimal points: Auto

Date format: 23 Jun 2020 17:20

Range value: 0 to 150

✓

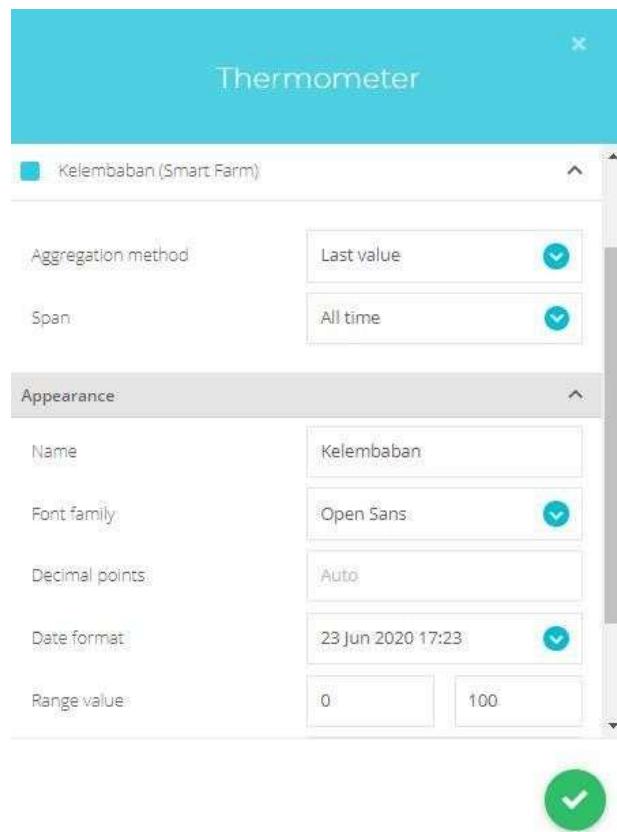


12. Lalu, pilih **Widget Thermometer** ke - 2 kemudian pilih **Add Variables** dan pilih **Smart Farm** kemudian pilih **Variabel Kelembaban** dan ikuti ketentuannya seperti pada gambar dan kemudian klik **Tombol Centang Hijau** ketika sudah selesai

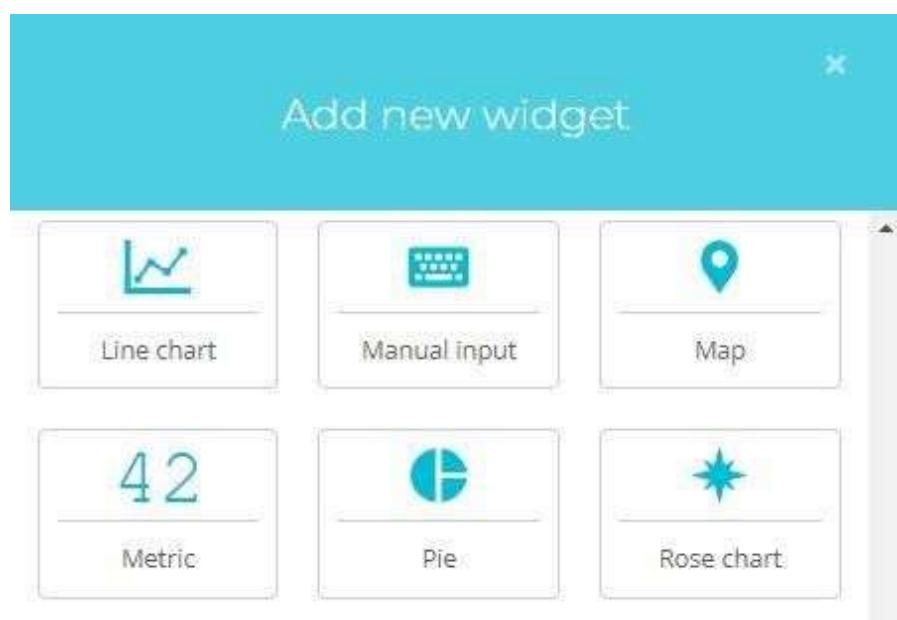
The first screenshot shows the "Add new widget" dialog with various options like Line chart, Manual input, Map, Metric, Pie, Rose chart, Scatter, Slider, Switch, Tank, Text, and Thermometer. The Thermometer option is selected.

The second screenshot shows the "Thermometer" creation screen with a "Data" tab and an "Add Variables" button. It also shows a "Widget Creation Select Variables" modal for "Smart Farm" with 8 variables listed: Amonia, Kelembaban, Ketinggian Air, and LED.

The third screenshot shows a detailed view of the "Widget Creation Select Variables" modal for "Smart Farm" with 8 variables listed: Amonia, Kelembaban, Ketinggian Air, and LED. The "Kelembaban" variable is highlighted with a yellow border.



13. Lalu, pilih **Widget Metric** kemudian pilih **Add Variables** dan pilih **Smart Farm** kemudian pilih **Variabel Amonia** dan ikuti ketentuannya seperti pada gambar dan kemudian klik **Tombol Centang Hijau** ketika sudah selesai





### Metric

BACK

Data

Add Variables

### Widget Creation

#### Select Variables

Search device Name

Smart Farm 8 Variables

### Widget Creation

#### Select Variables

Search device Name

Smart Farm 8 Variables

Amonia

Kelembaban

✓

### Metric

Data

Amonia (Smart Farm)

Aggregation method: Last value

Span: All time

Appearance

Name: Amonia

Use the HTML editor

Show last updated info

Font family: Open Sans

Decimal points: Auto

✓

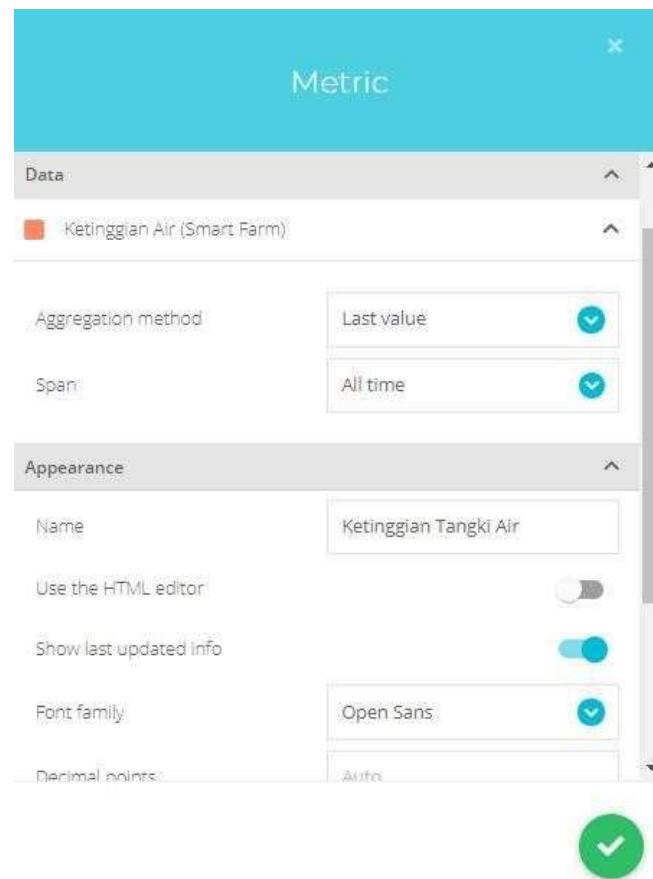


14. Lalu, pilih **Widget Metric** ke - 2 kemudian pilih **Add Variables** dan pilih **Smart Farm** kemudian pilih **Variabel Ketinggian** dan ikuti ketentuannya seperti pada gambar dan kemudian klik **Tombol Centang Hijau** ketika sudah selesai

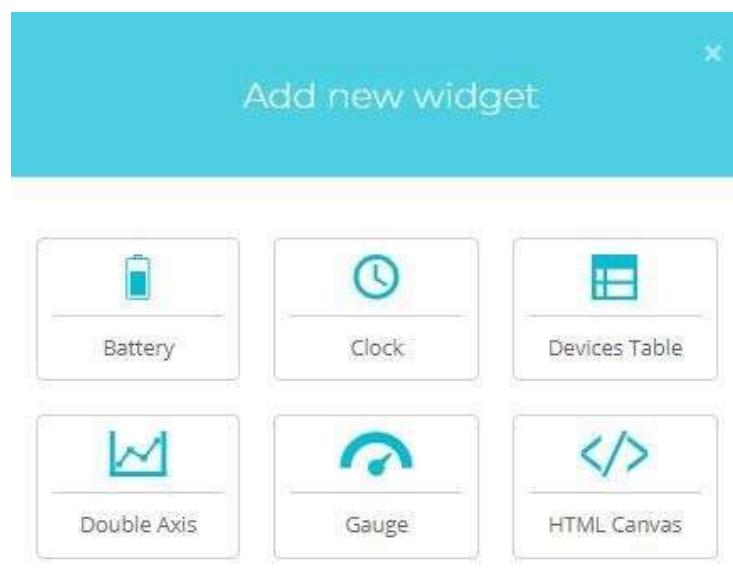
The first screenshot shows the 'Add new widget' interface with various options like Histogram, Image, Indicator, Line chart, Manual input, Map, Metric, Pie, and Rose chart. The 'Metric' option is selected.

The second screenshot shows the 'Metric' creation screen with a 'Data' tab and an 'Add Variables' button. To its right is the 'Widget Creation - Select Variables' dialog, which lists 'Smart Farm' with 8 variables, including 'Ketinggian Air' which is currently selected.

The third screenshot shows the 'Widget Creation - Select Variables' dialog again, but now 'Ketinggian Air' is highlighted with a yellow border. A green checkmark icon is visible in the bottom right corner of this dialog.



- 
15. Lalu, pilih **Widget Double Axis** kemudian pilih **Add Variables** dan pilih **Smart Farm** kemudian pilih **Variabel Suhu dan Kelembaban** dan ikuti ketentuannya seperti pada gambar dan kemudian klik **Tombol Centang Hijau** ketika sudah selesai





The screenshots illustrate the process of selecting variables for a Double Axis widget in the Ubidots platform.

**Screenshot 1:** The first step shows the "Double Axis" configuration screen. It includes a "Data" section with a "Add Variables" button and a "Widget Creation" section titled "Select Variables". A search bar labeled "Search device" and a dropdown menu labeled "Name" are also present.

**Screenshot 2:** The second step shows the "Widget Creation" screen with the "Select Variables" title. It lists variables under the "Smart Farm" device, with "Suhu" selected and highlighted with a yellow border. Other variables shown include "Pompa", "Water", and "Amonia".

