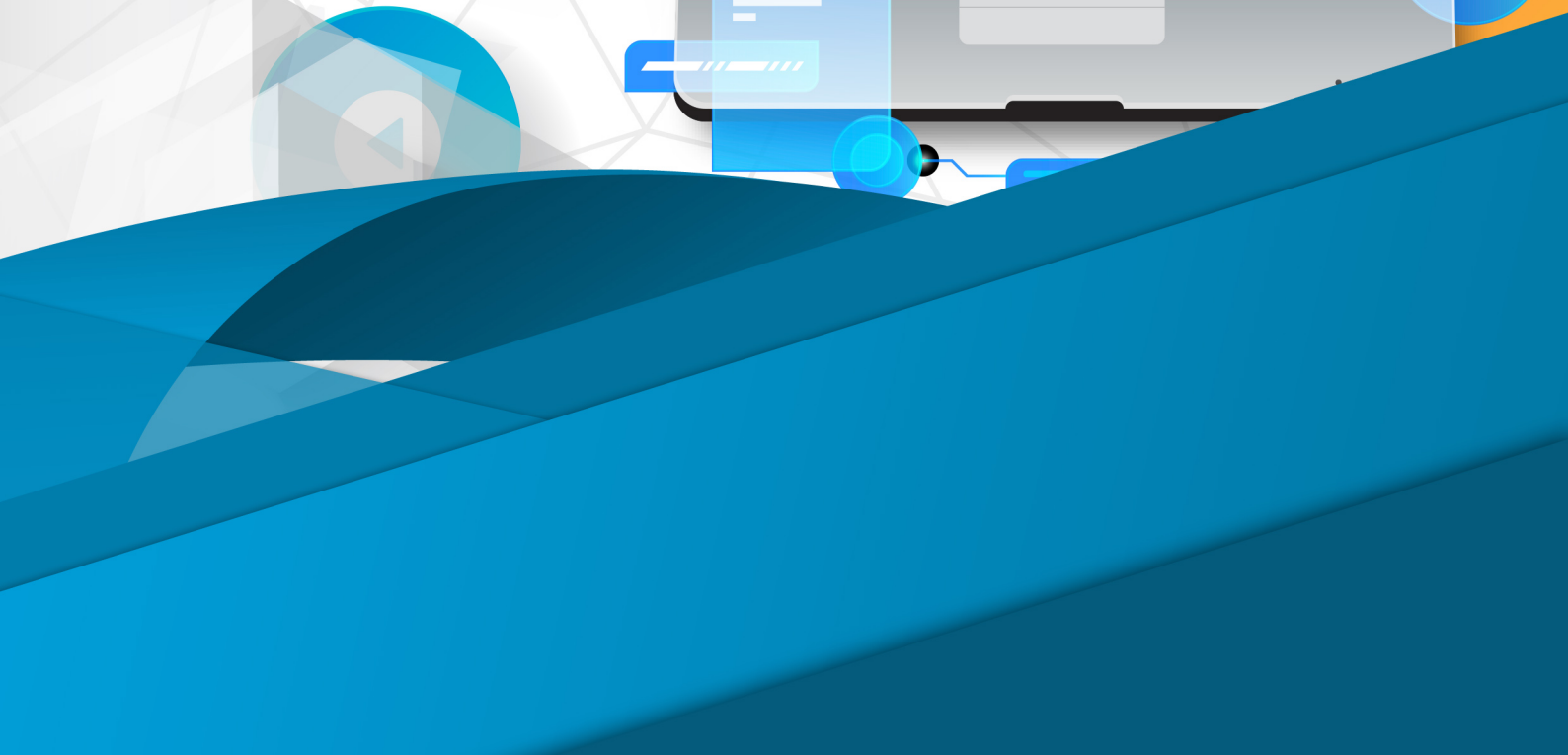
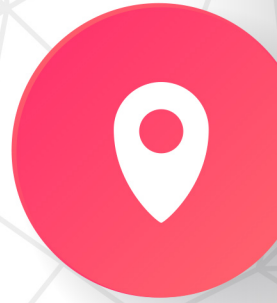


