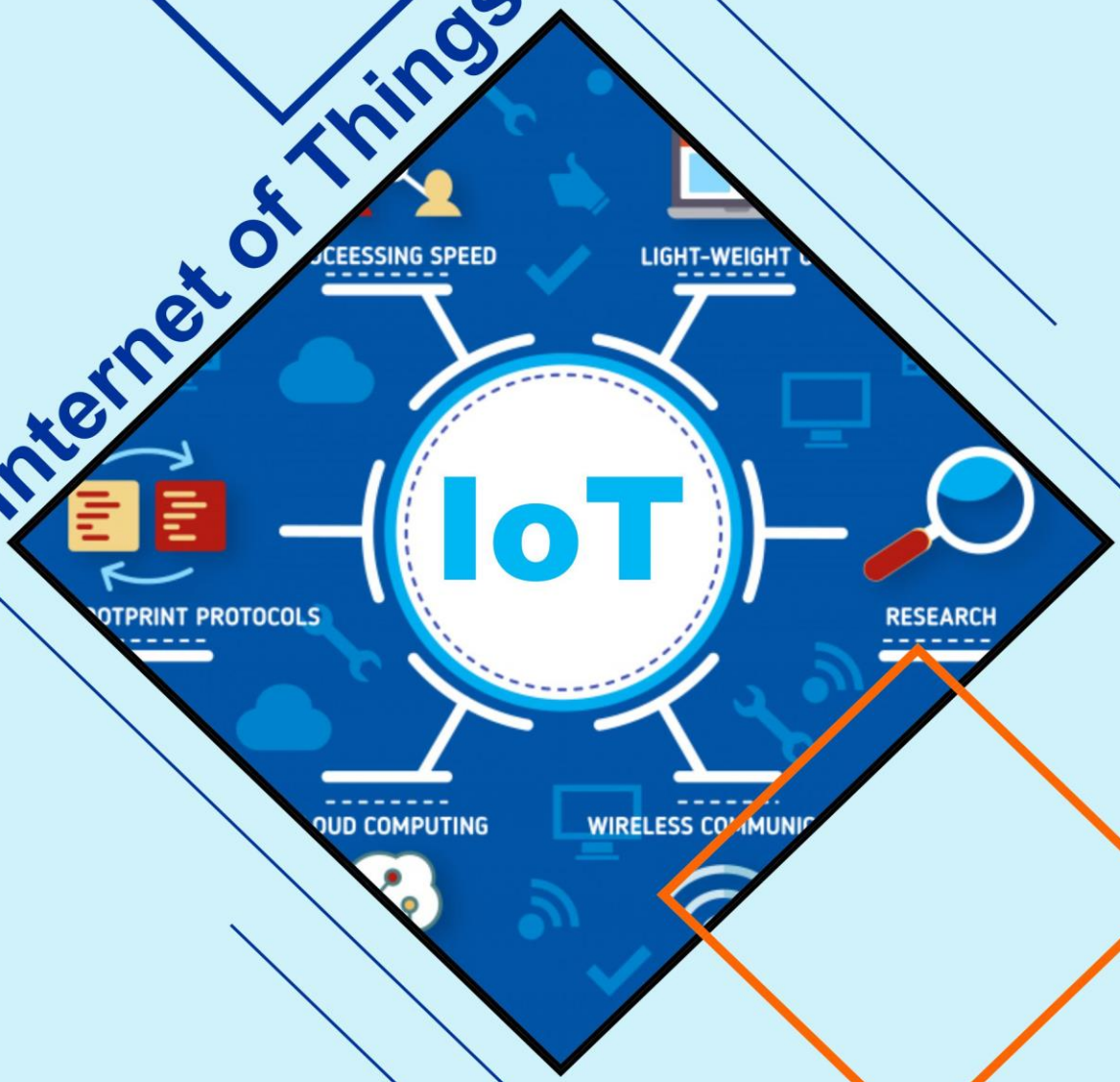
TRAINER KIT

IoT

Internet of Things

Internet of Things



ARDUTECH

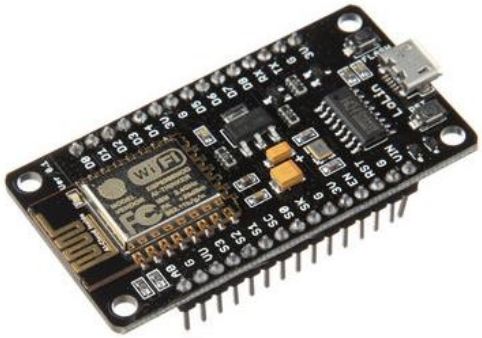
DAFTAR ISI:

Nodemcu V3

- **Konfigurasi Nodemcu V3**

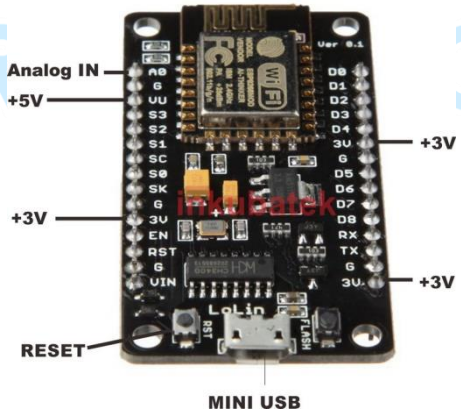
  NodeMCU adalah sebuah board elektronik yang berbasis chip ESP8266 dengan kemampuan menjalankan fungsi mikrokontroler dan juga koneksi internet (WiFi). Terdapat beberapa pin I/O sehingga dapat dikembangkan menjadi sebuah aplikasi monitoring maupun controlling pada proyek IoT.

  Bentuk fisik dari modul NodeMCU V3 tampak pada gambar berikut :

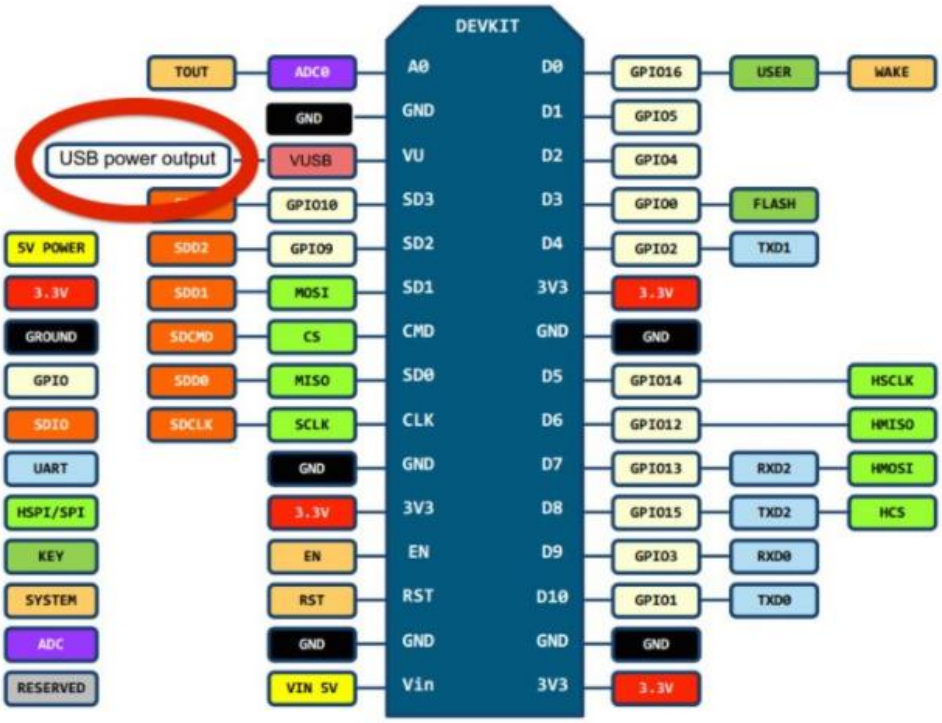


  Perhatikan pada NodeMCU V3 ini terdapat port USB (mini USB) sehingga akan memudahkan dalam pemrogramannya nanti.

- **Spesifikasi :**
  - Tegangan kerja : 3,3 V
  - Flash memori : 16 MB
  - Terintegrasi dengan protocol TCP/IP
  - Processor : Tensilica L106 32 bit
  - Kecepatan : 80 – 160 Mhz
  - Jumlah pin Digital I/O : 11 (D0 – D10)



- **Konfigurasi pin NodeMCu V3 :**



- **Pemrograman NodeMCU V3**
  NodeMCU V3 dapat diprogram dengan *compiler*-nya Arduino, menggunakan Arduino IDE. Tentu saja platform pemrogramannya memakai bahasa C.
  Dasar pemrograman Arduino dapat Anda pelajari di "TUTORIAL PEMROGRAMAN NODEMCU ESP8266 DENGAN ARDUINO.pdf" file pdf ada di **CD**.

- **Instalasi Driver NodeMCU V3**
  Board NodeMCU V3 support Win XP, Vista, Win 7/8/10. Prosedur instalasi silakan baca di **CD "INSTALASI DRIVER NodeMCU V3.pdf"**.

- **Seting Arduino IDE untuk Nodemcu V3**
  Detail langkahnya bisa Anda buka "Instalasi Software Compiler.pdf" file ada di CD.

- **Library Arduino untuk Aplikasi IoT**
  Sebelum kita menggunakan Arduino IDE untuk membuat program aplikasi IoT, beberapa library perlu kita tambahkan. **File library** dan **cara menambahkan libray** silakan buka di CD.

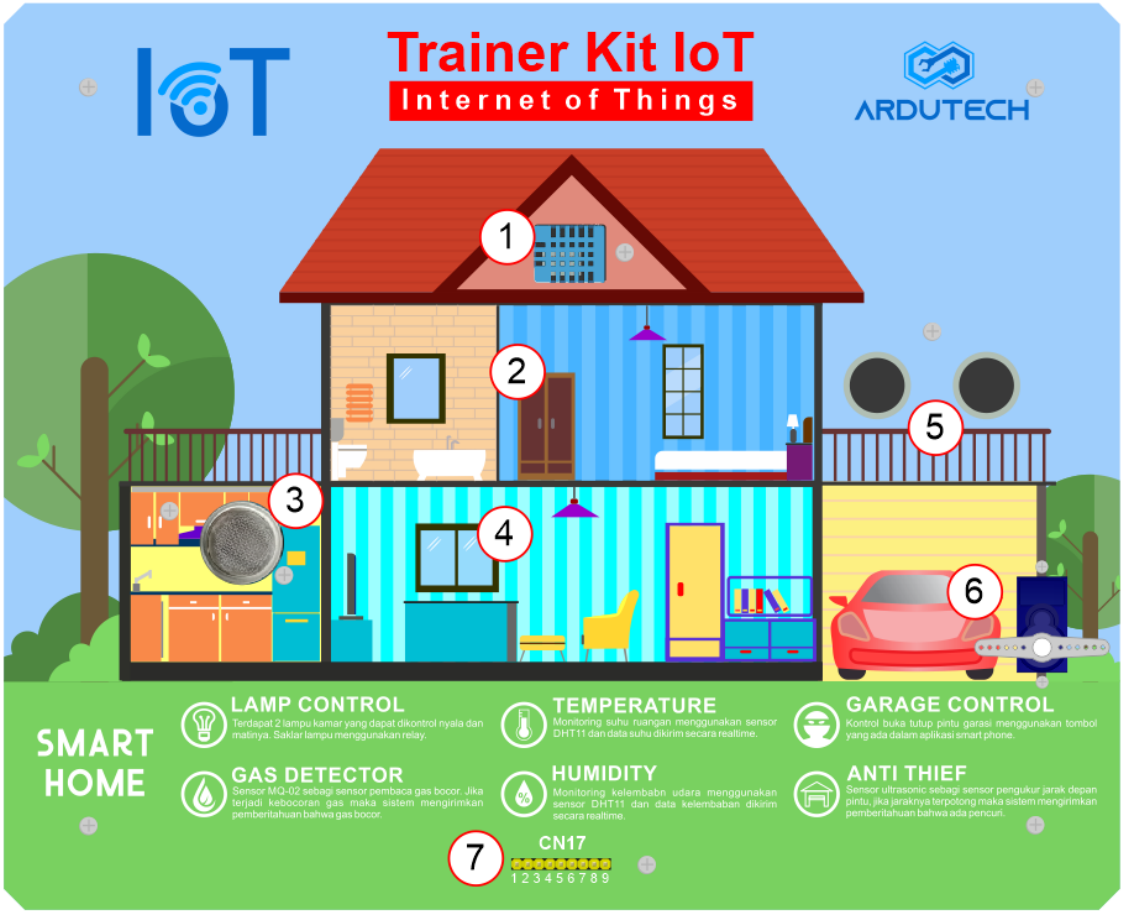
Modul Trainer Kit IoT Ardutech

Modul **Trainer Kit IoT** *made in* Ardutech terdiri dari board NodeMCU V3 dan beberapa peripheral penunjang aplikasi IoT.

Gambar Panel Atas dan Panel Bawah

Bagian atas merupakan aplikasi IoT smarthome yang akan dipakai pada project ke-10. Untuk project 1 s/d 9 semua menggunakan modul Bawah dengan keterangan sebagai berikut:

Panel Atas

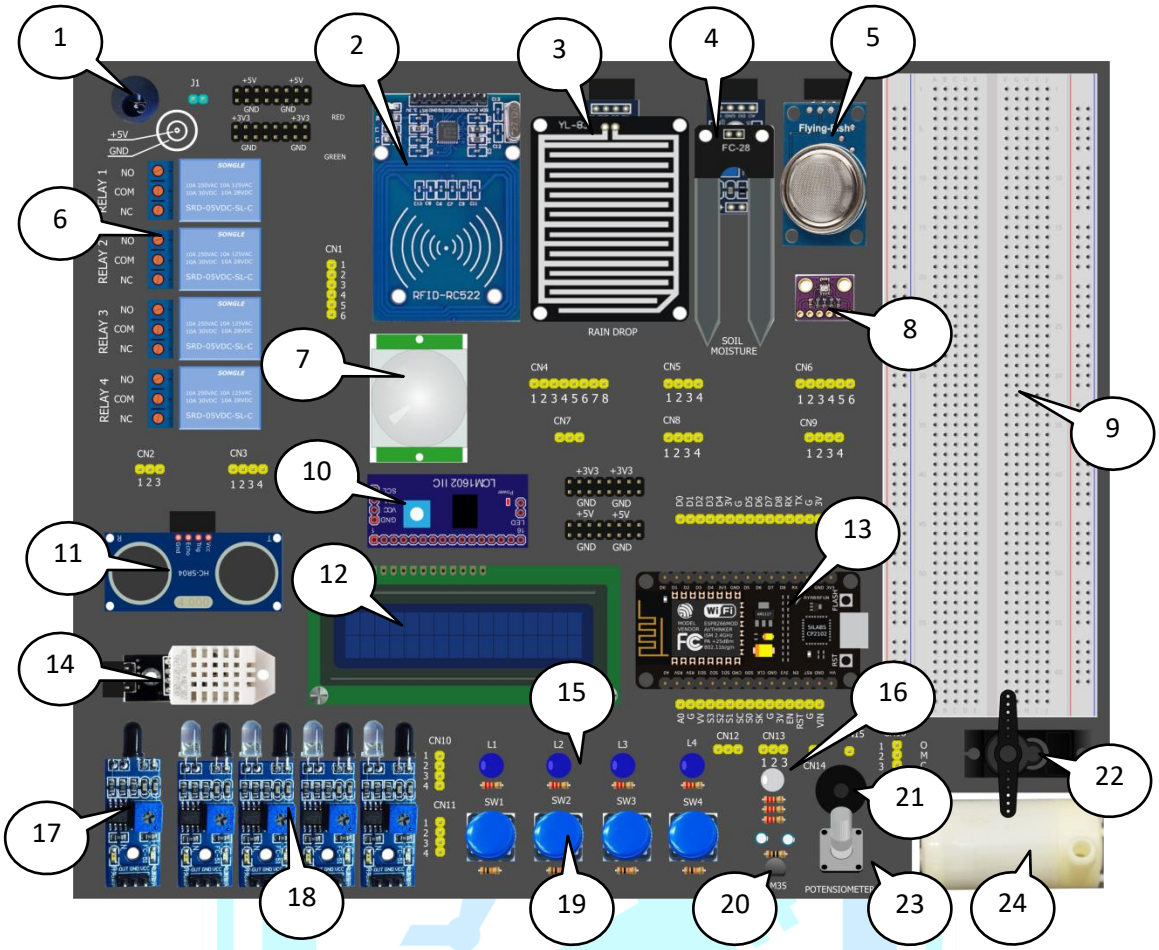


Keterangan :

Nomor	Keterangan
1	Sensor DHT11
2	Ruangan Atas / Lampu 1
3	Sensor Gas MQ-2
4	Ruangan Bawah / Lampu 2
5	Sensor Jarak Ultrasonic HC-SR04
6	Motor Servo
7	Konektor CN17

Penjelasan fungsi dan keterangan konektor akan dibahas pada project 16 smart home.

Panel Bawah



Keterangan :

Nomor	Keterangan
1	Konektor DC 5V/2A, untuk supply tegangan sensor J1 Jumper terhubung → NodeMCU power supply dari Adaptor J1 Jumper tidak terhubung → NodeMCU power supply dari USB
2	Module RFID
3	Sensor Raindrop
4	Sensor Soil moisture (sensor kelembaban tanah)
5	Sensor Gas MQ-2
6	Relay 4 channel
7	Sensor gerak PIR
8	Sensor tekanan atmosfir, ketinggian dan suhu
9	Breadboard
10	Module I2C LCD
11	Senor ultrasonic SR04
12	LCD 2x16
13	NodeMCU V3

14	Sensor suhu dan kelembaban DHT22
15	4 LED, Aktif HIGH
16	LED RGB
17	Flame sensor
18	4 Obstacle sensor
19	4 Tombol push ON, aktif HIGH
20	Sensor suhu LM35
21	Buzzer
22	Motor servo MG 90S
23	Potensiometer
24	Pompa air 5V

PENTING:

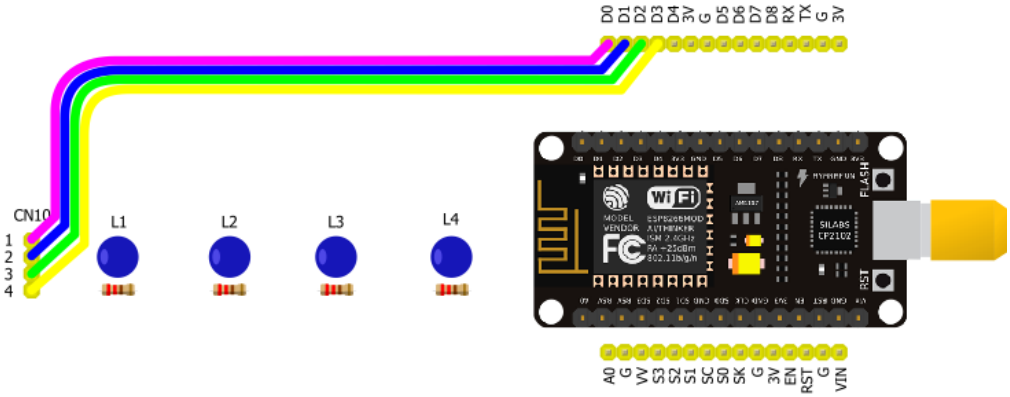
Setiap project yang menggunakan tegangan 5v maka gunakan adaptor 5v. Positip dan negatip adaptor jangan sampai terbalik.

Tes modul Nodemcu V3

Sebelum praktek pastikan :

1. **Driver usb NodeMCU V3 sudah terinstal.**
2. **Seting Arduino IDE untuk NodeMCU V3.**
3. **Library Arduino sudah ditambahkan.**

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - Kabel mikro USB
  - 4 LED
  - 4 Kabel jumper female – female
- **Skematik:**



Keterangan:

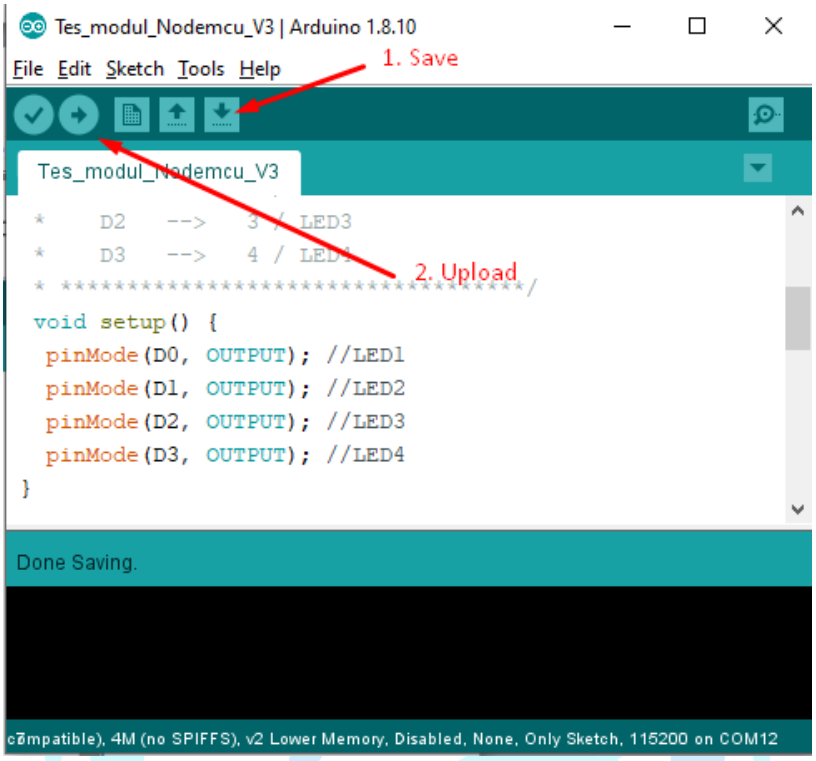
NodeMCU	CN10 / LED
D0	1
D1	2
D2	3
D3	4

- **Kebutuhan software:**
  - Arduino IDE
- **Program:**

```
/************************************
 * Program : Tes modul Nodemcu V3
 * Input :
 * Output : 4 LED
 * Trainer Kit IoT Ardutek
 * www.ardutech.com
 ************************************
 * NODEMCU  |    Koneksi CN10
 *    D0    -->   1 / LED1
 *    D1    -->   2 / LED2
 *    D2    -->   3 / LED3
 *    D3    -->   4 / LED4
 * ***********************************/
 void setup() {
  pinMode(D0, OUTPUT); //LED1
  pinMode(D1, OUTPUT); //LED2
  pinMode(D2, OUTPUT); //LED3
  pinMode(D3, OUTPUT); //LED4
}

void loop() {
  digitalWrite(D0, HIGH);
  digitalWrite(D1, LOW);
  digitalWrite(D2, LOW);
  digitalWrite(D3, LOW);
  delay(1000);
  digitalWrite(D0, LOW);
  digitalWrite(D1, HIGH);
  digitalWrite(D2, LOW);
  digitalWrite(D3, LOW);
  delay(1000);
  digitalWrite(D0, LOW);
  digitalWrite(D1, LOW);
  digitalWrite(D2, HIGH);
  digitalWrite(D3, LOW);
  delay(1000);
  digitalWrite(D0, LOW);
  digitalWrite(D1, LOW);
  digitalWrite(D2, LOW);
  digitalWrite(D3, HIGH);
```

```
    delay(1000);
  }
```

Ketik program diatas, kemudian simpan (***Save***) kemudian ***Upload*** ke board NodeMCU V3. Klik ***Upload***.



- **Jalannya Alat:**
  Menghidupkan LED 1 s/d LED 4 secara bergantian, sehingga LED terlihat seperti berjalan.

Tes Koneksi Nodemcu V3 dengan WiFi

Selanjutnya kita akan coba koneksikan NodeMCU V3 ini dengan jaringan internet WiFi.
Sebelumnya siapkan hotspot (WiFi) anda. Dapat memakai modem atau HP dengan system kemanan WPA2 PSK (bukan WiFi yang login dengan Web).
Catat nama jaringan WiFi dan passwordnya, nantinya kita gunakan untuk menulis program.
Misalnya :

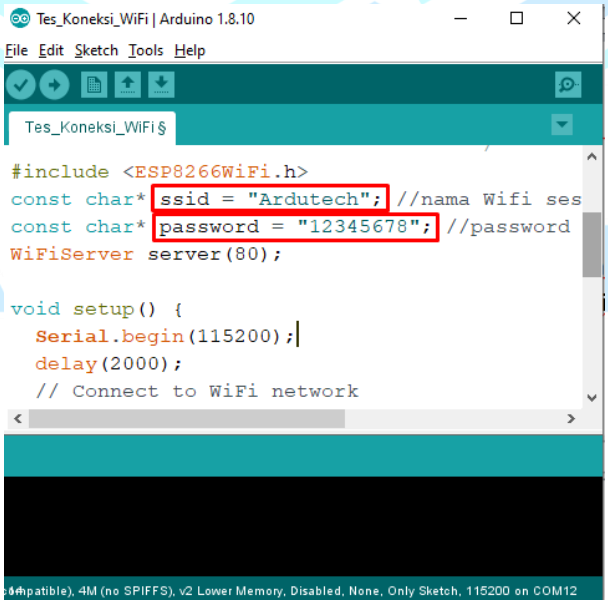
- **Nama : Ardutech**
- **Password : 12345678**

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - Kabel mikro USB
- **Kebutuhan software:**
  - Arduino IDE
- **Program:**

```
/************************************
 * Program : Tes Koneksi WiFi NodeMCU V3
 * Input : -
 * Output : Serial Monitor
 * Trainer Kit IoT Ardutek
 * www.ardutech.com
```

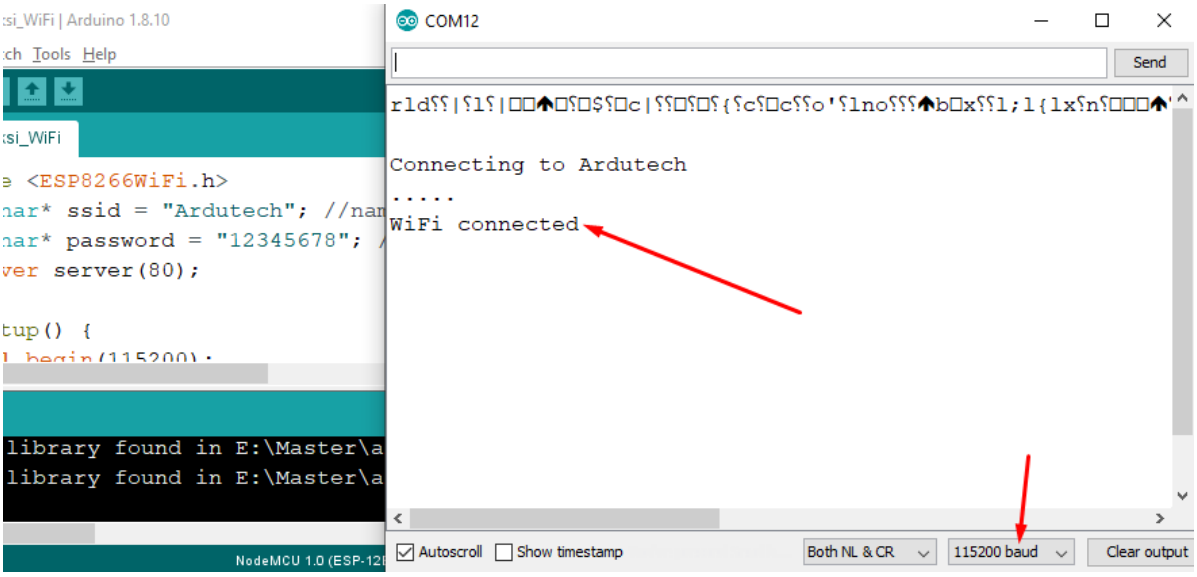
```
 * ********************************/
#include <ESP8266WiFi.h>
const char* ssid = "Ardutech"; //nama Wifi sesuaikan dengan WiFi anda
const char* password = "12345678"; //password WiFi sesuaikan dengan WiFi anda
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(2000);
  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
}

void loop() {
}
```

Ganti nama WiFi (ssid) dan password sesuai dengan jaringan anda.

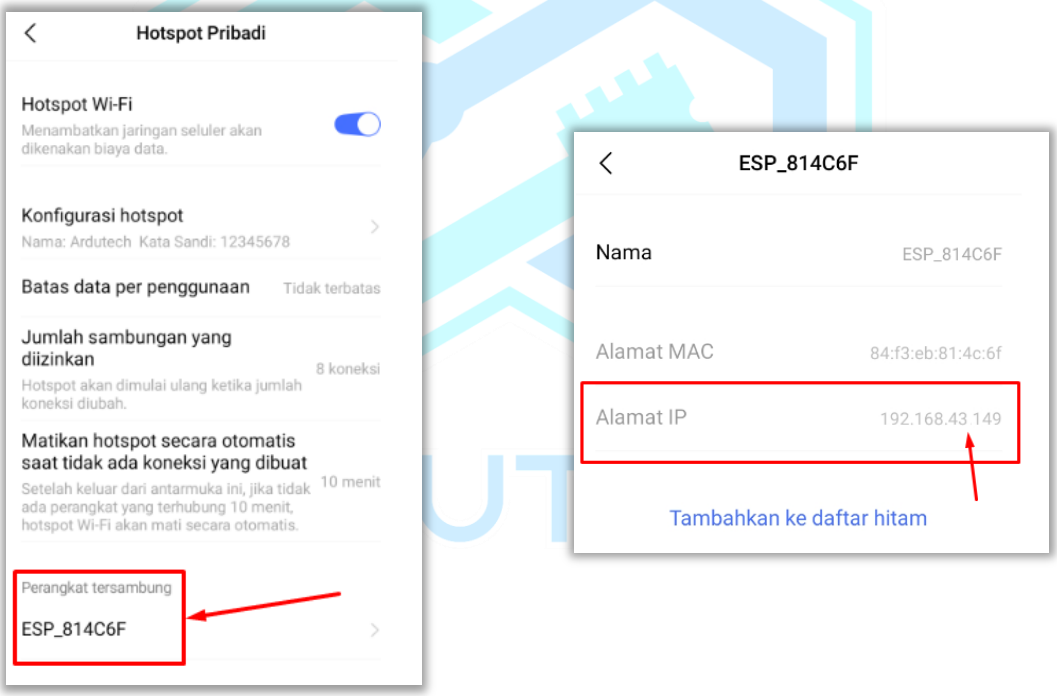


Simpan (***Save***) kemudian ***Upload***.

- **Jalannya Alat:**
  Buka **Serial Monitor** dengan **baudrate 115200** (***Tools Serial Monitor***) maka akan tampil status koneksi NodeMCU dengan WiFi “WiFi connected”.



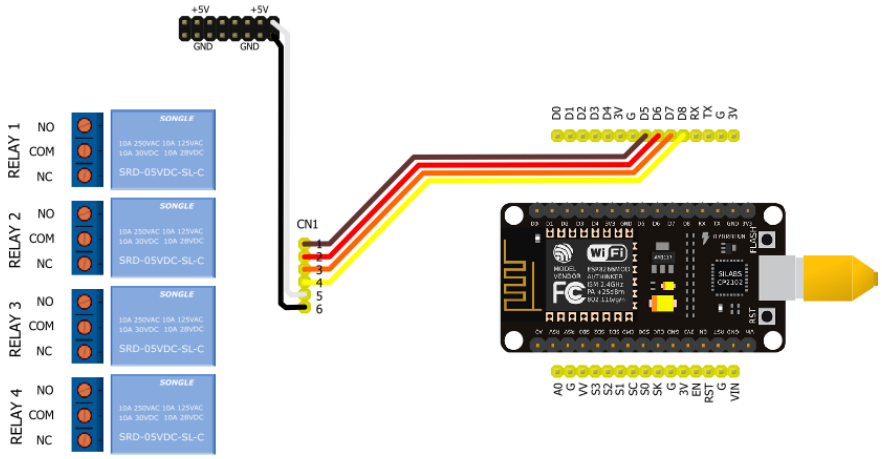
Jika menggunakan hotspot hp bisa anda lihat status koneksinya di setingan hotspot pribadi. Untuk melihat detail nya bisa anda klik nama perangkatnya maka akan mucul alamat MAC dan aalamat IP NodeMCU.



Jika tidak terhubung dengan jaringan WiFi, silakan anda reset dengan menekan tombol RST di NodeMCU dan pastikan nama ssid dan password benar (besar kecil tulisan berpengaruh).