**Screenshot 3:** The third step shows the "Widget Creation" screen with the "Select Variables" title. It lists variables under the "Smart Farm" device, with "Kelembaban" selected and highlighted with a yellow border. Other variables shown include "Amonia", "Ketinggian Air", and "LED".



**Double Axis**

**Data**

- Suhu (Smart Farm)

Aggregation method: Last value

Span: Last values

Display the last: 500 Value(s)

Type: Line

Y-Axis: Right Y-Axis

**Kelembaban (Smart Farm)**

Aggregation method: Last value

Span: Last values

Display the last: 500 Value(s)

Type: Line

Y-Axis: Left Y-Axis

+ Add Variables

**Appearance**

Name: Grafik Suhu dan Kelembaban

Decimal points: Auto

Show legend: Off

Date format: 23Jun 2020 17:37

Display X-Axis data zoom: On

X-Axis label: None

Custom style: Add style

**Double Axis**

**Left Y-Axis**

Y-axis name: Left Y-Axis

Position: Left

Y-axis range: 0 100

Use SI prefix: Off

**Right Y-Axis**

Y-axis name: Right Y-Axis

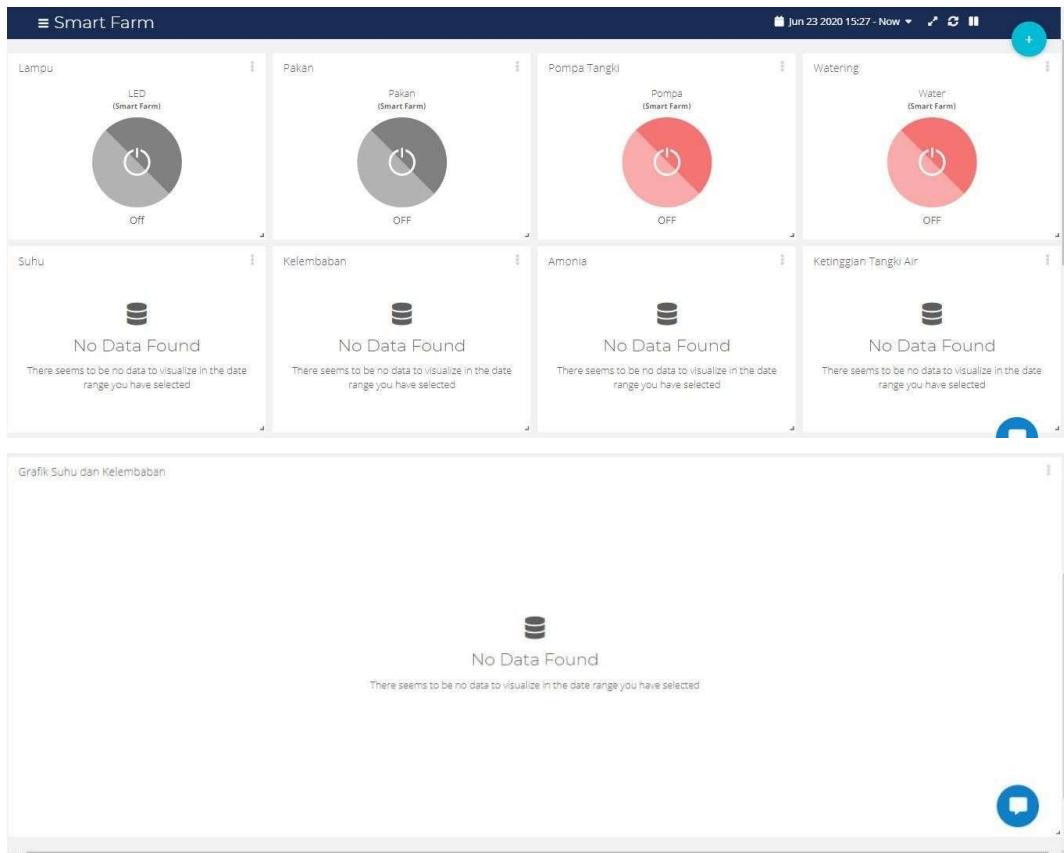
Position: Right

Y-axis range: 0 150

✓



16. Kemudian atur **Dashboard** dan **Widget** yang telah dibuat hingga memiliki tampilan seperti gambar di bawah:



### C. Konfigurasi Program pada NodeMCU di Arduino IDE

1. Hubungkan DHT11, HCSR-04, MQ-135, LED, Motor Servo dan Relay pada Board NodeMCU seperti ketentuan dibawah:

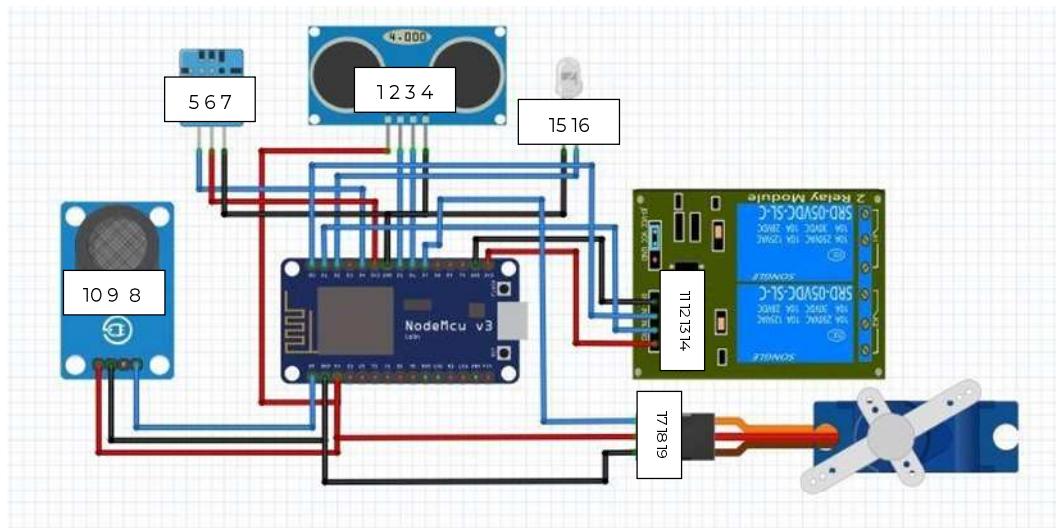
| No | Pin HC-SR04 | Pin NodeMCU |
|----|-------------|-------------|
| 1  | VCC         | V USB       |
| 2  | Trigger     | D5          |
| 3  | Echo        | D6          |
| 4  | GND         | GND         |

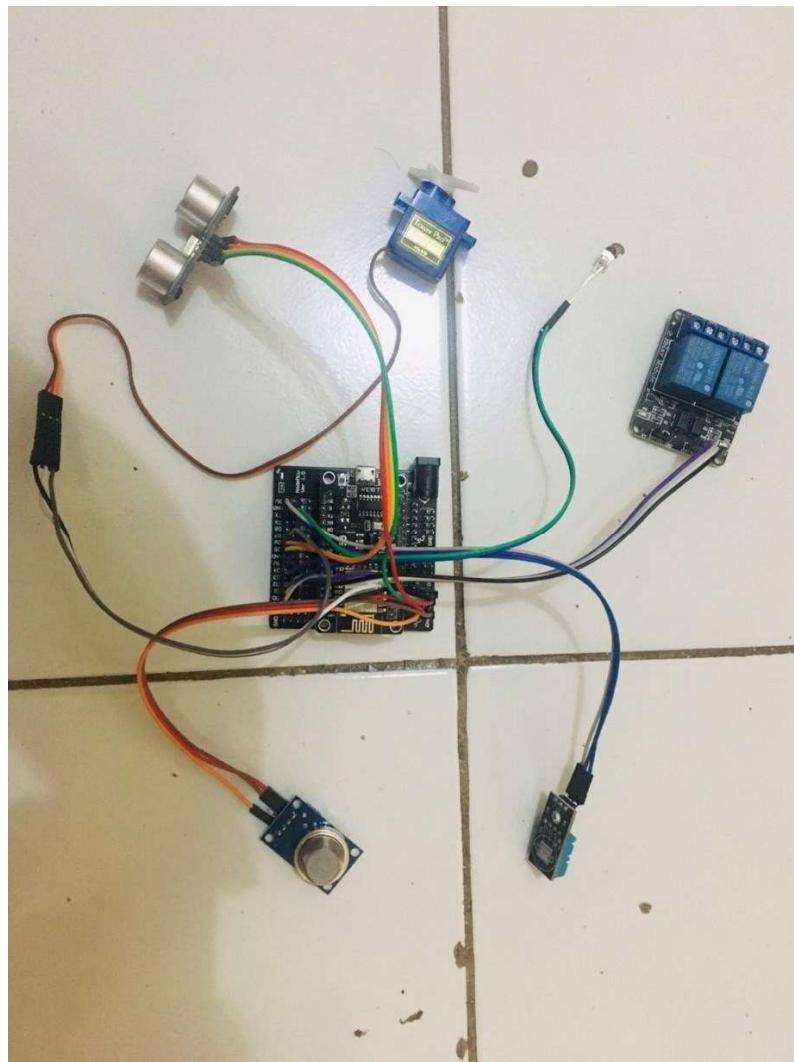
  

| No | Pin DHT11 | Pin NodeMCU |
|----|-----------|-------------|
| 5  | Pin Data  | D4          |
| 6  | VCC DHT11 | 3.3V        |
| 7  | GND DHT11 | GND         |