PANDUAN SESI PRAKTIKUM CERTIFICATE IN INTERNET OF THINGS PRACTITIONER



PANDUAN SESI PRAKTIKUM

**CERTIFIED INTERNET of THINGS
PRACTITIONER**

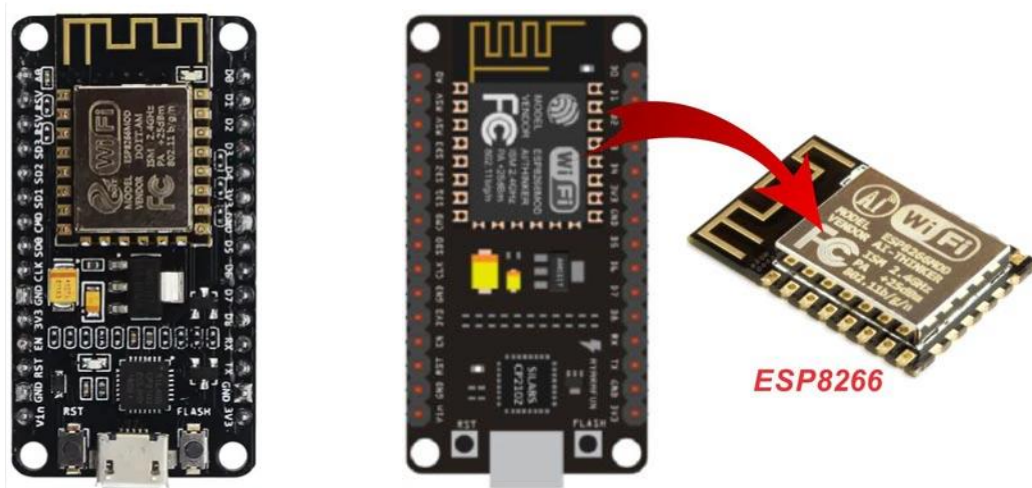


DAFTAR ISI

DAFTAR ISI	i
1. PENGENALAN KOMPONEN	1
1.1 Mikrokontroler NodeMCU	1
1.2 Papan sirkuit (breadboard)	4
1.3 Resistor.....	5
1.4 Kabel jumper.....	7
1.5 LED (Light Emitting Diode)	8
1.6 Sensor Suhu dan Kelembapan - DHT11	10
1.7 Sensor Gerak – PIR.....	11
2. PENGENALAN PERANGKAT LUNAK ARDUINO IDE, APLIKASI BLYNK, DAN THINGSPEAK	14
2.1 Perangkat Lunak Arduino IDE	14
2.2 Aplikasi Blynk.....	19
2.3 Aplikasi Thingspeak.....	24
3. PRAKTIKUM	27
3.1 Praktikum 1 : LED berkedip	27
3.2 Praktikum 2 : Kontrol LED menggunakan Blynk	30
3.3 Praktikum 3 : Kontrol LED menggunakan Thingspeak.....	36
3.4 Praktikum 4 : Pengukuran suhu dan kelembapan menggunakan DHT11	51
3.5 Praktikum 5 : Pengukuran suhu dan kelembapan menggunakan DHT11 & Blynk	55
3.6 Praktikum 6 : Pengukuran suhu dan kelembapan menggunakan DHT11 & Thingspeak	60
3.7 Praktikum 7 : Deteksi gerak menggunakan sensor PIR	69
3.8 Praktikum 8 : Deteksi gerak menggunakan sensor PIR & Blynk	71
3.9 Praktikum 9 : Deteksi gerak menggunakan sensor PIR & Thingspeak.....	77

1. PENGENALAN KOMPONEN

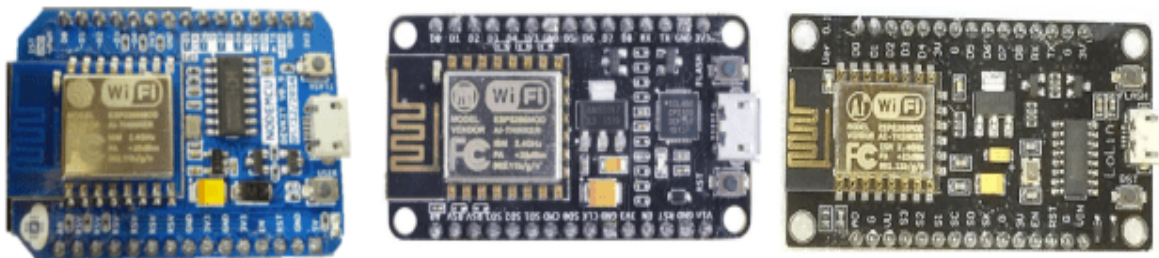
1.1 Mikrokontroler NodeMCU



NodeMCU ESP8266 merupakan modul turunan pengembangan dari modul platform IoT (Internet of Things) keluarga ESP8266 tipe ESP-12. Secara fungsi modul ini hampir serupa dengan platform modul arduino namun dikhususkan untuk "Connected to Internet".

Untuk saat ini modul NodeMCU sudah terdapat 3 tipe versi antara lain :

Versi NodeMCU ESP8266



Versi 0.9

Versi 1.0 (official)

Versi 1.0 (unofficial)

NodeMCU 0.9

Pada versi ini (v0.9) merupakan versi pertama yang memiliki memori flash 4 MB sebagai (System on Chip) SoC-nya dan ESP8266 yang digunakan yaitu ESP-12. Kelemahan dari versi ini yaitu dari segi ukuran modul board lebar, sehingga apabila ingin membuat protipe menggunakan modul versi ini pada breadboard, pin-nya akan habis digunakan hanya untuk modul ini.