Project IoT 1 : Kontrol 4 Relay dengan Web Server

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - 4 Channel Relay
  - Kabel micro USB
  - 6 Kabel jumper female – female
- **Skematik:**



Keterangan:

CN1 / Relay	NodeMCU
1/Rl1	D5
2/Rl2	D6
3/Rl3	D7
4/Rl4	D8
5/VCC	+5V
6/GND	GND

- **Kebutuhan Software:**
  - Arduino IDE
  - Laptop/Computer/hand phone
- **Program:**

  Ganti nama WiFi (ssid) dan password sesuai dengan jaringan anda.

```
/************************************
 * Program : Project 1 Kontrol 4 Relay
 * Input   : -
 * Output  : 4 Relay
 * Iot Starter Kit Inkubatek
 * www.ardutech.com
 ************************************
 *KONEKSI NODEMCU VS LED
 * NODEMCU  |   Relay
 *    D5   -->    1
 *    D6   -->    2
 *    D7   -->    3
 *    D8   -->    4
 *    5V   -->    5
 *    GND  -->    6
 ************************************/
#include <ESP8266WiFi.h>
const char* ssid = "Ardutech";
const char* password = "12345678";
#define Relay1 D5
#define Relay2 D6
```

```
#define Relay3 D7
#define Relay4 D8

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(Relay1, OUTPUT);
  pinMode(Relay2, OUTPUT);
  pinMode(Relay3, OUTPUT);
  pinMode(Relay4, OUTPUT);
  delay(10);
  digitalWrite(Relay1, LOW);
  digitalWrite(Relay2, LOW);
  digitalWrite(Relay3, LOW);
  digitalWrite(Relay4, LOW);

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");
}

void loop() {
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();
  if (request.indexOf("/relay1on") > 0)  {
```

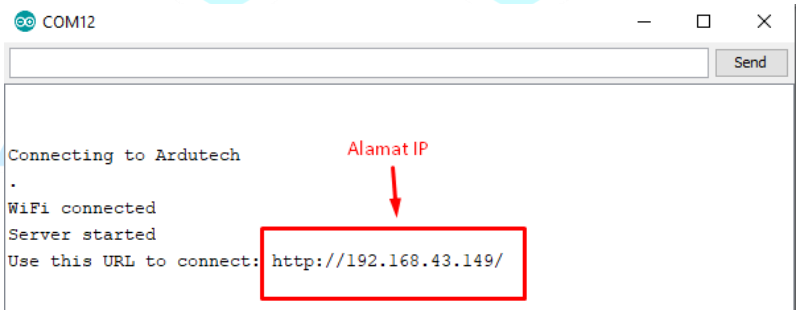
```
  digitalWrite(Relay1, HIGH);
}
if (request.indexOf("/relay1off") >0)  {
  digitalWrite(Relay1, LOW);
}

 if (request.indexOf("/relay2on") > 0)  {
  digitalWrite(Relay2, HIGH);
}
if (request.indexOf("/relay2off") >0)  {
  digitalWrite(Relay2, LOW);
}
  if (request.indexOf("/relay3on") >0)  {
  digitalWrite(Relay3, HIGH);

}
if (request.indexOf("/relay3off") > 0)  {
  digitalWrite(Relay3, LOW);
}
 if (request.indexOf("/relay4on") > 0)  {
  digitalWrite(Relay4, HIGH);
}
if (request.indexOf("/relay4off") > 0)  {
  digitalWrite(Relay4, LOW);
}

client.println ("<!DOCTYPE html> <html>\n");
client.println ("<head><meta name=\"viewport\" content=\"width=device-width,
 initial-scale=1.0, user-scalable=no\">\n");
client.println ("<title>Kontrol 4 Relay</title>\n");
client.println ("<style>html { font-family: Helvetica; display: inline-block;
 margin: 0px auto; text-align: center;color: #444444;}\n");
client.println ("body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto
 30px;} h3 {color: #444444;margin-bottom: 50px;}\n");
client.println (".button {display: block;width: 80px;background-color:
 #1abc9c;border: none;color: white;padding: 13px 30px;text-decoration:
 none;font-size: 25px;margin: 0px auto 35px;cursor: pointer;border-radius:
 4px;}\n");
client.println (".button-on {background-color: #1abc9c;}\n");
client.println (".button-on:active {background-color: #16a085;}\n");
client.println (".button-off {background-color: #34495e;}\n");
client.println (".button-off:active {background-color: #2c3e50;}\n");
client.println ("p {font-size: 14px;color: #888;margin-bottom: 10px;}\n");
client.println ("</style>\n");
client.println ("</head>\n");
client.println ("<body>\n");
client.println ("<h1>Kontrol 4 Relay Web Server (STA Mode)</h1>\n");
client.println ("<h2>www.tokotronik.com</h2>\n");

if(digitalRead(Relay1))
{ client.println ("<p>Relay 1 Status: ON</p><a class=\"button button-off\"
 href=\"/relay1off\">OFF</a>\n");}
else
```

```
    { client.println ("<p>Relay 1 Status: OFF</p><a class=\"button button-on\"
     href=\"/relay1on\">ON</a>\n");}

    if(digitalRead(Relay2))
    { client.println ("<p>Relay 2 Status: ON</p><a class=\"button button-off\"
     href=\"/relay2off\">OFF</a>\n");}
    else
    { client.println ("<p>Relay 2 Status: OFF</p><a class=\"button button-on\"
     href=\"/relay2on\">ON</a>\n");}

    if(digitalRead(Relay3))
    { client.println ("<p>Relay 3 Status: ON</p><a class=\"button button-off\"
     href=\"/relay3off\">OFF</a>\n");}
    else
    { client.println ("<p>Relay 3 Status: OFF</p><a class=\"button button-on\"
     href=\"/relay3on\">ON</a>\n");}

    if(digitalRead(Relay4))
    { client.println ("<p>Relay 4 Status: ON</p><a class=\"button button-off\"
     href=\"/relay4off\">OFF</a>\n");}
    else
    { client.println ("<p>Relay 4 Status: OFF</p><a class=\"button button-on\"
     href=\"/relay4on\">ON</a>\n");}

    client.println ("</body>\n");
    client.println ("</html>\n");
  }
```

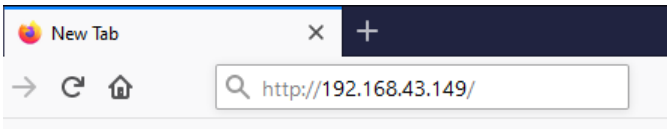
Simpan (***Save***) kemudian ***Upload***.

- **Jalannya Alat:**

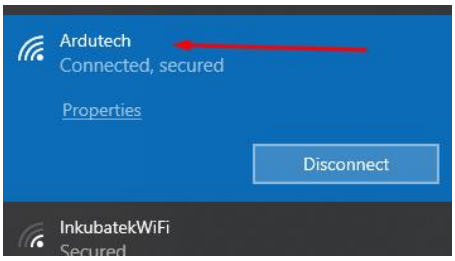
  Buka serial monitor, sehingga menampilkan status koneksi dan alamat IP NodeMCU.



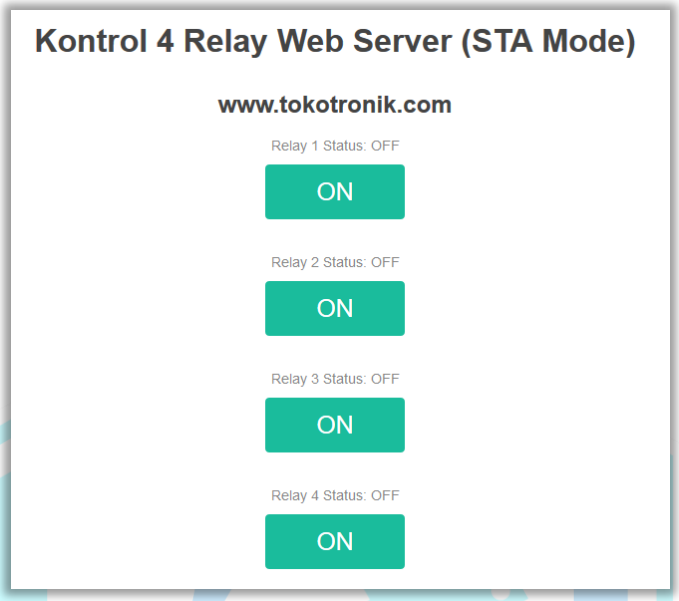
Copy alamat IP nya, pada contoh diatas alamatnya : 192.168.43.149 kemudian paste di web browser anda.



Akses alamat IP tersebut dijaringan WiFi yang sama contoh jaringan WiFi kami Ardutech, sehingga di computer yang kami gunakan terhubung di jaringan Ardutech (sama dengan NodeMCU).

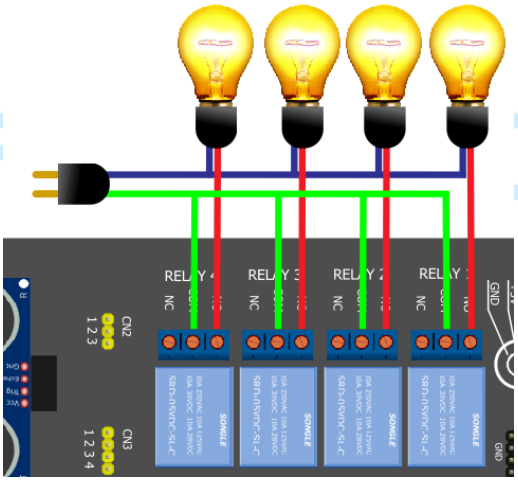


Kemudian ENTER maka akan tampil :



Tekan tombol ON atau OFF pada masing – masing relay maka relay yang bersesuaian akan ON atau OFF.

Apabila ingin mengontrol lampu AC maka bisa anda siapkan 4 buah lampu AC beserta kabel dan stecker (jack), kemudian buatlah rangkaian seperti ini :



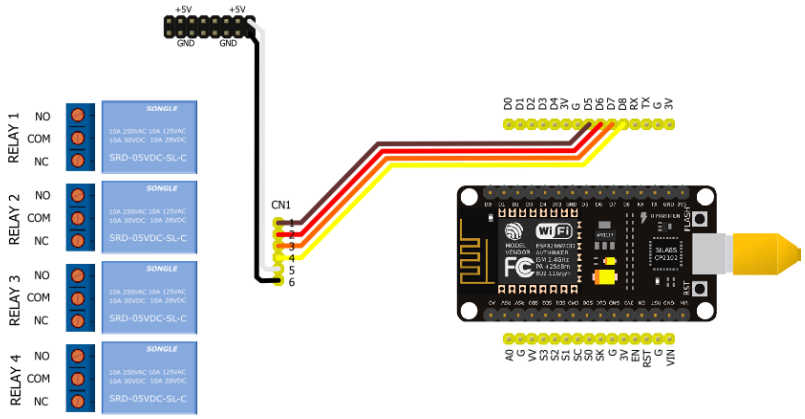
Hati – hati dan perhatikan dengan teliti karena kita akan berhubungan dengan tegangan AC 220V. Anda dapat juga menggantinya dengan lampu DC.

Project 2: Kontrol 4 Relay dengan Aplikasi Android

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - 4 Channel Relay

- Kabel micro USB
- 6 Kabel jumper female – female

- **Skematik:**



Keterangan:

CN1 / Relay	NodeMCU
1/Rl1	D5
2/Rl2	D6
3/Rl3	D7
4/Rl4	D8
5/VCC	+5V
6/GND	GND

- **Kebutuhan Software:**
  - Arduino IDE
  - Smartphone : install aplikasi “Kontrol 4 Relay.Apk “ (file apk ada di CD)
- **Program:**

Ganti nama WiFi (ssid) dan password sesuai dengan jaringan anda.

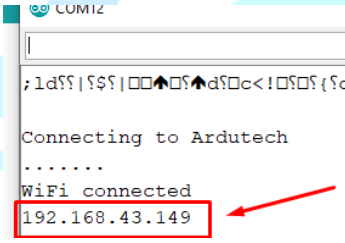
```
/************************************
 * Program : Project 2 Kontrol 4 Relay
 * Input   : -
 * Output  : 4 Relay
 * Iot Starter Kit Inkubatek
 * www.ardutech.com
 ************************************
 *KONEKSI NODEMCU VS RELAY
 * NODEMCU  |   CN1 RELAY
 *    D5    -->    1 / RL1
 *    D6    -->    2 / RL2
 *    D7    -->    3 / RL3
 *    D8    -->    4 / R4
 *    5V    -->    5 / +5V
 *    GND   -->    6 / GND
 ************************************/
#include <ESP8266WiFi.h>
const char* ssid = "Ardutech";
const char* password = "12345678";
```

```
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  pinMode(D5, OUTPUT);
  pinMode(D6, OUTPUT);
  pinMode(D7, OUTPUT);
  pinMode(D8, OUTPUT);

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  server.begin();
  Serial.println(WiFi.localIP());
}
//=======================================
void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }
  // Read the first line of the request
  String req = client.readStringUntil('\r');
  Serial.println(req);
  if (req.indexOf("/R1/1") != -1){
    digitalWrite(D5,HIGH);
  }
  else if (req.indexOf("/R1/0") != -1){
    digitalWrite(D5,LOW);
  }
  else if (req.indexOf("/R2/1") != -1){
    digitalWrite(D6,HIGH);
  }
  else if (req.indexOf("/R2/0") != -1){
    digitalWrite(D6,LOW);
  }
  else if (req.indexOf("/R3/1") != -1){
    digitalWrite(D7,HIGH);
  }
  else if (req.indexOf("/R3/0") != -1){
```

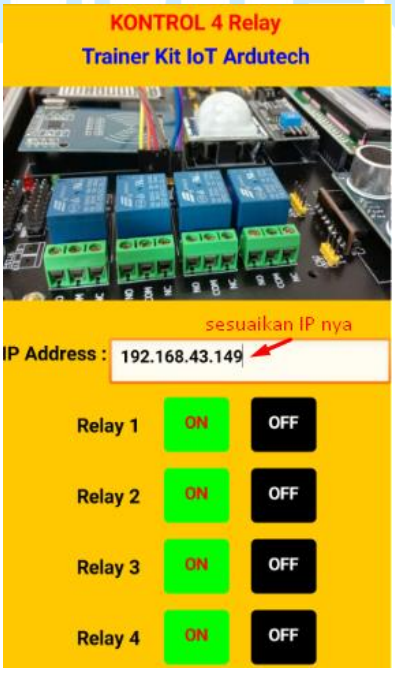
```
    digitalWrite(D7,LOW);
  }
   else if (req.indexOf("/R4/1") != -1){
    digitalWrite(D8,HIGH);
  }
  else if (req.indexOf("/R4/0") != -1){
    digitalWrite(D8,LOW);
  }
  else {
    Serial.println("invalid request");
    client.stop();
    return;
  }

  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println("");
  client.print("Success");
  Serial.println("Client disonnected");
}
```

Simpan (***Save***) kemudian ***Upload***.

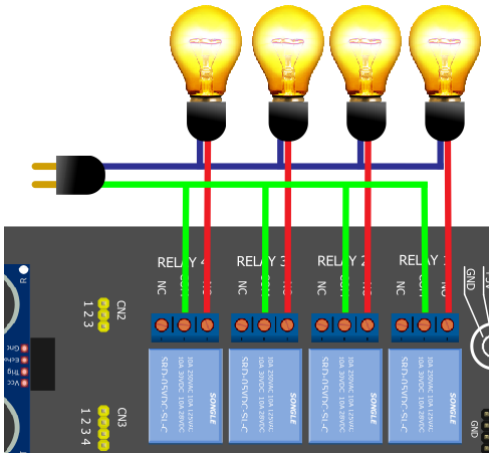
- **Jalannya Alat:**
  Buka serial monitor, sehingga menampilkan status koneksi dan alamat IP NodeMCU.



Sekarang buka aplikasi Android “Kontrol 4 Relay”. Masukan alamat IP nya pada kolom IP Address, pada contoh diatas alamatnya : 192.168.43.149.



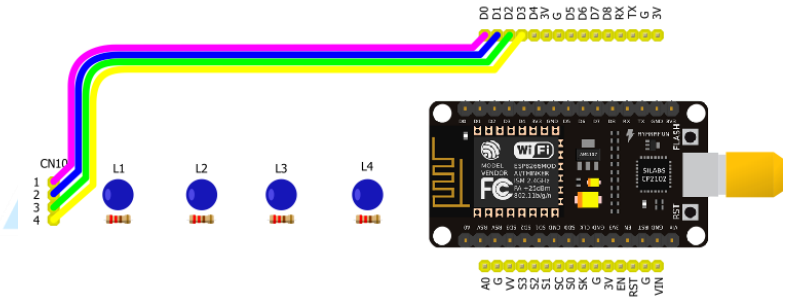
Tekan tombol ON atau OFF pada masing – masing relay maka relay yang bersesuaian akan ON atau OFF. Apabila ingin mengontrol lampu AC maka bisa anda siapkan 4 buah lampu AC beserta kabel dan stecker (jack), kemudian buatlah rangkaian seperti ini :



Hati – hati dan perhatikan dengan teliti karena kita akan berhubungan dengan tegangan AC 220V. Anda dapat juga menggantinya dengan lampu DC.

Project 3: Kontrol 4 LED dengan Suara

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - 4 LED
  - Kabel micro USB
  - 4 Kabel jumper female – female
- **Skematik:**



Keterangan:

CN10 / LED	NodeMCU
1/LED1	D0
2/LED2	D1
3/LED3	D2
4/LED4	D3

- **Kebutuhan Software:**
  - Arduino IDE
  - Smartphone : install aplikasi “Voice Lamp.Apk “ (file apk ada di CD)
- **Program:**

  Ganti nama WiFi (ssid) dan password sesuai dengan jaringan anda.

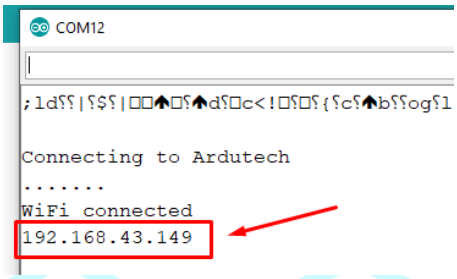
```
/***********************************
 * Program : Project 3 Kontrol 4 LED
 * Input   : -
 * Output  : 4 LED
 * Iot Starter Kit Inkubatek
 * www.ardutech.com
 ***********************************
 *KONEKSI NODEMCU VS LED
 * NODEMCU  |    CN10 LED
 *    D0   -->    1 / LED 1
 *    D1   -->    2 / LED 2
 *    D2   -->    3 / LED 3
 *    D3   -->    4 / LED 1
 ***********************************/
#include <ESP8266WiFi.h>
WiFiClient client;
WiFiServer server(80);
const char* ssid = "Ardutech";
const char* password = "12345678";
String  command =""; // Command received from Android device

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

  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);
  digitalWrite(D0,LOW);
  digitalWrite(D1,LOW);
  digitalWrite(D2,LOW);
  digitalWrite(D3,LOW);

  connectWiFi();
  server.begin();
}

void loop()
{
    client = server.available();
    if (!client) return;
    command = checkClient ();

    if (command == "Nyalakan%20lampu%201" || command == "nyalakan%20lampu%201" ||
   command == "Nyalakan%20lampu%20satu"|| command == "nyalakan%20lampu%20satu")
   digitalWrite(D0,HIGH);
    else if (command == "Matikan%20lampu%201" || command == "matikan%20lampu%201"||
   command == "Matikan%20lampu%20satu"|| command == "matikan%20lampu%20satu")
   digitalWrite(D0,LOW);
    else if (command == "Nyalakan%20lampu%202" || command == "nyalakan%20lampu%202"
   || command == "Nyalakan%20lampu%20dua"|| command == "nyalakan%20lampu%20dua")
   digitalWrite(D1,HIGH);
```

```
    else if (command == "Matikan%20lampu%202" || command == "matikan%20lampu%202"||
   command == "Matikan%20lampu%20dua"|| command == "matikan%20lampu%20dua")
   digitalWrite(D1,LOW);
    else if (command == "Nyalakan%20lampu%203" || command == "nyalakan%20lampu%203"
   || command == "Nyalakan%20lampu%20tiga"|| command == "nyalakan%20lampu%20tiga")
   digitalWrite(D2,HIGH);
    else if (command == "Matikan%20lampu%203" || command == "matikan%20lampu%203"||
   command == "Matikan%20lampu%20tiga"|| command == "matikan%20lampu%20tiga")
   digitalWrite(D2,LOW);
    else if (command == "Nyalakan%20lampu%204" || command == "nyalakan%20lampu%204"
   || command == "Nyalakan%20lampu%20empat"|| command == "nyalakan%20lampu%20empat")
   digitalWrite(D3,HIGH);
    else if (command == "Matikan%20lampu%204" || command == "matikan%20lampu%204"||
   command == "Matikan%20lampu%20empat"|| command == "matikan%20lampu%20empat")
   digitalWrite(D3,LOW);

    sendBackEcho(command); // send command echo back to android device
    command = "";
}
/* connecting WiFi */
void connectWiFi()
{
  Serial.println("Connecting to WIFI");
  WiFi.begin(ssid, password);
  while ((!(WiFi.status() == WL_CONNECTED)))
  {
    delay(300);
    Serial.print("..");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("NodeMCU Local IP is : ");
  Serial.println((WiFi.localIP()));
}

/* check command received from Android Device */
String checkClient (void)
{
  while(!client.available()) delay(1);
  String request = client.readStringUntil('\r');
  Serial.println(request);
  request.remove(0, 5);
  request.remove(request.length()-9,9);
  Serial.println(request);
  return request;
}

/* send command echo back to android device */
void sendBackEcho(String echo)
{
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println("");
```

```
  client.print("Success");
  delay(1);
  Serial.println("Client disonnected");
  Serial.println("");
}
```

Simpan (***Save***) kemudian ***Upload***.

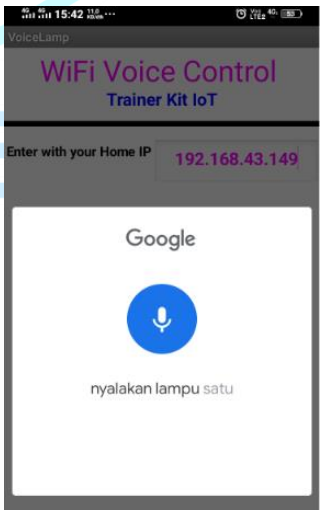
- **Jalannya Alat:**
  Buka serial monitor, sehingga menampilkan status koneksi dan alamat IP NodeMCU.



Sekarang buka aplikasi Android "Voice Lamp". Masukan alamat IP nya pada kolom IP Address, pada contoh diatas alamatnya : 192.168.43.149. Tekan tombol mic untuk perintah kontrol LED sebagai berikut:

Perintah	Kontrol
Nyalakan lampu satu	LED 1 menyala
Matikan lampu satu	LED 1 padam
Nyalakan lampu dua	LED 2 menyala
Matikan lampu dua	LED 2 padam

Perintah	Kontrol
Nyalakan lampu tiga	LED 3 menyala
Matikan lampu tiga	LED 3 padam
Nyalakan lampu empat	LED 4 menyala
Matikan lampu empat	LED 4 padam

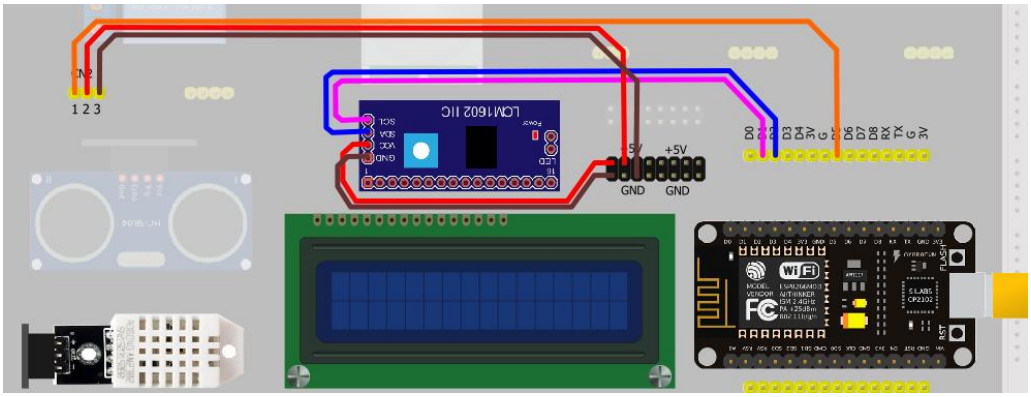


Project 4: Monitoring Suhu dan Kelembaban dengan Thingspeak

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - DHT22 (library DHT harus sudah di tambahkan, cara menambahkan library lihat tutorial "Cara menambahkan library ke Arduino IDE.pdf" ada di CD)

- LCD 2x16
- Kabel micro USB
- 3 dan 4 Kabel jumper female – female

- **Skematik:**



Keterangan:

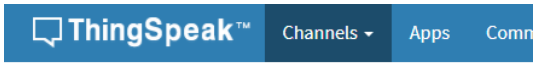
DHT22	CN12 / DHT22	NodeMCU
	1/OUT	D5
	2/+5V	+5V
	3/GND	GND

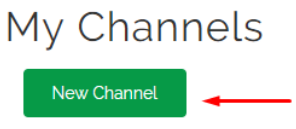
LCD I2C	Konektor 4 pin	NodeMCU
	SCL	D1
	SDA	D2
	VCC	+5V
	GND	GND

- **Kebutuhan Software:**
  - Arduino IDE
  - Laptop/computer : Thingspeak (tutorial Thingspeak lihat CD "Tutorial memulai thingspeak.pdf")

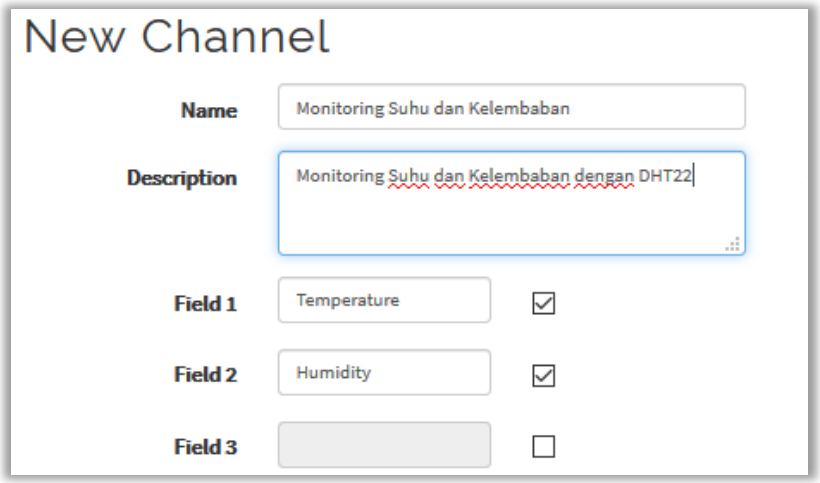
Buat channel baru pada thingspeak:

- Jika sudah punya akun thingspeak langsung saja kita buat channel baru. Klik "***New Chanel***"

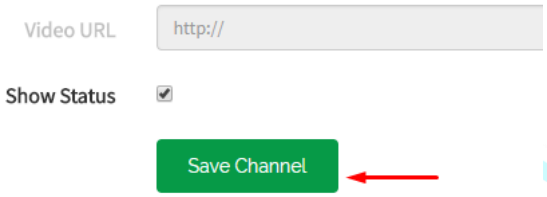




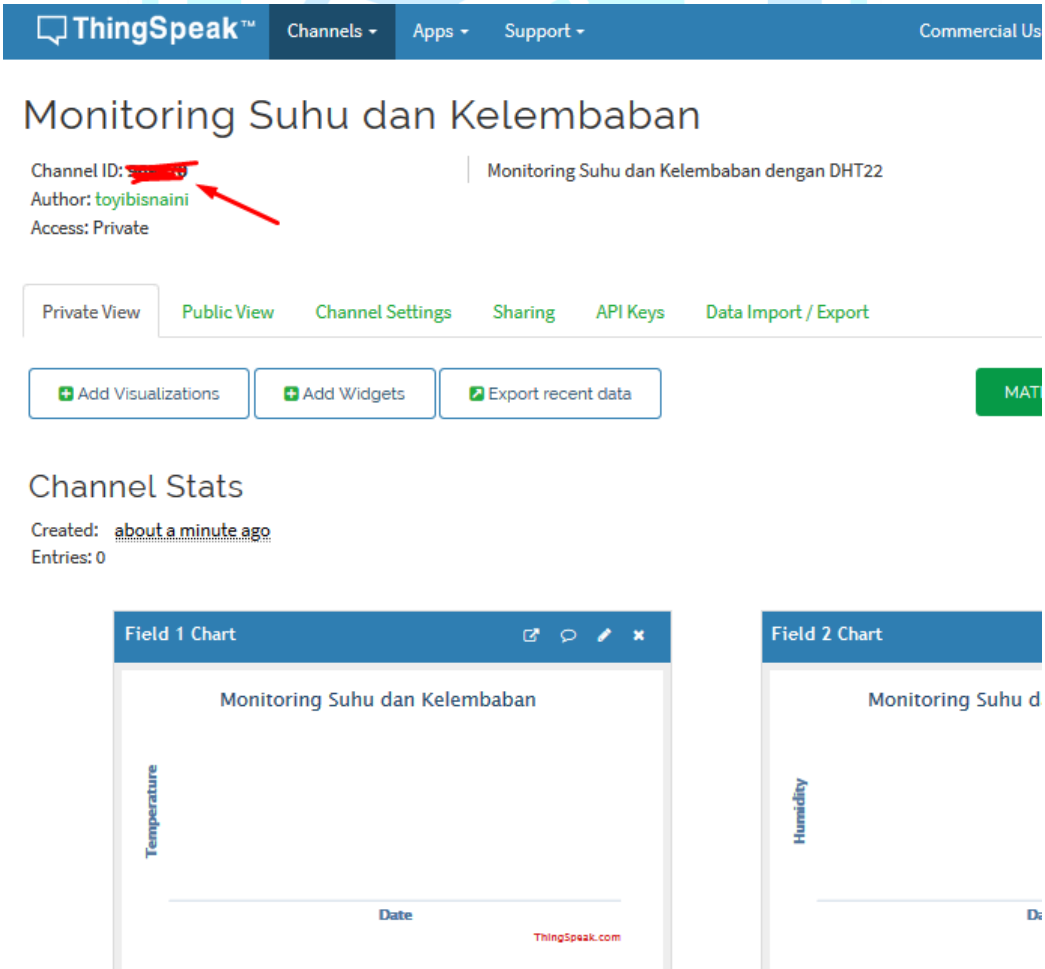
- Berikutnya isi kolom ***Name, Description dan Field 1***.



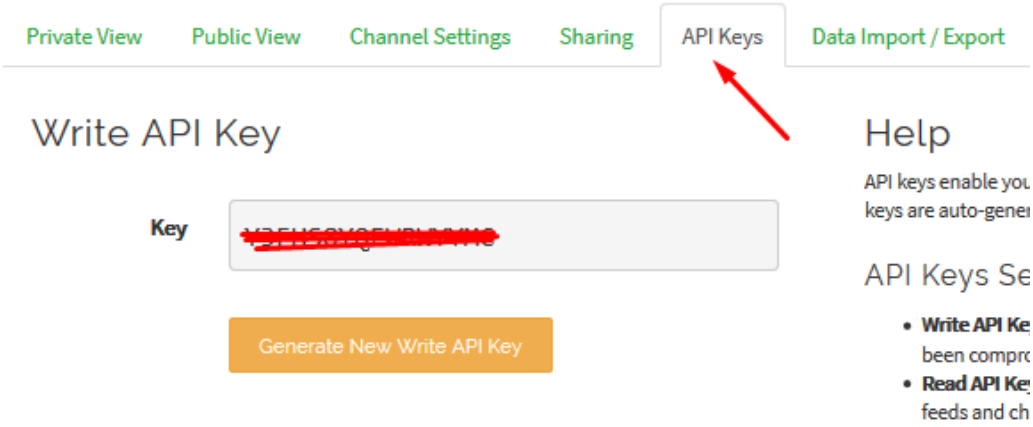
- OK, selanjutnya klik "***Save Channel***"



- Berikutnya akan tampil sebuah grafik (chart), Perhatikan dan catat Channel ID-nya , nanti kita pakai untuk membuat programnya



- Langkah selanjutnya kita cek API , klik tab ***API Keys***.



- Perhatikan dan catat kode API Keys-nya , nanti kita pakai untuk membuat programnya.
- Channel thingspeak siap.

- **Program:**

  Sesuaikan dengan setingan Anda:
  - Nama WiFi (ssid)
  - Password (pass)
  - Channel ID (myChannelNumber)
  - Write API Key (myWriteAPIKey)

```
char ssid[] = "namaWiFi";
char pass[] = "passwordWiFi";
WiFiClient  client;

unsigned long myChannelNumber = 000000;
const char * myWriteAPIKey = "75DSG3UWHQKMBL9W";
```

```
/************************************
 * Program : Project 4 Monitoring suhu dan Kelemababan dengan Thingspeak
 * Input   : DHT22
 * Output  : LCD
 * Iot Starter Kit Inkubatek
 * www.ardutech.com
 ************************************
 * KONEKSI NODEMCU dg LCD
 * NODE MCU  |  LCD I2C
 *    D1   -->    SCL
 *    D2   -->    SDA
 *    power: 5V GND
 ************************************
 * KONEKSI NODEMCU dg DHT22
 * NODEMCU  |   DHT22
 *    D3   -->    1
 *    +5   -->    2
 *   GND   -->    3
 ************************************/
#include <DHT.h>
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
#include <Wire.h>
```

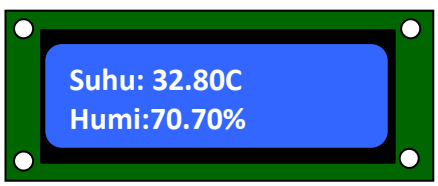
```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 16, 2);

char ssid[] = "Ardutech";
char pass[] = "12345678";
WiFiClient  client;

unsigned long myChannelNumber = 909403;
const char * myWriteAPIKey = "75DSG3UWHQKMBL9W";

#define DHTPIN D3
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

byte delaySend;

void setup() {
  lcd.begin();
  lcd.backlight();
  lcd.print("Koneksi WiFi");
  lcd.setCursor(0,1);
  lcd.print(ssid);
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);
  if(WiFi.status() != WL_CONNECTED){
    Serial.println(" ");
    Serial.println(" ");
    Serial.print("Attempting to connect to SSID: ");
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);
      Serial.print(".");
      delay(5000);
    }
    Serial.println("\nConnected.");
    lcd.print(" Oke...!");
  }
  delay(2000);
  lcd.clear();
  dht.begin();
}
//======================
void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("DHT22 tidak terbaca... !");
    lcd.setCursor(0,0);
    lcd.print("     DHT22    ");
    lcd.setCursor(0,1);
    lcd.print(" tidak terbaca");
    return;
```

```
  }
  lcd.setCursor(0,0);
  lcd.print("Suhu:");
  lcd.print(t);
  lcd.print("C  ");
  lcd.setCursor(0,1);
  lcd.print("Humi:");
  lcd.print(h);
  lcd.print("%  ");
  delay(1000);
  delaySend++;
  if (delaySend<20)return;

  lcd.clear();
  lcd.print("  Update DHT22");
  lcd.setCursor(0,1);
  lcd.print("ke Thingspeak...");
  ThingSpeak.setField(1,t);
  ThingSpeak.setField(2,h);
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
    lcd.clear();
    lcd.print("Successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
    lcd.clear();
    lcd.print("ERROR.....");
  }
  delaySend=0;
  delay(3000);
}
```

Simpan (***Save***) kemudian ***Upload***.

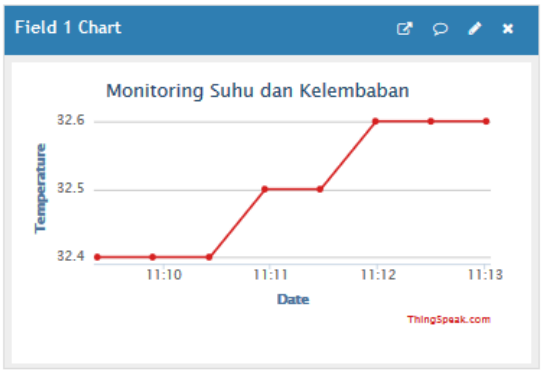
- **Jalannya Alat:**
  - LCD menampilkan koneksi WiFi, tunggu sampai OKE.

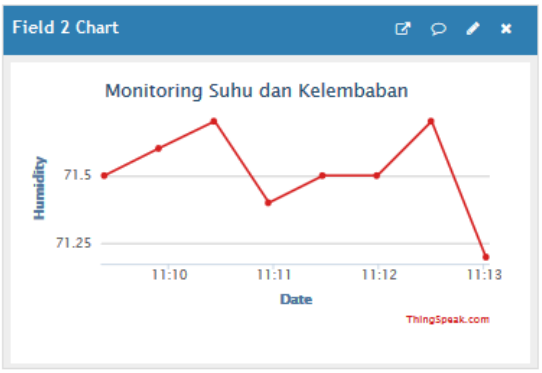
  - Selanjutnya LCD menampilkan nilai pembacaan suhu dan kelembaban



  - Buka web ***thingspeak.com*** dan masuk ke akun anda dengan user name + password yang sudah dibuat.