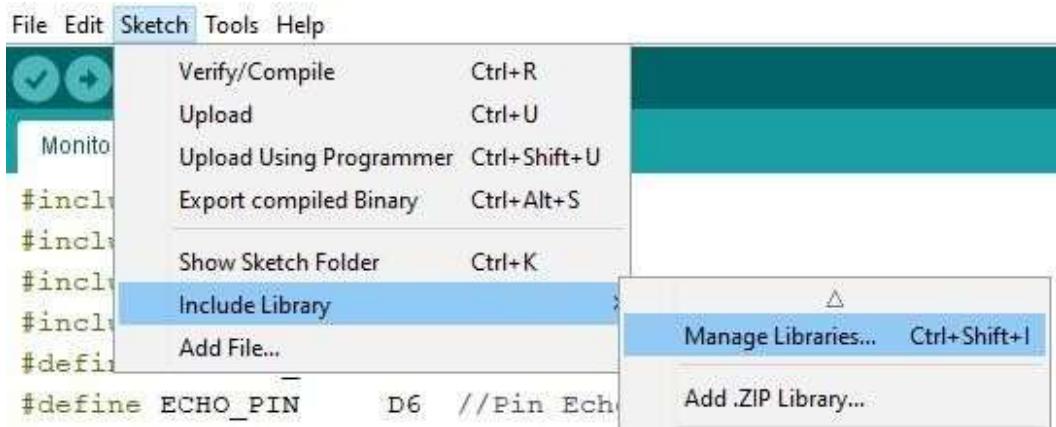


| No | Pin MQ-135      | Pin NodeMCU |
|----|-----------------|-------------|
| 8  | Pin Data        | A0          |
| 9  | GND MQ-135      | Gnd         |
| 10 | VCC MQ-135      | 3.3V        |
| No | Pin Relay       | Pin NodeMCU |
| 11 | GND Relay       | Gnd         |
| 12 | Pin IN1         | D0          |
| 13 | Pin IN2         | D1          |
| 14 | VCC Relay       | 3.3V        |
| No | Pin LED         | Pin NodeMCU |
| 15 | Kaki Pendek     | Gnd         |
| 16 | Kaki Panjang    | D2          |
| No | Pin Motor Servo | Pin NodeMCU |
| 17 | Pin Data        | D7          |
| 18 | VCC             | 5V / V USB  |
| 19 | GND             | Gnd         |





2. Buka Arduino IDE, kemudian install Library yang diperlukan yaitu **DHT**, **NewPing**, **Servo**, **MQUnifiedSensor**, **PubSubClient** dan **Ubidots MQTT for ESP8266** versi terbaru seperti gambar di bawah:





**NewPing**  
by Tim Eckel Version 1.9.1 **INSTALLED**  
**A library that makes working with ultrasonic sensors easy.** When I first received an ultrasonic sensor I was not happy with how poorly it performed. I soon realized the problem was not the sensor, it was the available ping and ultrasonic libraries causing the problem. The NewPing library totally fixes these problems, adds many new features, and breathes new life into these very affordable distance sensors.  
[More info](#)

**DHT sensor library**  
by Adafruit Version 1.3.8 **INSTALLED**  
**Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors** Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors

**Ubidots**  
by Cristian Arrieta  
**Ubidots library for the GPRS module SIM900** Ubidots library for the GPRS module SIM900  
[More info](#)  
Version 4.0.1

**Ubidots MQTT for ESP8266**  
by Jose Garcia , Mateo Velez Version 1.1.0 **INSTALLED**  
**Library for sending data to the Ubidots cloud using ESP8266 based systems** Library for sending data to the Ubidots cloud using ESP8266 based systems  
[More info](#)

**pubsubclient**  
Modules for WLAN-Client, NTP, OTA, MQTT on ESP32/ESP8266 compatible with muwerk scheduler  
Modules for WLAN-Client, NTP, OTA, MQTT on ESP32/ESP8266 compatible with muwerk scheduler, requires libraries usdt, muwerk, Arduino\_JSON and PubSubClient.  
[More info](#)

**PubSubClient**  
by Nick O'Leary Version 2.8.0 **INSTALLED**  
**A client library for MQTT messaging.** MQTT is a lightweight messaging protocol ideal for small devices. This library allows you to send and receive MQTT messages. It supports the latest MQTT 3.1.1 protocol and can be configured to use the older MQTT 3.1 if needed. It supports all Arduino Ethernet Client compatible hardware, including the Intel Galileo/Edison, ESP8266 and TI CC3000.  
[More info](#)

**MQUnifiedsensor**  
by Miguel Califa , Yerson Carrillo, Ghiryd Contreras Version 2.0.1 **INSTALLED**  
**This library allows you to read the MQ sensors very easily.** This library allows an Arduino/Genuino/ESP8266 board to read MQ Sensors (Air quality meter) references: MQ2, MQ3, MQ4, MQ5, MQ6, MQ7, MQ8, MQ9, MQ131, MQ135, MQ303A, MQ309A.  
[More info](#)  
Select version

**Servo**  
Built-In by Michael Margolis, Arduino Version 1.1.6 **INSTALLED**  
**Allows Arduino/Genuino boards to control a variety of servo motors.** This library can control a great number of servos. It makes careful use of timers: the library can control 12 servos using only 1 timer. On the Arduino Due you can control up to 60 servos.  
[More info](#)



### 3. Kemudian, *copy* dan *paste source code* di bawah ke Arduino IDE

```
#include "UbidotsESPMQTT.h" //Library untuk Ubidots MQTT ESP
#include <NewPing.h> //Library untuk HC-SR04
#include <DHT.h> //Library untuk DHT11
#include <MQUnifiedsensor.h> // Library untuk MQ0135
#include <Servo.h> //Library untuk Servo

#define TRIGGER_PIN D5 //Pin Trigger HC-SR04 pada NodeMCU
#define ECHO_PIN      D6 //Pin Echo HC-SR04 pada NodeMCU
#define MAX_DISTANCE 250 //Maksimum Pembacaan Jarak (cm)
#define DHTPIN D4      //Define Pin DHT
#define DHTTYPE DHT11 //Define Jenis DHT
#define Pompa D0      //Define Pin Relay 1
#define Minum D1      //Define Pin Relay 2
#define LED D2        //Define Pin LED

#define placa "ESP8266" //Define jenis board yang digunakan
#define Voltage_Resolution 5 //Tegangan yang digunakan
#define pin A0 //Pin yang digunakan untuk MQ-135
#define type "MQ-135" //Jenis MQ yang digunakan
#define ADC_Bit_Resolution 10 //Resolusi Bit ADC
#define RatioMQ135CleanAir 3.6 //Nilai Udara dianggap bersih

#define TOKEN "BBFF-tCGUs5jrHw1pIilk5949NHLbQ6FGRu" //Token dari
Ubidots
#define WIFI_NAME "KILLUA" //SSID Wi-Fi
#define WIFI_PASS "nada124@" //Password Wi-Fi

MQUnifiedsensor MQ135(placa, Voltage_Resolution,
ADC_Bit_Resolution, pin, type); //Aktifkan fungsi MQ
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); //Aktifkan
Fungsi HCSR-04 //Aktifkan fungsi HC-SR04
DHT dht(DHTPIN, DHTTYPE); //Aktifkan Fungsi DHT
Servo servo; //Aktifkan fungsi Servo
Ubidots client(TOKEN); //Aktifkan fungsi Ubidots dengan Token

unsigned long durasiKirim = 0;
unsigned long jedaKirim = 2000;

//Untuk menerima perintah dari Dashboard Ubidots dengan isi Topik
dan Nilai Pesannya
void callback(char* topic, byte* payload, unsigned int length) {
//Mencetak Topik dan Nilai Pesan yang masuk dari Dashboard Ubidots
Serial.print("Pesan Diterima: [");
Serial.print(topic);
Serial.print("] ");
for (int i=0;i<length;i++) {
    Serial.print((char)payload[i]);
}

//Kondisi apabila Topik yang masuk LED dan Membaca Nilai Pesan
yang dikirimkan untuk menyalakan LED
if(strcmp(topic,"/v1.6/devices/smart-farm/led/1v")==0) {
    if((char)payload[0]=='1'){
        digitalWrite(LED, HIGH);
    }
}
```



```
        }
    else{
        digitalWrite(LED, LOW);
    }
Serial.println();

}

//Kondisi apabila Topik yang masuk Relay 1 dan Membaca Nilai Pesan
//yang dikirimkan untuk menyalakan Pompa
if(strcmp(topic,"/v1.6/devices/smart-farm/relay1/lv")==0) {
    if((char)payload[0]=='0'){
        digitalWrite(Pompa, LOW);
    }
    else{
        digitalWrite(Pompa, HIGH);
    }
    Serial.println();
}

//Kondisi apabila Topik yang masuk Relay 2 dan Membaca Nilai Pesan
//yang dikirimkan untuk Menyalakan Air
if(strcmp(topic,"/v1.6/devices/smart-farm/relay2/lv")==0) {
    if((char)payload[0]=='0'){
        digitalWrite(Minum, LOW);
    }
    else{
        digitalWrite(Minum, HIGH);
    }
    Serial.println();
}

//Kondisi apabila Topik yang masuk Servo dan Membaca Nilai Pesan
//yang dikirimkan untuk Pemberian Pakan
if(strcmp(topic,"/v1.6/devices/smart-farm/servo/lv")==0) {
    if((char)payload[0]=='1'){
        servo.write(180); //Menggerakkan Servo ke Sudut 180 apabila
pesan yang masuk bernilai 1
    }
    else{
        servo.write(0); //Menggerakkan Servo ke Sudut 0 apabila pesan
yang masuk bernilai 0
    }
    Serial.println();
}

void setup() {
    Serial.begin(115200); //Baudrate untuk Serial Komunikasi
    client.ubidotsSetBroker("industrial.api.ubidots.com"); //Set
Broker Ubidots
    client.setDebug(true);
    client.wifiConnection(WIFINAME, WIFIPASS); //Mencoba Koneksi
dengan Wi-Fi
    client.begin(callback); //Aktifkan fungsi Callback
    pinMode(LED, OUTPUT); //Set LED sebagai Output
```



```
pinMode(Pompa, OUTPUT); //Set Relay 1 sebagai Output
pinMode(Minum, OUTPUT); //Set Relay 2 sebagai Output
digitalWrite(LED, LOW); //Set LED dengan Nilai LOW
digitalWrite(Pompa, HIGH); //Set Relay 1 dengan Nilai HIGH
digitalWrite(Minum, HIGH); //Set Relay 2 dengan Nilai HIGH
client.ubidotsSubscribe("smart-farm","led"); //Subscribe ke
Topik LED dan Device Smart-Farm
client.ubidotsSubscribe("smart-farm","relay1"); //Subscribe ke
Topik Relay1 dan Device Smart-Farm
client.ubidotsSubscribe("smart-farm","relay2"); //Subscribe ke
Topik Relay2 dan Device Smart-Farm
client.ubidotsSubscribe("smart-farm","servo"); //Subscribe ke
Topik Servo dan Device Smart-Farm
dht.begin(); //Memulai fungsi DHT
servo.attach(13); //Memulai fungsi Servo pada Pin 13
MQ135.setRegressionMethod(1); //_PPM = a*ratio^b //Set Method
yang digunakan untuk MQ-135
MQ135.init(); //Memulai fungsi MQ-135
//Fungsi untuk Kalibrasi MQ-135
float calcR0 = 0;
for(int i = 1; i<=10; i++)
{
    MQ135.update();
    calcR0 += MQ135.calibrate(RatioMQ135CleanAir);
    Serial.print(".");
}
MQ135.setR0(calcR0/10);
}

void loop() {
    unsigned long time = millis(); //Fungsi Millis
    MQ135.update(); //Update Nilai MQ-135
    MQ135.setA(102.2 ); MQ135.setB(-2.473); //Set Nilai Sampling
untuk NH4 /Amonia
    float NH4 = MQ135.readSensor(); //Hasil Pembacaan nilai Amonia
    int jarak = sonar.ping_cm(); //Hasil Pembacaan Jarak
    float h = dht.readHumidity(); //Hasil Pembacaan Humidity
    float t = dht.readTemperature(); //Hasil Pembacaan Suhu

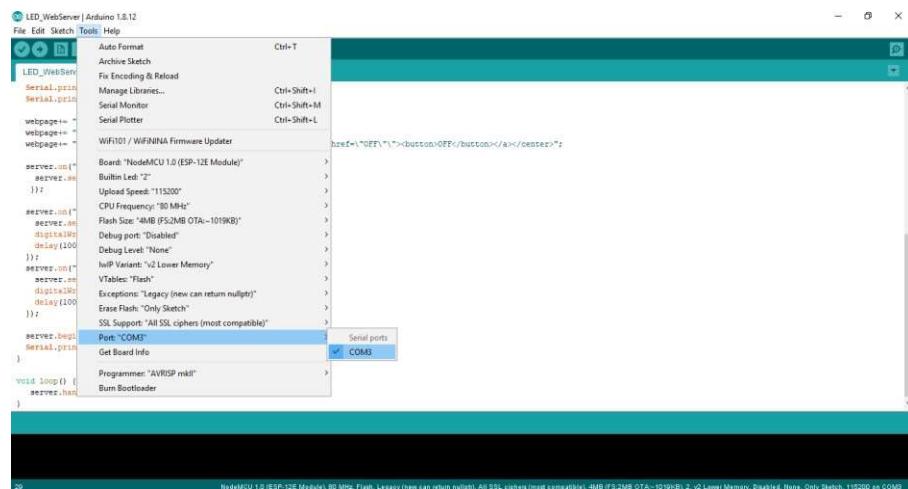
    //Reconnect apabila Sambungan Koneksi NodeMCU terputus dan re-
subscribe topik dan device yang ada
    if(!client.connected()){
        client.reconnect();
        client.ubidotsSubscribe("smart-farm","led");
        client.ubidotsSubscribe("smart-farm","relay1");
        client.ubidotsSubscribe("smart-farm","relay2");
        client.ubidotsSubscribe("smart-farm","servo");
    }

    //Kondisi untuk mengirimkan hasil pembacaan sensor ke Device
    Ubidots setiap 2 detik
    if ((unsigned long)(time - durasiKirim) >= jedaKirim)
    {
        //Mengirimkan hasil pembacaan sensor ke Device Ubidots
        client.add("ketinggian", jarak);
        client.add("suhu", t);
```



```
        client.add("kelembaban", h);
        client.add("amonia", NH4);
        client.ubidotsPublish("smart-farm");
        durasiKirim = millis();
    }
    client.loop();
}
```