NodeMCU 1.0

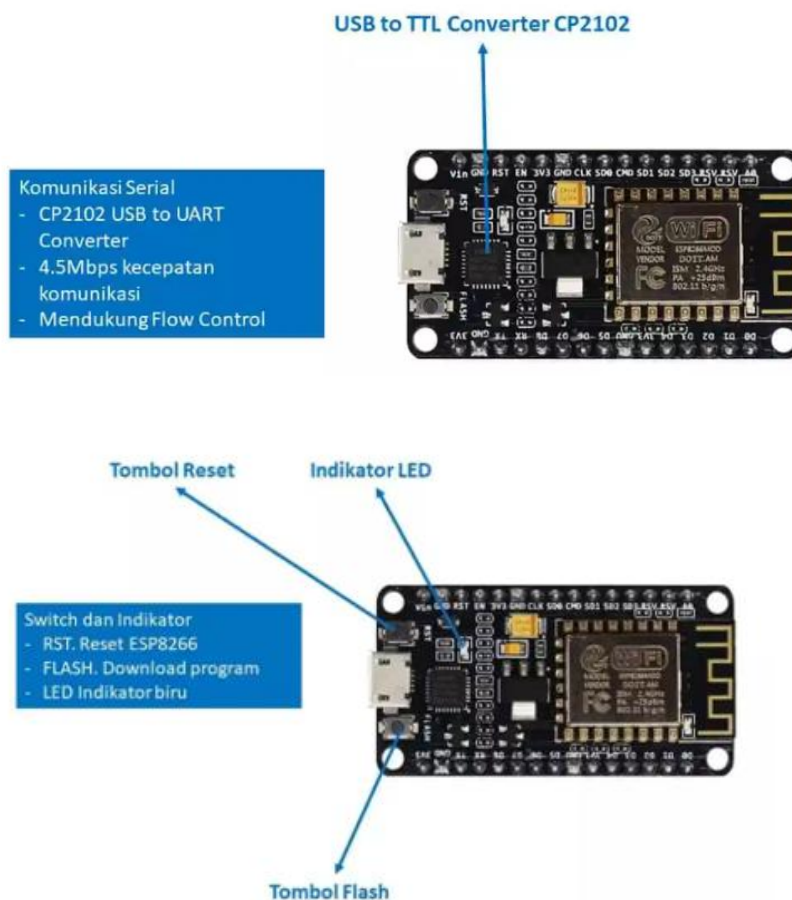
Versi ini merupakan pengembangan dari versi 0.9. Dan pada versi 1.0 ini ESP8266 yang digunakan yaitu tipe ESP-12E yang dianggap lebih stabil dari ESP-12. Selain itu ukuran board modulnya diperkecil sehingga compatible digunakan membuat prototipe projek di breadboard. Serta

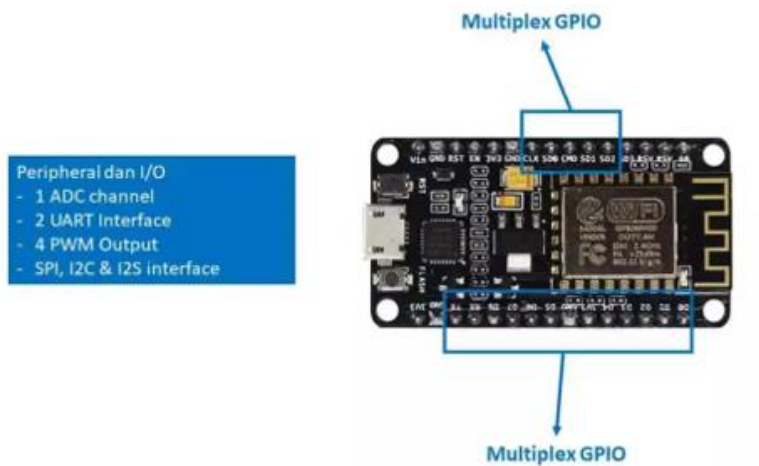
terdapat pin yang dikhususkan untuk komunikasi SPI (Serial Peripheral Interface) dan PWM (Pulse Width Modulation) yang tidak tersedia di versi 0.9.

NodeMCU 1.0 (unofficial board)

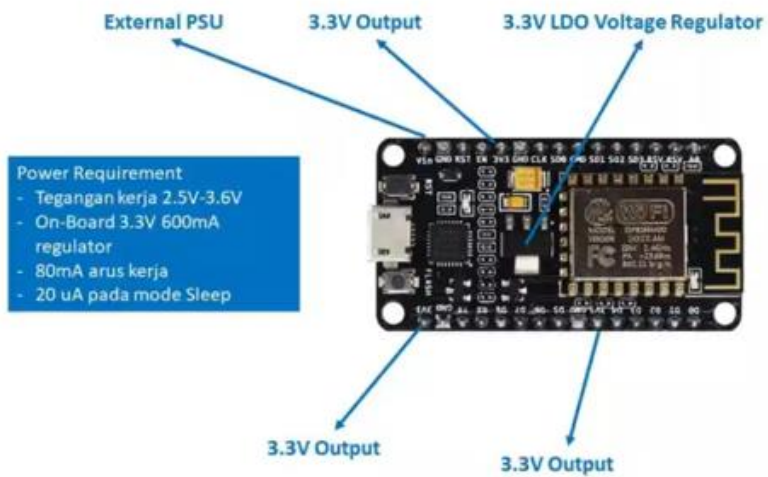
Dikatakan unofficial board dikarenakan produk modul ini diproduksi secara tidak resmi terkait persetujuan dari Developer Official NodeMCU. Perbedaannya tidak begitu mencolok dengan versi 1.0 (official board) yaitu hanya penambahan V usb power output.