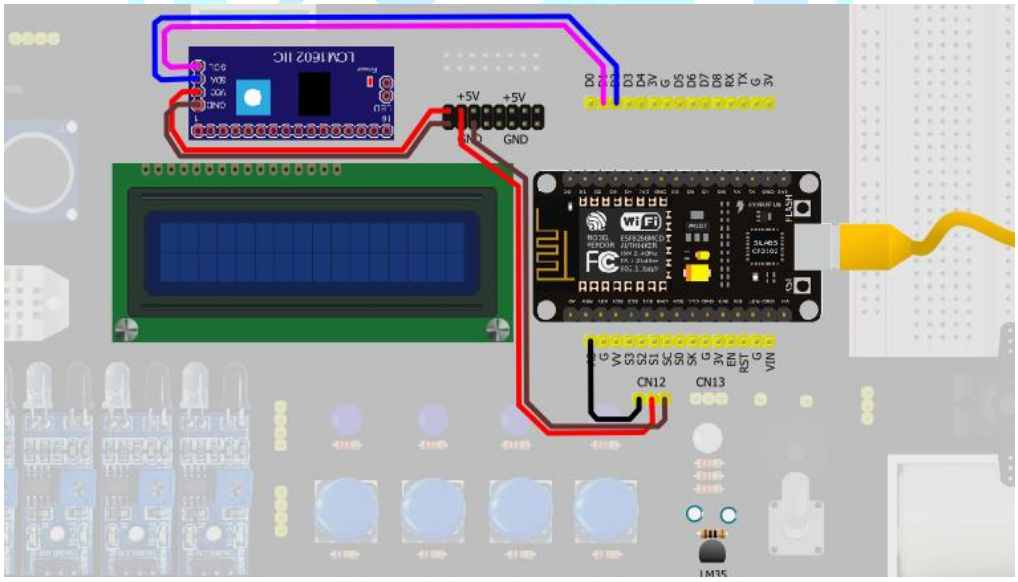
- Buka channel “Monitoring Suhu dan Kelembaban”
- Setelah 20 detik data suhu dan kelembaban dikirim ke channel thingspeak.





Project 5: Monitoring Suhu Adafruit IO

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - Sensor Suhu LM35
  - LCD 2x16
  - Kabel micro USB
  - 3 dan 4 Kabel jumper female – female
- **Skematik:**



Keterangan:

LM35	CN12 / LM35	NodeMCU
	1/OUT	A0
	2/+5V	+5V
	3/GND	GND

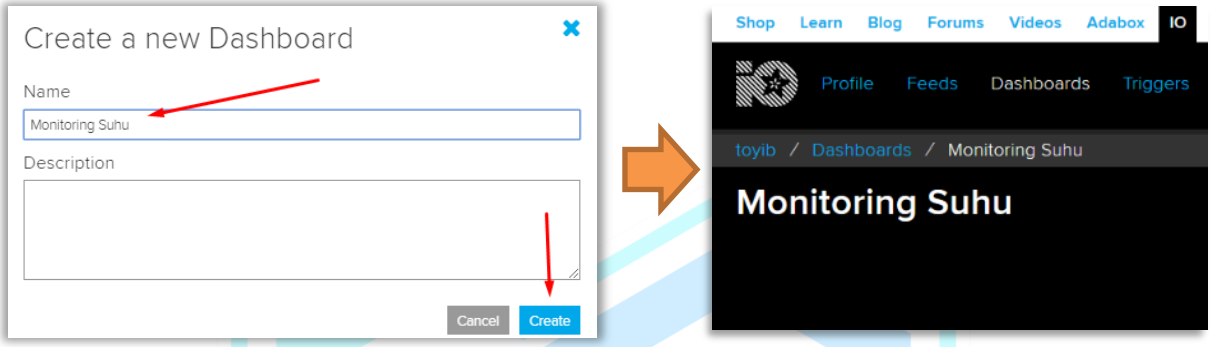
LCD I2C	Konektor 4 pin	NodeMCU
	SCL	D1

	SDA	D2
	VCC	+5V
	GND	GND

- **Kebutuhan Software:**
  - Arduino IDE
  - Laptop/computer : Adafruit IO (tutorial Adafruit IO lihat CD "Tutorial memulai Adafruit.pdf")

  Buat Dashboard:

  - Buat dashboard dengan nama "***Monitoring Suhu***"



- **Program:**

  Sesuaikan dengan setingan Anda:
  - SSID (nama WiFi Anda)
  - PASS (password WiFi Anda)
  - AIO_USERNAME (user name bisa lihat https://accounts.adafruit.com)
  - AIO_KEY(diisi active Key Anda)

```
/************************************************
 * Program : Project 5 Monitoring suhu dengan Adafruit IO
 * Input   : LM35
 * Output  : LCD
 * Server  : Adafruit IO
 * Trainer Kit IoT
 * www.ardutech.com
 ************************************
 * KONEKSI NODEMCU dg LCD
 * NODE MCU  |  LCD I2C
 *    D1   -->    SCL
 *    D2   -->    SDA
 *    power: 5V GND
 ************************************
 * KONEKSI NODEMCU dg LM35
 * NODEMCU  |   LM35
 *    A0   -->    1
 *    +5   -->    2
 *   GND   -->    3
 ************************************************/
#include <ESP8266WiFi.h>
```

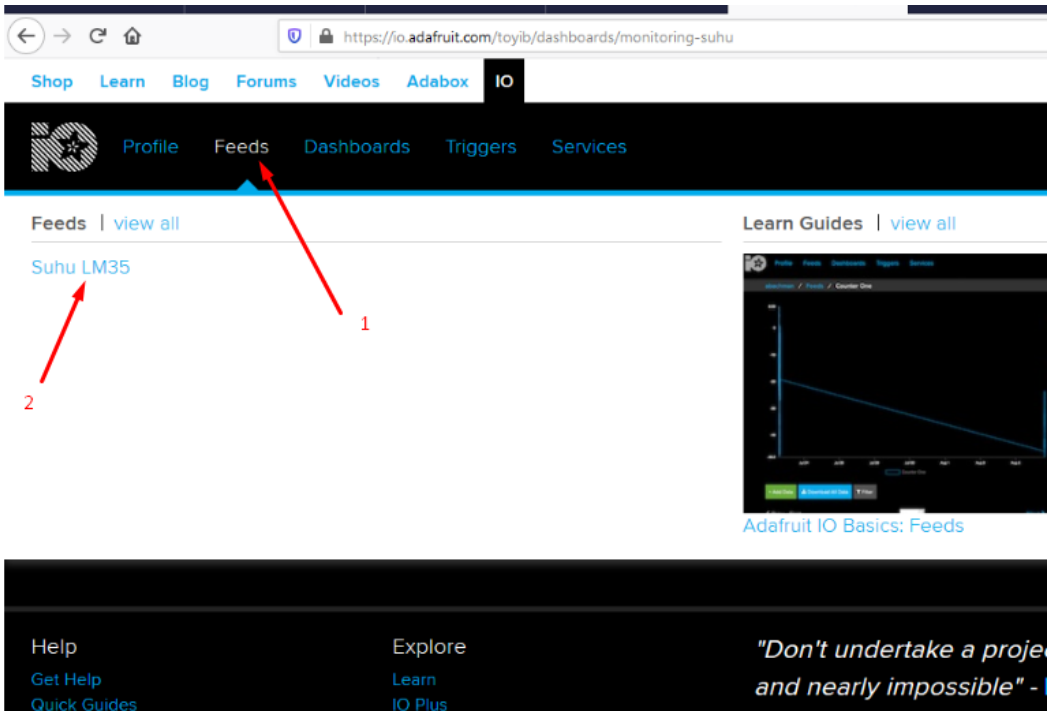
```
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 16, 2);
// Lamp pin
const int lamp_pin = 4;
int analogValue;
float millivolts,celsius;
byte delaySend;

/************************ WiFi Access Point *********************************/
#define SSID "-----namaWiFi-------"
#define PASS "-----passwordWiFi---"

/************************ Adafruit.io Setup *********************************/
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                   // use 8883 for SSL
#define AIO_USERNAME    "----namapengguna-----"//username
#define AIO_KEY         "------AIOKEY---------"

/************ Global State (you don't need to change this!) ******************/
// Create an ESP8266 WiFiClient class to connect to the MQTT server.
WiFiClient client;
// Setup the MQTT client class by passing in the WiFi client and MQTT server and
   login details.
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME,
   AIO_USERNAME, AIO_KEY);

/****************************** Feeds ***************************************/
// Setup a feed called 'photocell' for publishing.
// Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>
Adafruit_MQTT_Publish suhu = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Suhu
   LM35");

void setup() {
  lcd.begin();
  lcd.backlight();
  lcd.print("Koneksi WiFi");
  lcd.setCursor(0,1);
  lcd.print(SSID);
  pinMode(lamp_pin, OUTPUT);
  Serial.begin(115200);
  delay(10);
  Serial.println(F("Adafruit MQTT demo"));
  // Connect to WiFi access point.
  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(SSID);
  WiFi.begin(SSID, PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```
    Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
  lcd.print(" Oke...!");
  delay(3000);
  lcd.clear();
}

void loop() {
  MQTT_connect();
  analogValue = analogRead(A0);
  millivolts = (analogValue/1024.0) * 3300;
  celsius = millivolts/10;
  lcd.setCursor(0,0);
  lcd.print("Suhu:");
  lcd.print(celsius);
  lcd.print("C   ");
  delay(1000);
  delaySend++;
  if (delaySend<20)return;//tunda kirim data 20detik
  delaySend=0;
  lcd.setCursor(0,0);
  lcd.print("kirim data suhu");
  Serial.print(F("\nSending celsius val "));
  Serial.print(celsius);
  Serial.print("...");

  if (! suhu.publish(celsius)) {
    Serial.println(F("Failed"));
  } else {
    Serial.println(F("OK!"));
  }
  mqtt.processPackets(1000);
  lcd.clear();
}

// Function to connect and reconnect as necessary to the MQTT server.
// Should be called in the loop function and it will take care if connecting.
void MQTT_connect() {
  int8_t ret;
  // Stop if already connected.
  if (mqtt.connected()) {
    return;
  }
  Serial.print("Connecting to MQTT... ");
  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
       Serial.println(mqtt.connectErrorString(ret));
       Serial.println("Retrying MQTT connection in 10 seconds...");
       mqtt.disconnect();
       delay(10000);  // wait 10 seconds
```

```
      retries--;
      if (retries == 0) {
        // basically die and wait for WDT to reset me
        while (1);
      }
  }
  Serial.println("MQTT Connected!");
}
```

Simpan (***Save***) kemudian ***Upload***.

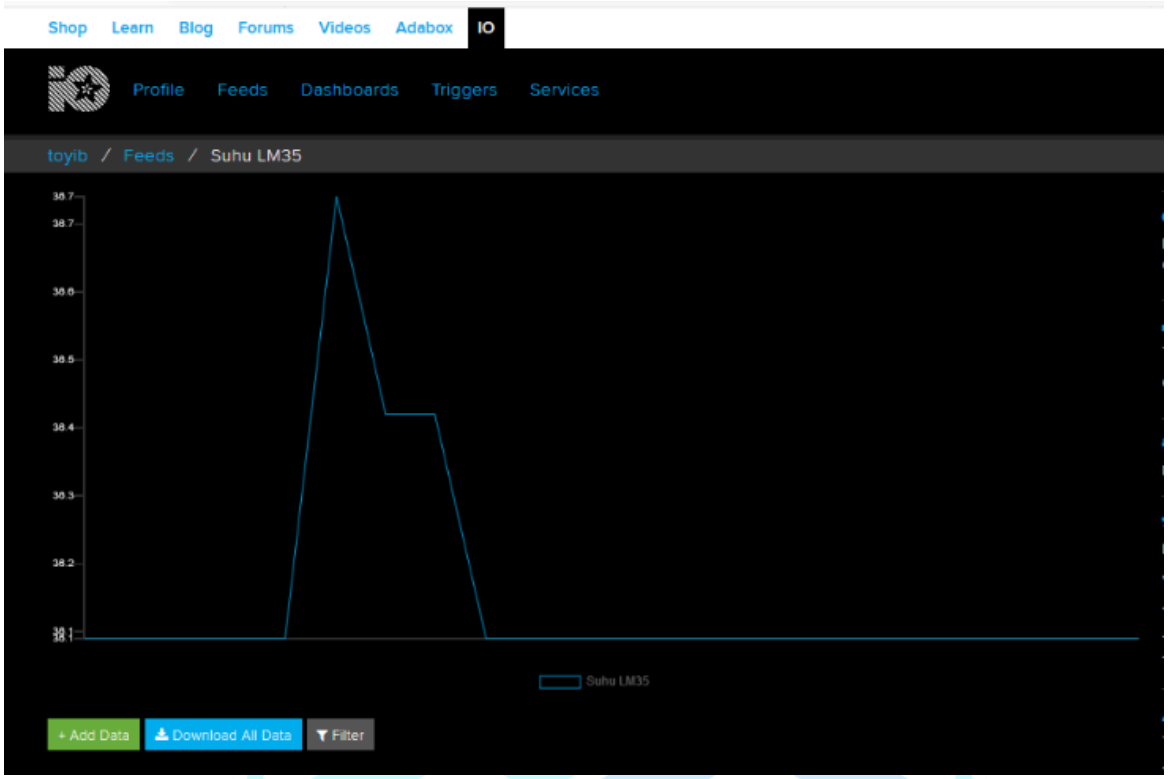
Selanjutnya kembali ke Adafruit IO buka Feeds kemudian klik Suhu LM35



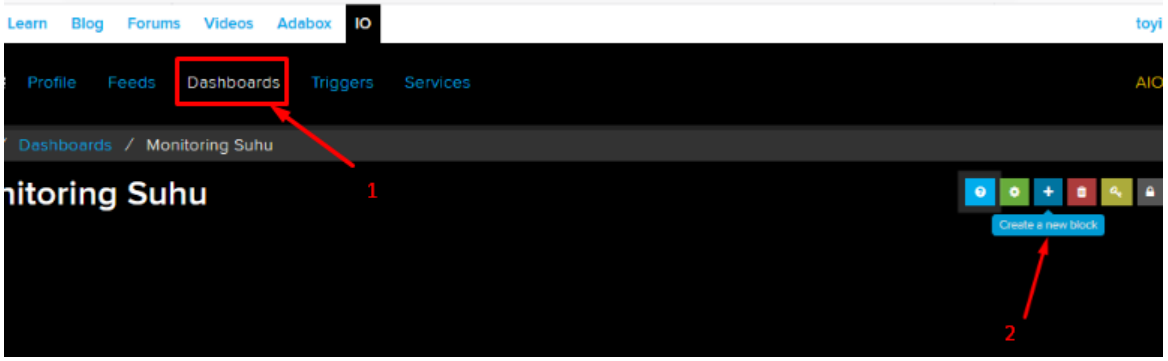
Feeds “Suhu LM35” dibuat dari program.

```
/****************************** Feeds ***************************************/
// Setup a feed called 'photocell' for publishing.
// Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>
Adafruit_MQTT_Publish suhu = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Suhu LM35");

void setup() {
  lcd.begin();
```

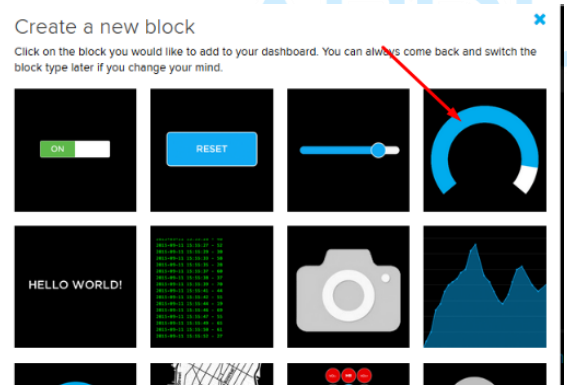
Klik Feeds Suhu LM35, sehingga menampilkan suhu yang terbaca berupa grafik perubahan suhu.

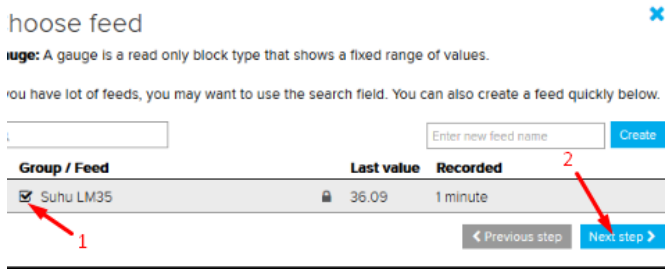


Buka dashboard, pilih button Create a new block (+)



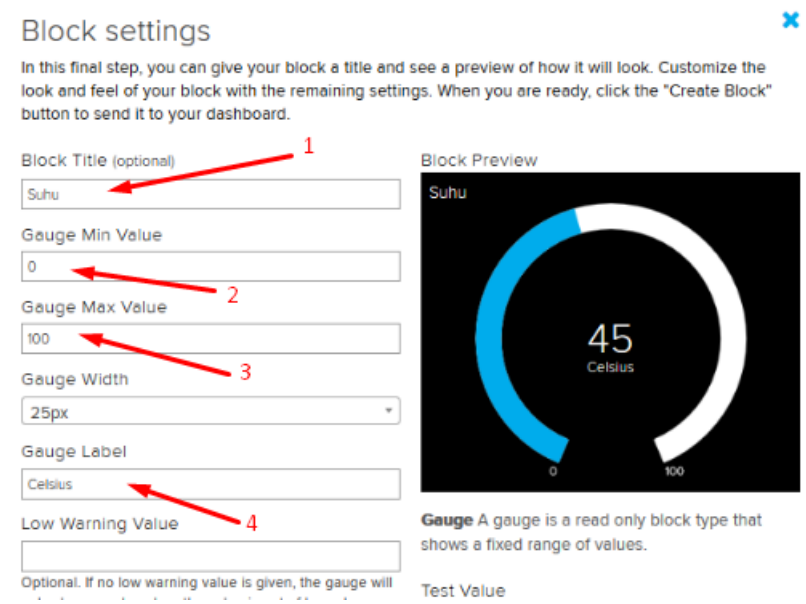
Pilih gauge, selanjutnya beri centang pada cek box Suhu LM35. Kemudian pilih Next step



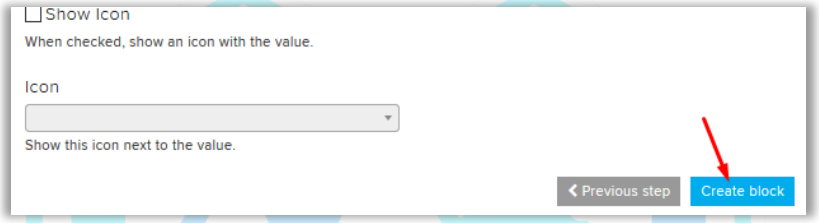


Berikutnya masuk pada Block setting ganti keterangan dan parameter gauge

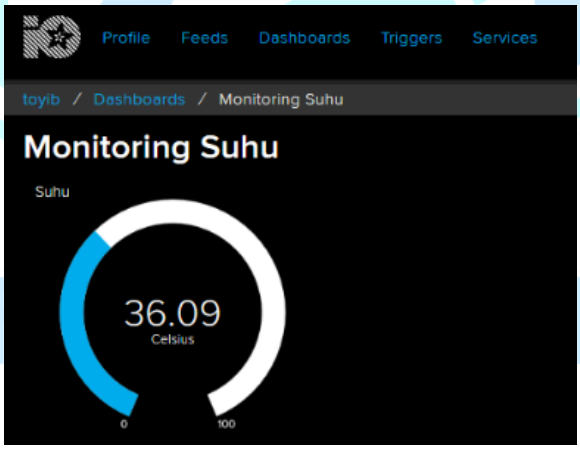
1. Block Title : Suhu
2. Gauge Min Value : 0
3. Gauge Max Value : 100
4. Gauge Width : 25px



Jika sudah terisi klik Create block.

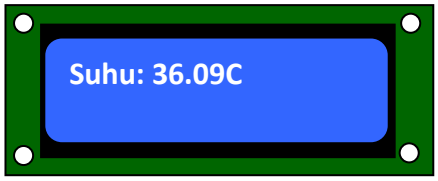


Jika sudah maka dashboard akan menampilkan suhu LM35 dalam bentuk gauge.

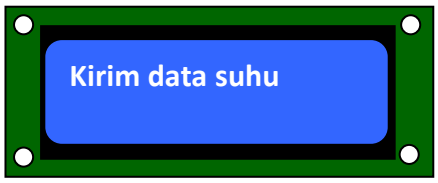


- **Jalannya Alat:**
  - LCD menampilkan koneksi WiFi, tunggu sampai OKE.

  - Selanjutnya LCD menampilkan nilai pembacaan suhu

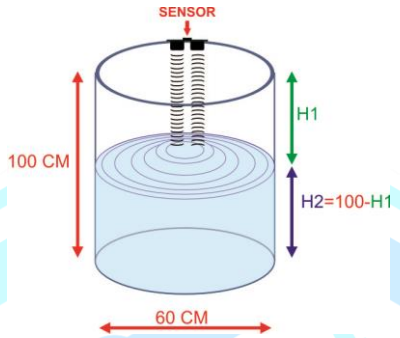


- Setelah 20 detik kirim data suhu

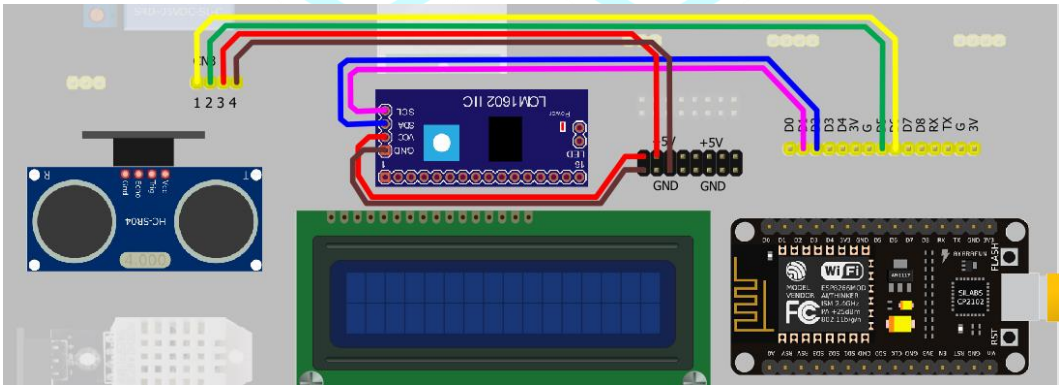


Project 6: Monitoring Level Cairan dengan RemotXY

Prinsip dasar : kita ukur tinggi permukaan air, trus dikalikan dengan luas alas tangki, nah ketemu deh volumenya.



- **Kebutuhan Hardware:**
  - NodeMCU V3
  - Sensor Ultrasonic HC-SR04
  - LCD 2x16
  - Kabel micro USB
  - 4 dan 4 Kabel jumper female – female
- **Skematik:**



Keterangan:

HC-SR04	CN3	NodeMCU
	1/ ECHO	D6
	2/TRIG	D5
	3/+5V	+5V
	4/GND	GND

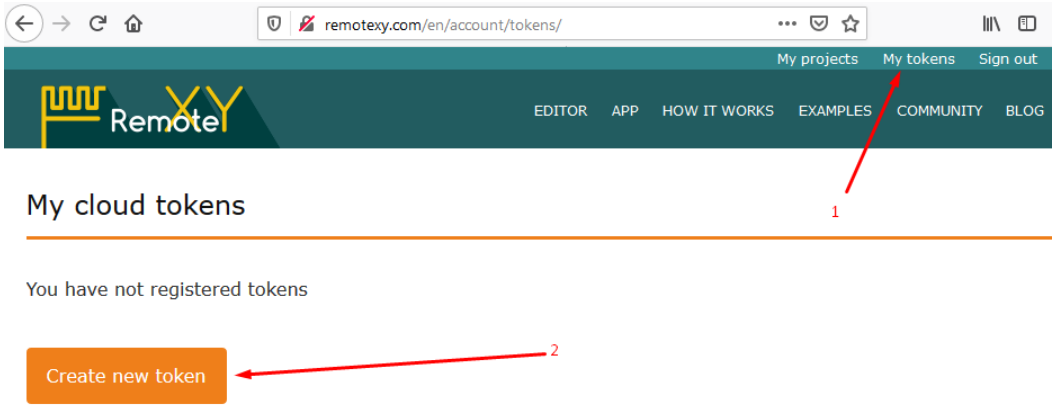
LCD I2C	Konektor 4 pin	NodeMCU

	SCL	D1
	SDA	D2
	VCC	+5V
	GND	GND

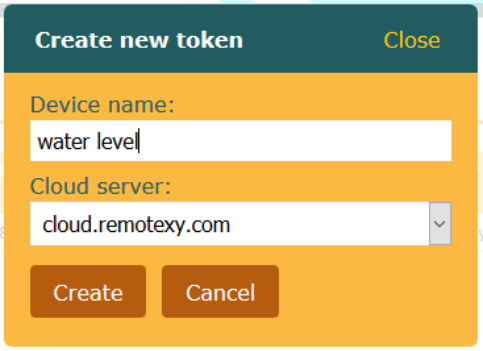
- **Kebutuhan Software:**
  - Arduino IDE
  - Laptop/computer : RemoteXY (tutorial RemoteXY lihat CD "Tutorial memulai RemoteXY.pdf")
  - Smartphone : RemoteXY (install dari play store atau file APK dari CD)

  RemotXY:

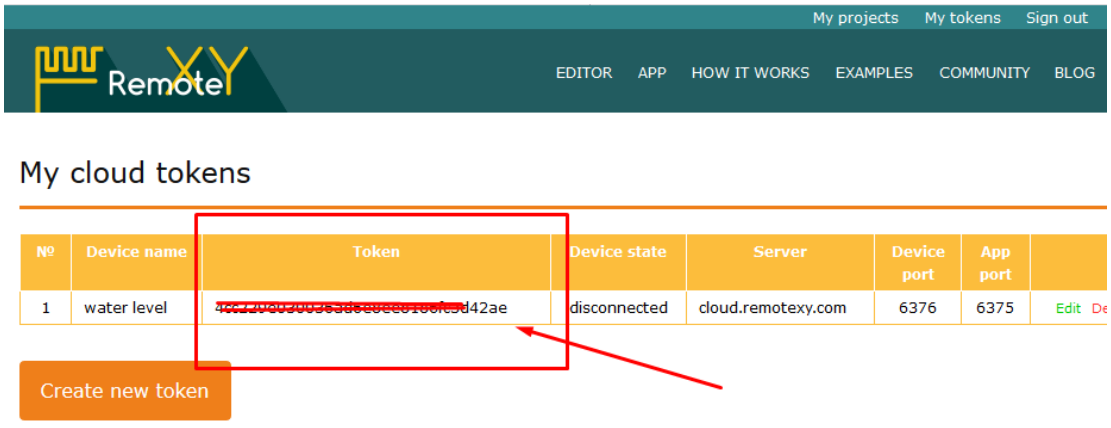
  - Buka remote XY dilaptop/computer lewat web www.remotexy.com
  - Buat token klik **My token** kemudian klik button **Create new token**



  - Buat token dengan nama "***water level***"

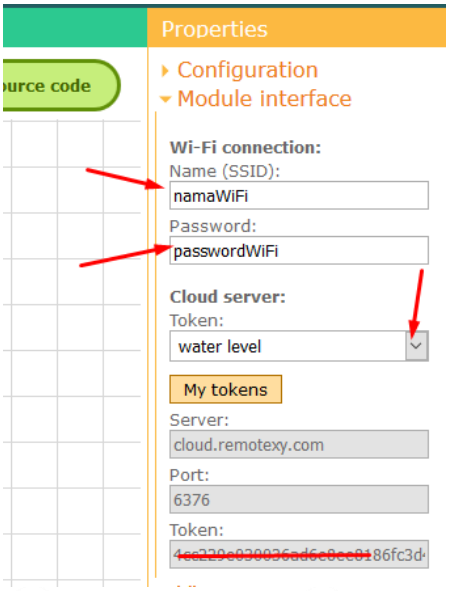


  - Catat kode token yang nanti akan kita masukkan pada program dan aplikasi smartphone

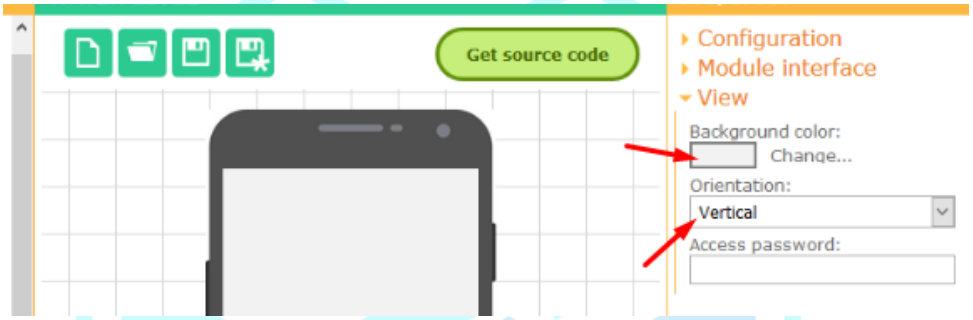


  - Selanjutnya kembali pada tab editor.

- Seting koneksi WiFi (SSID dan password) dan pilih token water level.

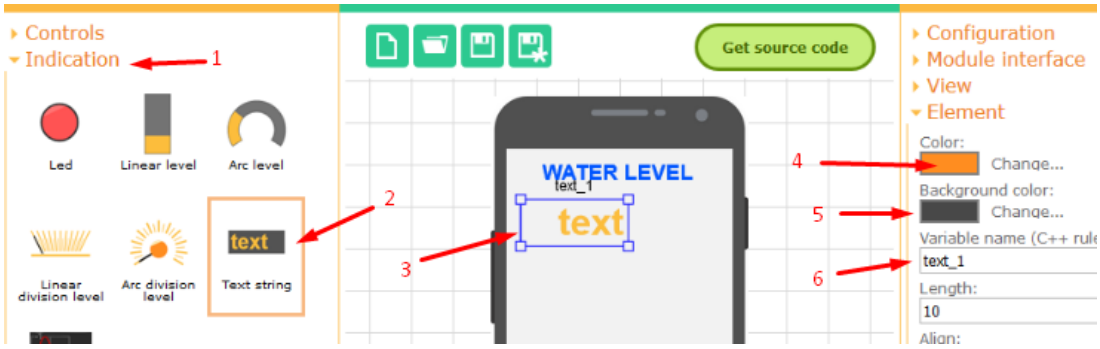


- Setting background color sesuai selera kemudian orientation pilih vertical.

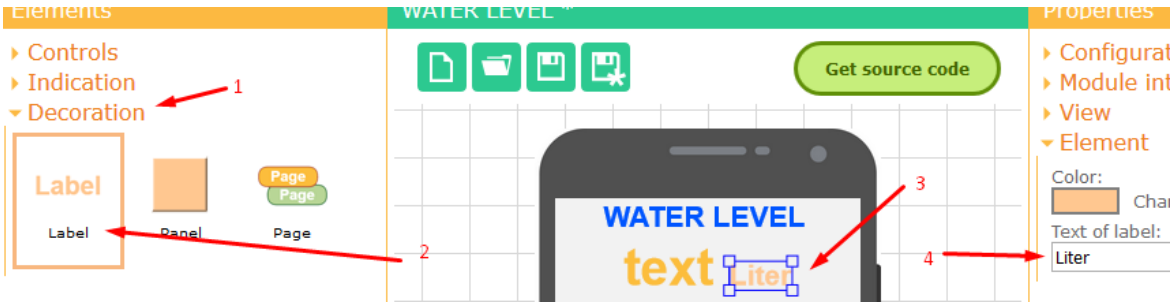


- Buat label tulisan water level pada display atas. Seret label → posisikan tengah atas → ganti warna biru → text of label di ganti WATER LEVEL.

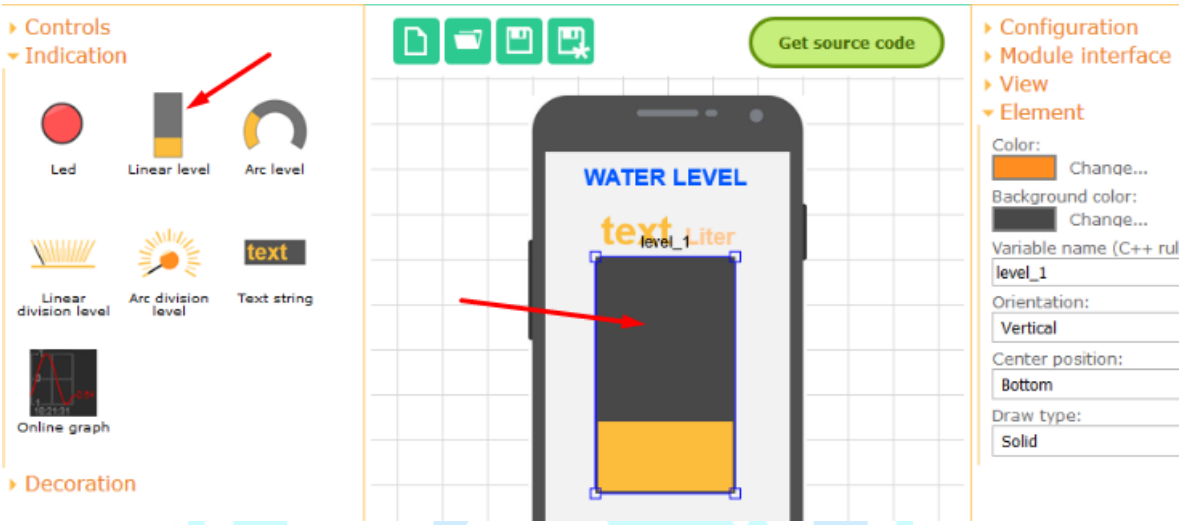
- Buat text string. Klik indication → Seret Text string → posisikan dibawah WATER LEVEL → ganti warna orange → text of label di ganti WATER LEVEL.