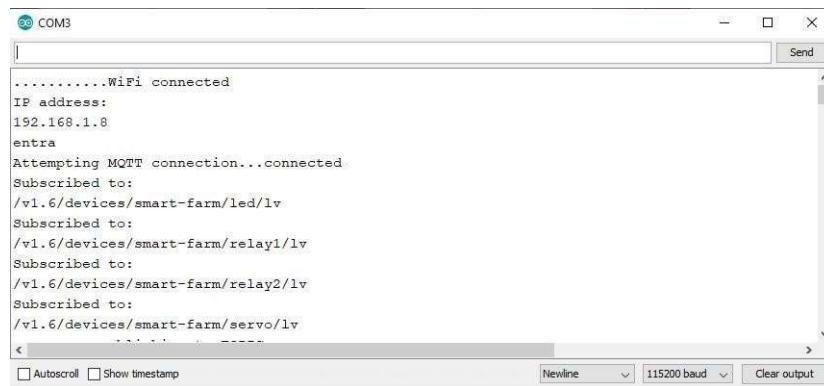
4. Kemudian hubungkan Board NodeMCU dengan Laptop menggunakan Kabel USB. Selanjutnya, pilih Menu **Tools -> Port**. Lalu, pilih Port yang tersedia seperti gambar di bawah:



5. Kemudian, klik **Button Upload** yang berbentuk seperti tanda panah untuk melakukan **Compile** dan **Upload Program** ke Board NodeMCU.



6. Setelah program selesai di-upload, buka **Serial Monitor** dan apabila tampilan sudah seperti gambar di bawah, maka NodeMCU berhasil terhubung ke Ubidots





7. Lalu, bukalah Halaman **Dashboard Ubidots** dan apabila tampilan sudah seperti gambar di bawah, maka Data dari Sensor berhasil dikirimkan ke *Cloud Storage* milik Ubidots





## PROJECT VI

### MEMBUAT WEATHER STATION DENGAN MENGGUNAKAN NODEMCU DAN ADAFRUIT IO

#### - DESKRIPSI :

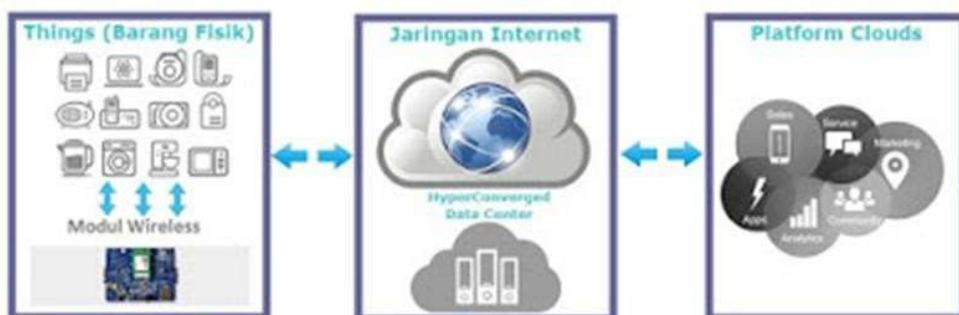
Project ini bertujuan untuk membuat sebuah *Weather Station* yang dapat dikendalikan dan dimonitoring melalui Platform Adafruit IO dan juga *Board NodeMCU* yang saling terhubung.

#### - BACAAN SINGKAT :

##### A. *Internet of Things*

*Internet of Things (IoT)* merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari koneksi internet yang tersambung secara terus-menerus. *Internet of Things* digunakan untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara *independent*. Salah satu contohnya adalah *smarthome* yang dapat memantau dan mengontrol kondisi rumah melalui *smartphone* dengan bantuan koneksi internet.

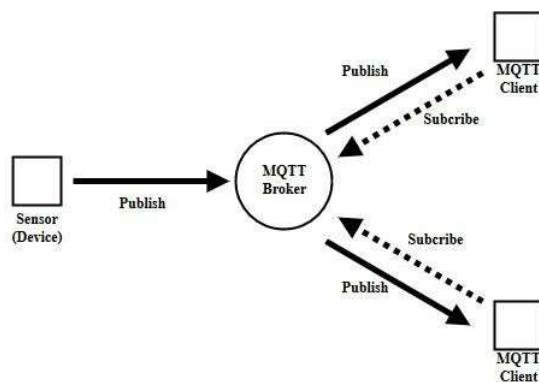
Pada dasarnya, perangkat *Internet of Things* terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komunikasi dan *server* sebagai pengumpul informasi yang diterima dari *sensor* dan dianalisa. Gambar di bawah menunjukkan gambaran cara kerja *Internet of Things* yang terdiri dari perangkat fisik, jaringan internet, dan *cloud database / server*.





### B. Message Queuing Telemetry Transport

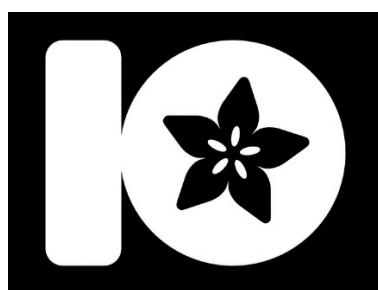
Protokol MQTT (*Message Queuing Telemetry Transport*) merupakan protokol pesan yang sangat sederhana dan juga ringan. Protokol MQTT menggunakan arsitektur *publish* dan *subscribe* yang dirancang secara terbuka dan mudah untuk diimplementasikan. Selain itu, MQTT mampu menangani ribuan pengguna jarak jauh dengan hanya menggunakan satu *server*. MQTT menggunakan *bandwidth* jaringan yang sedikit, selain itu juga menggunakan sumber daya perangkat yang sedikit pula. Pada protokol ini berbasis M2M (*Machine to Machine*) yang artinya dapat berkomunikasi antar mesin atau perangkat, bisa dua perangkat atau lebih. Arsitektur dari protokol MQTT dapat dilihat pada Gambar 2.2.



Gambar 2.2 Arsitektur Protokol MQTT

Pada gambar di atas, terlihat terdiri dari 2 komponen utama, yaitu *MQTT client* dan *MQTT broker*. *MQTT client* merupakan protokol yang berada pada perangkat, seperti halnya Arduino. *MQTT broker* berfungsi untuk menangani *publish* dan juga *subscribe* data.

### C. Adafruit IO

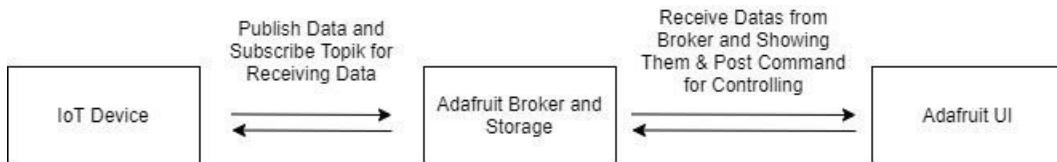


Adafruit IO adalah salah satu Platform IoT yang memberikan kemudahan dalam membuat sebuah layanan yang menggunakan konsep IoT. Berbagai fitur



yang dimiliki oleh Adafruit IO memberikan kemudahan bagi pengguna untuk bisa membuat sebuah layanan IoT salah satunya adalah *Cloud Storage* yang memberikan kemudahan dalam penyimpanan data IoT dan juga *Dashboard* yang memberikan kemudahan dalam membuat sebuah UI untuk *monitoring* atau *controlling* sebuah *IoT Device*.