Spesifikasi	NodeMCU		
	0.9	1.0 (Official board)	1.0 (Unofficial board)
Vendor Pembuat	Amica	Amica	LoLin
Tipe ESP8266	ESP12	ESP-12E	ESP-12E
USB port	Micro Usb		
GPIO Pin	11	13	13
ADC	1 pin (10 bit)		
Usb to Serial Converter	CH340G	CP2102	CH340G
Power Input	5 Vdc		
Ukuran Module	47 x 31 mm	47 x 24 mm	57 x 30 mm

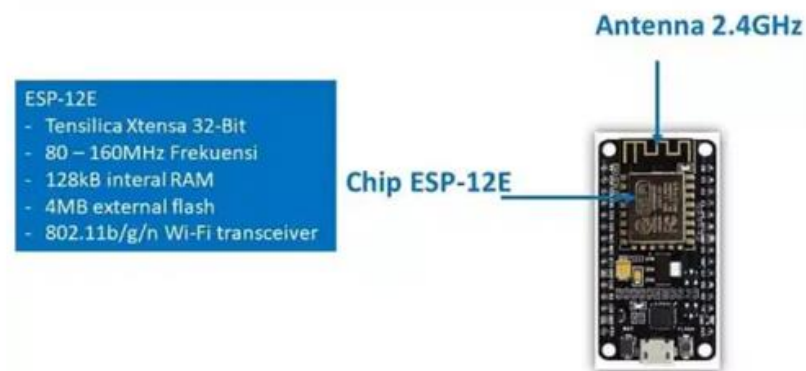




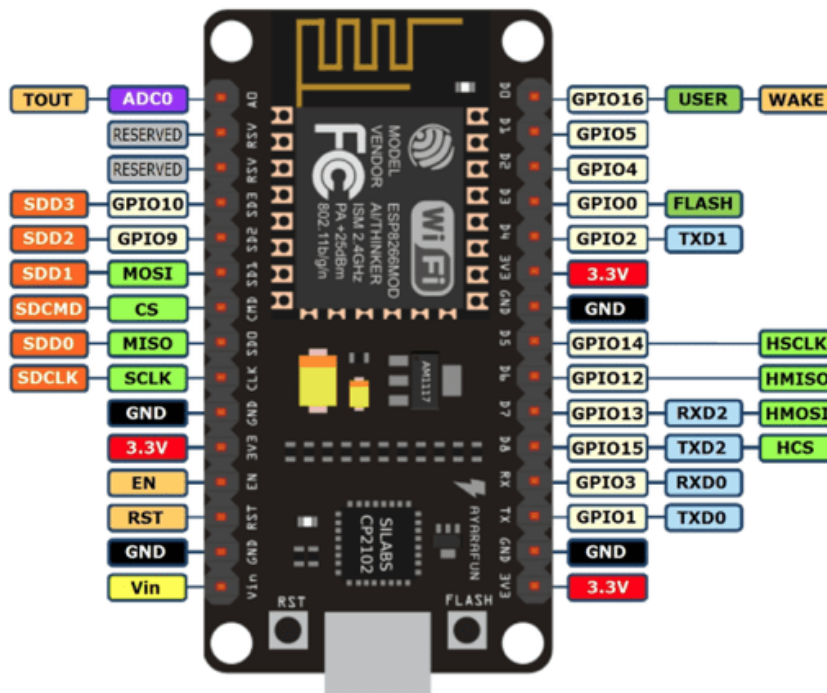
- Peripheral dan I/O
- 1 ADC channel
 - 2 UART Interface
 - 4 PWM Output
 - SPI, I2C & I2S interface



- Power Requirement
- Tegangan kerja 2.5V-3.6V
 - On-Board 3.3V 600mA regulator
 - 80mA arus kerja
 - 20 uA pada mode Sleep



- ESP-12E
- Tensilica Xtensa 32-Bit
 - 80 – 160MHz Frekuensi
 - 128kB internal RAM
 - 4MB external flash
 - 802.11b/g/n Wi-Fi transceiver