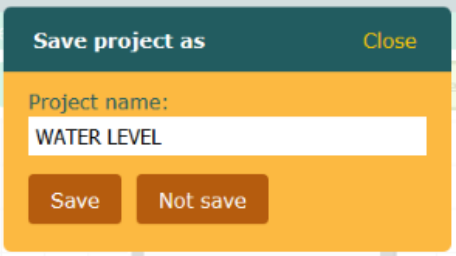
- Selanjutnya buat satuan liter



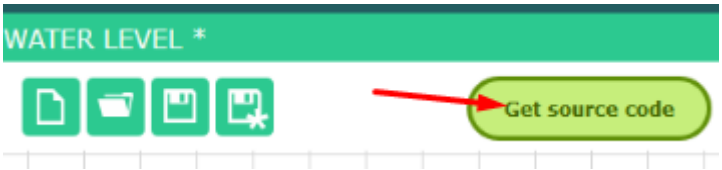
- Berikutnya tampilkan bar perubahan level air. Seret linier level → atur posisi ditengah.



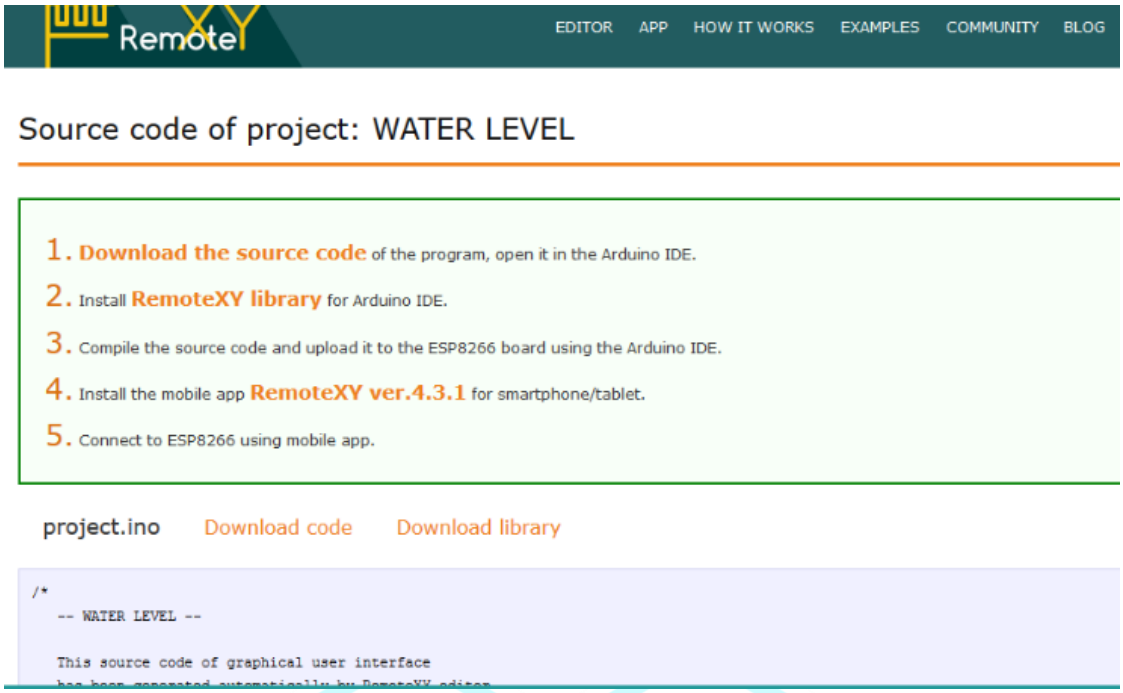
- Simpan project dengan nama WATER LEVEL



- **Program:**
  - Source code sudah dibuat otomatis oleh remoteXY, tetapi masih perlu modifikasi
  - OKE, buka program klik Get source code



  - Source code bisa anda copy-paste ke Arduino IDE



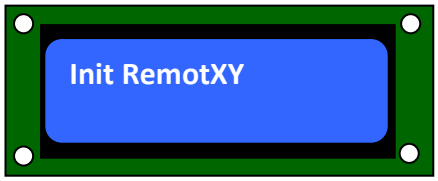
- Selanjutnya modifikasi Source code di atas, yang perlu kita sesuaikan SSID, password dan token. Silahkan Anda tambahkan source code yang belum termasuk di source code bawaan remoteXY, seperti sensor ultrasonic dan display LCD.

```
#define REMOTEXY_MODE__ESP8266WIFI_LIB_CLOUD
#include <ESP8266WiFi.h>
#include <RemoteXY.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 16, 2);
const int TRIGPIN=D5;
const int ECHOPIN=D6;
unsigned int jarak,timer,T_air,La;
unsigned int Vol;

// RemoteXY connection settings
#define REMOTEXY_WIFI_SSID "Ardutech"
#define REMOTEXY_WIFI_PASSWORD "12345678"
#define REMOTEXY_CLOUD_SERVER "cloud.remotexy.com"
#define REMOTEXY_CLOUD_PORT 6376
#define REMOTEXY_CLOUD_TOKEN "token_sesuaikan_dari_remoteXY"

#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
  { 255,0,0,12,0,60,0,8,16,1,
  129,0,37,66,0,6,17,0,129,0,
  10,1,42,6,190,87,65,84,69,82,
  32,76,69,86,69,76,0,67,2,10,
  8,24,13,2,26,11,129,0,36,13,
  13,6,17,76,105,116,101,114,0,66,
  1,13,23,36,61,2,26 };
```

```
struct {
  char text_1[11];  // string UTF8 end zero
  int8_t level_1; // =0..100 level position
  uint8_t connect_flag;  // =1 if wire connected, else =0
} RemoteXY;
#pragma pack(pop)
//----------------------------BACA SENSOR ULTRSONIC
void Baca_SRF04(){
  digitalWrite(TRIGPIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGPIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGPIN, LOW);
  timer = pulseIn(ECHOPIN, HIGH);
  jarak= timer/58;

  if (jarak>100)jarak=100;
  T_air=100-jarak;
  Vol=La*T_air/1000;
}

void setup() {
  pinMode(ECHOPIN, INPUT);
  pinMode(TRIGPIN, OUTPUT);
  lcd.begin();
  lcd.backlight();
  lcd.print("Init RemotXY");
  RemoteXY_Init ();
  delay(3000);
  lcd.clear();
  // tinggi tangki 1 m (100 cm)
  // diameter alas 60 cm
  // Luas alas = 3.14x30x30 = 2826 cm2
  La=2826;
}

void loop()
{
  RemoteXY_Handler ();
  Baca_SRF04();
  lcd.setCursor(0,0);
  lcd.print("Water Level");
  lcd.setCursor(0,1);
  lcd.print("Level:");
  lcd.print(Vol);
  lcd.print(" L   ");
  dtostrf(Vol, 0, 0, RemoteXY.text_1);
  int level=map(Vol,0,280,0,100);
  RemoteXY.level_1 = level;
  delay(2000);
}
```

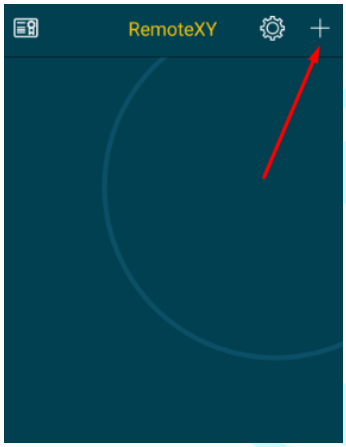
Simpan (***Save***) kemudian ***Upload***.

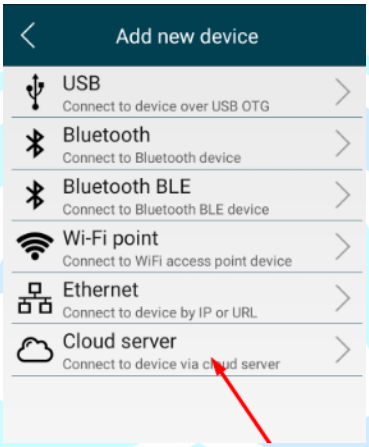
- **Jalannya Alat:**
  - LCD menampilkan inisialisasi RemotXY.

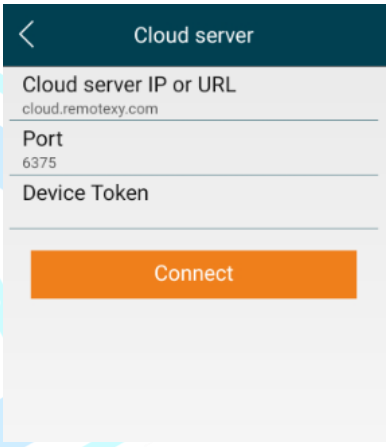


  - Selanjutnya LCD menampilkan nilai pembacaan Water Level

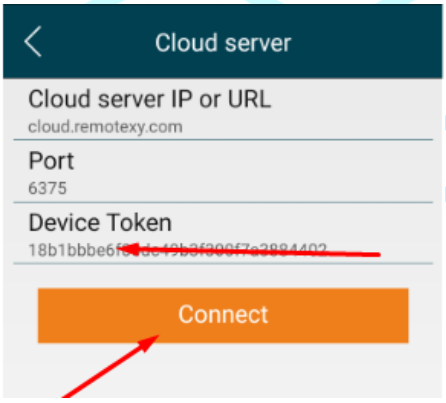
  - Selanjutnya buka aplikasi RemoteXY yang ada di smart phone Anda:



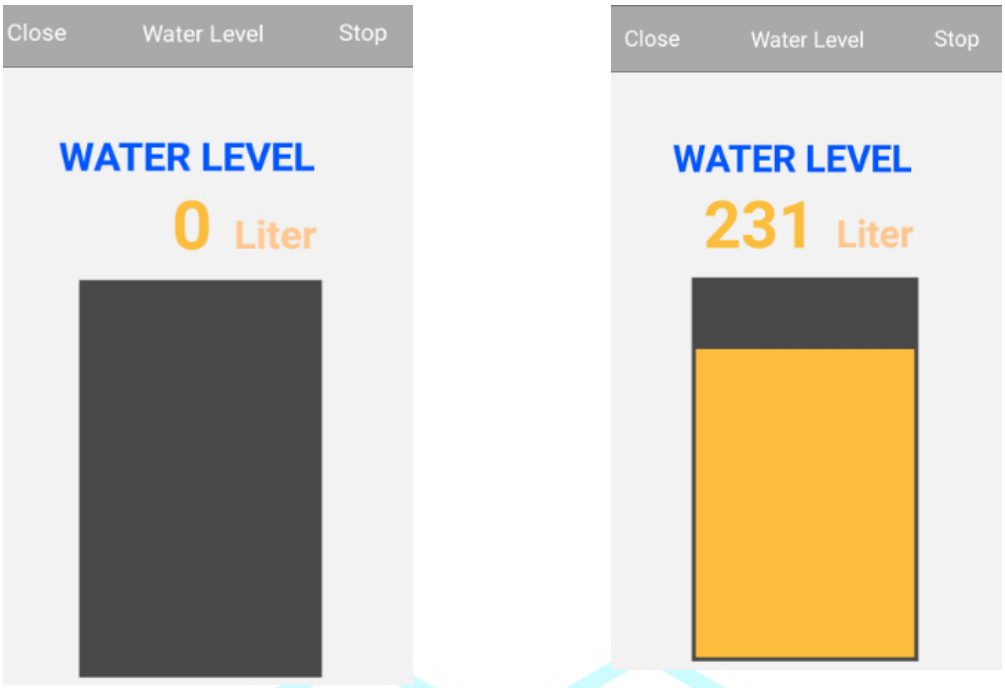




  - Masukan token

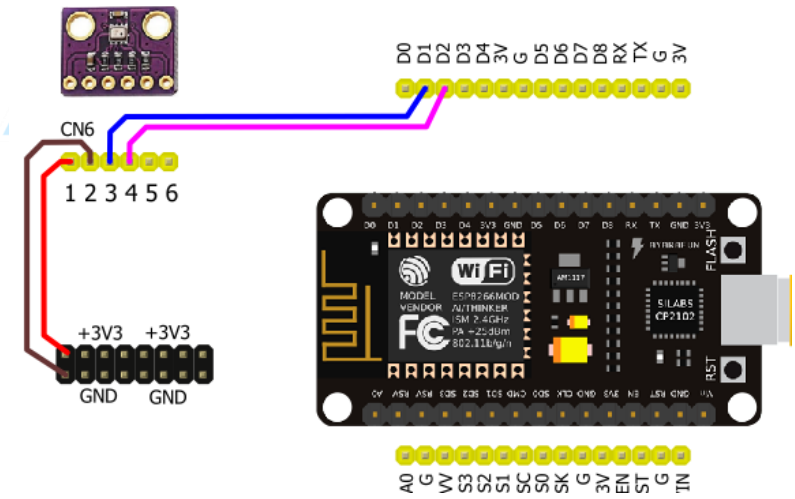


  - Tampilan saat air masih kosong dan terisi



Project 7: Monitoring Tekanan Udara, Suhu & Ketinggian dengan Web Server

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - Module Sensor BME280
  - Kabel micro USB
  - 4 Kabel jumper female – female
- **Skematik:**



Keterangan:

CN6 / BME280	NodeMCU
1/VCC	3V3
2/GND	GND

3/SCL	D1
4/SDA	D2

- **Kebutuhan Software:**
  - Arduino IDE
  - Laptop/Computer/hand phone
- **Program:**

  Ganti nama WiFi (ssid) dan password sesuai dengan jaringan anda.

```
/************************************
 * Program : Project 7 Monitoring suhu, tekanan udara dan ketinggian
 * Input   : BME280
 * Output  : -
 * Trainer Kit IoT Ardutech
 * www.ardutech.com
 ************************************
 * KONEKSI NODEMCU dg sensor BME280
 * NODE MCU  |  CN6/BME280
 *    3V3    -->   1/VCC
 *    GND    -->   2/GND
 *    D1     -->   3/SCL
 *    D2     -->   4/SDA
 ************************************/
#include <ESP8266WebServer.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#define SEALEVELPRESSURE_HPA (1013.25)

Adafruit_BMP280 bmp;
float temperature, pressure, altitude;
/*Put your SSID & Password*/
const char* ssid = "Ardutech";  // Enter SSID here
const char* password = "12345678";  //Enter Password here
ESP8266WebServer server(80);

void setup() {
  Serial.begin(115200);
  delay(100);
  bmp.begin(0x76);
  Serial.println("");
  Serial.print("Connecting to: ");
  Serial.println(ssid);
  //connect to your local wi-fi network
  WiFi.begin(ssid, password);
  //check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected..!");
```

```
  Serial.print("Got IP: ");  Serial.println(WiFi.localIP());
  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);
  server.begin();
  Serial.println("HTTP server started");
}

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

void handle_OnConnect() {
  temperature = bmp.readTemperature();
  pressure = bmp.readPressure() / 100.0F;
  altitude = bmp.readAltitude(SEALEVELPRESSURE_HPA);
  server.send(200, "text/html", SendHTML(temperature,pressure,altitude));
}

void handle_NotFound(){
  server.send(404, "text/plain", "Not found");
}

String SendHTML(float temperature,float pressure,float altitude){
  String ptr = "<!DOCTYPE html>";
  ptr +="<html>";
  ptr +="<head>";
  ptr +="<title>ESP8266 Weather Station</title>";
  ptr +="<meta name='viewport' content='width=device-width, initial-scale=1.0'>";
  ptr +="<link href='https://fonts.googleapis.com/css?family=Open+Sans:300,400,600'
   rel='stylesheet'>";
  ptr +="<style>";
  ptr +="html { font-family: 'Open Sans', sans-serif; display: block; margin: 0px
   auto; text-align: center;color: #444444;}";
  ptr +="body{margin: 0px;} ";
  ptr +="h1 {margin: 50px auto 30px;} ";
  ptr +=".side-by-side{display: table-cell;vertical-align: middle;position:
   relative;}";
  ptr +=".text{font-weight: 600;font-size: 19px;width: 200px;}";
  ptr +=".reading{font-weight: 300;font-size: 50px;padding-right: 25px;}";
  ptr +=".temperature .reading{color: #F29C1F;}";
  ptr +=".humidity .reading{color: #3B97D3;}";
  ptr +=".pressure .reading{color: #26B99A;}";
  ptr +=".altitude .reading{color: #955BA5;}";
  ptr +=".superscript{font-size: 17px;font-weight: 600;position: absolute;top:
   10px;}";
  ptr +=".data{padding: 10px;}";
  ptr +=".container{display: table;margin: 0 auto;}";
  ptr +=".icon{width:65px}";
  ptr +="</style>";
  ptr +="</head>";
  ptr +="<body>";
  ptr +="<h1>ESP8266 Weather Station</h1>";
  ptr +="<div class='container'>";
```

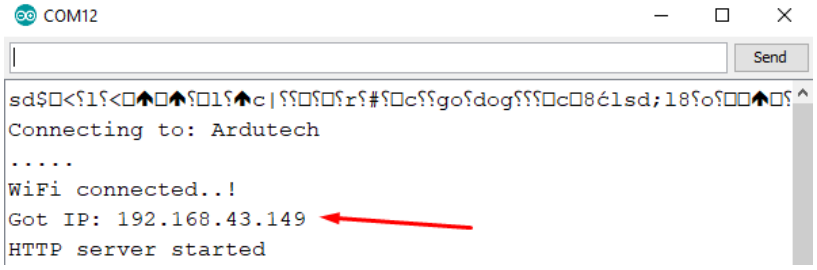
```
ptr +="<div class='data temperature'>";
ptr +="<div class='side-by-side icon'>";
ptr +="<svg enable-background='new 0 0 19.438 54.003'height=54.003px id=Layer_1
 version=1.1 viewBox='0 0 19.438 54.003'width=19.438px x=0px xml:space=preserve
 xmlns=http://www.w3.org/2000/svg xmlns:xlink=http://www.w3.org/1999/xlink
 y=0px><g><path d='M11.976,8.82v-
 2h4.084V6.063C16.06,2.715,13.345,0,9.996,0H9.313C5.965,0,3.252,2.715,3.252,6.063v3
 0.982";
ptr
 +="C1.261,38.825,0,41.403,0,44.286c0,5.367,4.351,9.718,9.719,9.718c5.368,0,9.719-
 4.351,9.719-9.718";
ptr +="c0-2.943-1.312-5.574-3.378-7.355V18.436h-3.914v-2h3.914v-2.808h-4.084v-
 2h4.084V8.82H11.976z M15.302,44.833";
ptr +="c0,3.083-2.5,5.583-5.583,5.583s-5.583-2.5-5.583-5.583c0-2.279,1.368-
 4.236,3.326-5.104V24.257C7.462,23.01,8.472,22,9.719,22";
ptr
 +="s2.257,1.01,2.257,2.257V39.73C13.934,40.597,15.302,42.554,15.302,44.833z'fill=#
 F29C21 /></g></svg>";
ptr +="</div>";
ptr +="<div class='side-by-side text'>Temperature</div>";
ptr +="<div class='side-by-side reading'>";
ptr +=(int)temperature;
ptr +="<span class='superscript'>&deg;C</span></div>";
ptr +="</div>";
ptr +="<div class='data pressure'>";
ptr +="<div class='side-by-side icon'>";
ptr +="<svg enable-background='new 0 0 40.542 40.541'height=40.541px id=Layer_1
 version=1.1 viewBox='0 0 40.542 40.541'width=40.542px x=0px xml:space=preserve
 xmlns=http://www.w3.org/2000/svg xmlns:xlink=http://www.w3.org/1999/xlink
 y=0px><g><path d='M34.313,20.271c0-0.552,0.447-1,1-1h5.178c-0.236-4.841-2.163-
 9.228-5.214-12.593l-3.425,3.424";
ptr +="c-0.195,0.195-0.451,0.293-0.707,0.293s-0.512-0.098-0.707-0.293c-0.391-0.391-
 0.391-1.023,0-1.414l3.425-3.424";
ptr +="c-3.375-3.059-7.776-4.987-12.634-
 5.215c0.015,0.067,0.041,0.13,0.041,0.202v4.687c0,0.552-0.447,1-1,1s-1-0.448-1-
 1V0.25";
ptr +="c0-0.071,0.026-0.134,0.041-
 0.202C14.39,0.279,9.936,2.256,6.544,5.385l3.576,3.577c0.391,0.391,0.391,1.024,0,1.
 414";
ptr +="c-0.195,0.195-0.451,0.293-0.707,0.293s-0.512-0.098-0.707-0.293L5.142,6.812c-
 2.98,3.348-4.858,7.682-5.092,12.459h4.804";
ptr +="c0.552,0,1,0.448,1,1s-0.448,1-
 1,1H0.05c0.525,10.728,9.362,19.271,20.22,19.271c10.857,0,19.696-8.543,20.22-
 19.271h-5.178";
ptr +="C34.76,21.271,34.313,20.823,34.313,20.271z M23.084,22.037c-0.559,1.561-
 2.274,2.372-3.833,1.814";
ptr +="c-1.561-0.557-2.373-2.272-1.815-3.833c0.372-1.041,1.263-1.737,2.277-
 1.928L25.2,7.202L22.497,19.05";
ptr +="C23.196,19.843,23.464,20.973,23.084,22.037z'fill=#26B999 /></g></svg>";
ptr +="</div>";
ptr +="<div class='side-by-side text'>Pressure</div>";
ptr +="<div class='side-by-side reading'>";
ptr +=(int)pressure;
```

```
  ptr +="<span class='superscript'>hPa</span></div>";
  ptr +="</div>";
  ptr +="<div class='data altitude'>";
  ptr +="<div class='side-by-side icon'>";
  ptr +="<svg enable-background='new 0 0 58.422 40.639'height=40.639px id=Layer_1
   version=1.1 viewBox='0 0 58.422 40.639'width=58.422px x=0px xml:space=preserve
   xmlns=http://www.w3.org/2000/svg xmlns:xlink=http://www.w3.org/1999/xlink
   y=0px><g><path d='M58.203,37.754l0.007-0.004L42.09,9.935l-0.001,0.001c-0.356-
   0.543-0.969-0.902-1.667-0.902";
  ptr +="c-0.655,0-1.231,0.32-1.595,0.808l-0.011-0.007l-0.039,0.067c-0.021,0.03-
   0.035,0.063-0.054,0.094L22.78,37.692l0.008,0.004";
  ptr +="c-0.149,0.28-0.242,0.594-
   0.242,0.934c0,1.102,0.894,1.995,1.994,1.995v0.015h31.888c1.101,0,1.994-
   0.893,1.994-1.994";
  ptr +="C58.422,38.323,58.339,38.024,58.203,37.754z'fill=#955BA5 /><path
   d='M19.704,38.674l-0.013-0.004l13.544-23.522L25.13,1.156l-
   0.002,0.001C24.671,0.459,23.885,0,22.985,0";
  ptr +="c-0.84,0-1.582,0.41-2.051,1.038l-0.016-0.01L20.87,1.114c-0.025,0.039-
   0.046,0.082-0.068,0.124L0.299,36.851l0.013,0.004";
  ptr
   +="C0.117,37.215,0,37.62,0,38.059c0,1.412,1.147,2.565,2.565,2.565,2.565v0.015h16.989c-
   0.091-0.256-0.149-0.526-0.149-0.813";
  ptr +="C19.405,39.407,19.518,39.019,19.704,38.674z'fill=#955BA5 /></g></svg>";
  ptr +="</div>";
  ptr +="<div class='side-by-side text'>Altitude</div>";
  ptr +="<div class='side-by-side reading'>";
  ptr +=(int)altitude;
  ptr +="<span class='superscript'>m</span></div>";
  ptr +="</div>";
  ptr +="</div>";
  ptr +="</body>";
  ptr +="</html>";
  return ptr;
}
```

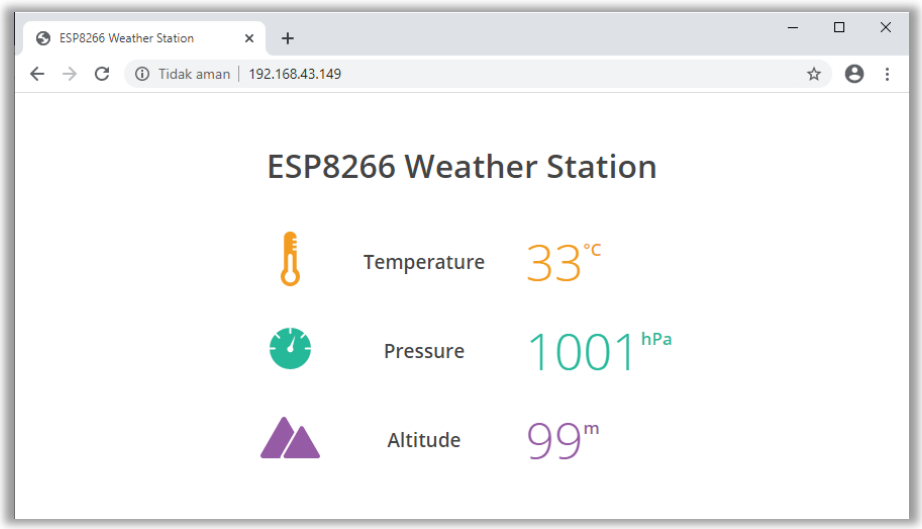
Simpan (***Save***) kemudian ***Upload***.

- **Jalannya Alat:**

  Buka serial monitor, sehingga menampilkan status koneksi dan alamat IP NodeMCU.

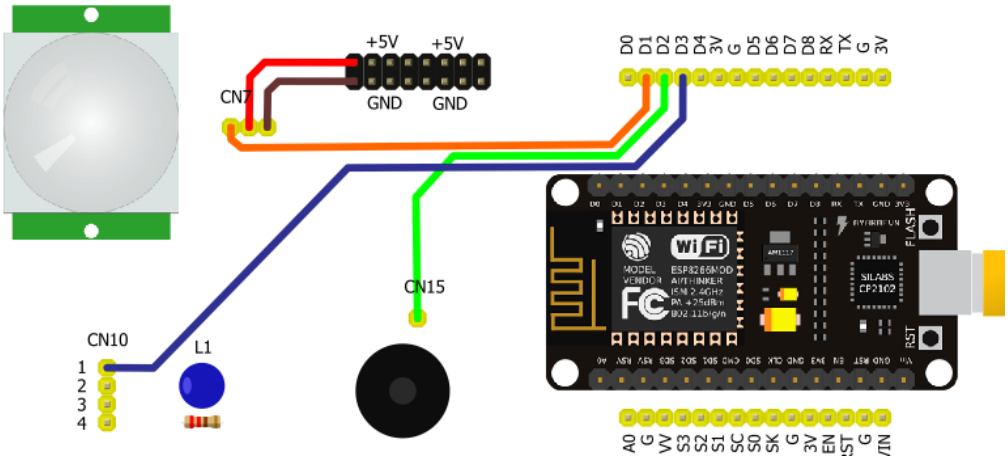


  Buka alamat IP tersebut menggunakan browser yang ada di laptop/smart phone anda.



Project 8: Alert System Anti Maling dengan Telegram

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - Sensor PIR
  - Buzzer
  - 1 LED
  - Kabel micro USB
  - 1, 3 dan 4 Kabel jumper female – female
- **Skematik:**



Keterangan:

PIR	CN7 / PIR	NodeMCU
	1/OUT	D1
	2/+5V	+5V
	3/GND	GND

BUZZUR	CN15/BUZZER	NodeMCU

	1/BUZZER +	D2

LED	CN10/LED	NodeMCU
	1/LED1	D3

- **Kebutuhan Software:**
  - Arduino IDE
  - Smart phone : Telegram (tutorial Telegram lihat CD "Tutorial memulai Telegram.pdf")

- **Program:**

  Ganti nama WiFi (ssid), password, Bot Token dan Chat ID anda.

```
/*********************************************
 * Program : Project 8 ALERT SYSTEM ANTI MALING
 * Input   : PIR
 * Output  : LED
 *           BUZZER
 * Server  : Telegram
 * Iot Starter Kit Inkubatek
 * www.ardutech.com
 ************************************
 * NODEMCU  |    Koneksi
 *    D1   -->   OUT PIR
 *    D2   -->   BUZZER
 *    D3   -->   LED1
 ************************************************/
#include "CTBot.h"
CTBot myBot;

String ssid = "Ardutech";     // REPLACE mySSID WITH YOUR WIFI SSID
String pass = "12345678"; // REPLACE myPassword YOUR WIFI PASSWORD, IF ANY
String token = "1048951628:AAGmdVvkCMmlTUAa7e1Epf9vVUC21SbH63s";   // REPLACE myToken
   WITH YOUR TELEGRAM BOT TOKEN
long chartIDbot = 780527634;

int PIRPIN = D1;
int BUZZPIN = D2;
int LEDPIN = D3;

void setup() {
 Serial.begin(115200);
 Serial.println();
 Serial.println();
 Serial.println("Starting TelegramBot...");
 myBot.wifiConnect(ssid, pass);
 myBot.setTelegramToken(token);
   // check if all things are ok
 if (myBot.testConnection())  Serial.println("\ntestConnection OK");
 else Serial.println("\ntestConnection NOK");

 pinMode(BUZZPIN, OUTPUT);
```

```
 pinMode(LEDPIN, OUTPUT);
 digitalWrite(BUZZPIN,LOW);
 digitalWrite(LEDPIN,LOW);
 pinMode(PIRPIN, INPUT_PULLUP);
}

void loop() {
   TBMessage msg;
  if(digitalRead(PIRPIN)==HIGH){
    msg.text="PERINGATAN ADA MALING";
    Serial.println(msg.text);
    while(digitalRead(PIRPIN)){
      digitalWrite(BUZZPIN, HIGH);
      digitalWrite(LEDPIN, HIGH);
      delay(300);
      digitalWrite(BUZZPIN, LOW);
      digitalWrite(LEDPIN, LOW);
      delay(100);
    }
    myBot.sendMessage(chartIDbot, msg.text);  // notify the sender
   }
   delay(1000);
}
```

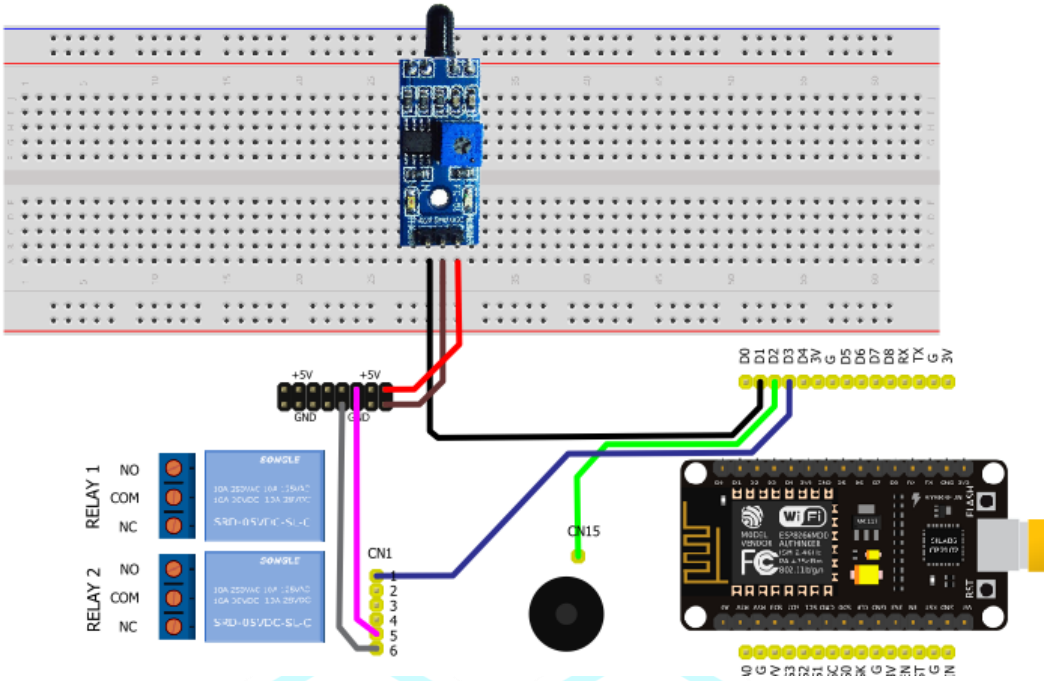
Simpan (***Save***) kemudian ***Upload***.

- **Jalannya Alat:**
  - Silahkan coba gerakkan benda (tangan anda) di depan sensor PIR, maka sensor akan mendeteksi dan alat akan mengirim pesan Telegram ke HP anda.

  - Catatan: Sensor PIR ini sangat sensitive, jadi sebaiknya di area depan sensor PIR jangan ada benda2 yang bergerak

Project 9: Detector Kebakaran dengan Telegram

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - MQ-02
  - Buzzer
  - Breadboard
  - Kabel micro USB
  - 1, 3 dan 3 Kabel jumper female – female
- **Skematik:**



Keterangan:

SENSOR API	KONEKTOR 3 PIN	NodeMCU
	1/DO	D1
	2/GND	GND
	3/VCC	+5V

BUZZUR	CN15/BUZZER	NodeMCU
	1/BUZZER +	D2

RELAY	CN1/RELAY	NodeMCU
	1/RELAY1	D3

Catatan: Pemasangan sensor api menggunakan Breadboard

- **Kebutuhan Software:**
  - Arduino IDE
  - Smart phone : Telegram (tutorial Telegram lihat CD “Tutorial memulai Telegram.pdf”)

- **Program:**

  Ganti nama WiFi (ssid), password, Bot Token dan Chat ID anda.

```
/*********************************************
 * Program : PROJECT 9 DETEKTOR KEBAKARAN
 * Input   : FLAME SENSOR
 * Output  : RELAY
 *           BUZZER
 * Server  : Telegram
 * Trainer Kit IoT Ardutech
 * www.ardutech.com
 ************************************
```

```
 * NODEMCU  |    Koneksi
 *    D1   -->   OUT FLAME SENSOR
 *    D2   -->   BUZZER
 *    D3   -->   RELAY
 ************************************************/
#include "CTBot.h"
CTBot myBot;

String ssid = "Ardutech";     // REPLACE mySSID WITH YOUR WIFI SSID
String pass = "12354678"; // REPLACE myPassword YOUR WIFI PASSWORD, IF ANY
String token = "1048951628:AAGmdVvkCMmlTUAa7e1Epf9vVUC21SbH63s";   // REPLACE myToken
   WITH YOUR TELEGRAM BOT TOKEN
long chartIDbot = 780527634;

int FLAMEPIN = D1;
int BUZZPIN = D2;
int RELAYPIN = D3;
String msg = "KEBAKARAN TERDETEKSI";

void setup() {
 Serial.begin(115200);
 Serial.println();
 Serial.println();
 Serial.println("Starting TelegramBot...");
 myBot.wifiConnect(ssid, pass);
 myBot.setTelegramToken(token);
   // check if all things are ok
 if (myBot.testConnection())  Serial.println("\ntestConnection OK");
 else Serial.println("\ntestConnection NOK");

 pinMode(BUZZPIN, OUTPUT);
 pinMode(RELAYPIN, OUTPUT);
 digitalWrite(BUZZPIN,LOW);
 digitalWrite(RELAYPIN,LOW);
 pinMode(FLAMEPIN, INPUT_PULLUP);
}

void loop() {
  if(digitalRead(FLAMEPIN)==LOW){
    Serial.println(msg);
    digitalWrite(RELAYPIN, HIGH);
    while(!digitalRead(FLAMEPIN)){
      digitalWrite(BUZZPIN, HIGH);
      delay(300);
      digitalWrite(BUZZPIN, LOW);
      delay(100);
    }
    myBot.sendMessage(chartIDbot, msg);  // notify the sender
    digitalWrite(RELAYPIN, LOW);
   }
   delay(1000);
}
```

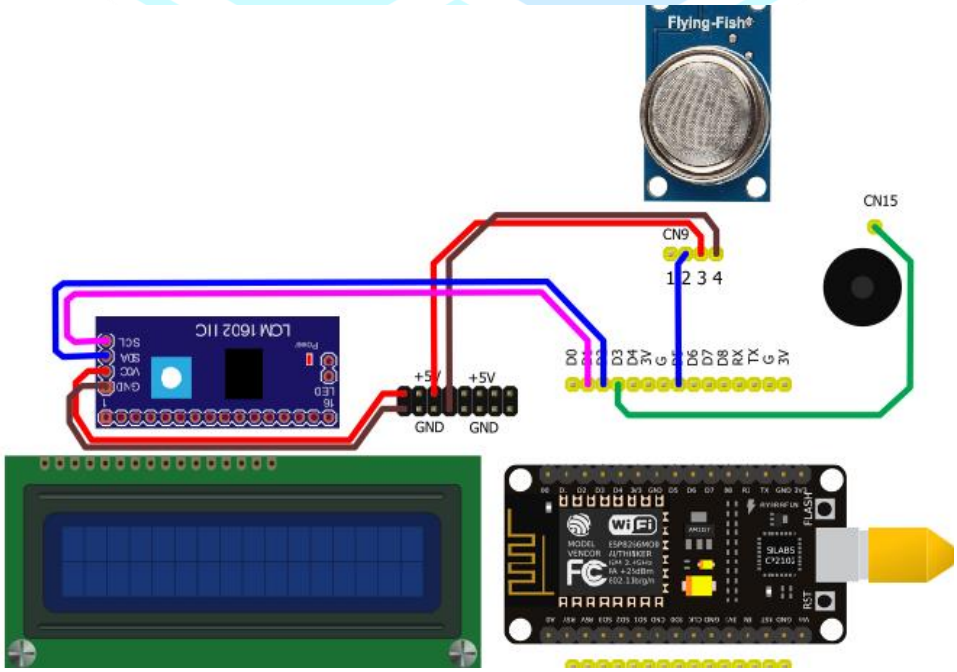
Simpan (***Save***) kemudian ***Upload***.