#### D. Cara Kerja



Cara kerja dari sistem ini adalah *IoT Device* mengambil Data melalui sensor yang digunakan pada sistem, kemudian Data tersebut dikirimkan ke *MQTT Broker* dan data disimpan pada *Cloud Storage* dan kemudian Data tersebut akan ditampilkan pada Dashboard yang telah dibuat pada Adafruit IO.

#### - KOMPONEN :

1. Board NodeMCU V3
2. Kabel Jumper
3. Breadboard / Expansion Board
4. Kabel USB / Data
5. DHT 11
6. Relay 2 Channel
7. LED
8. Sensor MQ-135
9. Adafruit IO Platform
10. Web Browser
11. Arduino IDE



## - LANGKAH-LANGKAH

### A. Konfigurasi Device pada Ubidots

1. Buka halaman **Adafruit IO** pada Browser : <https://io.adafruit.com/>
2. Setelah itu, pilih **Get Started for Free** apabila belum memiliki akun



3. Lalu isi kolom Registrasi sesuai dengan Perintah yang ada dan selanjutnya klik **Create Account**

SIGN UP

The best way to shop with Adafruit is to create an account which allows you to shop faster, track the status of your current orders, review your previous orders and take advantage of our other member benefits.

FIRST NAME

LAST NAME

EMAIL

USERNAME

Username is viewable to the public on the forums, Adafruit IO, and elsewhere.

PASSWORD

**CREATE ACCOUNT**

HAVE AN ADAFRUIT ACCOUNT?  
**SIGN IN**

4. Setelah Registrasi, kalian akan tiba di Halaman Akun kalian seperti gambar di bawah

My Account

Profile  
Addresses  
Security & Privacy  
Payment Methods  
Subscriptions  
Order History  
My Wishlists  
Manage Gift Certificates  
Product Notifications  
Newsletter Settings  
Change Password  
Services

We think you are in the Jakarta time zone, if this isn't correct, please choose the correct time zone below.

FIRST NAME

LAST NAME

EMAIL

USERNAME

GRAVATAR EMAIL

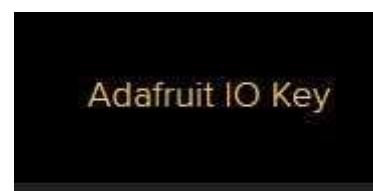
FAVORITE COLOR



5. Selanjutnya, pada bagian **Menu**, kalian dapat memilih **Menu IO** untuk menuju ke Halaman Pembuatan Dashboard untuk Keperluan IoT

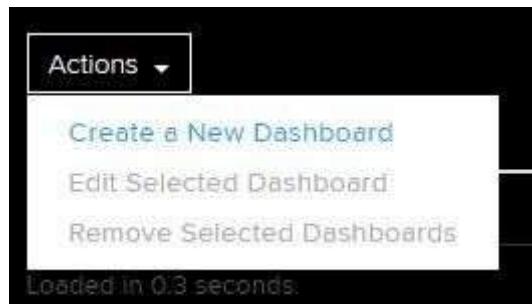
Shop   Learn   Blog   Forums   Videos   Adabox   IO

6. Selanjutnya ketika tiba di **Halaman Dashboard**, kalian dapat mengklik **Adafruit IO Key** pada bagian Pojok kanan atas untuk mendapatkan **Key** dan **Username** dari Akun Kalian untuk digunakan pada Program Arduino IDE



|                                 |   |
|---------------------------------|---|
| Username                        | <input type="text" value="nrkholid"/>                         |
| Active Key                      | <input type="text" value="aio_SnkQB4U7YVWJDM8TaghpbWYJ2dZE"/> |
| <button>REGENERATE KEY</button> |   |

7. Selanjutnya kembali lagi ke **Halaman Dashboard**, kemudian klik **Action** dan pilih **Create a New Dashboard**



8. Lalu buatlah sebuah **Dashboard** dengan ketentuan sebagai berikut:

Create a new Dashboard ✖

Name

Description

Cancel Create



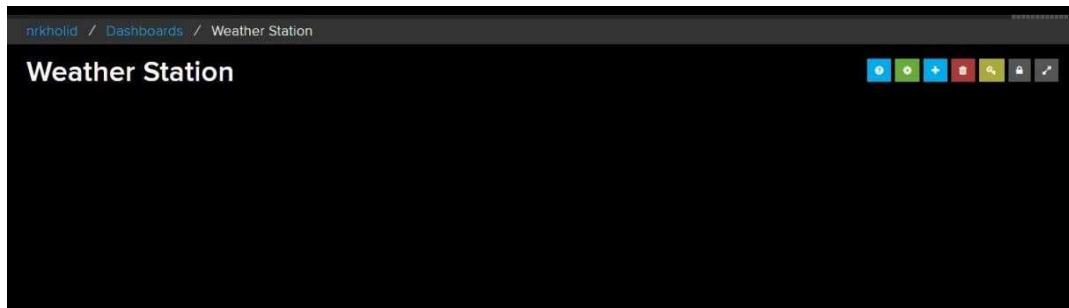
9. Apabila sudah selesai, maka hasilnya akan seperti gambar di bawah:

| Actions ▾                                  |                   |
|--|-------------------|
| <input type="checkbox"/> Name              | Key               |
| <input type="checkbox"/> Weather Station   | weather-station   |
| <input type="checkbox"/> Welcome Dashboard | welcome-dashboard |

Loaded in 0.3 seconds.

10. Kemudian klik **Dashboard ‘Weather Station’** yang telah kalian buat.

Lalu kalian akan dibawa ke Halaman **Dashboard ‘Weather Station’** dengan tampilan sebagai berikut



11. Selanjutnya, buatlah sebuah **Widget** dengan klik Tombol (+) berwarna biru di pojok kanan atas



12. Kemudian, pilih **Widget Switch** seperti gambar dibawah:





13. Setelah itu, buatlah beberapa Feeds seperti **LED**, **Relay1**, **Relay2**, **Suhu**, **Kelembaban**, **CO**, dan **CO2** seperti gambar di bawah:

led Create

relay1 Create

relay2 Create

suhu Create

kelembaban Create

co Create

CO2 Create

14. Apabila sudah, maka tampilan akhirnya akan seperti ini:

Choose feed ✖

**Toggle:** A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

Create

| Group / Feed                          | Last value | Recorded   |
|---------------------------------------|------------|------------|
| <input type="checkbox"/> CO           |            | 4 minutes  |
| <input type="checkbox"/> CO2          |            | 4 minutes  |
| <input type="checkbox"/> kelembaban   |            | 4 minutes  |
| <input type="checkbox"/> led          | OFF        | 6 minutes  |
| <input type="checkbox"/> relay1       |            | 4 minutes  |
| <input type="checkbox"/> relay2       |            | 4 minutes  |
| <input type="checkbox"/> suhu         |            | 4 minutes  |
| <input type="checkbox"/> TekananUdara |            | 4 minutes  |
| <input type="checkbox"/> Welcome Feed |            | 34 minutes |

◀ Previous step Next step ▶



15. Setelah membuat Feeds, selanjutnya pilih Feed **LED** dan kemudian klik **Next Step** dan ikuti ketentuannya seperti pada gambar lalu klik **Create Block**

Choose feed

**Toggle:** A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

| Group / Feed                            | Last value | Recorded   |
|---|------------|------------|
| □ CO                                    | 🔒          | 4 minutes  |
| □ CO2                                   | 🔒          | 4 minutes  |
| □ kelembaban                            | 🔒          | 4 minutes  |
| <input checked="" type="checkbox"/> led | OFF        | 7 minutes  |
| □ relay1                                | 🔒          | 5 minutes  |
| □ relay2                                | 🔒          | 5 minutes  |
| □ suhu                                  | 🔒          | 4 minutes  |
| □ TekananUdara                          | 🔒          | 4 minutes  |
| □ Welcome Feed                          | 🔒          | 35 minutes |

[← Previous step](#) [Next step →](#)

Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

|                                  |               |
|----------------------------------|---------------|
| Block Title (optional)           | Block Preview |
| <input type="text" value="LED"/> |               |
| Button On Text                   |               |
| <input type="text" value="ON"/>  |               |
| Button Off Text                  |               |
| <input type="text" value="OFF"/> |               |

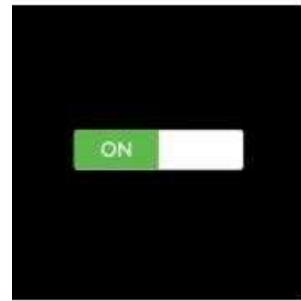
**Toggle** A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

|                 |  |
|-----------------|--|
| Test Value      | <input type="text" value="45"/>  |
| Published Value | <div style="background-color: #ccc; width: 100px; height: 10px; margin-bottom: 5px;"></div> <small>0 bytes</small> |

[← Previous step](#) [Create block](#)



16. Selanjutnya, buatlah lagi **Widget Switch ke – 2** lalu pilih Feed **Relay1** dan kemudian ikut ketentuan pada gambar di bawah lalu klik **Create Block**



### Choose feed

**Toggle:** A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

| Group / Feed                               | Last value | Recorded   |
|--|------------|------------|
| <input type="checkbox"/> CO                | 🔒          | 5 minutes  |
| <input type="checkbox"/> CO2               | 🔒          | 5 minutes  |
| <input type="checkbox"/> kelembaban        | 🔒          | 5 minutes  |
| <input type="checkbox"/> led               | OFF        | 7 minutes  |
| <input checked="" type="checkbox"/> relay1 | 🔒          | 6 minutes  |
| <input type="checkbox"/> relay2            | 🔒          | 5 minutes  |
| <input type="checkbox"/> suhu              | 🔒          | 5 minutes  |
| <input type="checkbox"/> TekananUdara      | 🔒          | 5 minutes  |
| <input type="checkbox"/> Welcome Feed      | 🔒          | 36 minutes |

[◀ Previous step](#) [Next step ▶](#)

### Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)

Block Preview  
**Lampu Rumah**

| Button On Text | Button Off Text |
|----------------|-----------------|
| ON             | OFF             |

**Toggle** A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

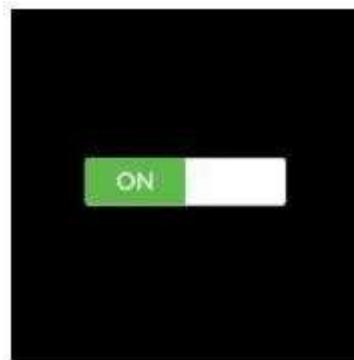


Test Value  
45

Published Value  
0 bytes

[◀ Previous step](#) [Create block](#)