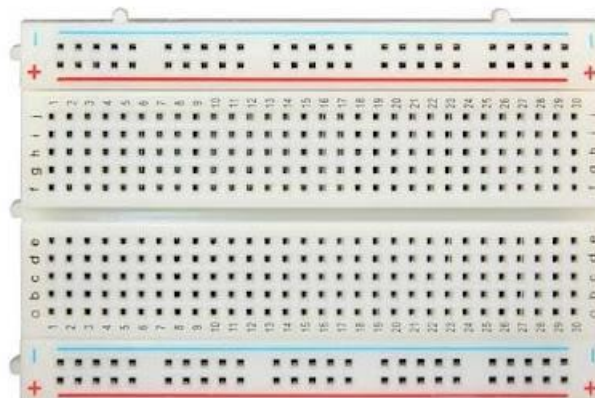
1.2 Papan sirkuit (breadboard)

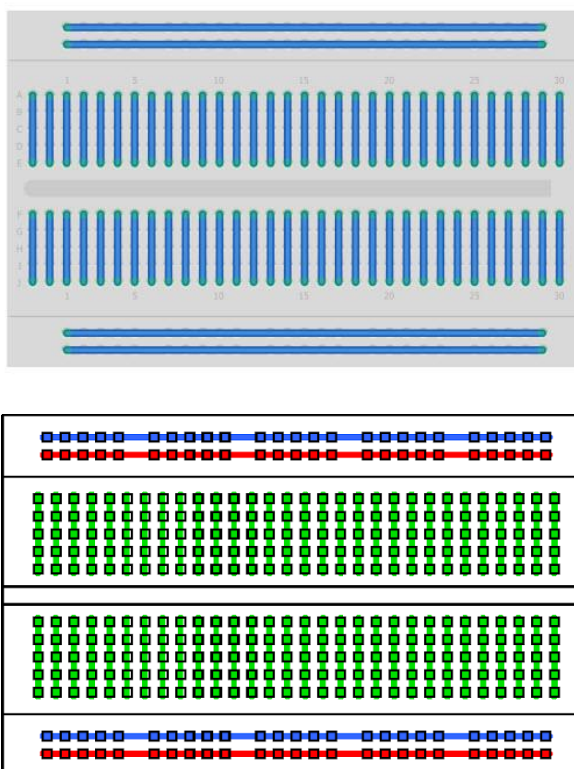
Breadboard atau projectboard adalah dasar konstruksi sebuah sirkuit elektronik yang merupakan bagian prototipe dari suatu rangkaian elektronik yang belum disolder sehingga masih dapat dirubah skema atau pengantian komponen.

Jenis-jenis breadboard ditentukan berdasarkan banyak lubang yang terdapat pada papan itu, misal breadboard 400 lubang, 170 lubang, dan lain sebagainya

Hal terpenting yang harus diketahui sebelum menggunakan project board ini yaitu memahami dengan baik bagaimana jalur yang saling terhubung antara satu lubang dengan lainnya.

Gambar di bawah memperlihatkan hubungan antar lubang pada salah satu jenis breadboard 400 titik.



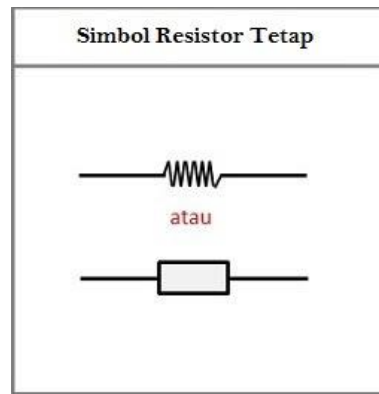


Tampak bahwa deretan lubang di bagian atas dan bawah ditandai dengan garis merah dan biru. Deretan lubang yang ditandai garis merah menunjukkan jalur positif untuk catudaya, sedangkan yang ditandai garis biru merupakan jalur negatif untuk catudaya. Lubang-lubang di bagian tengah terbagi dalam dua kelompok, yaitu kelompok atas dan kelompok bawah. Hubungan antar-lubangnya adalah berderet kebawah.

1.3 Resistor



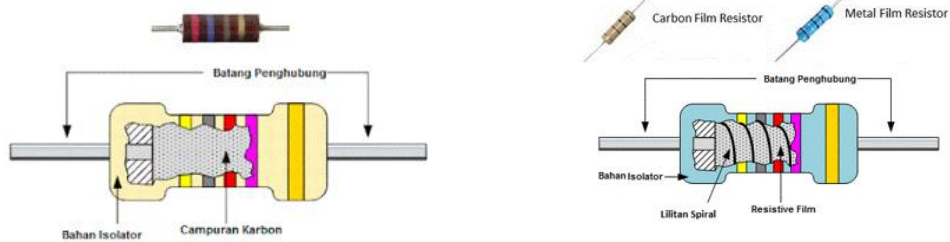
Resistor adalah komponen dasar elektronika yang umumnya digunakan pada rangkaian elektronika ataupun rangkaian listrik lainnya dengan fungsi utama yaitu menghambat/membatasi jumlah arus input atau arus yang mengalir masuk ke dalam satu rangkaian, dimana kemampuan resistor dalam membatasi arus masuk sesuai dengan spesifikasi resistor tersebut. Sesuai dengan namanya resistor bersifat resistif dan umumnya terbuat dari bahan karbon.