- **Jalannya Alat:**
  - Silahkan coba nyalakan api di depan sensor, maka sensor akan mendeteksi dan alat akan mengirim pesan Telegram ke HP anda.

  - Catatan: Sensor PIR ini sangat sensitive, jadi sebaiknya di area depan sensor PIR jangan ada benda2 yang bergerak

Project 10: Detector Kebocoran Gas LPG dengan Blynk

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - Module Sensor MQ-02
  - LCD I2C
  - Kabel micro USB
  - 3 dan 4 Kabel jumper female – female
- **Skematik:**



Keterangan:

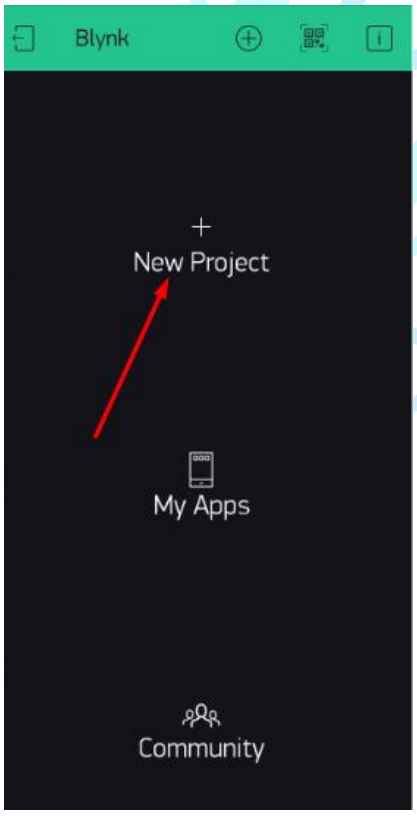
MQ-02	CN9 / MQ-02	NodeMCU
	1/OUT ANALOG	-

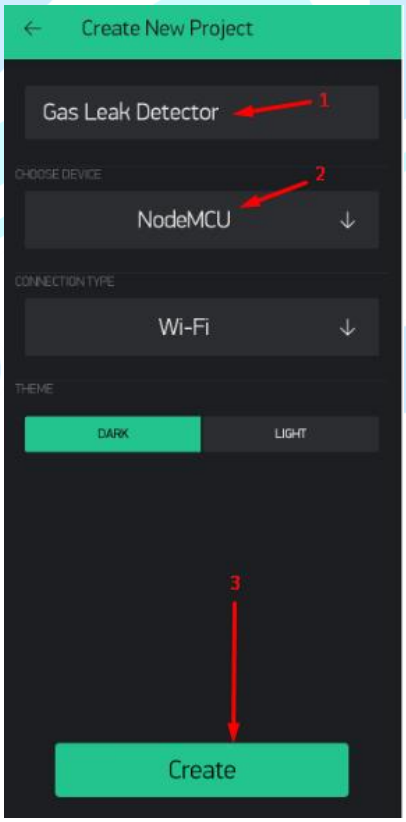
	2/OUT DIGITAL	D5
	3/+5V	+5V
	4/GND	GND

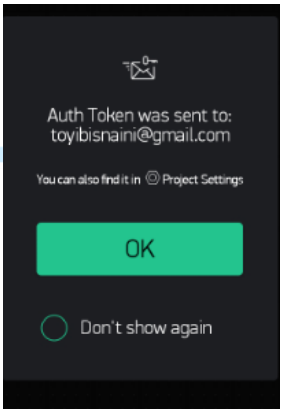
BUZZUR	CN15	NodeMCU
	1/BUZZER +	D3

LCD I2C	Konektor 4 pin	NodeMCU
	SCL	D1
	SDA	D2
	VCC	+5V
	GND	GND

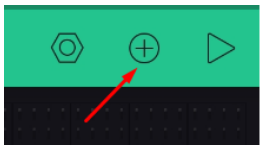
- **Kebutuhan Software:**
  - Arduino IDE
  - Smart phone : Blynk (tutorial Blynk lihat CD “Tutorial memulai Blynk.pdf”)
    Buat project di Blynk dengan nama Gas Leak Detector. Device NodeMCU.

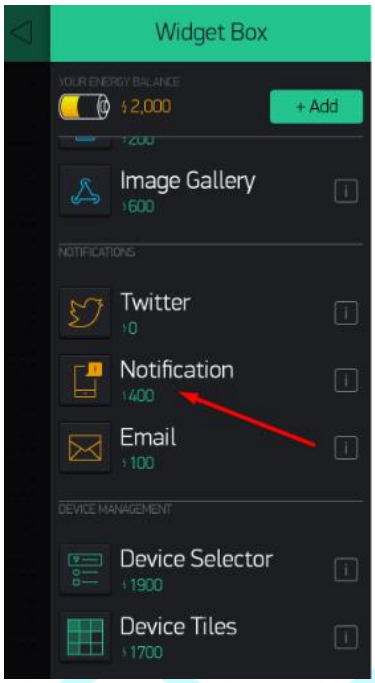


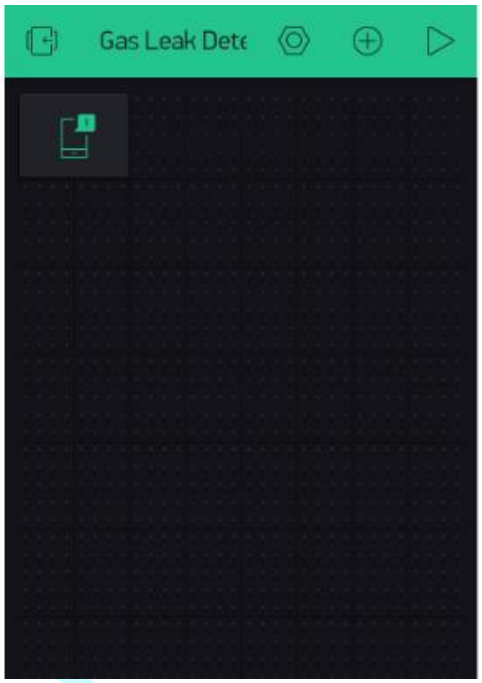




Buat notifikasi di Blynk, klik button (+), sehingga masuk Widget Box, skrol kebawah pilih Notification.







Blynk siap.

- **Program:**

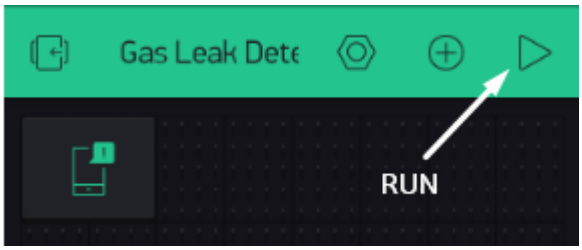
  Ganti nama WiFi (ssid), password dan token (auth) anda.

```
/*************************************************
 * Program : Project 10 GAS LEAK DETECTOR
 * Input   : MQ02
 * Output  : LCD
 *           BUZZER
 * Server  : Blynk
 * Trainer Kit IoT Ardutech
 * www.ardutech.com
 ***********************************
 * NODEMCU  |    Koneksi
 *    D1    -->   SCL LCD I2C
 *    D2    -->   SDA LCD I2C
 *    D5    -->   MQ02
 *    D3    -->   BUZZER
 *************************************************/
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 16, 2);

char auth[] = "-htbXm6E3Lizwp915_No7P516Ywa7nrj";//ganti token
char ssid[] = "Ardutech";//ganti nama wifi anda
char pass[] = "12345678";//ganti password wifi anda

#define BUZZ D3
#define MQ2Pin D5
int MQ2Value;
```

```
void setup()
{
  Serial.begin(115200);
  lcd.begin();
  lcd.backlight();
  lcd.print("Koneksi WiFi");
  lcd.setCursor(0,1);
  lcd.print(ssid);
  delay(10);
  Blynk.begin(auth, ssid, pass);
  lcd.print(" OK!");
  delay(2000);
  lcd.clear();
  pinMode(MQ2Pin, INPUT);
  pinMode(BUZZ, OUTPUT);
}

void loop(){
  Blynk.run();
  MQ2Value = digitalRead(MQ2Pin);
  if(MQ2Value){
    lcd.setCursor(0, 0);
    lcd.print("  LPG GAS LEAK");
    lcd.setCursor(0, 1);
    lcd.print("DETECTOR SYSTEM");
    delay(1000);
  }
  else{
    lcd.clear();
    lcd.print(">Gas terdeteksi<");
    Blynk.notify("Kebocoran Gas terdeteksi.....!");
    while(!MQ2Value){
      lcd.noBacklight();
      digitalWrite(BUZZ,HIGH);
      delay(200);
      digitalWrite(BUZZ,LOW);
      lcd.backlight();
      delay(200);
      MQ2Value = digitalRead(MQ2Pin);
    }
    lcd.clear();
    delay(1000);
  }
}
```

Simpan (***Save***) kemudian ***Upload***.

- **Jalannya Alat:**
  - Jalankan Blynk, klik run.

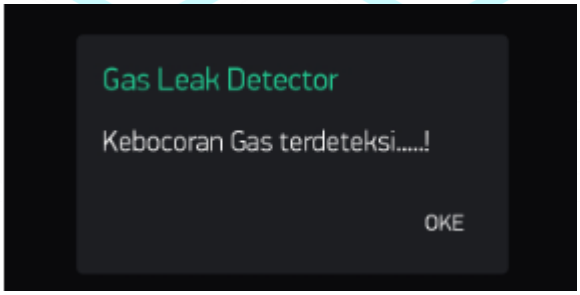


- LCD menampilkan status koneksi wifi. Tunggu sampai koneksi OK.

- Selanjutnya LCD menampilkan tampilan awal.

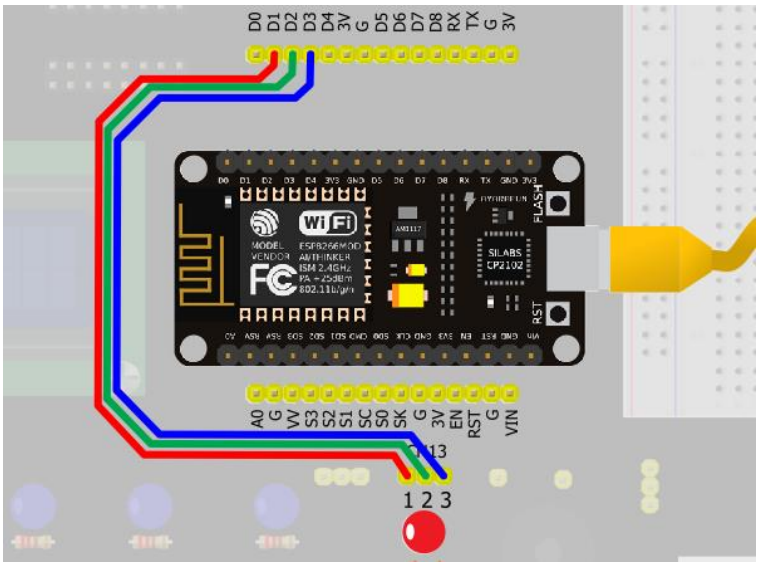
- Selanjutnya untuk membuat gangguan pada sensorn MQ-02 gunakan gas (korek gas atau LPG) maka LCD menampilkan “>Gas terdeteksi<” dan buzzer akan berbunyi selama gas masih terbaca oleh sensor MQ-02.

- Blynk akan menapilkan notifikasi berupa “Kebocoran Gas terdeteksi….!”.



Project 11: Kontrol LED RGB dengan Blynk

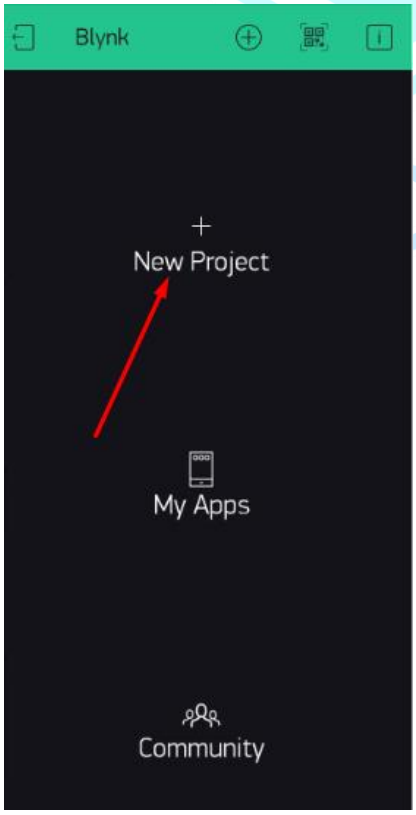
- **Kebutuhan Hardware:**
  - NodeMCU V3
  - LED RGB
  - Kabel micro USB
  - 3 Kabel jumper female – female
- **Skematik:**

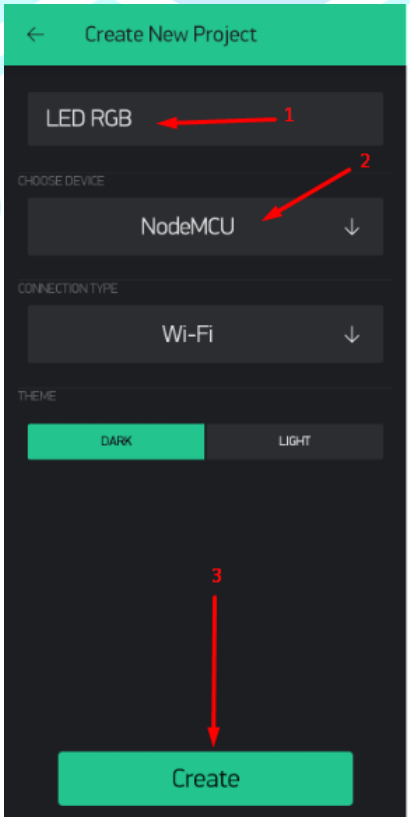


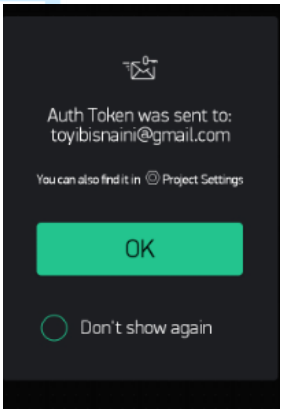
Keterangan:

LED RGB	CN13 / RGB	NodeMCU
	1/RED	D1
	2/GREEN	D2
	3/BLUE	D3

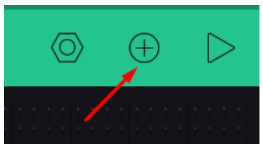
- **Kebutuhan Software:**
  - Arduino IDE
  - Smart phone : Blynk (tutorial Blynk lihat CD “Tutorial memulai Blynk.pdf”)
    Buat project di Blynk dengan nama Kontrol RGB. Device NodeMCU.

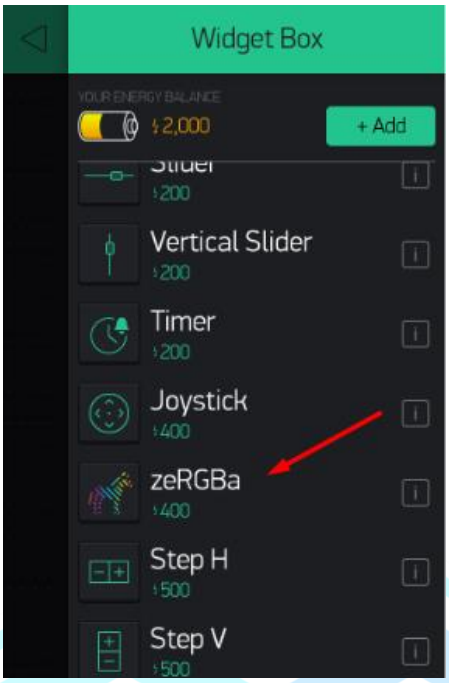


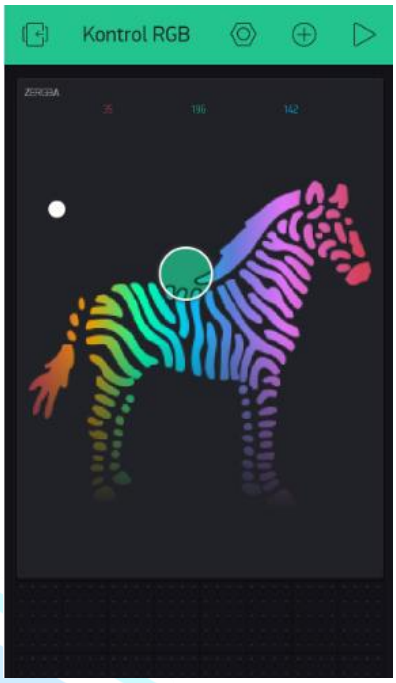




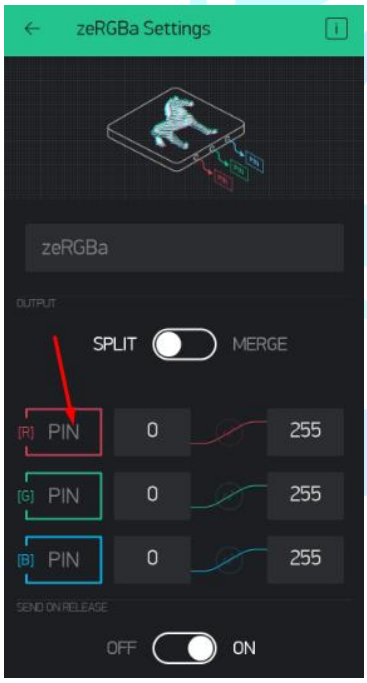
Buat widget zeRGBa di Blynk, klik button (+), sehingga masuk Widget Box, skrol kebawah pilih zeRGBa, sesuaikan ukurannya sesuai selera.

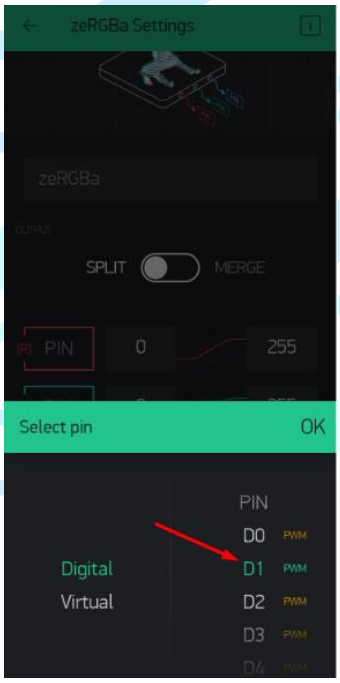


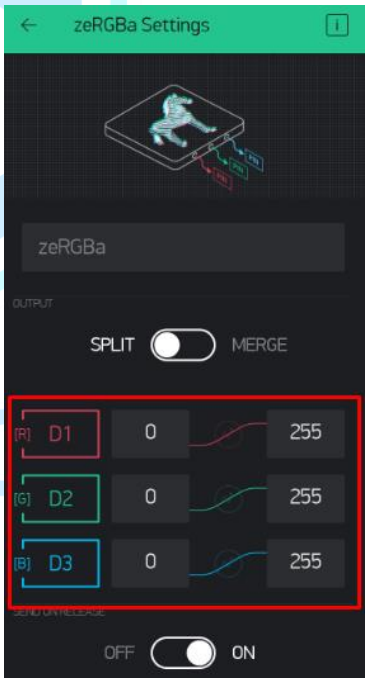




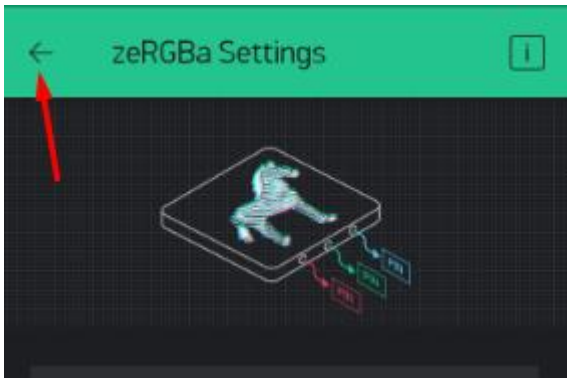
Seting kontrol LED RGB, LED R dikontrol D0, LED G dikontrol D1, LED B dikontrol D3.







Kembali ke tampilan awal.



Blynk siap.

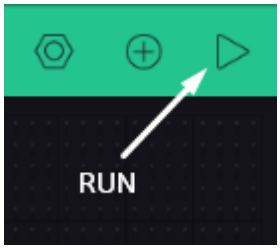
- **Program:**

  Ganti nama WiFi (ssid), password dan token (auth) anda.

```
/*************************************************
 * Program : Project 11 Kontrol LED RGB
 * Input   : -
 * Output  : LED RGB
 * Server  : Blynk
 * Trainer Kit IoT Ardutech
 * www.ardutech.com
 ***********************************
 * NODEMCU  |    Koneksi
 *    D1   -->   LED R
 *    D2   -->   LED G
 *    D3   -->   LED B
 *************************************************/
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
//Ganti dengan Token anda
char auth[] = "-htbXm6E3Lizwp915_No7P516Ywa7nrj";//ganti auth token anda
// Ganti dengan WiFi anda
char ssid[] = "Ardutech";//ganti nama wifi anda
char pass[] = "12345678";//ganti password wifi anda

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
}

void loop()
{
  Blynk.run();
}
```

  Simpan (***Save***) kemudian ***Upload***.

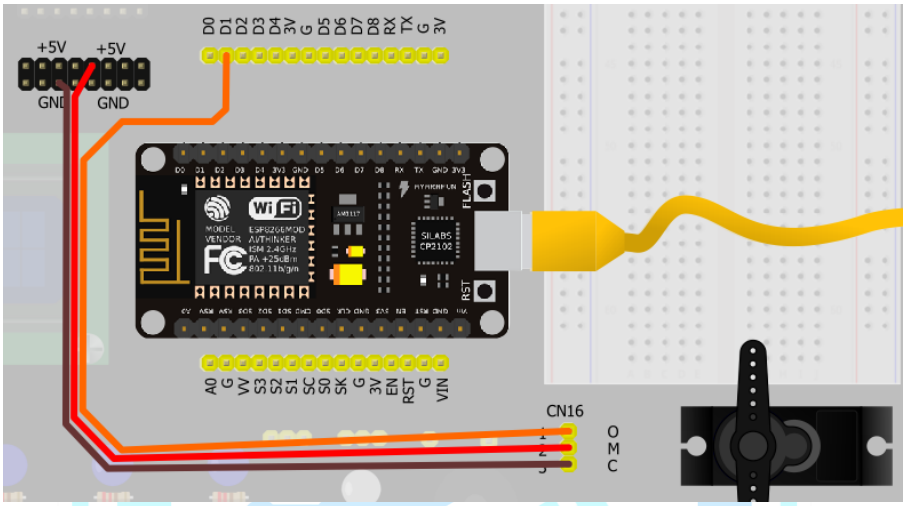
- **Jalannya Alat:**
  - Jalankan Blynk, klik run.



  - Geser dengan jari lingkaran putih untuk menentukan warna LED RGB, perhatikan LED RGB di board **Trainer Kit IoT ardutech** juga berubah warna menyesuaikan warna yang kita pilih

Project 12: Cat Foot Feeding Blynk

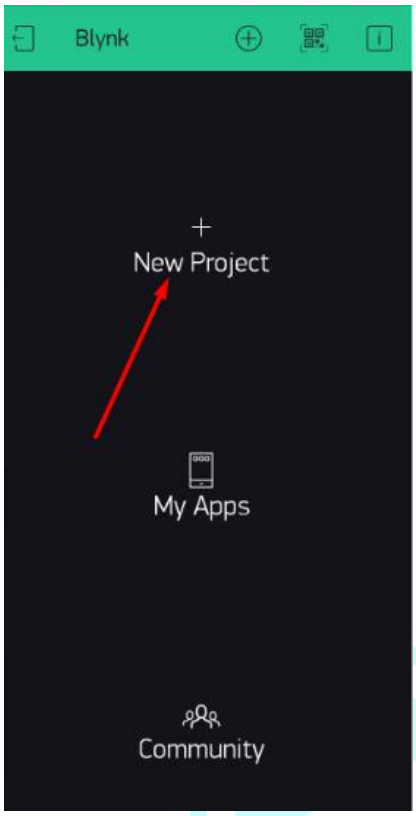
- **Kebutuhan Hardware:**
  - NodeMCU V3
  - Servo MG90S
  - Kabel micro USB
  - 3 Kabel jumper female – female
- **Skematik:**

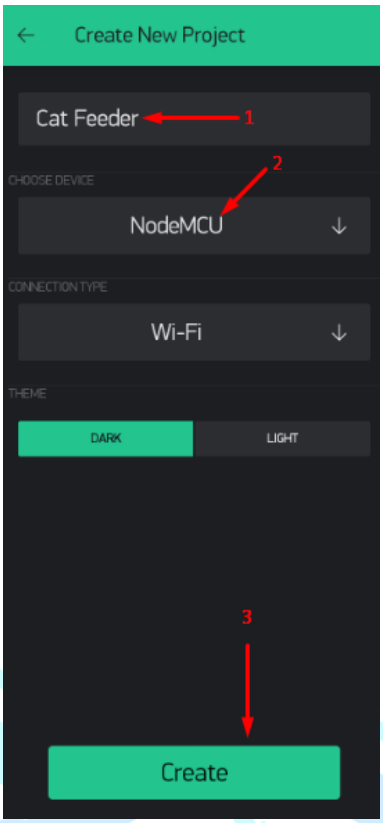


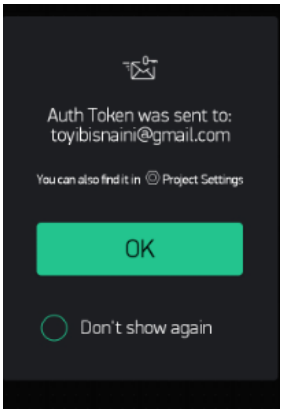
Keterangan:

Servo MG90S	CN16 / Servo	NodeMCU
	1/Orange/In Servo	D1
	2/Red/5V	+5V
	3/Brown/GND	GND

- **Kebutuhan Software:**
  - Arduino IDE
  - Smart phone : Blynk (tutorial Blynk lihat CD "Tutorial memulai Blynk.pdf")
    Buat project di Blynk dengan nama Cat Feeder. Device NodeMCU.



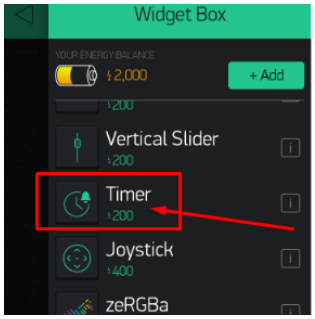


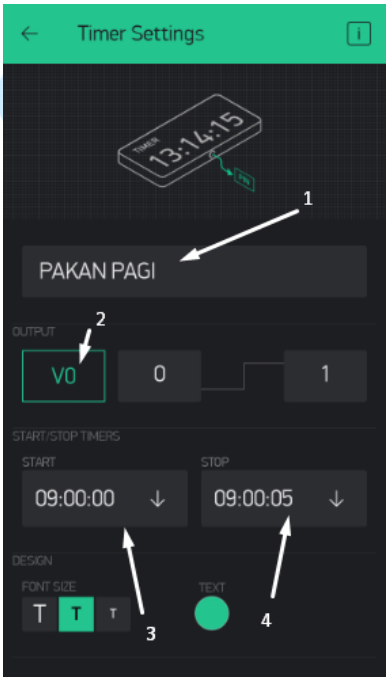


Buat widget di Blynk, klik button (+), sehingga masuk Widget Box.

1. Timer tambah 3 timer

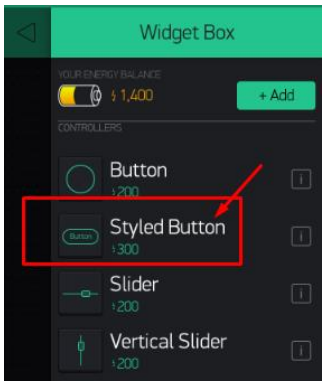
WIDGET	LABEL	OUTPUT	START TIMER	STOP TIMER
Timer 1	PAKAN PAGI	V0 : 0 → 1	09:00:00	09:00:05
Timer 2	PAKAN SIANG	V1 : 0 → 1	13:00:00	13:00:05
Timer 3	PAKAN SORE	V2 : 0 → 1	16:00:00	16:00:05

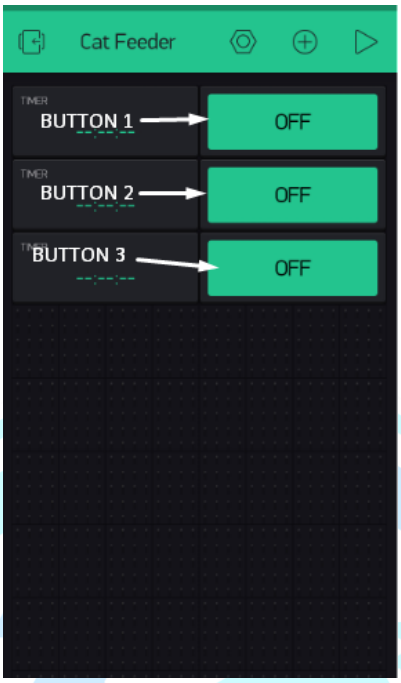


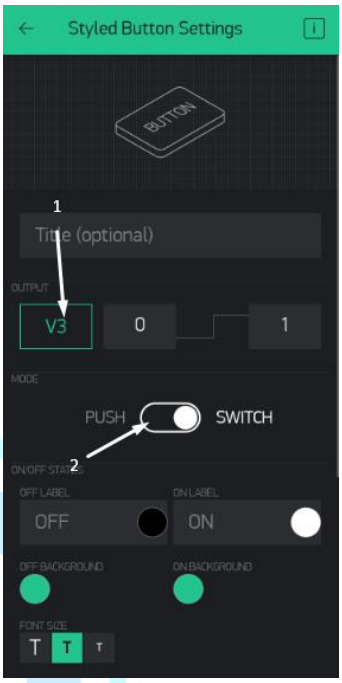


2. Styled Button tambah 3 styled button

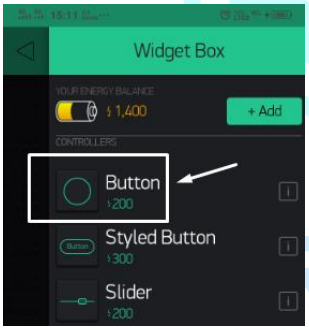
WIDGET	OUTPUT	MODE
Styled Button 1	V3 : 0 → 1	Switch
Styled Button 2	V4 : 0 → 1	Switch
Styled Button 3	V5 : 0 → 1	Switch

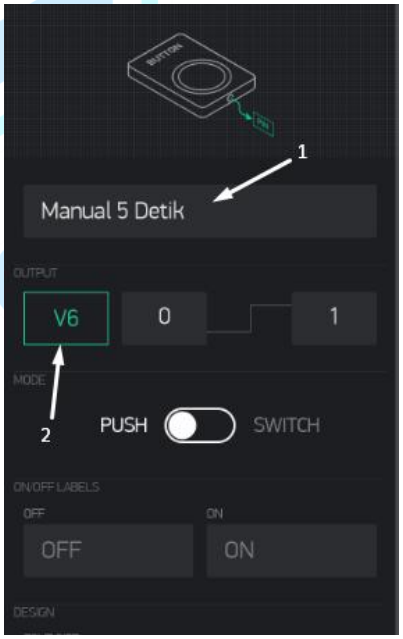






3. 1 Button, yang perlu disesuaikan **Label** = Manual 5 Detik, **OUTPUT** = V6 0→ 1.





Hasil akhirnya sebagai berikut

Blynk siap.

- **Program:**

  Ganti nama WiFi (ssid), password dan token (auth) anda.