17. Selanjutnya, buatlah lagi **Widget Switch ke – 3** lalu pilih **Feed Relay2** dan kemudian ikut ketentuan pada gambar di bawah lalu klik **Create Block**



### Choose feed



**Toggle:** A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

| <input type="text"/> Group / Feed          | Last value | Recorded   |
|--|------------|------------|
| <input type="checkbox"/> CO                | 🔒          | 7 minutes  |
| <input type="checkbox"/> CO2               | 🔒          | 7 minutes  |
| <input type="checkbox"/> kelembaban        | 🔒          | 7 minutes  |
| <input type="checkbox"/> led               | 🔒 OFF      | 9 minutes  |
| <input type="checkbox"/> relay1            | 🔒          | 7 minutes  |
| <input checked="" type="checkbox"/> relay2 | 🔒          | 7 minutes  |
| <input type="checkbox"/> suhu              | 🔒          | 7 minutes  |
| <input type="checkbox"/> TekananUdara      | 🔒          | 7 minutes  |
| <input type="checkbox"/> Welcome Feed      | 🔒          | 37 minutes |

[◀ Previous step](#) [Next step ➤](#)

Block settings ✖

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)

Lampu Rumah Dalam

Button On Text

ON

Button Off Text

OFF

Block Preview



**Toggle** A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

Test Value

45

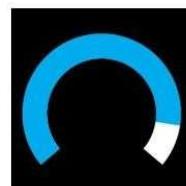
Published Value



0 bytes

[◀ Previous step](#)[Create block](#)

**18.** Selanjutnya, buatlah **Widget Gauge** lalu pilih Feed **Suhu** dan kemudian ikut ketentuan pada gambar di bawah lalu klik **Create Block**

Choose feed ✖

**Gauge:** A gauge is a read only block type that shows a fixed range of values.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.



Enter new feed name

[Create](#)

| Group / Feed                             | Last value | Recorded   |
|--|------------|------------|
| <input type="checkbox"/> CO              | 🔒          | 9 minutes  |
| <input type="checkbox"/> CO2             | 🔒          | 9 minutes  |
| <input type="checkbox"/> kelembaban      | 🔒          | 9 minutes  |
| <input type="checkbox"/> led             | 🔒 OFF      | 11 minutes |
| <input type="checkbox"/> relay1          | 🔒          | 9 minutes  |
| <input type="checkbox"/> relay2          | 🔒          | 9 minutes  |
| <input checked="" type="checkbox"/> suhu | 🔒          | 9 minutes  |
| <input type="checkbox"/> TekananUdara    | 🔒          | 9 minutes  |
| <input type="checkbox"/> Welcome Feed    | 🔒          | 40 minutes |

[◀ Previous step](#)[Next step ➤](#)



## Block settings



In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

## Block Title (optional)

## Gauge Min Value

## Gauge Max Value

## Gauge Width

## Gauge Label

## Low Warning Value

Optional. If no low warning value is given, the gauge will only change color when the value is out of bounds.

## Low Warning Value

Optional. If no low warning value is given, the gauge will only change color when the value is out of bounds.

## High Warning Value

Optional. If no high warning value is given, the gauge will only change color when the value is out of bounds.

## Decimal Places

Number of decimal places to display when value is a number. Defaults to 2.

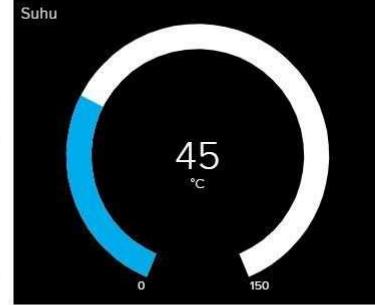
 Show Icon

When checked, show an icon with the value.

## Icon

Show this icon next to the value.

## Block Preview



**Gauge** A gauge is a read only block type that shows a fixed range of values.

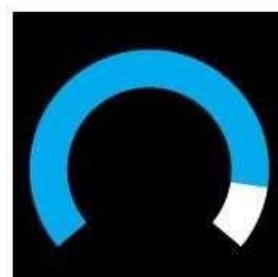
## Test Value

**Gauge** A gauge is a read only block type that shows a fixed range of values.

## Test Value

[Previous step](#) [Create block](#)

- 19.** Selanjutnya, buatlah **Widget Gauge ke - 2** lalu pilih Feed **Kelembaban** dan kemudian ikut ketentuan pada gambar di bawah lalu klik **Create Block**



Choose feed ✖

**Gauge:** A gauge is a read only block type that shows a fixed range of values.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

| Group / Feed                                   | Last value | Recorded   |
|--|------------|------------|
| <input type="checkbox"/> CO                    | 🔒          | 13 minutes |
| <input type="checkbox"/> CO2                   | 🔒          | 13 minutes |
| <input checked="" type="checkbox"/> kelembaban | 🔒          | 13 minutes |
| <input type="checkbox"/> led                   | 🔒 OFF      | 15 minutes |
| <input type="checkbox"/> relay1                | 🔒          | 13 minutes |
| <input type="checkbox"/> relay2                | 🔒          | 13 minutes |
| <input type="checkbox"/> suhu                  | 🔒          | 13 minutes |
| <input type="checkbox"/> TekananUdara          | 🔒          | 13 minutes |
| <input type="checkbox"/> Welcome Feed          | 🔒          | 44 minutes |

[← Previous step](#) [Next step →](#)

Block settings ✖

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)

Gauge Min Value

Gauge Max Value

Gauge Width

Gauge Label

Block Preview

**Kelembaban**

45 %

0 100

## Low Warning Value

Optional: If no low warning value is given, the gauge will only change color when the value is out of bounds.

**Gauge** A gauge is a read only block type that shows a fixed range of values.

## High Warning Value

Optional: If no high warning value is given, the gauge will only change color when the value is out of bounds.

## Test Value

## Decimal Places

Number of decimal places to display when value is a number. Defaults to 2.

 Show Icon

When checked, show an icon with the value.

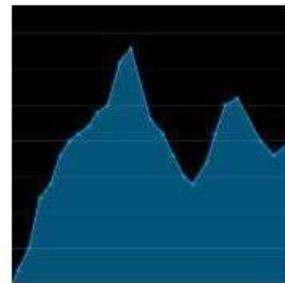
## Icon

Show this icon next to the value.

[← Previous step](#) [Create block](#)



20. Selanjutnya, buatlah **Widget Line Chart** lalu pilih Feed **CO** dan kemudian ikut ketentuan pada gambar di bawah lalu klik **Create Block**



### Choose up to 5 feeds

Line Chart: The line chart is used to graph one or more feeds.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

| Group / Feed                           | Last value | Recorded      |        |
|--|------------|---------------|--------|
| <input checked="" type="checkbox"/> CO | ?          | about 22 h... | 1 of 5 |
| <input type="checkbox"/> CO2           | ?          | about 22 h... |        |
| <input type="checkbox"/> kelembaban    | ?          | about 22 h... |        |
| <input type="checkbox"/> led           | OFF        | about 22 h... |        |
| <input type="checkbox"/> relay1        | ?          | about 22 h... |        |
| <input type="checkbox"/> relay2        | ?          | about 22 h... |        |
| <input type="checkbox"/> suhu          | ?          | about 22 h... |        |
| <input type="checkbox"/> TekananUdara  | ?          | about 22 h... |        |
| <input type="checkbox"/> Welcome Feed  | ?          | about 22 h... |        |

[« Previous step](#) [Next step »](#)

### Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

|  |               |
|--|---------------|
| Block Title (optional)                           | Block Preview |
| <input type="text" value="CO"/>                  |               |
| Show History                                     |               |
| <input type="button" value="Live (no history)"/> |               |
| X-Axis Label                                     |               |
| <input type="text" value="Waktu"/>               |               |
| Y-Axis Label                                     |               |
| <input type="text" value="Nilai"/>               |               |
| Y-Axis Minimum                                   |               |
| <input type="text"/>                             |               |
| Leave blank to automatically detect.             |               |
| Y-Axis Maximum                                   |               |
| <input type="text"/>                             |               |
| Leave blank to automatically detect.             |               |



Decimal Places  
0  
Number of decimal places to display, defaults to 4.

Raw Data Only

When checked, do not show aggregate data ever. If the chart history includes more than 640 data points, only the 640 most recent will be shown.

Stepped Line

Use a stepped line graph. Useful for representing logic levels.

[◀ Previous step](#) [Create block](#)

21. Selanjutnya, buatlah **Widget Line Chart ke - 2** lalu pilih Feed **CO2** dan kemudian ikut ketentuan pada gambar di bawah lalu klik **Create Block**



### Choose up to 5 feeds



**Line Chart:** The line chart is used to graph one or more feeds.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

| <input type="text"/>                    | <input type="text"/> Enter new feed name | <a href="#">Create</a>                                  |
|---|--|---|
| Group / Feed                            | Last value                               | Recorded  |
| <input type="checkbox"/> CO             | 🔒  | about 22 h...   |
| <input checked="" type="checkbox"/> CO2 | 🔒  | about 22 h... <span style="float: right;">1 of 5</span> |
| <input type="checkbox"/> kelembaban     | 🔒  | about 22 h...   |
| <input type="checkbox"/> led            | 🔒 OFF                                    | about 22 h...   |
| <input type="checkbox"/> relay1         | 🔒  | about 22 h...   |
| <input type="checkbox"/> relay2         | 🔒  | about 22 h...   |
| <input type="checkbox"/> suhu           | 🔒  | about 22 h...   |
| <input type="checkbox"/> TekananUdara   | 🔒  | about 22 h...   |
| <input type="checkbox"/> Welcome Feed   | 🔒  | about 22 h...   |

[◀ Previous step](#) [Next step ➤](#)



## Block settings



In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)

CO2

Show History

Live (no history)

X-Axis Label

Waktu

Y-Axis Label

Nilai

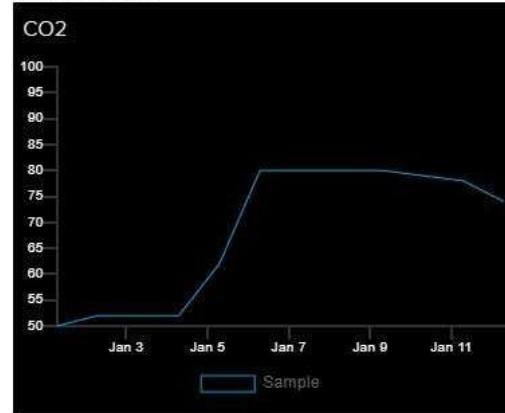
Y-Axis Minimum

Leave blank to automatically detect.