Resistor tetap merupakan jenis resistor yang nilainya sudah tertulis pada badan resistor dengan menggunakan kode warna ataupun angka. Resistor ini banyak digunakan sebagai penghambat arus listrik secara permanen.

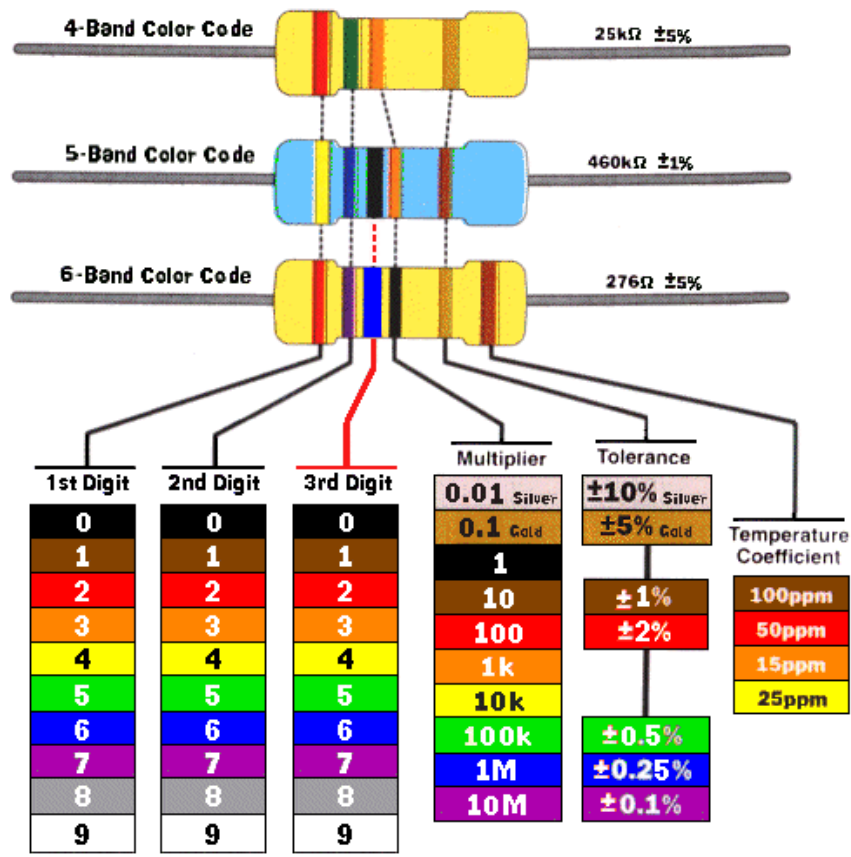


Resistor Tetap Selalu memiliki Kode Angka atau Kode Warna di Badannya



Cara membaca kode warna resistor ditemukan pada tahun 1920-an, nilai resistansi serta toleransi pada resistor ditampilkan berdasarkan deretan pita berwarna yang dilukis pada badan resistor.

Dua pita pertama menentukan nilai dari resistansi, pita ketiga menentukan faktor pengali, yang akan memberikan nilai resistansi dan terakhir, pita keempat menentukan nilai toleransi.



1.4 Kabel jumper



Jenis jenis kabel jumper yang paling umum adalah sebagai berikut:

- Kabel Jumper Male to Male

Jenis yang pertama adalah kabel jumper *male male*. Kabel jumper male to male adalah jenis yang sangat cocok untuk kamu yang mau membuat rangkaian elektronik di *breadboard*.



- Kabel Jumper Male to Female

Kabel jumper male female memiliki ujung konektor yang berbeda pada tiap ujungnya, yaitu *male* dan *female*.



- Kabel Jumper Female to Female

Jenis kabel jumper yang terakhir adalah kabel *female to female*. Kabel ini sangat cocok untuk menghubungkan antar komponen yang memiliki *header male*. contohnya seperti sensor ultrasonik *HC-SR04*, sensor suhu *DHT*, dan masih banyak lagi.