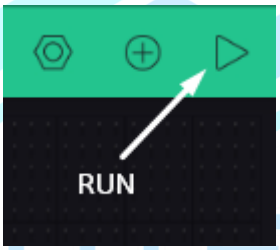
```
/*************************************************
 * Program : Project 12 CAT FOOT FEEDING
 * Input   : -
 * Output  : Servo
 * Server  : Blynk
 * Trainer Kit IoT Ardutech
 * www.ardutech.com
 ***********************************
 * NODEMCU  |    Koneksi
 *    D1    -->   Servo
 *************************************************/
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>
Servo servo;
byte tb1,tb2,tb3=0;

// Ganti dengan token anda yang dikirim via email.
char auth[] = "-htbXm6E3Lizwp915_No7P516Ywa7nrj";

// Ganti dengan WiFi dan Password anda
char ssid[] = "Ardutech";
char pass[] = "12345678";

//-------------KONTROL PAKAN PAGI
BLYNK_WRITE(V0){
  int pagi = param.asInt();
```

```
  Serial.print("pakan pagi:");
  Serial.println(pagi);
  kontrolServo(pagi,tb1);
}
BLYNK_WRITE(V3){
  tb1 = param.asInt();
  Serial.print("tb1:");
  Serial.println(tb1);
}
//-------------KONTROL PAKAN SIANG
BLYNK_WRITE(V1){
  int siang = param.asInt();
  Serial.print("pakan siang:");
  Serial.println(siang);
  kontrolServo(siang,tb2);
}
BLYNK_WRITE(V4){
  tb2 = param.asInt();
  Serial.print("tb2:");
  Serial.println(tb2);
}
//-------------KONTROL PAKAN SORE
BLYNK_WRITE(V2){
  int sore = param.asInt();
  Serial.print("pakan sore:");
  Serial.println(sore);
  kontrolServo(sore,tb3);
}
BLYNK_WRITE(V5){
  tb3 = param.asInt();
  Serial.print("tb3:");
  Serial.println(tb3);
}
//-------------KONTROL PAKAN MANUAL
BLYNK_WRITE(V6){
  int value = param.asInt();
  if(value){
    kontrolServo(value,HIGH);
    Serial.println("manual ON");
    delay(5000);
    kontrolServo(!value,HIGH);
    Serial.println("manual OFF");
  }
}

//-------------KONTROL SERVO
void kontrolServo(byte buttonState, byte tombol){
  if(buttonState==HIGH && tombol==HIGH){
    servo.write(0);
    delay(100);
  }
  else {
    delay(100);
```

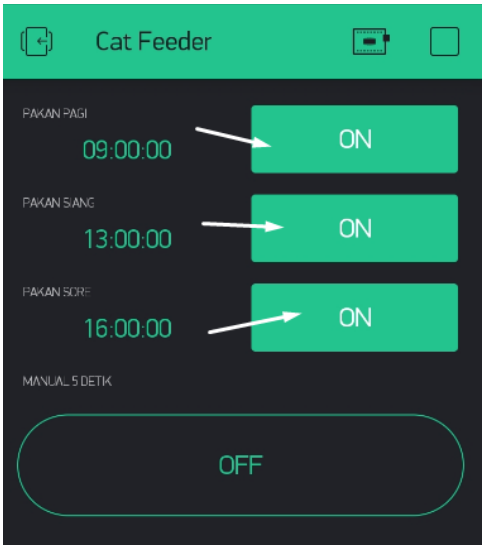
```
    servo.write(90);
  }
}

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  servo.attach(D1);
  servo.write(90);
  Blynk.virtualWrite(V1, LOW);
  Blynk.virtualWrite(V3, LOW);
  Blynk.virtualWrite(V5, LOW);
  Blynk.setProperty(V1, "offLabel", "OFF");
  Blynk.setProperty(V3, "offLabel", "OFF");
  Blynk.setProperty(V5, "offLabel", "OFF");
}

void loop()
{
  Blynk.run();
}
```

Simpan (***Save***) kemudian ***Upload***.

- **Jalannya Alat:**
  - Jalankan Blynk, klik run.



  - Aktifkan pakan pagi, siang dan sore dengan menekan tombol disamping pakan, sehingga semua tombol pakan posisi ON.

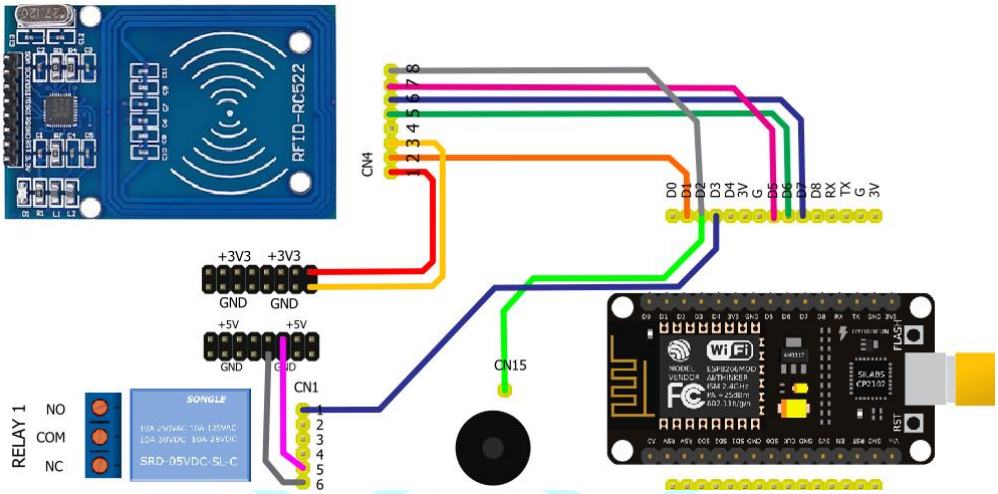


  - Pakan akan keluar sesuai dengan jadwal jam yang sudah dibuat

- Tekan tombol manual untuk membuka tutup pakan secara manual. Tutup pakan akan membuka selama 5 detik (servo=0), setelah 5 detik tutup pakan akan menutup lagi (servo=90).

Project 13: Kunci Brankas dengan RFID dan Blynk

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - RFID-RC522
  - 1 RELAY
  - BUZZER
  - Kabel micro USB
  - 7, 3 dan 1 Kabel jumper female – female
- **Skematik:**



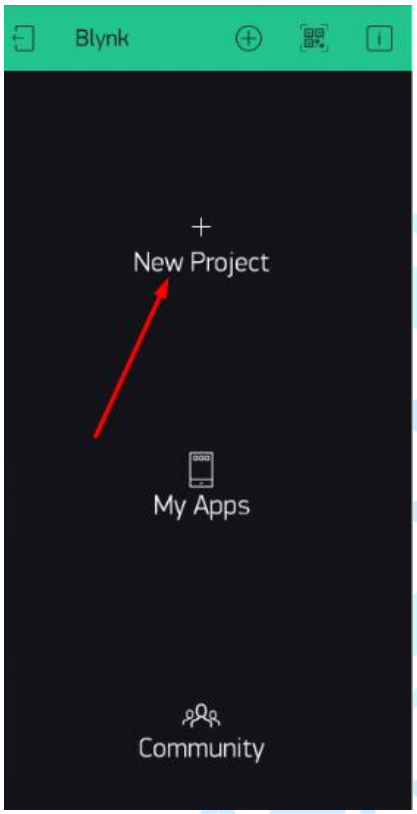
Keterangan:

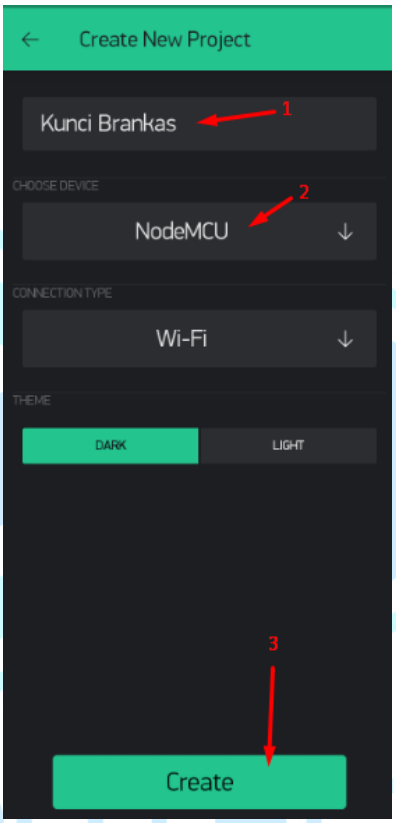
RFID-RC522	CN4/RFID	NodeMCU
	1/3.3V	3V3
	2/RST	D1
	3/GND	GND
	4/IRQ	-
	5/MISO	D6
	6/MOSI	D7
	7/SCK	D5
	8/SDA	D2

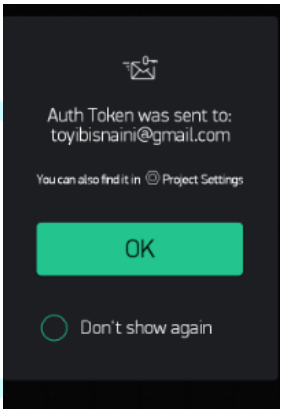
BUZZUR	CN15/BUZZER	NodeMCU
	1/BUZZER +	D8

RELAY	CN1/RELAY	NodeMCU
	1/RELAY1	D3
	5/VCC	+5V
	6/GND	GND

- **Kebutuhan Software:**
  - Arduino IDE
  - Smart phone : Blynk (tutorial Blynk lihat CD "Tutorial memulai Blynk.pdf")
    Buat project di Blynk dengan nama Kunci Brankas. Device NodeMCU.

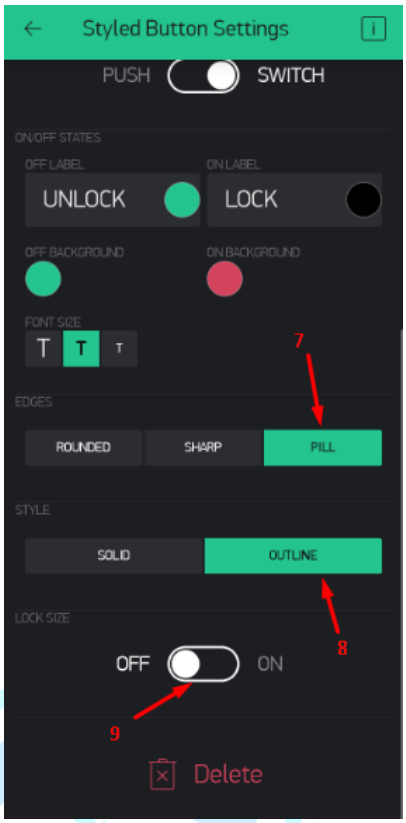




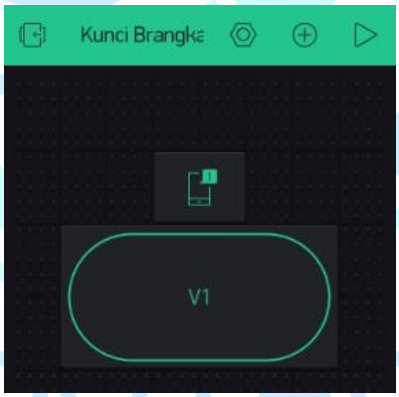


Buat widget di Blynk, klik button (+), sehingga masuk Widget Box, pilih:

1. Notification
2. Styled Button, yang perlu disesuaikan **OUTPUT** = V1 0→ 1, **MODE** = SWITCH, **OFF LABEL** = UNLOCK(GREEN), **ON LABEL**=LOCK(BLACK), **OFF BACKGROUND** = GREEN, **ON BACKGROUND** = RED, **EDGES** = PILL, **STYLE** = OUTLINE, **LOCK SIZE**=OFF.



Blynk siap.



- **Program:**

  Ganti nama WiFi (ssid), password dan token (auth) anda.

```
/*************************************************
 * Program : PROJECT 13 KUNCI BRANKAS
 * Input   : SENSOR RFID-RC522
 * Output  : RELAY
 *           BUZZER
 * Server  : Blynk
 * Trainer Kit IoT Ardutech
 * www.ardutech.com
 *************************************
 * NODEMCU  |    Koneksi
 *    D1    -->   RFID-RST
 *    D2    -->   RFID-SDA (SS)
 *    D7    -->   RFID-MOSI
 *    D6    -->   RFID-MISO
 *    D5    -->   RFID-SCK
```

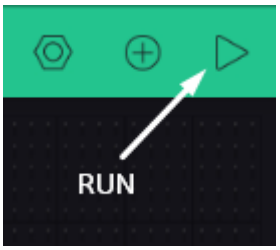
```
 *    D8   -->   BUZZER
 *    D3   -->   RELAY
 ***************************************************/
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SPI.h>
#include <MFRC522.h>

char auth[] = "-htbXm6E3Lizwp915_No7P516Ywa7nrj";
char ssid[] = "Ardutech";
char pass[] = "12345678";

#define SS_PIN D2
#define RST_PIN D1

#define RELAY_PIN D3
#define BUZZ_PIN D8

MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class
MFRC522::MIFARE_Key key;
byte nuidPICC[4];

byte lock;
String ID_TAG;
String ID1="cb4d8122";

BLYNK_WRITE(V1){
  lock=param.asInt();
  Serial.print("lock:");
  Serial.println(lock);
}

void readRFID(byte *buffer, byte bufferSize){
  ID_TAG="";
  for(byte i = 0;i<bufferSize; i++)
  {
    ID_TAG=ID_TAG+String(buffer[i], HEX);
  }
}

void setup() {
  pinMode(BUZZ_PIN,OUTPUT);
  pinMode(RELAY_PIN,OUTPUT);
  digitalWrite(RELAY_PIN,LOW);
  Serial.begin(115200);
  SPI.begin(); // Init SPI bus
  rfid.PCD_Init(); // Init MFRC522
  Serial.println();
  Serial.println();
  Blynk.begin(auth, ssid, pass);
  delay(1000);
  Serial.println();
```

```
  Serial.println();
  Serial.println("KUNCI BRANKAS");
}

void loop() {
  Blynk.run();
  if ( !rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial())
    return;
    delay(100);
    readRFID(rfid.uid.uidByte, rfid.uid.size);

//-------aktifkan komentar dibawah untuk melihat ID kartu
//    Serial.print("In hex: ");
//    Serial.println(ID_TAG);
//-----------------------------------------
    if(ID_TAG==ID1){
      Serial.println("Kartu Terdaftar ");
      Blynk.notify("Brankas diakses ID terdaftar, tekan tombol UNLOCK untuk
   membuka");
      buzzer(2, 100, 100);
      while(!lock){
        delay(100);
        Blynk.run();
      }
      buzzer(1, 500, 500);
      digitalWrite(RELAY_PIN,HIGH);
      Blynk.notify("Tekan tombol LOCK untuk mengunci brankas");
      while(lock){
        delay(100);
        Blynk.run();
      }
      buzzer(1, 500, 500);
      digitalWrite(RELAY_PIN,LOW);
    }
    else{
      Serial.println("Kartu Salah");
      Blynk.notify("Brankas diakses dengan ID tidak terdaftar");
      buzzer(50, 200, 50);
      delay(3000);
  }
}

void printHex(byte *buffer, byte bufferSize) {
  ID_TAG="";
  for (byte i = 0; i < bufferSize; i++) {
    Serial.print(buffer[i], HEX);
    ID_TAG=ID_TAG+String(buffer[i], HEX);
  }
}
void buzzer(byte ulang, int jedaOn, int jedaOff){
  for(int j=0;j<ulang;j++){
    digitalWrite(BUZZ_PIN,HIGH); delay(jedaOn);
    digitalWrite(BUZZ_PIN,LOW);  delay(jedaOff);
```

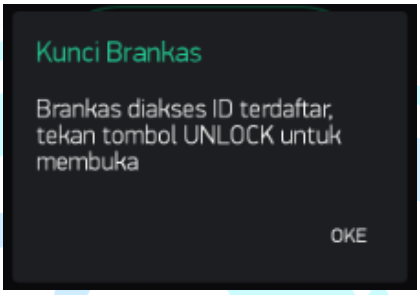
```
  }
}
```

Simpan (***Save***) kemudian ***Upload***.

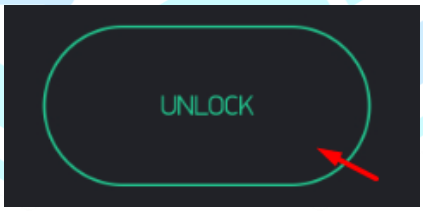
- **Jalannya Alat:**
  - Jalankan Blynk, klik run.



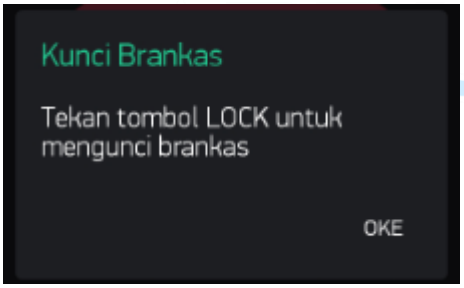
  - Tempelkan Tag RFID pada RFID reader (RFID-RC522) secara otomatis muncul notifikasi pada blynk “Brankas diakses ID terdaftar, tekan tombol UNLOCK untuk membuka”.

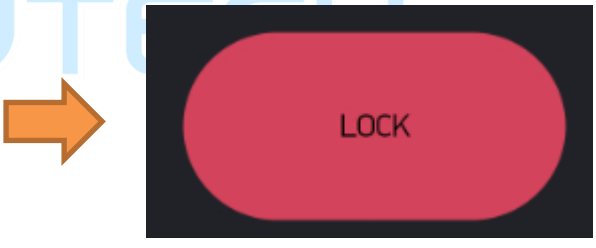


  - Selanjutnya buka brankas dengan menekan tombol UNLOCK di Blynk.

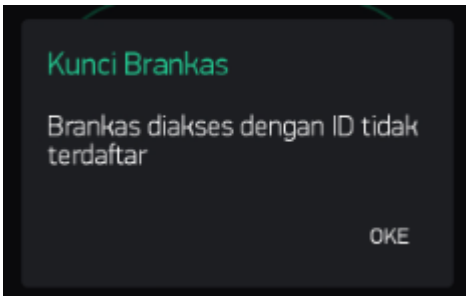


  - Setelah selesai, kunci kembali brankas dengan menekan tombol LOCK di Blynk



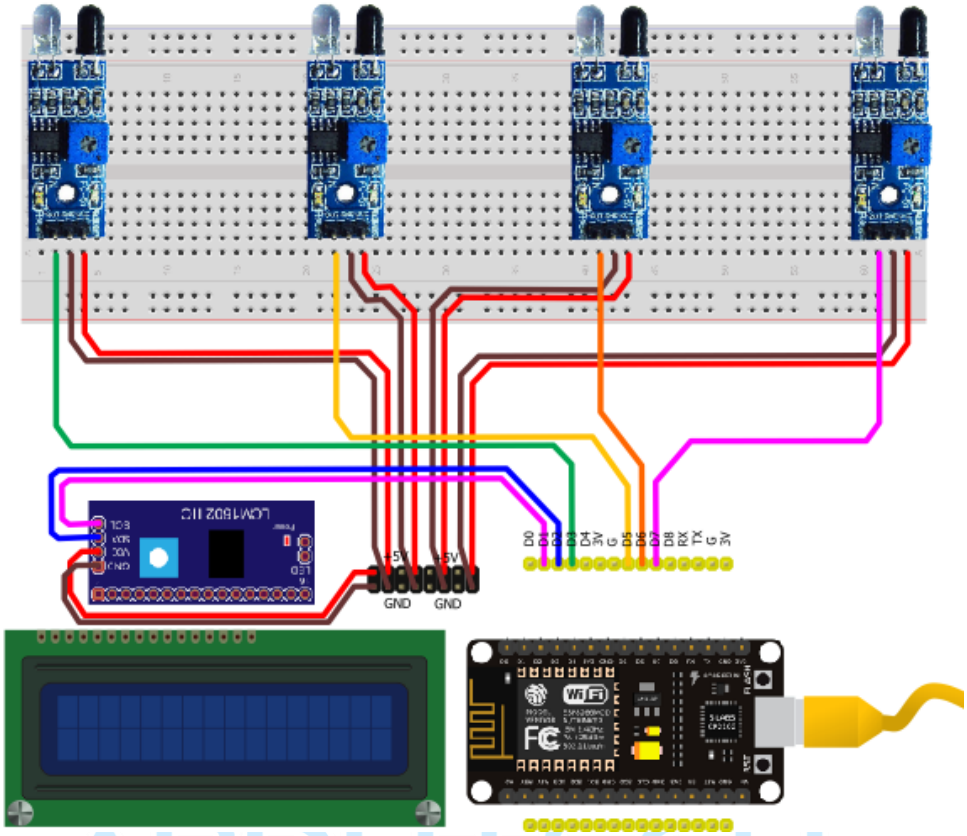


  - Jika Tag RFID tidak terdaftar maka Blynk mengirimkan notifiasi “Brankas diakses dengan ID tidak terdaftar” dan buzzer aka berbunyi selama 50x.



Project 14: Smart Parking System Blynk

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - 4 Infrared Sensor Obstacle
  - LCD I2C
  - breadboard
  - Kabel micro USB
  - 4, 3, 3, 3 dan 3 Kabel jumper female – female
- **Skematik:**



Catatan: Pemasangan sensor api menggunakan Breadboard

Keterangan:

IR SENSOR OBSTACLE 1	KONEKTOR 3 PIN	NodeMCU
	1/OUT	D3
	2/GND	GND
	3/VCC	+5V

IR SENSOR OBSTACLE 2	KONEKTOR 3 PIN	NodeMCU
	1/OUT	D5
	2/GND	GND
	3/VCC	+5V

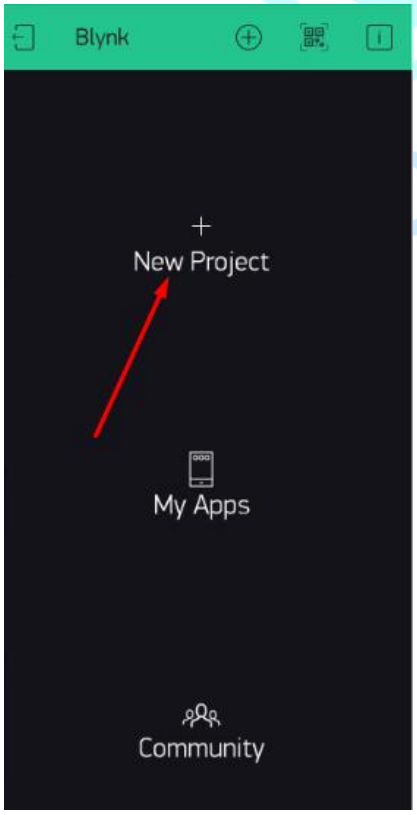
IR SENSOR OBSTACLE 3	KONEKTOR 3 PIN	NodeMCU
	1/OUT	D6
	2/GND	GND
	3/VCC	+5V

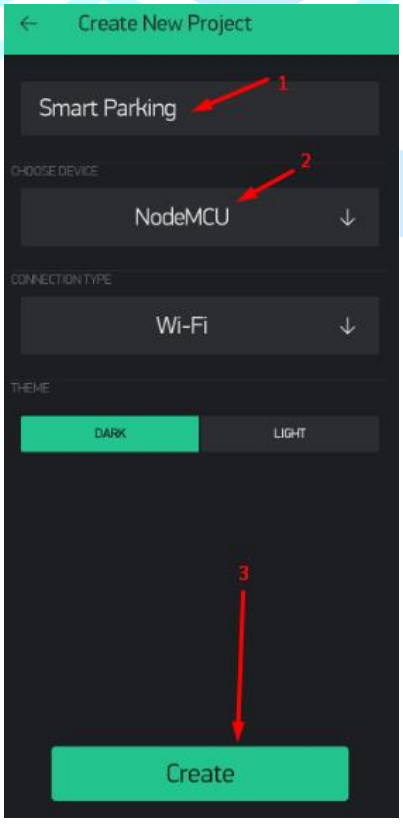
IR SENSOR OBSTACLE 4	KONEKTOR 3 PIN	NodeMCU
	1/OUT	D7
	2/GND	GND
	3/VCC	+5V

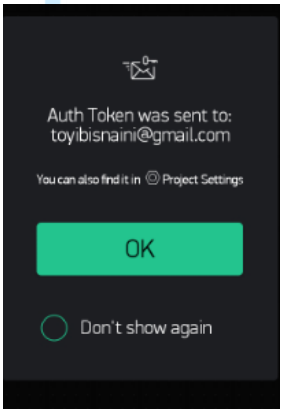
LCD I2C	Konektor 4 pin	NodeMCU
	SCL	D1
	SDA	D2
	VCC	+5V
	GND	GND

- **Kebutuhan Software:**
  - Arduino IDE
  - Smart phone : Blynk (tutorial Blynk lihat CD "Tutorial memulai Blynk.pdf")

    Buat project di Blynk dengan nama Cat Feeder. Device NodeMCU.

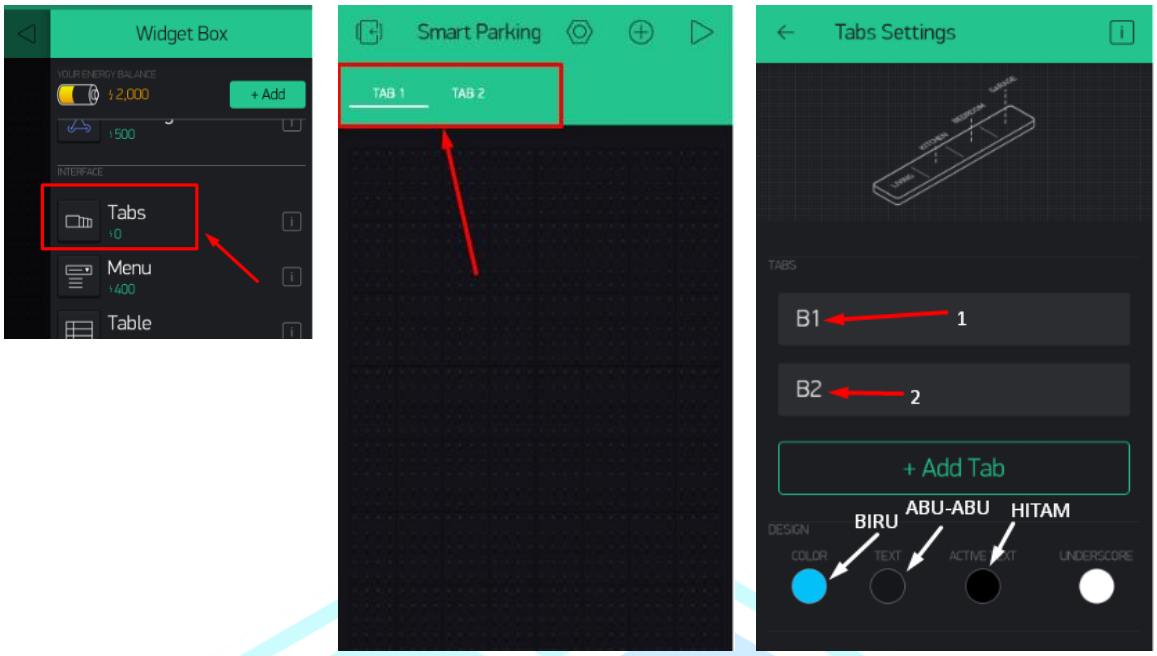






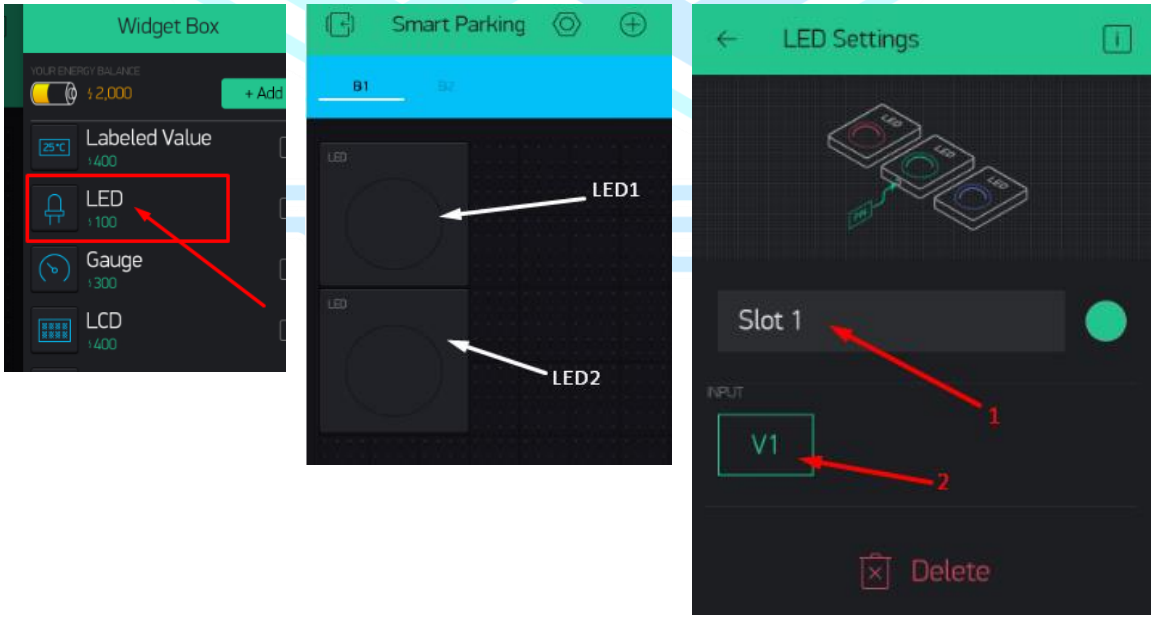
Buat widget di Blynk, klik button (+), sehingga masuk Widget Box.

1. Tabs sesuaikan setingan berikut: TABS **Tab 1** = B1, **Tab 2** = B2, **DESIGN COLOR** = BIRU, **TEXT =** ABU-ABU, **ACTIVE TEXT** = HITAM.



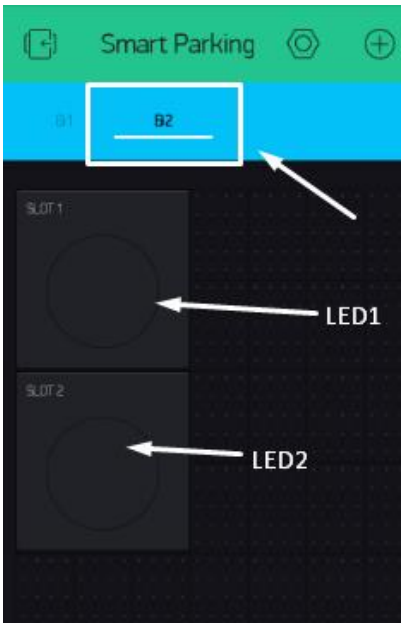
2. 2 LEDE di B1 dengan setingan

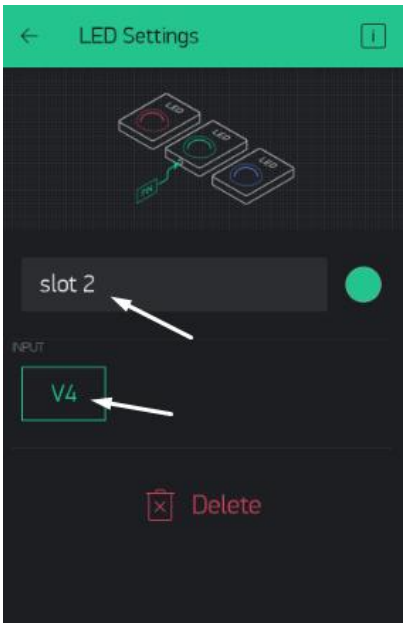
WIDGET	LABEL	INPUT
LED 1	SLOT 1	V1
LED 2	SLOT 2	V2



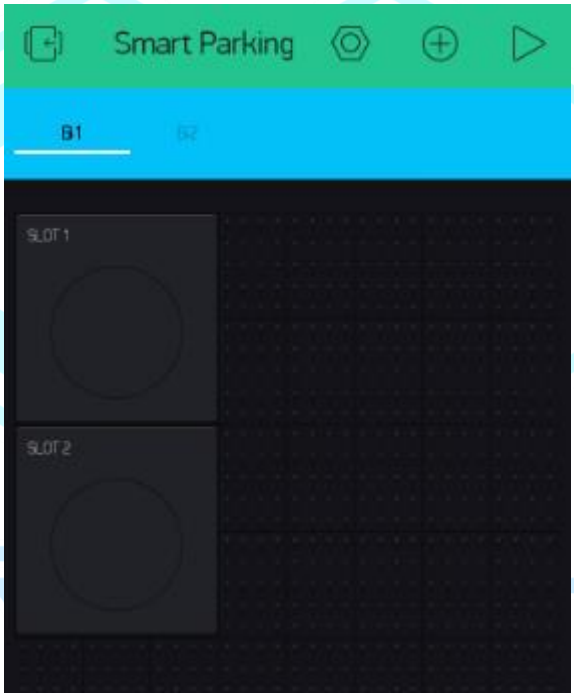
3. 2 LEDE di B2 dengan setingan

WIDGET	LABEL	INPUT
LED 1	SLOT 1	V3
LED 2	SLOT 2	V4





Hasil akhirnya sebagai berikut



Blynk siap.