Y-Axis Maximum

Leave blank to automatically detect.

Block Preview



Line Chart The line chart is used to graph one or more feeds.

Decimal Places

0

Number of decimal places to display, defaults to 4.

Raw Data Only

When checked, do not show aggregate data ever. If the chart history includes more than 640 data points, only the 640 most recent will be shown.

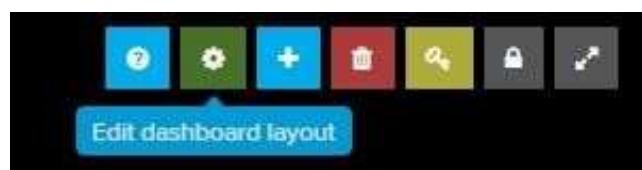
Stepped Line

Use a stepped line graph. Useful for representing logic levels.

[◀ Previous step](#)

[Create block](#)

22. Setelah selesai, kemudian Editlah **Dashboard Layout** yang telah berisikan **Widget** yang dibuat dengan klik Tombol **Edit Dashboard** pada pojok kanan atas seperti gambar di bawah





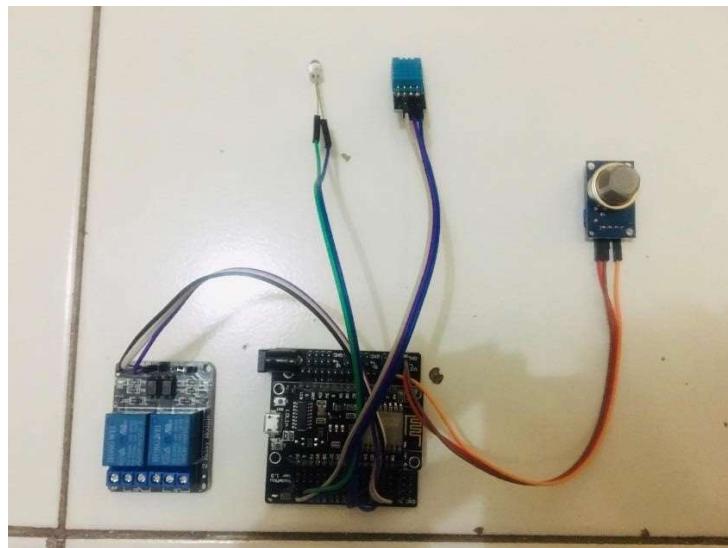
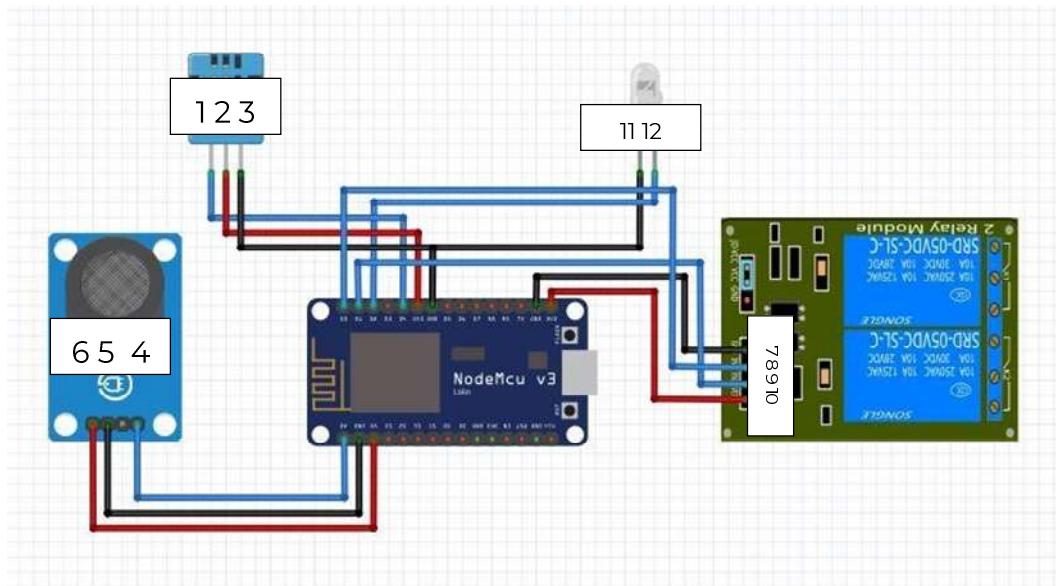
23. Editlah **Dashboard Layout** yang telah ada hingga tampilan berbentuk seperti gambar di bawah



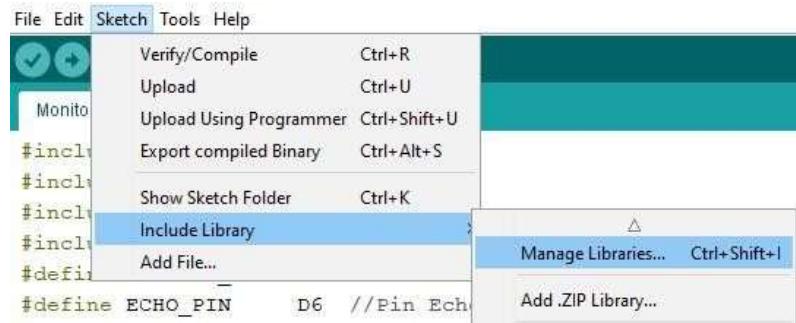
### B. Konfigurasi Program pada NodeMCU di Arduino IDE

1. Hubungkan DHT11, MQ-135, LED, dan Relay pada Board NodeMCU seperti ketentuan dibawah:

| No | Pin DHT11    | Pin NodeMCU |
|----|--------------|-------------|
| 1  | Pin Data     | D4          |
| 2  | VCC DHT11    | 3.3V        |
| 3  | GND DHT11    | GND         |
| No | Pin MQ-135   | Pin NodeMCU |
| 4  | Pin Data     | A0          |
| 5  | GND MQ-135   | Gnd         |
| 6  | VCC MQ-135   | 3.3V        |
| No | Pin Relay    | Pin NodeMCU |
| 7  | GND Relay    | Gnd         |
| 8  | Pin IN1      | D0          |
| 9  | Pin IN2      | D1          |
| 10 | VCC Relay    | 3.3V        |
| No | Pin LED      | Pin NodeMCU |
| 11 | Kaki Pendek  | Gnd         |
| 12 | Kaki Panjang | D2          |



2. Buka Arduino IDE, kemudian install Library yang diperlukan yaitu **DHT**, **MQUnifiedSensor**, **Adafruit IO Arduino** versi terbaru seperti gambar di bawah:





DHT sensor library  
by Adafruit Version 1.3.8 **INSTALLED**  
**Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors** Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors  
[More info](#)

Type All Topic All MQUnified  
**MQUnifiedsensor**  
by Miguel Califa , Yersson Carrillo, Ghordy Contreras Version 2.0.1 **INSTALLED**  
**This library allows you to read the MQ sensors very easily.** This library allows an Arduino/Genuino/ESP8266 board to read MQ Sensors (Air quality meter) references: MQ2, MQ3, MQ4, MQ5, MQ6, MQ7, MQ8, MQ9, MQ131, MQ135, MQ303A, MQ309A.  
[More info](#)  
Select version [Install](#)

**Adafruit IO Arduino**  
by Adafruit  
**Arduino library to access Adafruit IO.** Arduino library to access Adafruit IO using the Adafruit AirLift, ESP8266, ESP32, M0 WINC1500, WICED, MKR1000, Ethernet, or FONA hardware.  
[More info](#)  
Version 3.6.3 [Install](#)

Dependencies for library Adafruit IO Arduino:3.6.3 X  
The library **Adafruit IO Arduino:3.6.3** needs some other library dependencies currently not installed:  

- **Adafruit MQTT Library**
- **ArduinoHttpClient**
- **Adafruit NeoPixel**
- **Adafruit Si7021 Library**
- **Adafruit SGP30 Sensor**
- **Adafruit BME280 Library**
- **Adafruit LIS3DH**
- **Adafruit BusIO**
- **Adafruit VEML6070 Library**
- **Adafruit ILI9341**
- **Adafruit GFX Library**
- **Adafruit STMPE610**
- **Adafruit TouchScreen**
- **AUnit**
- **ESP32Servo**

Would you like to install also all the missing dependencies?

[Install all](#) [Install 'Adafruit IO Arduino' only](#) [Cancel](#)