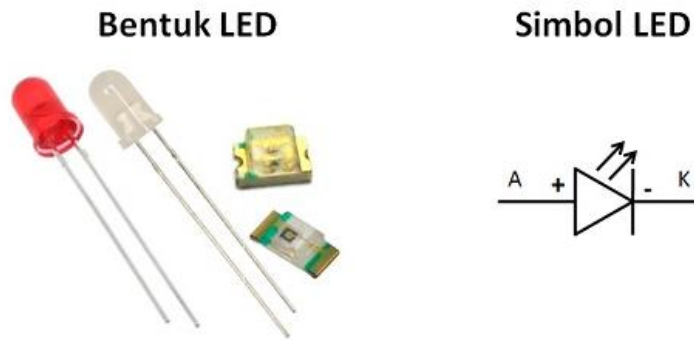


1.5 LED (Light Emitting Diode)

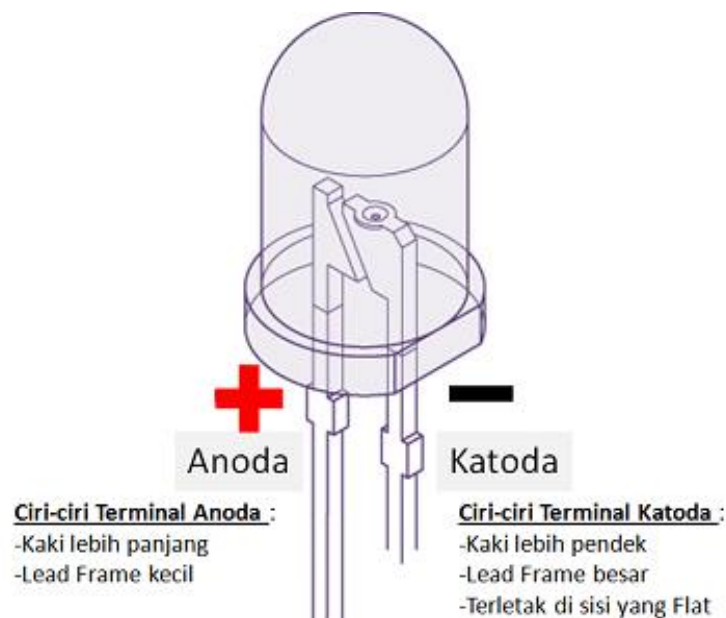


Light Emitting Diode atau LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering kita jumpai pada Remote Control TV ataupun Remote Control perangkat elektronik lainnya.

Bentuk LED mirip dengan sebuah bohlam (bola lampu) yang kecil dan dapat dipasangkan dengan mudah ke dalam berbagai perangkat elektronika. Berbeda dengan Lampu Pijar, LED tidak memerlukan pembakaran filamen sehingga tidak menimbulkan panas dalam menghasilkan cahaya. Oleh karena itu, saat ini LED (Light Emitting Diode) yang bentuknya kecil telah banyak digunakan sebagai lampu penerang dalam LCD TV yang mengganti lampu tube.



Untuk mengetahui polaritas terminal Anoda (+) dan Katoda (-) pada LED. Ciri-ciri terminal Anoda pada LED adalah kaki yang lebih panjang dan juga Lead Frame yang lebih kecil. Sedangkan ciri-ciri terminal Katoda adalah kaki yang lebih pendek dengan Lead Frame yang besar serta terletak di sisi yang flat.



Masing-masing Warna LED (Light Emitting Diode) memerlukan tegangan maju (Forward Bias) untuk dapat menyalakannya. Tegangan Maju untuk LED tersebut tergolong rendah sehingga memerlukan sebuah resistor untuk membatasi arus dan tegangannya agar tidak merusak LED yang bersangkutan.

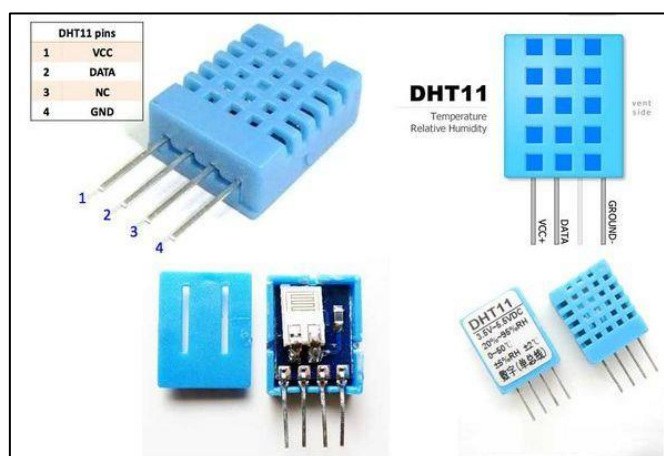
Warna	Tegangan Maju @20mA
Infra Merah	1,2V
Merah	1,8V
Jingga	2,0V
Kuning	2,2V
Hijau	3,5V
Biru	3,6V
Putih	4,0V