- **Program:**

  Ganti nama WiFi (ssid), password dan token (auth) anda.

```
/*************************************************
 * Program : Project 14 SMART PARKING
 * Input   : IR SENSOR OBSTACLE
 * Output  : 4 LED
 * Server  : Blynk
 * Trainer Kit IoT Ardutech
 * www.ardutech.com
 ***********************************
 * NODEMCU  |    Koneksi
 *    D1   -->   SCL LCD I2C
 *    D2   -->   SDA LCD I2C
```

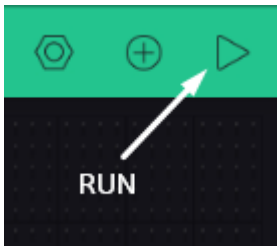
```
 *    D3   -->   Obstacle 1
 *    D5   -->   Obstacle 2
 *    D6   -->   Obstacle 3
 *    D7   -->   Obstacle 4
 ************************************************/
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);

#define S1 D3
#define S2 D5
#define S3 D6
#define S4 D7

char auth[] = "-htbXm6E3Lizwp915_No7P516Ywa7nrj";
char ssid[] = "Ardutech";
char pass[] = "12345678";

WidgetLED slot1(V1);
WidgetLED slot2(V2);
WidgetLED slot3(V3);
WidgetLED slot4(V4);

byte send1=0;
byte send2=0;
byte send3=0;
byte send4=0;

void setup() {
  pinMode(S1, INPUT_PULLUP);
  pinMode(S2, INPUT_PULLUP);
  pinMode(S3, INPUT_PULLUP);
  pinMode(S4, INPUT_PULLUP);
  lcd.begin();
  lcd.backlight();
  lcd.print("Koneksi WiFi");
  lcd.setCursor(0,1);
  Serial.begin(115200);
  Serial.println();
  Serial.println();
  delay(10);
  Blynk.begin(auth, ssid, pass);
  lcd.print(" Oke...!");
  delay(2000);
  lcd.clear();
  lcd.print("SMART PARKING");
  delay(2000);
  lcd.clear();
  slot1.off();
  slot2.off();
  slot3.off();
```

```
  slot4.off();
}

void loop() {
  Blynk.run();
  lcd.setCursor(0,0);
  lcd.print("B1 1:");
  lcd.setCursor(0,1);
  lcd.print("   2:");
  //--------------------------
  if(digitalRead(S1) == LOW) {
    lcd.setCursor(5,0);
    lcd.print("X");
    sendBlynk();
    send1=0;
  }
  else{
    lcd.setCursor(5,0);
    lcd.print("O");
    sendBlynk();
    send1=1;
  }

  //--------------------------
  if(digitalRead(S2) == LOW) {
    lcd.setCursor(5,1);
    lcd.print("X");
    sendBlynk();
    send2=0;
  }
  else{
    lcd.setCursor(5,1);
    lcd.print("O");
    sendBlynk();
    send2=1;
  }
  lcd.setCursor(8,0);
  lcd.print("B2 1:");
  lcd.setCursor(8,1);
  lcd.print("   2:");
  //--------------------------
  if(digitalRead(S3) == LOW) {
    lcd.setCursor(13,0);
    lcd.print("X");
    sendBlynk();
    send3=0;
  }
  else{
    lcd.setCursor(13,0);
    lcd.print("O");
    sendBlynk();
    send3=1;
  }
```

```
  //---------------------------
  if(digitalRead(S4) == LOW) {
    lcd.setCursor(13,1);
    lcd.print("X");
    sendBlynk();
    send4=0;
  }
  else{
    lcd.setCursor(13,1);
    lcd.print("O");
    sendBlynk();
    send4=1;
  }
  delay(300);
}

void sendBlynk(){
  if(send1!=digitalRead(S1)){
    if (send1)slot1.on();
    else slot1.off();
    Serial.print("send1:");
    Serial.println(send1);
    delay(500);
  }
  if(send2!=digitalRead(S2)){
    if (send2)slot2.on();
    else slot2.off();
    Serial.print("send2:");
    Serial.println(send2);
    delay(500);
  }
  if(send3!=digitalRead(S3)){
    if (send3)slot3.on();
    else slot3.off();
    Serial.print("send3:");
    Serial.println(send3);
    delay(500);
  }
  if(send4!=digitalRead(S4)){
    if (send4)slot4.on();
    else slot4.off();
    Serial.print("send4:");
    Serial.println(send4);
    delay(500);
  }
}
```

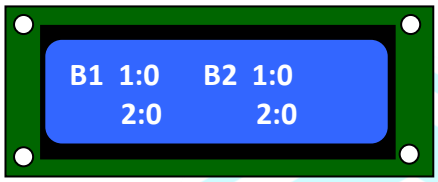
Simpan (***Save***) kemudian ***Upload***.

- **Jalannya Alat:**
  - Jalankan Blynk, klik run.

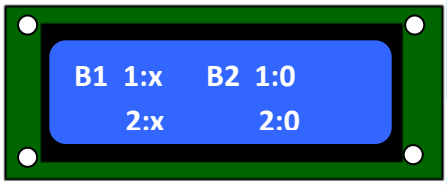


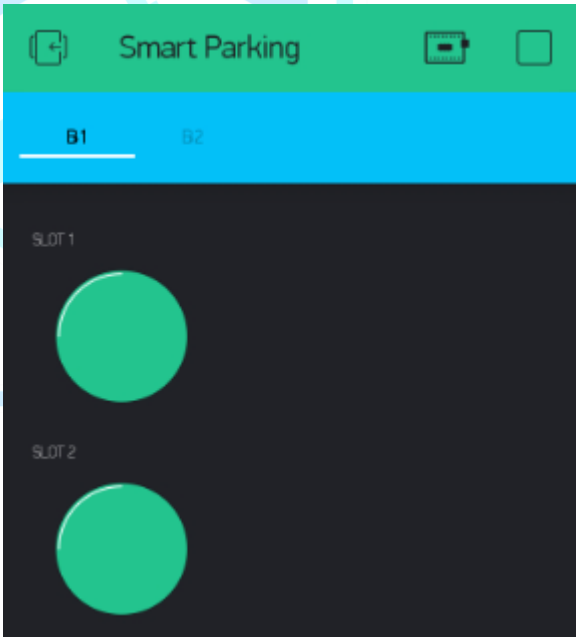
- LCD menampilkan koneksi WiFi, tunggu sampai OKE.

- Selanjutnya LCD menampilkan status parker. B1 (Basement 1) terdiri dari 2 slot (1 dan 2). B2 (Basement 2) terdiri dari 2 slot (1 dan 2).



- Jika B1 terisi penuh maka tampilan pada LCD dan Blynk sebagai berikut:





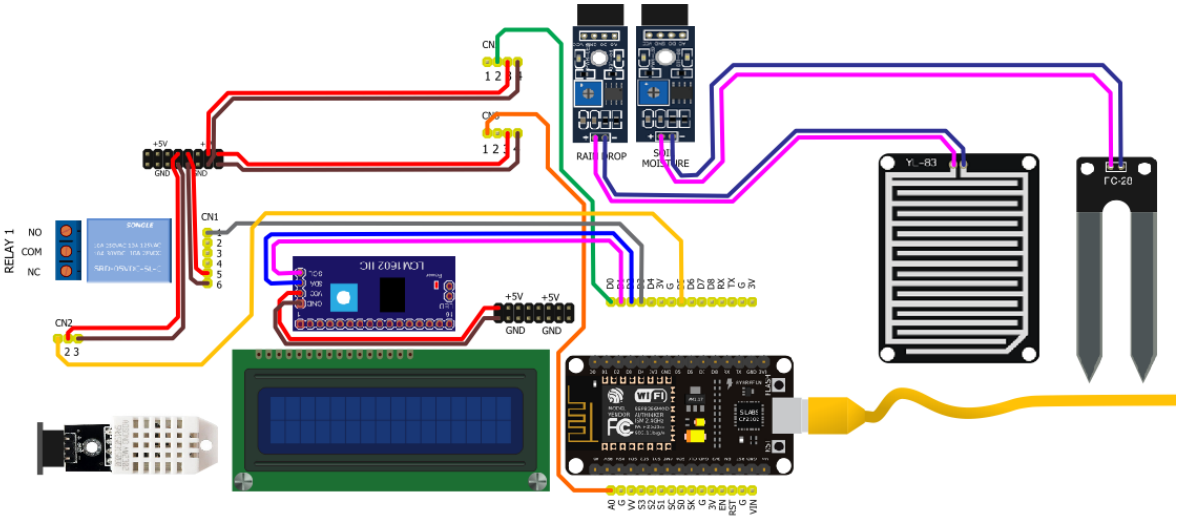
- Tanda 0 menandakan slot masih kosong dan tanda x slot sudah terisi.

Project 15: Smart Garden Adafruit IO

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - Sensor suhu dan kelembaban DHT22
  - Sensor hujan / rain drop
  - Sensor Kelembaban tanah / soil moisture
  - LCD 2x16

- Pompa air DC 5V (kontrol ON/OFF dengan relay 1)
- 1 Relay
- Kabel micro USB
- 2, 2, 3, 3, 3, 3 dan 4 Kabel jumper female – female

- **Skematik:**



Catatan: Sensor raindrop dan sensor kelembaban tanah dilepas dari soketnya, kemudian anda hubungkan antara sensor dengan modul komparatornya.

Keterangan:

DHT22	CN2 / DHT22	NodeMCU
	1/OUT	D5
	2/+5V	+5V
	3/GND	GND

SOIL MOISTURE	CN8/SOIL MOISTURE	NodeMCU
	1/AO	A0
	2/DO	-
	3/VCC	+5V
	4/GND	GND

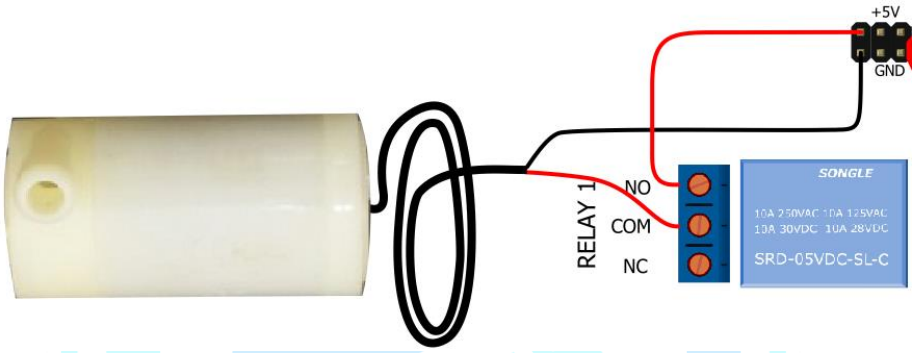
RAIN DROP	CN5/RAIN DROP	NodeMCU
	1/AO	-
	2/DO	D0
	3/VCC	+5V
	4/GND	GND

RELAY	CN1/RELAY	NodeMCU
	1/RELAY1	D3

	5/VCC	+5V
	6/GND	GND

LCD I2C	Konektor 4 pin	NodeMCU
	SCL	D1
	SDA	D2
	VCC	+5V
	GND	GND

Pengkabelan pompa air DC 5V dengan Relay:



- **Kebutuhan Software:**
  - Arduino IDE
  - Laptop/computer : Adafruit IO (tutorial Adafruit IO lihat CD "Tutorial memulai Adafruit.pdf")

- **Program:**

  Sesuaikan dengan setingan Anda:
  - SSID (nama WiFi Anda)
  - PASS (password WiFi Anda)
  - AIO_USERNAME (user name bisa lihat https://accounts.adafruit.com)
  - AIO_KEY(diisi active Key Anda)

```
/*************************************************
 * Program : Project 15 Smart Garden
 * Input   : DHT22
 *           Soil Moisture
 *           Rain Drop
 * Output  : LCD
 * actuator: Relay --> pump
 * Server  : Adafruit IO
 * Trainer Kit IoT Arduino
 * www.ardutech.com
 ***********************************
 * NODEMCU  |    Koneksi
 *    A0    -->   AO Soil moisture
 *    D0    -->   Raindrop
 *    D1    -->   SCL LCD I2C
 *    D2    -->   SDA LCD I2C
```

```
 *    D3   -->   Relay --> Pump
 *    D5   -->   DHT22
 ************************************************/
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <Wire.h>
#include <DHT.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 16, 2);

#define DHTPIN D5
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

#define SOILPIN A0
#define RAINPIN D0
#define PUMPPIN D3

byte humi, temp;
byte delaySend;
byte MODE=1;

/************************* WiFi Access Point *********************************/
#define SSID "-----namaWiFi-------"
#define PASS "-----passwordWiFi---"

/************************* Adafruit.io Setup *********************************/
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                   // use 8883 for SSL
#define AIO_USERNAME    "----namapengguna-----"//username
#define AIO_KEY         "------AIOKEY---------"// SAMA DENGAN SEBELUMNYA

/************ Global State (you don't need to change this!) ******************/
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME,
  AIO_USERNAME, AIO_KEY);

/****************************** Feeds ***************************************/
Adafruit_MQTT_Publish TEMP = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/temp
  DHT22");
Adafruit_MQTT_Publish HUMI = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/humi
  DHT22");
Adafruit_MQTT_Publish SOIL = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
  "/feeds/soilmoisture");
Adafruit_MQTT_Publish RAIN = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
  "/feeds/Raindrop");

Adafruit_MQTT_Subscribe MANUAL = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
  "/feeds/manual pump", MQTT_QOS_1);
Adafruit_MQTT_Subscribe PUMP = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
  "/feeds/onoffpump", MQTT_QOS_1);
```

```
void onoffMANUAL(char *data, uint16_t len) {
  String message = String(data);
  message.trim();
  if (message == "ON") {MODE=2;Serial.println("MODE: 2");}
  if (message == "OFF") {MODE=1;Serial.println("MODE: 1");}
}

void onoffPUMP(char *data, uint16_t len) {
  Serial.print("PUMP: ");
  Serial.println(data);
  String message = String(data);
  message.trim();
  if (message == "ON" && MODE==2) {digitalWrite(PUMPPIN, HIGH);}
  if (message == "OFF" && MODE==2) {digitalWrite(PUMPPIN, LOW);}
}

void setup() {
  pinMode(PUMPPIN, OUTPUT);
  digitalWrite(PUMPPIN,LOW);
  pinMode(RAINPIN, INPUT_PULLUP);
  lcd.begin();
  lcd.backlight();
  lcd.print("Koneksi WiFi");
  lcd.setCursor(0,1);
  lcd.print(SSID);
  Serial.begin(115200);
  delay(10);
  Serial.println();
  Serial.println();
  Serial.println("SMART GARDEN");
  // Connect to WiFi access point.
  Serial.print("Connecting to ");
  Serial.println(SSID);
  WiFi.begin(SSID, PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
  lcd.print(" Oke...!");
  delay(3000);
  lcd.clear();
  dht.begin();
  MANUAL.setCallback(onoffMANUAL);
  mqtt.subscribe(&MANUAL);
  PUMP.setCallback(onoffPUMP);
  mqtt.subscribe(&PUMP);
}

void loop() {
```

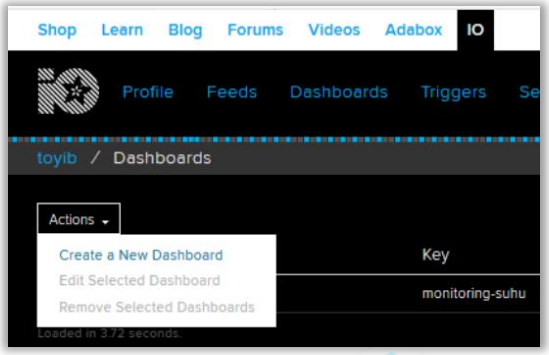
```
MQTT_connect();
humi = dht.readHumidity();
temp = dht.readTemperature();
if (isnan(humi) || isnan(temp)) {
  Serial.println("DHT22 tidak terbaca... !");
  lcd.setCursor(0,0);
  lcd.print("      DHT22");
  lcd.setCursor(0,1);
  lcd.print("tidak terbaca... !");
  return;
}
else{
lcd.setCursor(0,0);
lcd.print("T:");
lcd.print(temp);
lcd.print("C  ");
lcd.setCursor(0,1);
lcd.print("H:");
lcd.print(humi);
lcd.print("%  ");
}

word soilMoisture=analogRead(SOILPIN);
soilMoisture=map(soilMoisture, 0, 1023, 100, 0);

if(MODE==1){
  if(soilMoisture<50 && digitalRead(RAINPIN)==1){
    digitalWrite(PUMPPIN, HIGH);
  }
  else {
    digitalWrite(PUMPPIN, LOW);
  }
}

lcd.setCursor(6,0);
lcd.print("S:");
lcd.print(soilMoisture);
lcd.print("%  ");
lcd.setCursor(13,0);
lcd.print("M:");
if(MODE==1) lcd.print("1  ");
else lcd.print("2  ");
lcd.setCursor(6,1);
lcd.print("Pump ");
if(digitalRead(PUMPPIN)) lcd.print("ON   ");
else lcd.print("OFF   ");
Serial.println();
//-----------------------------------------SEND TEMP
Serial.print("Sending temperature: ");
Serial.print(temp);
Serial.print(" -->");
if (! TEMP.publish(temp)) {
  Serial.println("Failed");
```

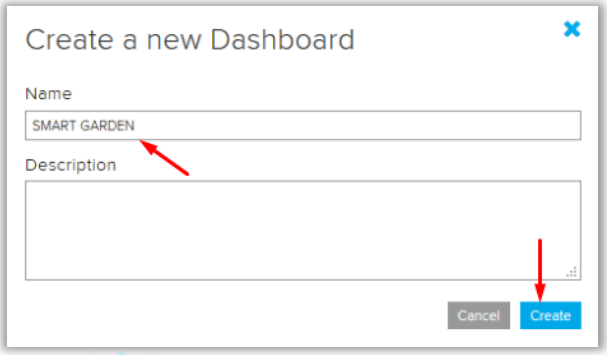
```
  } else {
    Serial.println("OK!");
  }
  //-------------------------------------SEND HUMI
  Serial.print("Sending humidity: ");
  Serial.print(humi);
  Serial.print(" -->");
  if (! HUMI.publish(humi)) {
    Serial.println("Failed");
  } else {
    Serial.println("OK!");
  }
  //-------------------------------------SEND SOIL MOISTURE
  Serial.print("Sending soil moisture: ");
  Serial.print(soilMoisture);
  Serial.print(" -->");
  if (! SOIL.publish(soilMoisture)) {
    Serial.println("Failed");
  } else {
    Serial.println("OK!");
  }
  //-------------------------------------SEND RAINDROP STATUS
  Serial.print("Sending rain drop status: ");
  Serial.print(digitalRead(RAINPIN));
  Serial.print(" -->");
  if (! RAIN.publish(digitalRead(RAINPIN))) {
    Serial.println("Failed");
  } else {
    Serial.println("OK!");
  }

  mqtt.processPackets(10000);
  lcd.begin();
  lcd.clear();
  lcd.print("kirim data....");
  delay(1000);
}

void MQTT_connect() {
  int8_t ret;
  if (mqtt.connected()) {
    return;
  }
  Serial.print("Connecting to MQTT... ");
  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) {
       Serial.println(mqtt.connectErrorString(ret));
       Serial.println("Retrying MQTT connection in 10 seconds...");
       mqtt.disconnect();
       delay(10000);  // wait 10 seconds
       retries--;
       if (retries == 0) {
         while (1);
```

```
        }
    }
    Serial.println("MQTT Connected!");
}
```

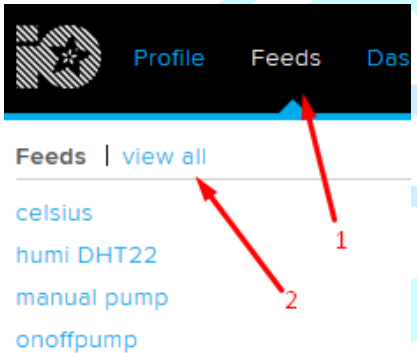
Simpan (***Save***) kemudian ***Upload***.

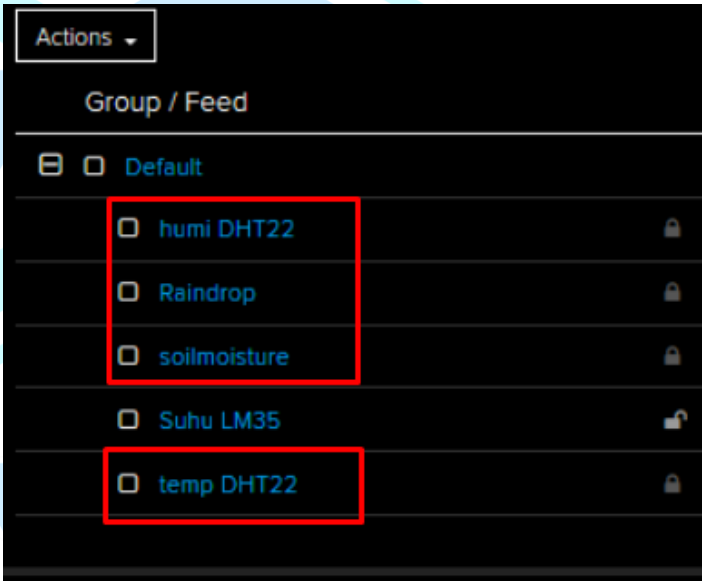
- Selanjutnya masuk ke Adafruit IO, buat dashboard dengan nama "***SMART GARDEN***"





- Selanjutnya buka Feeds → view all, seharusnya muncul humi DHT22, temp DHT22, Raindrop dan soilmoisture.

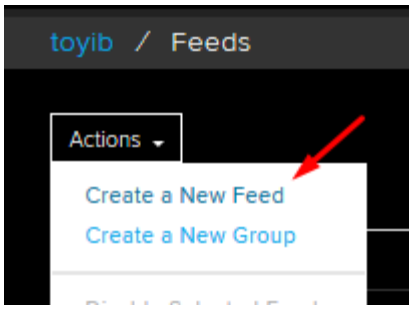




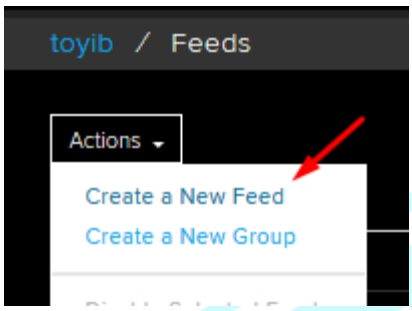
Feeds tersebut dibuat dari program berikut:

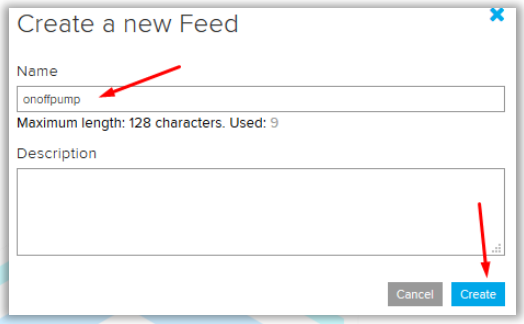
```
/*****************************/
qtt, AIO_USERNAME "/feeds/temp DHT22");
qtt, AIO_USERNAME "/feeds/humi DHT22");
qtt, AIO_USERNAME "/feeds/soilmoisture");
qtt, AIO_USERNAME "/feeds/Raindrop");
```

- Berikutnya kita buat feed untuk tombol ON/OFF pompa (manual pump dan onoffpump). Klik **Actions** pilih **Create a New Feed**. **Name** = manual pump, kemudian klik Create.

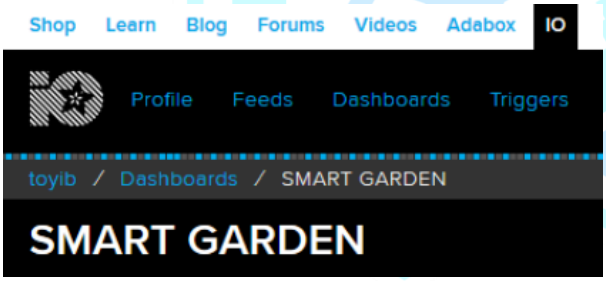


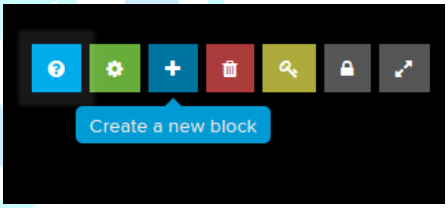
- Feed yang ke-2 **onoffpump**. Klik **Actions** pilih **Create a New Feed**. **Name** = onoffpump, kemudian klik Create.



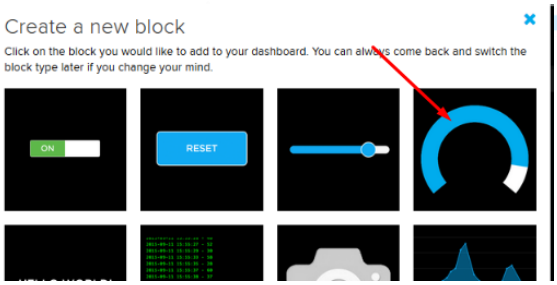


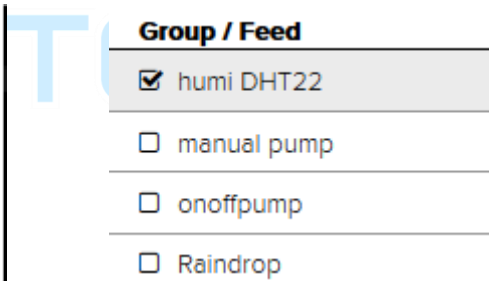
- Selanjutnya kita buat tampilan didashbord untuk smart garden, pertama buka dashboard **SMART GARDEN**, pilih button Create a new block (+)





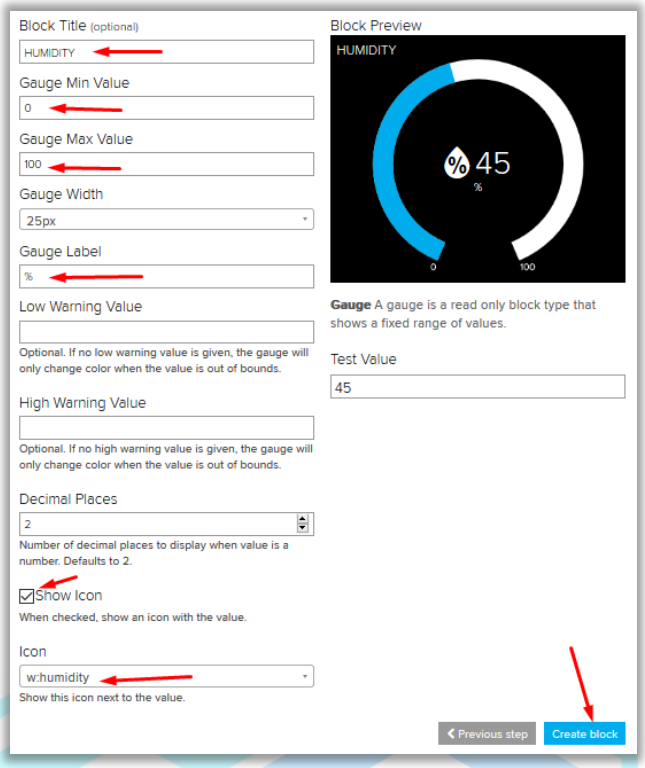
Pilih gauge, selanjutnya beri centang pada cek box humi DHT22. Kemudian pilih Next step





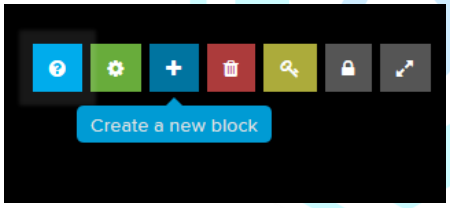
Berikutnya masuk pada Block setting ganti keterangan dan parameter gauge:

1. Block Title : HUMIDITY
2. Gauge Min Value : 0
3. Gauge Max Value : 100
4. Gauge Label : %
5. Centang Show Icon
6. Pilih icon w:humidity

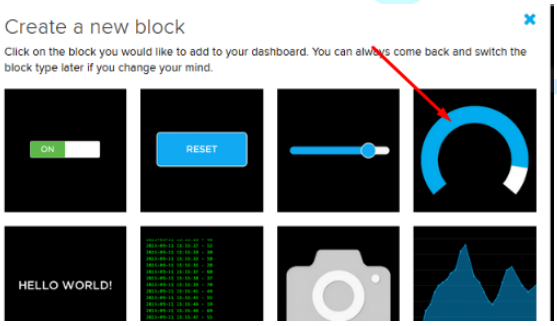


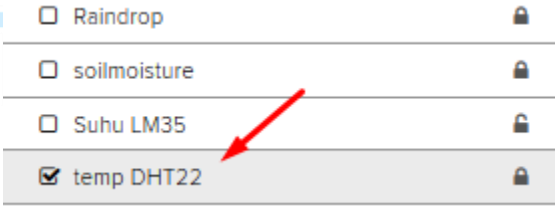
Jika sudah terisi klik Create block.

- Selanjutnya suhu (temp DHT22), pilih button Create a new block (+)



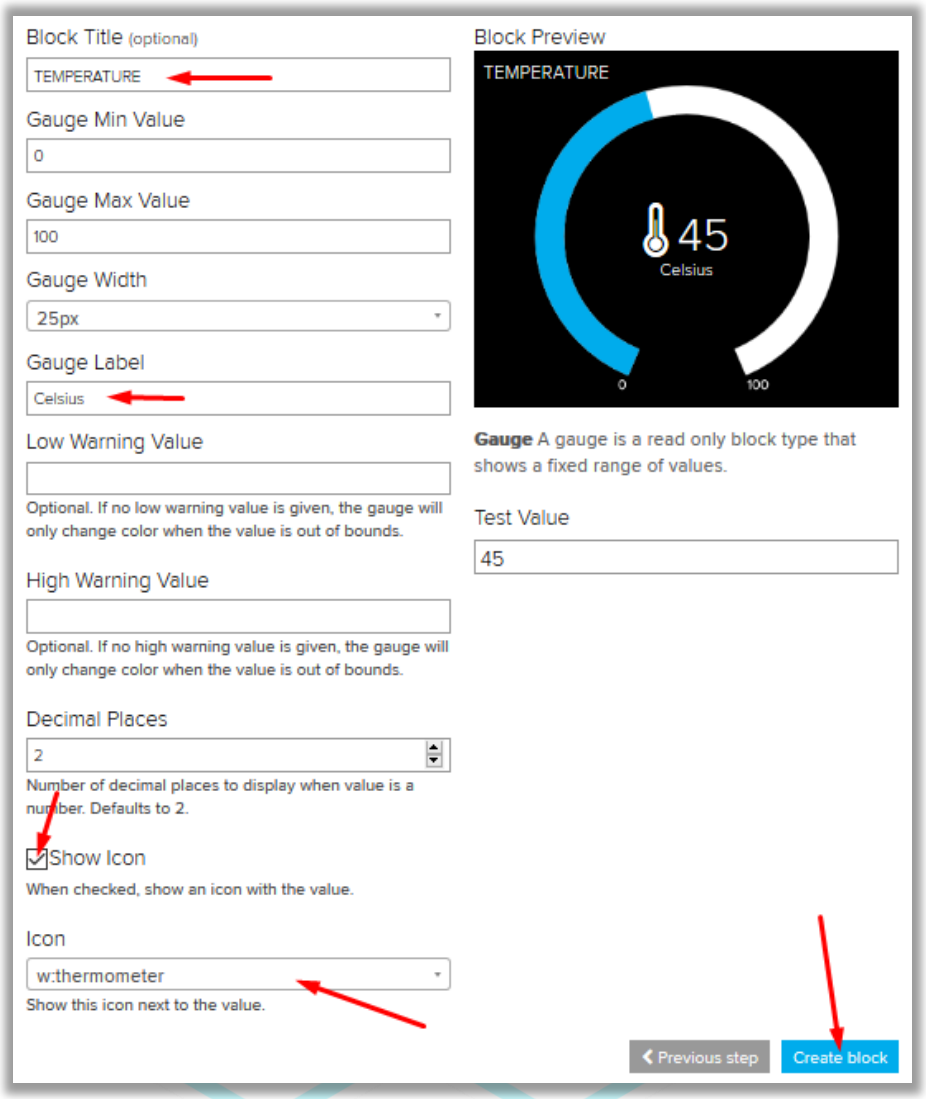
Pilih gauge, selanjutnya beri centang pada cek box humi DHT22. Kemudian pilih Next step





Berikutnya masuk pada Block setting ganti keterangan dan parameter gauge:

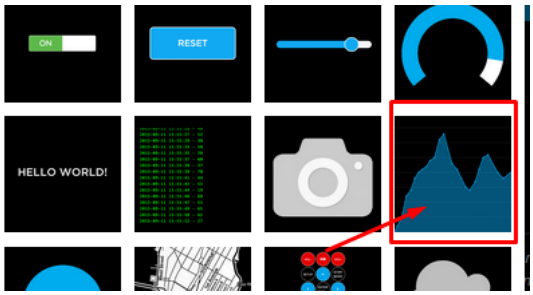
1. Block Title : TEMPERATURE
2. Gauge Label : Celsius
3. Centang Show Icon
4. Pilih icon w:thermometer

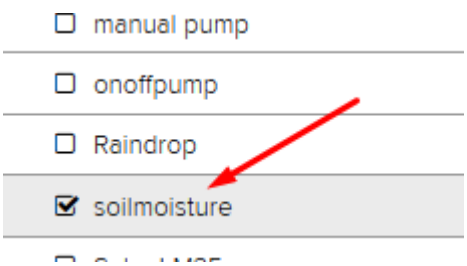


Jika sudah terisi klik Create block.

- Selanjutnya kelembaban tanah, pilih button Create a new block (+)

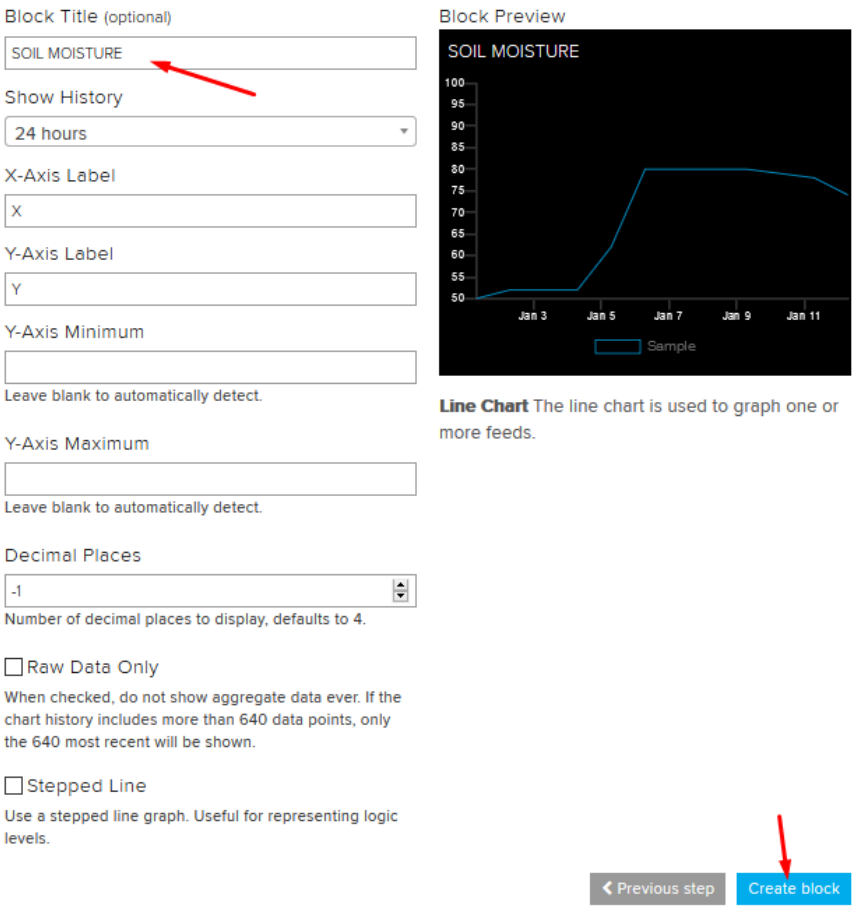
Pilih line chart, selanjutnya beri centang pada cek box soilmoisture. Kemudian pilih Next step





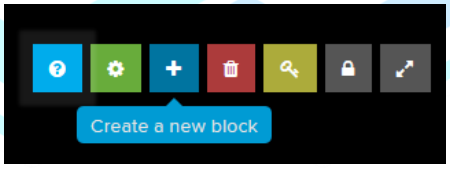
Berikutnya masuk pada Block setting ganti keterangan dan parameter line chart:

Block Title : SOIL MOISTURE

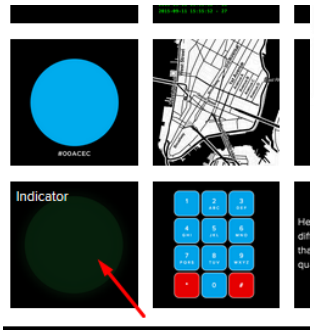


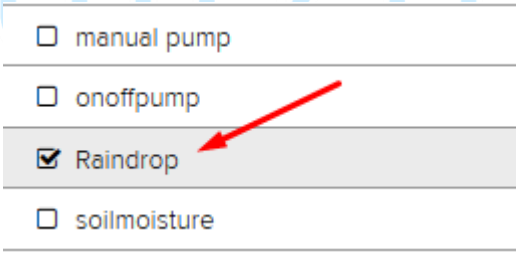
Jika sudah terisi klik Create block.