### 3. Kemudian, *copy dan paste source code* di bawah ke Arduino IDE

```
#include <ESP8266WiFi.h> //Library untuk Terhubung ke Wi-Fi
#include "Adafruit_MQTT.h" //Library Adafruit MQTT
#include "Adafruit MQTT Client.h" //Library MQTT Client
#include <DHT.h> //Library DHT11
#include <MQUnifiedsensor.h> // Library untuk MQ0135

#define WLAN_SSID          "KILLUA" //Nama Wi-Fi
#define WLAN_PASS          "nada124@" //Password
#define AIO_SERVER          "io.adafruit.com" //Broker atau Server
#define AIO_SERVERPORT      1883 //Port MQTT

#define AIO_USERNAME        "nrkholid" //Username Adafruit IO
#define AIO_KEY              "aio_SnkQ84U7YVWJDM8TaghpWYJ2dZE" //Key
Adafruit IO

#define Relay1 D0 //Pin Relay 1 di NodeMCU
#define Relay2 D1 //Pin Relay 2 di NodeMCU
#define LED D2 //Pin LED di NodeMCU
#define DHTPIN D4 //Pin DHT di NodeMCU
#define DHTTYPE DHT11 //Jenis DHT yang digunakan

#define placa "ESP8266" //Define jenis board yang digunakan
#define Voltage_Resolution 5 //Tegangan yang digunakan
#define pin A0 //Pin yang digunakan untuk MQ-135
#define type "MQ-135" //Jenis MQ yang digunakan
#define ADC_Bit_Resolution 10 //Resolusi Bit ADC
#define RatioMQ135CleanAir 3.6 //Nilai Udara dianggap bersih

DHT dht(DHTPIN, DHTTYPE); //Aktifkan fungsi DHT
MQUnifiedsensor MQ135(placa, Voltage_Resolution,
ADC_Bit_Resolution, pin, type); //Aktifkan fungsi MQ

char str_hum[8]; //Membuat Variabel Char str_hum dengan panjang 8 karakter
char str_temp[8]; //Membuat Variabel Char str_temp dengan panjang 8 karakter
char str_co[8]; //Membuat Variabel Char str_co dengan panjang 8 karakter
char str_co2[8]; //Membuat Variabel Char str_co2 dengan panjang 8 karakter
unsigned long durasiKirim = 0; //Untuk membantu fungsi Millis
unsigned long jedaKirim = 6000; //Untuk membantu fungsi Millis

WiFiClient client; //Mengaktifkan fungsi WiFi Client
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY); //Aktifkan fungsi MQTT dan check informasi ke Adafruit IO
Adafruit_MQTT_Subscribe lampu1 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/led"); //Fungsi untuk Subscribe ke Topik LED
Adafruit_MQTT_Subscribe lampu2 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/relay1"); //Fungsi untuk Subscribe ke Topik Relay1
```



```
Adafruit_MQTT_Subscribe lampu3 = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/relay2"); //Fungsi untuk Subscribe ke Topik
Relay2
Adafruit_MQTT_Publish suhu = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/suhu"); //Fungsi untuk Publish ke Topik Suhu
Adafruit_MQTT_Publish kelembaban = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/kelembaban"); //Fungsi untuk Publish ke Topik
Kelembaman
Adafruit_MQTT_Publish KCO = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/CO"); //Fungsi untuk Publish ke Topik CO
Adafruit_MQTT_Publish KCO2 = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/CO2"); //Fungsi untuk Publish ke Topik CO2

void MQTT_connect(); //Untuk terhubung ke Adafruit MQTT
void setup() {
    Serial.begin(115200); //Baudrate untuk Komunikasi Serial
    pinMode(LED,OUTPUT); //Set LED sebagai Output
    pinMode(Relay1,OUTPUT); //Set Relay1 sebagai Output
    pinMode(Relay2,OUTPUT); //Set Relay2 sebagai Output
    digitalWrite(LED, LOW); //Set Nilai LED
    digitalWrite(Relay1, HIGH); //Set Nilai Relay1
    digitalWrite(Relay2, HIGH); //Set Nilai Relay2
    dht.begin(); //Memulai fungsi DHT
    MQ135.setRegressionMethod(1); //_PPM = a*ratio^b //Set Method
yang digunakan untuk MQ-135
    MQ135.init(); //Memulai fungsi MQ-135
    //Fungsi untuk Kalibrasi MQ-135
    float calcR0 = 0;
    for(int i = 1; i<=10; i++)
    {
        MQ135.update();
        calcR0 += MQ135.calibrate(RatioMQ135CleanAir);
        Serial.print(".");
    }
    MQ135.setR0(calcR0/10);
    delay(10);
    //Untuk memulai terhubung dengan jaringan Wi-Fi
    WiFi.begin(WLAN_SSID, WLAN_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    //Subscribe Topik Lampu1, Lampu2, dan Lampu3
    mqtt.subscribe(&lampu1);
    mqtt.subscribe(&lampu2);
    mqtt.subscribe(&lampu3);
}

void loop() {
    unsigned long time = millis(); //Fungsi Millis
    MQTT_connect(); //Untuk fungsi reconnect ke Adafruit MQTT
    MQ135.update(); //Update nilai sensor MQ-135
    //Pembacaan Nilai CO oleh Sensor MQ-135
    MQ135.setA(605.18); MQ135.setB(-3.937);
    float CO = MQ135.readSensor();
    //Pembacaan Nilai CO2 oleh Sensor MQ-135
```



```
MQ135.setA(110.47); MQ135.setB(-2.862);
float CO2 = MQ135.readSensor();
//Pembacaan nilai Suhu dan Humidity oleh Sensor DHT11
float h = dht.readHumidity();
float t = dht.readTemperature();

//Konversi Variabel Float / Double ke String dengan 4 nilai di
depan koma dan 2 nilai di belakang koma
dtostrf(t, 4, 2, str_temp);
dtostrf(h, 4, 2, str_hum);
dtostrf(CO, 4, 2, str_co);
dtostrf(CO2, 4, 2, str_co2);

//Pengiriman Nilai Hasil Pembacaan Sensor ke Adafruit IO setiap
6 Detik
if ((unsigned long)(time - durasiKirim) >= jedaKirim)
{
    suhu.publish(str_temp);
    kelembaban.publish(str_hum);
    KCO.publish(str_co);
    KCO2.publish(str_co2);
    durasiKirim = millis();
}

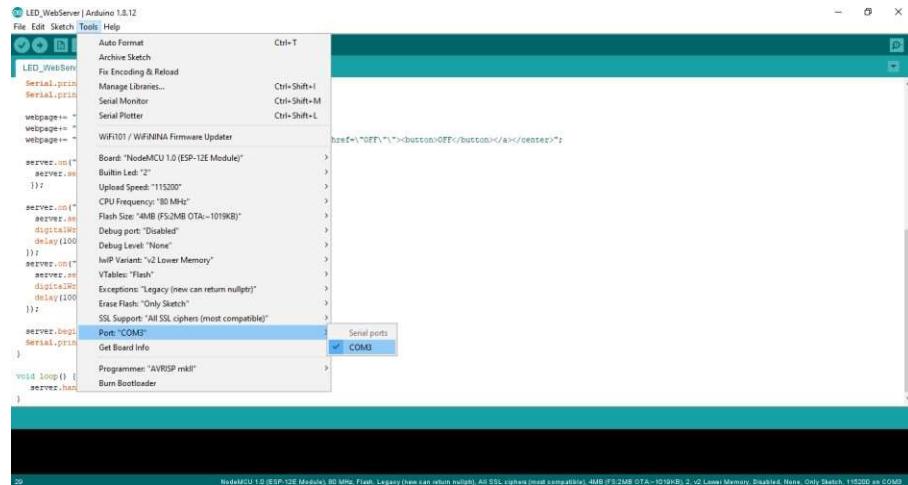
//Menjalankan fungsi subscribe
Adafruit_MQTT_Subscribe *subscription;
//Pembacaan Nilai berdasarkan Subscribe yang diterima
while ((subscription = mqtt.readSubscription(5000))) {
    //Kondisi untuk menghidupkan LED apabila subscribe yang diterima
    adalah LED
    if (subscription == &lampu1) {
        Serial.print(F("Got: "));
        Serial.println((char *)lampu1.lastread);
        if(strcmp((char*)lampu1.lastread,"ON")==0) {
            digitalWrite(LED,HIGH);
        }
        if(strcmp((char*)lampu1.lastread,"OFF")==0) {
            digitalWrite(LED,LOW);
        }
    }
    //Kondisi untuk menghidupkan Relay1 apabila subscribe yang
    diterima adalah Relay1
    if (subscription == &lampu2) {
        Serial.print(F("Got: "));
        Serial.println((char *)lampu2.lastread);
        if(strcmp((char*)lampu2.lastread,"ON")==0) {
            digitalWrite(Relay1,LOW);
        }
        if(strcmp((char*)lampu2.lastread,"OFF")==0) {
            digitalWrite(Relay1,HIGH);
        }
    }
    //Kondisi untuk menghidupkan Relay2 apabila subscribe yang
    diterima adalah Relay2
    if (subscription == &lampu3) {
        Serial.print(F("Got: "));
```



```
Serial.println((char *)lampa3.lastread);
if(strcmp((char*)lampa3.lastread,"ON")==0) {
    digitalWrite(Relay2,LOW);
}
if(strcmp((char*)lampa3.lastread,"OFF")==0) {
    digitalWrite(Relay2,HIGH);
}
}

//Fungsi untuk koneksi ke Adafruit MQTT
void MQTT_connect() {
    int8_t ret;
    if (mqtt.connected()) {
        return;
    }
    Serial.print("Connecting to MQTT... ");
    uint8_t retry = 3;
    while ((ret = mqtt.connect()) != 0) {
        Serial.println(mqtt.connectErrorString(ret));
        Serial.println("Reconnecting . . .");
        mqtt.disconnect();
        delay(5000);
        retry--;
        if (retries == 0) {
            while (1);
        }
    }
    Serial.println("MQTT Connected!");
}
```

4. Kemudian hubungkan Board NodeMCU dengan Laptop menggunakan Kabel USB. Selanjutnya, pilih Menu **Tools -> Port**. Lalu, pilih Port yang tersedia seperti gambar di bawah:



5. Kemudian, klik **Button Upload** yang berbentuk seperti tanda panah untuk melakukan **Compile** dan **Upload Program** ke Board NodeMCU.



- Setelah program selesai di-upload, bukalah **Adafruit Dashboard** yang telah kalian buat, apabila sudah seperti gambar di bawah artinya Data berhasil masuk ke **Adafruit IO**



## DAFTAR PUSTAKA

---

- [1] Zerynth Studio, “NodeMCU V3,” [Online]. Available: <https://docs.zerynth.com/latest/official/board.zerynth.nodemcu3/docs/index.html>. [Accessed: 28 Mei 2020]
- [2] Dickson Kho, “Pengertian LED (Light Emitting Diode) dan Cara Kerjanya,” [Online]. Available: <https://teknikelektronika.com/pengertian-led-light-emitting-diode-cara-kerja>. [ Accessed : 28 Mei 2020 ]
- [3] M. K. Arafat, “SISTEM PENGAMANAN PINTU RUMAH BERBASIS Internet Of Things (IoT) dengan ESP8266,” Jurnal Ilmiah Fakultas Teknik “Technologia,” vol. 7, no. 4, pp. 262–268, 2016.
- [4] M. K. Arafat, “SISTEM PENGAMANAN PINTU RUMAH BERBASIS Internet Of Things (IoT) dengan ESP8266,” Jurnal Ilmiah Fakultas Teknik “Technologia,” vol. 7, no. 4, pp. 262–268, 2016.
- [5] Blynk, “Intro” [Online]. Available: <https://docs.blynk.cc/>. [Accessed : 2 Juni 2020]
- [6] Ajie, “MENGUKUR SUHU DAN KELEMBABAN UDARA DENGAN SENSOR DHT11 DAN ARDUINO,” 2016. [Online]. Available: <http://saptaji.com/2016/08/10/mengukur-suhu-dan-kelembaban-udara-dengan-sensor-dht11-dan-arduino/>. [Accessed : 2 Juni 2020]
- [7] Elektronika Dasar, “Motor Servo,” 2020. [Online]. Available: <https://elektronika-dasar.web.id/motor-servo/>. [Accessed : 2 Juni 2020]
- [8] T. Budioko, “Sistem Monitor Suhu Jarak Jauh Berbasis Internet of Things Menggunakan Protokol MQTT,” Semin. Ris. Teknol. Inf., Yogyakarta, 2016, pp. 353–358.