- Selanjutnya sensor hujan, pilih button Create a new block (+)



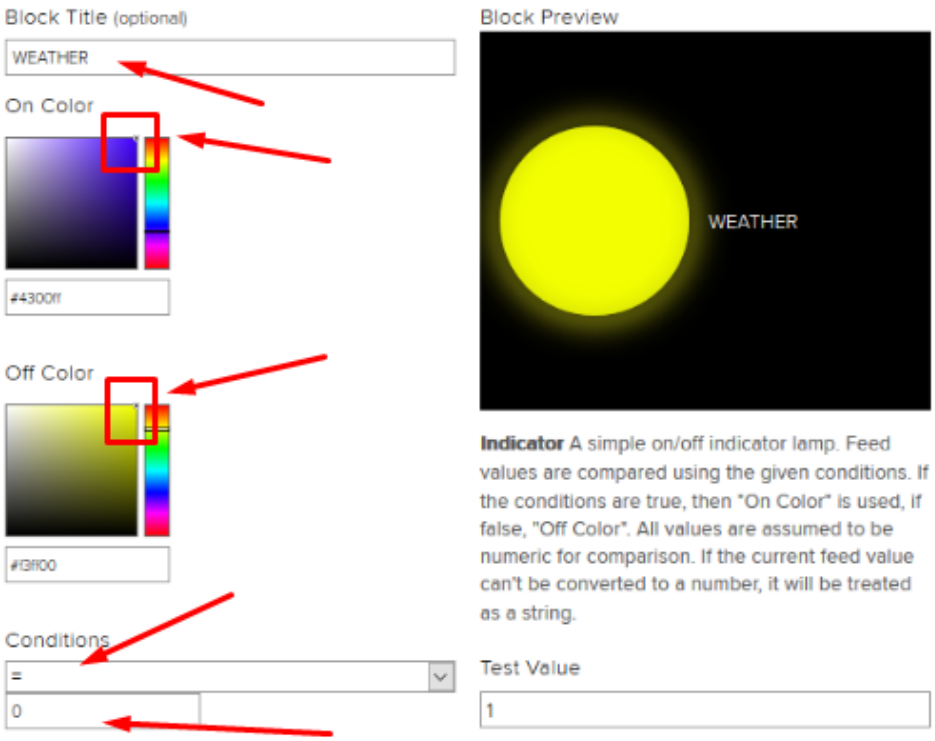
Pilih Indicator, selanjutnya beri centang pada cek box Raindrop. Kemudian pilih Next step





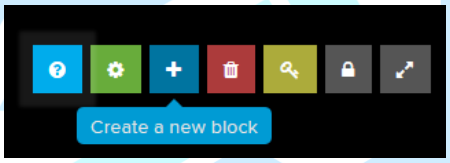
Berikutnya masuk pada Block setting ganti keterangan dan parameter Indicator:

1. Block Title : WEATHER
2. On Color : Biru tua
3. Off Color : Kuning
4. Conditions : = 0



Jika sudah terisi klik Create block.

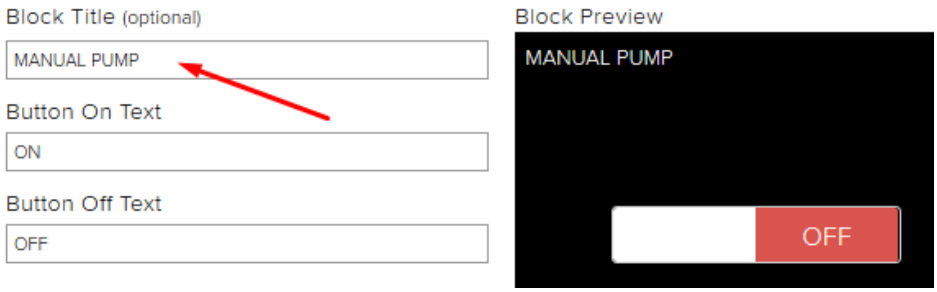
- Selanjutnya tombol manual pump, pilih button Create a new block (+)



Pilih Toggle, selanjutnya beri centang pada cek box manual pump. Kemudian pilih Next step.

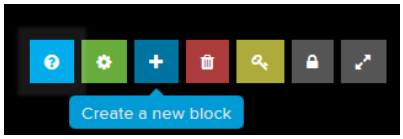
Berikutnya masuk pada Block setting ganti keterangan dan parameter Toggle:

Block Title : MANUAL PUMP

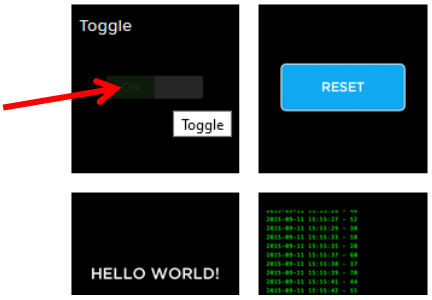


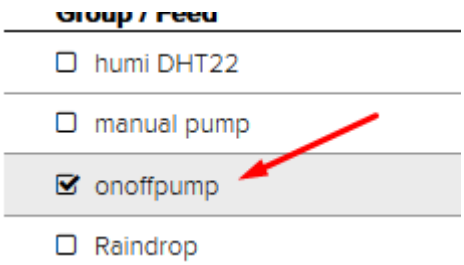
Jika sudah terisi klik **Create block**.

- Selanjutnya tombol pump control, pilih button Create a new block (+)



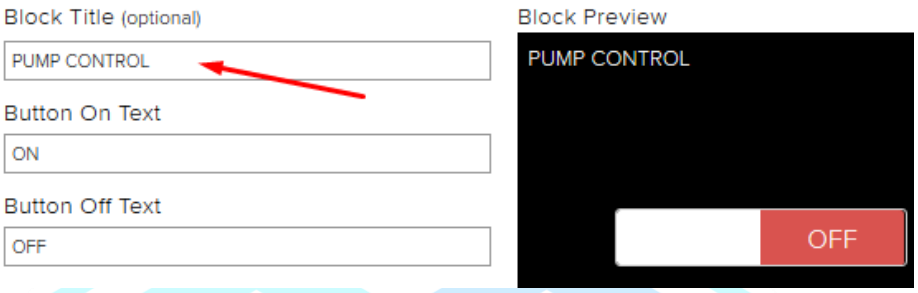
Pilih Toggle, selanjutnya beri centang pada cek box onoffpump. Kemudian pilih Next step





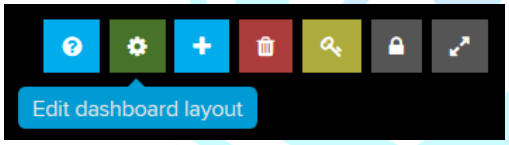
Berikutnya masuk pada Block setting ganti keterangan dan parameter Toggle:

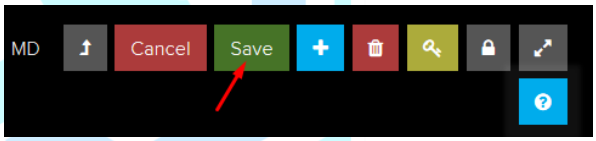
Block Title : MANUAL PUMP

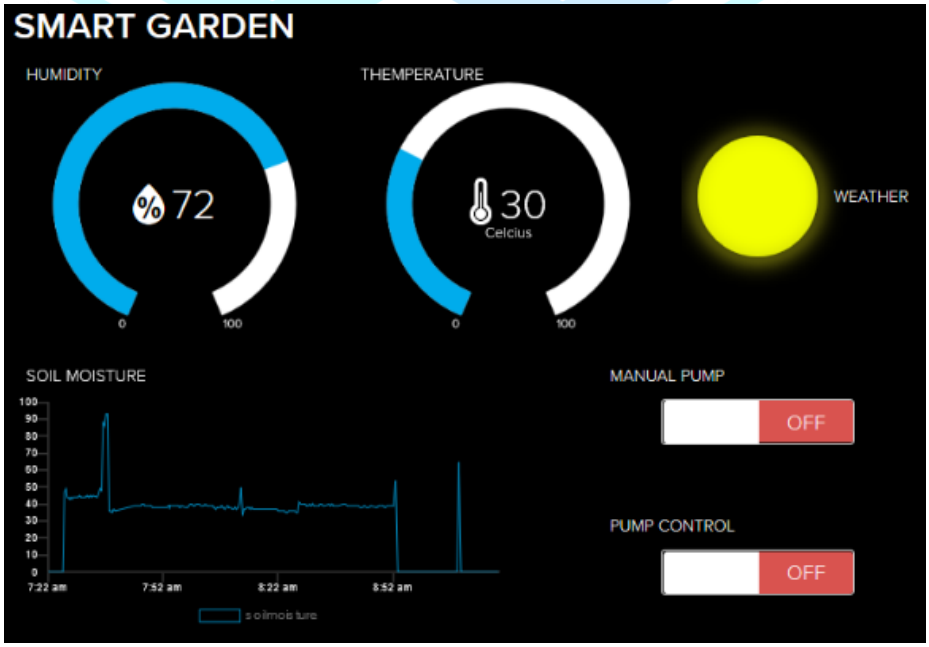


Jika sudah terisi klik **Create block**.

- Tahap terahir, silahkan Anda atur posisi dari masing masing block dengan klik Edit dashboard layout, jika sudah sesuai kemudian Anda save

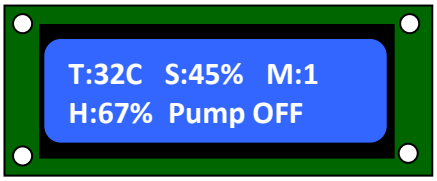






- **Jalannya Alat:**
  - LCD menampilkan koneksi WiFi, tunggu sampai OKE.

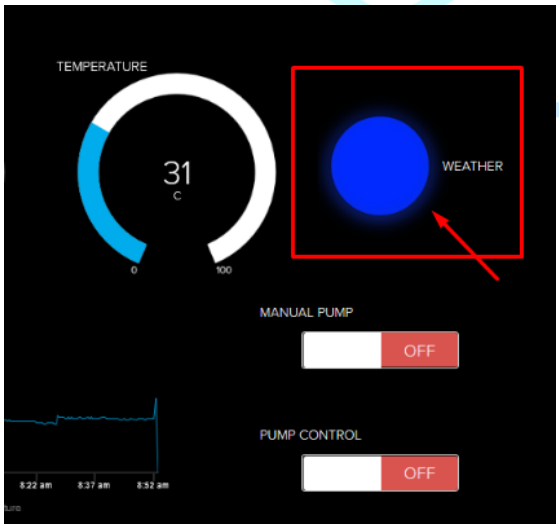
- Selanjutnya LCD menampilkan nilai pembacaan suhu, kelembaba udara, kelembaban tanah, status pompa dan Mode.

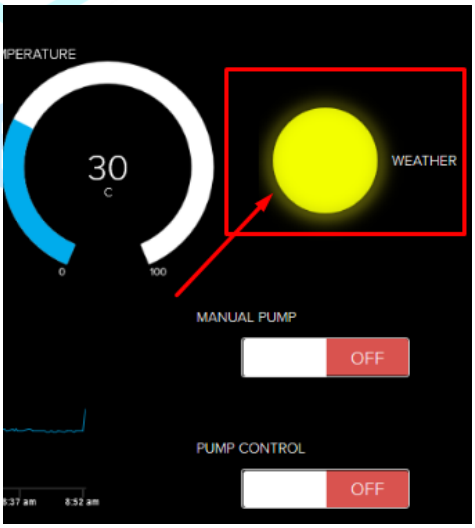


T = suhu yang terbaca oleh sensor DHT22
H = kelembaban udara yang terbaca oleh sensor DHT22
S = kelembaban tanah yang terbaca oleh sensor soil moisture
Pump OFF = adalah status dari pompa (ON / OFF)
M = mode 1 atau mode 2, mode 1 adalah system penyiraman tanah otomatis berdasarkan pembacaan sensor,mode 2 adalah system manual yang artinya pompa ON/OFF dikontrol oleh tombol manual yang ada di block SMART GARDEN Adafruit IO.

- Setelah 10 detik kirim data

- Indicator WEATHER berwarna biru jika terjadi hujan dan akan berwarna kuning jika tidak hujan.



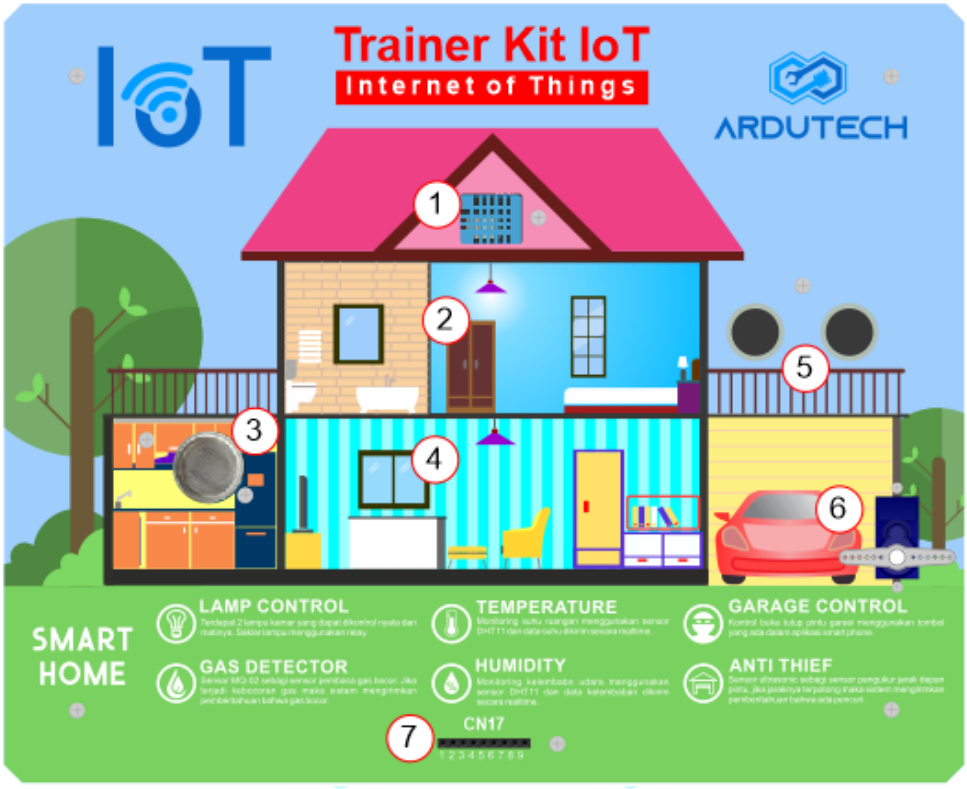


Project 16: Smart Home

- **Kebutuhan Hardware:**
  - NodeMCU V3
  - Sensor MQ-02

- Sensor DHT11
- Sensor Ultrasonic
- Servo SG90
- 2 LED
- Kabel micro USB
- 9 Kabel jumper female – female, 6 Kabel jumper male – female
  (semua hardware sudah dalam 1 panel kecuali board NodeMCU)

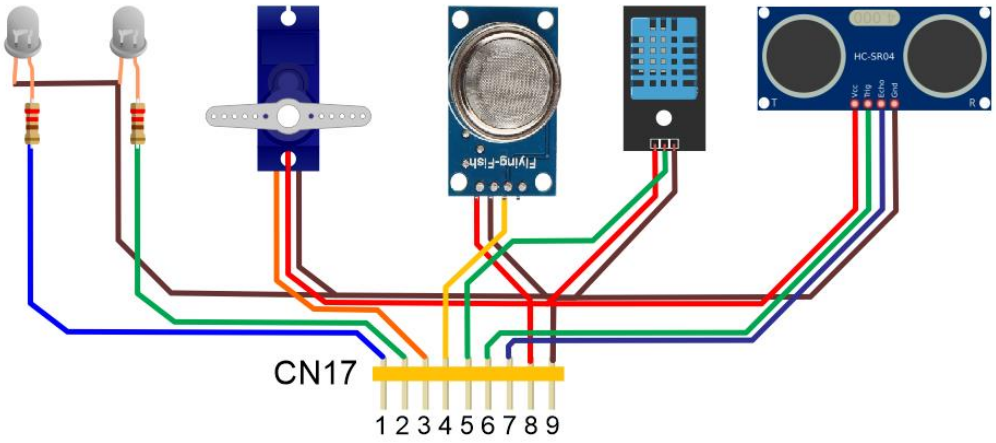
- **Panel Smart Home**



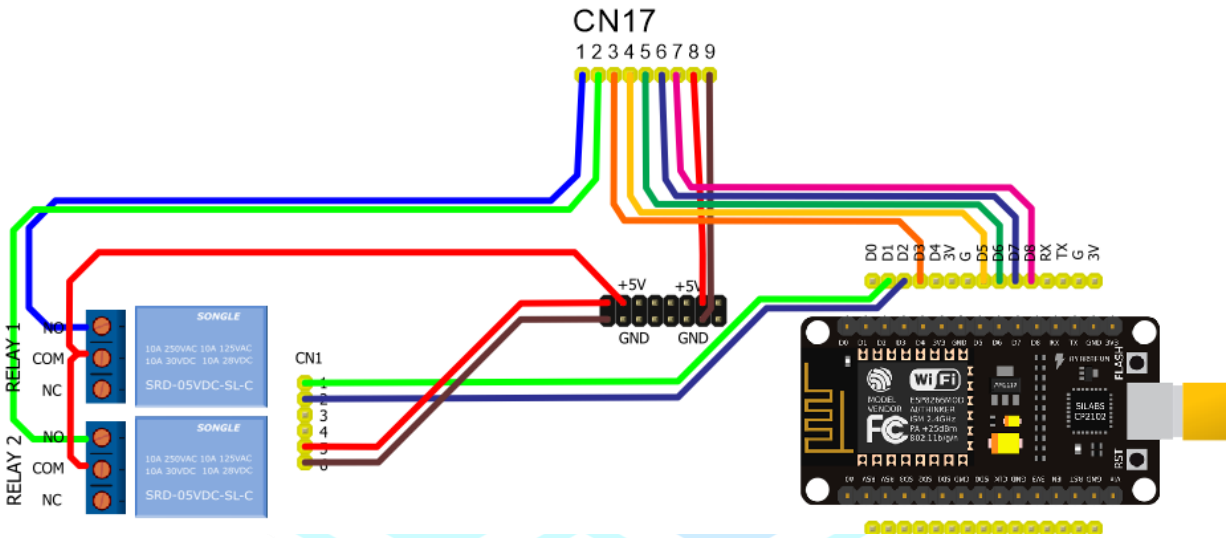
Keterangan :

Nomor	Keterangan	Fungsi
1	Sensor DHT11	Membaca suhu dan kelembaban udara
2	Ruangan Atas / Lampu 1	Menyalakan dan mematukan lampu ruang atas
3	Sensor Gas MQ-2	Membaca gas bocor
4	Ruangan Bawah / Lampu 2	Menyalakan dan mematikan lampu ruang bawah
5	Sensor Jarak Ultrasonic HC-SR04	Menetahui ada tidaknya pencuri
6	Motor Servo	Kontrol buka dan tutup pintu garasi
7	Konektor CN17	Konektor output dan input panel Smart Home

- **Skematik:**
  Skematik Panel smart home:



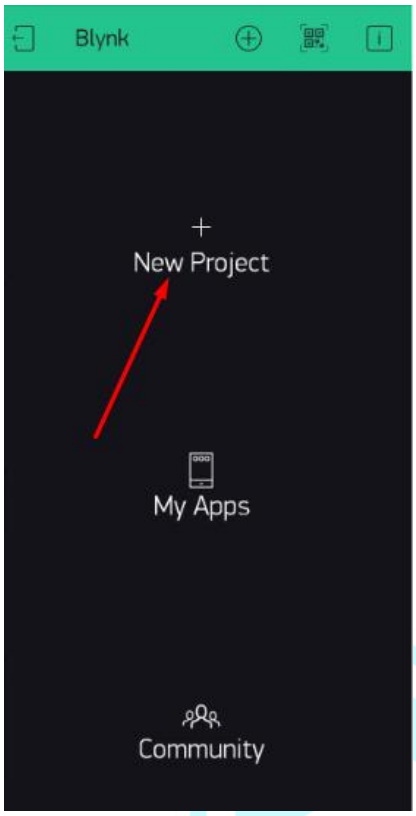
Koneksi panel smart home (CN17) dengan NodeMCU:

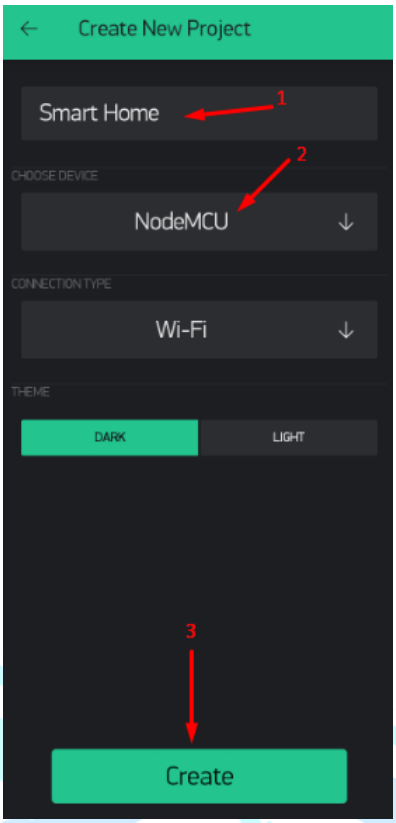


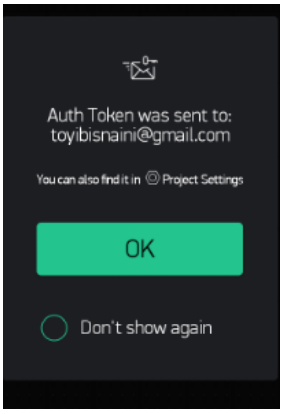
Keterangan:

PANEL SMART HOME	CN17	NodeMCU
	1/ RELAY 1 kontrol LED1	D1
	2/ RELAY 2 kontrol LED2	D2
	3/ SERVO	D3
	4/ DO MQ-02	D5
	5/ OUT DHT11	D6
	6/ TRIGER SR04	D7
	7/ ECHO SR04	D8
	8/VCC	+5V
	9/GND	GND

- **Kebutuhan Software:**
  - Arduino IDE
  - Smart phone : Blynk (tutorial Blynk lihat CD “Tutorial memulai Blynk.pdf”)
    Buat project di Blynk dengan nama SMART HOME. Device NodeMCU.



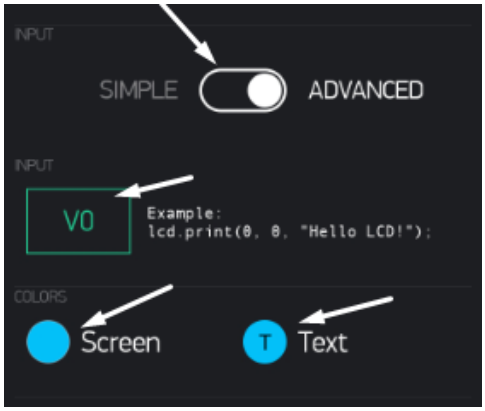




Buat display widget seperti berikut ini:

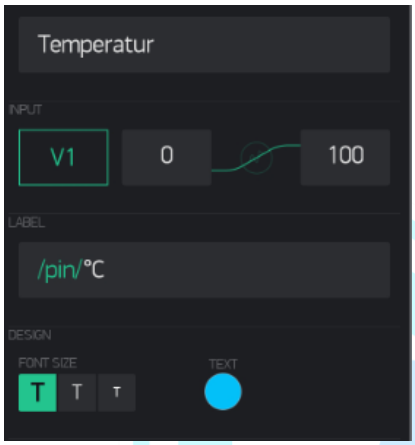
Setingan masing masing widget, klik widget sehingga masuk pada **setting widget**:

1. LCD



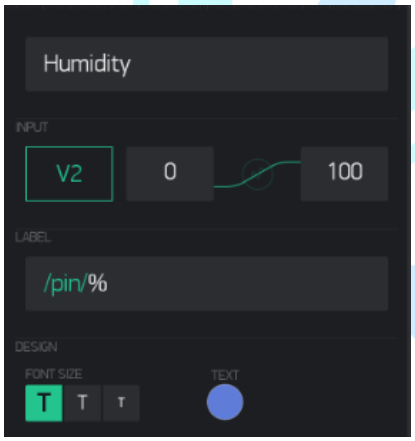
WIDGET	LCD
INPUT	ADVANCE
INPUT	V0
COLORS	BIRU

2. TEMPERATURE



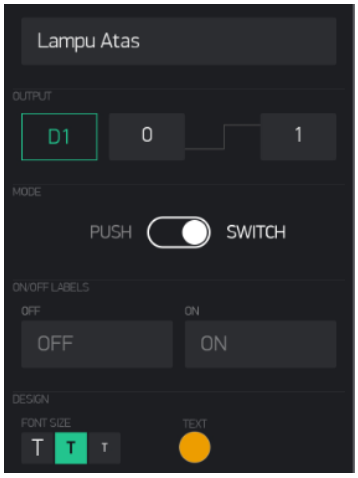
WIDGET	GAUGE
GAUGE	Temperature
INPUT	V1 : 0→100
LABEL	/pin/^{0}C
DESIGN	FONT SIZE : BESAR TEXT : BIRU

3. HUMIDITY



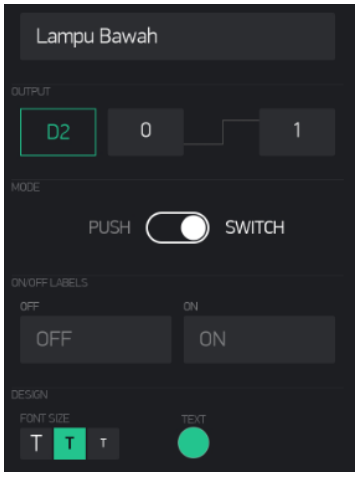
WIDGET	GAUGE
GAUGE	Humidity
INPUT	V2 : 0→100
LABEL	/pin/%
DESIGN	FONT SIZE : BESAR TEXT : UNGU

4. LAMPU ATAS



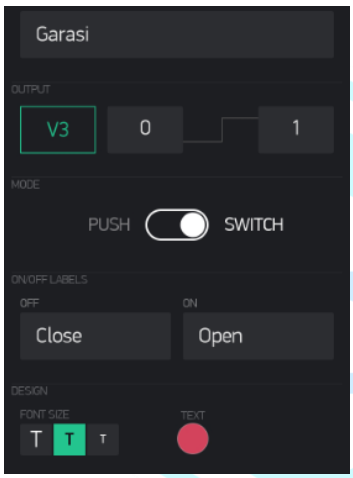
WIDGET	BUTTON
Button	Lampu Atas
INPUT	D1 : 0→1
MODE	SWITCH
DESIGN	FONT SIZE : SEDANG TEXT : ORANGE

5. LAMPU BAWAH



WIDGET	BUTTON
Button	Lampu Bawah
INPUT	D1 : 0→1
MODE	SWITCH
DESIGN	FONT SIZE : SEDANG TEXT : HIJAU

6. GARASI



WIDGET	BUTTON
Button	Garasi
INPUT	V3 : 0→1
MODE	SWITCH
ON/OFF LABELS	OFF: Close ON: Open
DESIGN	FONT SIZE : SEDANG TEXT : MERAH

7. NOTIFICATION tidak perlu seting

Blynk siap.

- **Program:**

  Ganti nama WiFi (ssid), password dan token (auth) anda.

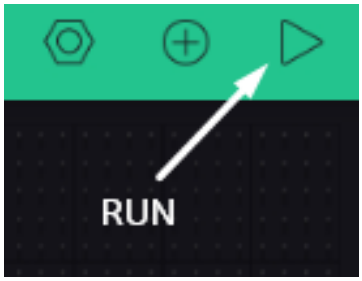
```
/*************************************************
 * Program : Project 11 Smart Garbage
 * Input   : DHT11, SR-04, MQ2
 * Output  : Buzzer, Servo, 2 LED
 * Server  : Blynk
 * Trainer Kit IoT Ardutech
 * www.ardutech.com
 ***********************************
 * NODEMCU  |    Koneksi
 *    D1   -->   Lampu 1
 *    D2   -->   Lampu 2
 *    D3   -->   Servo
 *    D5   -->   MQ2
 *    D6   -->   DHT11
 *    D7   -->   TRIG
 *    D8   -->   ECHO
 ***********************************/
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SPI.h>
#include <DHT.h>
#include <Servo.h>
#define MQ2Pin D5
#define DHTPIN D6
#define TRIGPIN D7
#define ECHOPIN D8
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
Servo servo;

//Ganti dengan Token anda
char auth[] = "-htbXm6E3Lizwp915_No7P516Ywa7nrj";
// Ganti dengan WiFi anda
char ssid[] = "Ardutech";
char pass[] = "12345678";

WidgetLCD lcd(V0);
byte humi, temp;
unsigned int jarak, tmr;
int f,MQ2Value;

BLYNK_WRITE(V3){
  int buttonState = param.asInt();
  if(buttonState==HIGH){
    servo.write(0);
    delay(100);
  }
  else if(buttonState==LOW){
    delay(100);
    servo.write(90);
```

```
  }
}
//=====================================
void sendDataDHT(){
  humi = dht.readHumidity();
  temp = dht.readTemperature();
  if (isnan(humi) || isnan(temp)) {
    Serial.println("DHT11 tidak terbaca... !");
  }
  else{
    Serial.println(temp);
    Serial.println(humi);
    Blynk.virtualWrite(1, temp);//virtual input V10 di Blynk (suhu)
    Blynk.virtualWrite(2, humi);//virtual input V11 di Blynk (kelembaban)
  }
}
//================================
void ukur_jarak(){
  digitalWrite(TRIGPIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGPIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGPIN, LOW);
  tmr = pulseIn(ECHOPIN, HIGH);
  jarak= tmr/58;
  Serial.print("Jarak=");
  Serial.println(jarak);
  if(jarak<10){
    Serial.println("==>Pencuri terdeteksi");
    lcd.clear();
    lcd.print(0, 0, "  WASPADA !!!   ");
    lcd.print(0, 1, "Ada pencuri....");
    Blynk.notify("Pencuri terdeteksi.....!");
    tampilanNormal();
  }
}
//================================
void getMQ2Value(void){
  MQ2Value = digitalRead(MQ2Pin);
  if (!MQ2Value)
  {
    Serial.println("==>Gas terdeteksi");
    lcd.clear();
    lcd.print(0, 0, "  WASPADA !!!   ");
    lcd.print(0, 1, "Gas Bocor....");
    Blynk.notify("Kebocoran Gas terdeteksi.....!");
    tampilanNormal();
  }
}
//============================
void tampilanNormal(){
  lcd.clear();
  lcd.print(0, 0, "== SMART HOME ==");
```

```
  lcd.print(0, 1, ">>Ardutech.com<<");
}

//==========================
void setup(){
  delay(2500);
  dht.begin();
  Serial.begin(115200);
  Serial.println();
  Serial.println();
  servo.attach(D3);
  Blynk.begin(auth, ssid, pass);
  tampilanNormal();
  delay(1000);
  pinMode(MQ2Pin, INPUT);
  pinMode(ECHOPIN, INPUT);
  pinMode(TRIGPIN, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  servo.write(90);
}
//============================
void loop(){
  Blynk.run();
  sendDataDHT();
  delay(300);
  ukur_jarak();
  delay(300);
  getMQ2Value();
  delay(300);
}
```

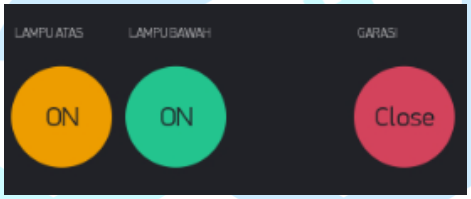
Simpan (***Save***) kemudian ***Upload***.

- **Jalannya Alat:**
  - Jalankan Blynk, klik run.



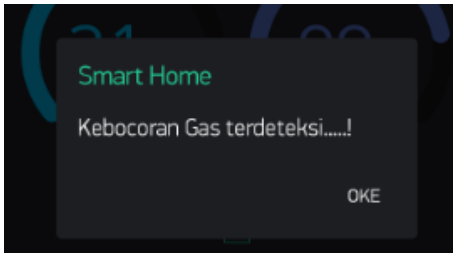
  - LCD menampilkan tampilan awal dan gauge menampilkan pembacaan suhu dan kelembaban.

- Silahkan anda kontrol lampu atas bawah dan garasi dengan menekan tombolnya.



- Notifikasi ada pencuri akan ditampilkan jika jarak terbaca ultrasonic lebih kecil dari 10cm

- Notifikasi kebocoran gas akan ditampilkan ada gas yang terbaca oleh sensor gas MQ-02



Trouble shooting

Trouble shooting dan lain sebagainya silakan cek juga di CD. Semua hardware dan project sudah di uji dan berjalan sesuai dengan fungsinya, jika tidak jalan silahkan anda cek:

1. **Instalasi** meliputi koneksi pin NodeMCU dengan sensor maupun actuator. Pastikan juga kabel masih normal, untuk cek kabel silahkan anda ukur dengan ohm meter apakah kabel masih bagus atau mungkin sudah putus.
2. **Power** pastikan power supply untuk sensor dengan inputan 5V mendapatkan 5V, begitu juga untuk sensor yang membutuhkan supply 3V3 tidak salah hubung.
3. **WiFi ERROR** tidak terhubung dengan jaringan, pastikan nama dan password benar. Gunakan jaringan WiFi dengan system keamanan WP2PSK, bukan WiFi dengan system keamanan harus login ke web.
4. **ERROR Koneksi dengan server** pastikan token/auth/API/AIO benar, semua karakter tertulis besar kecil tulisan harus sama. Pastikan alamat server benar.
5. **Test** jika semua sudah dipastikan dan hasilonya masih error, silahkan anda tes masing masing modul (yang berkaitan) dengan program tes yang sudah kita siapkan di CD pada folder **Hardware testing**.

Terimakasih.

Ardutech.com @2020