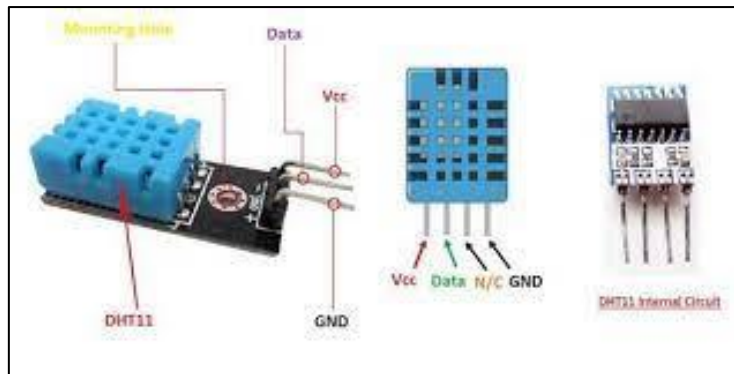
1.6 Sensor Suhu dan Kelembapan - DHT11

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

- **Pengukuran Kelembaban Udara**
 - Resolusi pengukuran: 16Bit
 - Repeatability: $\pm 1\%$ RH
 - Akurasi pengukuran: $25^{\circ}\text{C} \pm 5\%$ RH
 - Interchangeability: fully interchangeable
 - Waktu respon: $1 / e$ (63%) of 25°C 6 detik
 - Histeresis: $< \pm 0.3\%$ RH
 - Long-term stability: $< \pm 0.5\%$ RH / yr in
- **Pengukuran Temperatur**
 - Resolusi pengukuran: 16 Bit
 - Repeatability: $\pm 0.2^{\circ}\text{C}$
 - Range: At $25^{\circ}\text{C} \pm 2^{\circ}\text{C}$
 - Waktu Respon: $1 / e$ (63%) 10 detik
- **Karakteristik Kelistrikan**
 - Power supply: DC 3.5 – 5.5V
 - Konsumsi arus: measurement 0.3mA, standby $60\mu\text{A}$
 - Periode sampling : lebih dari 2 detik

Ada 2 macam tipe DHT11 yang umumnya sudah berupa modul, yakni DHT11 dengan 3 pin dan 4 pin. Pada modul DHT11 yang berkaki 4 ada satu pin yang tidak digunakan.

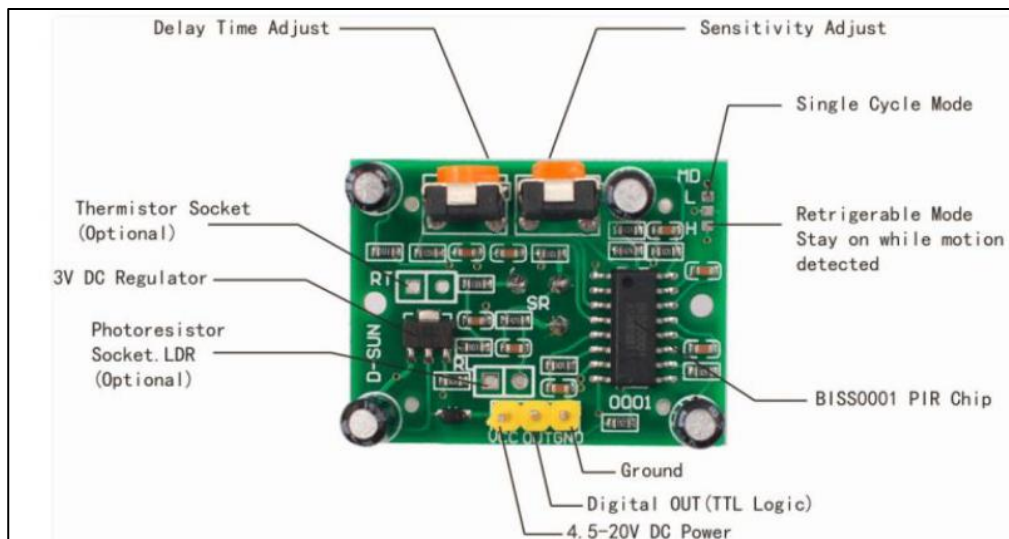


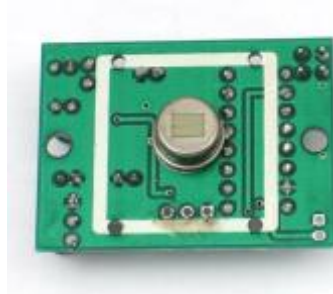


1.7 Sensor Gerak – PIR

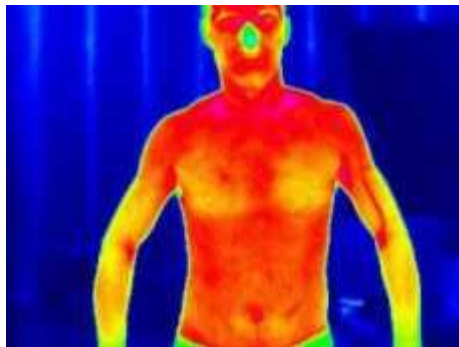


Sensor PIR merupakan sensor yang dapat mendeteksi pergerakan, dalam hal ini sensor PIR banyak digunakan untuk mengetahui apakah ada pergerakan manusia dalam daerah yang mampu dijangkau oleh sensor PIR. Sensor ini memiliki ukuran yang kecil, murah, hanya membutuhkan daya yang kecil, dan mudah untuk digunakan. Oleh sebab itu, sensor ini banyak digunakan pada skala rumah maupun bisnis. Sensor PIR ini sendiri merupakan kependekan dari “Passive InfraRed” sensor.








Pada umumnya sensor PIR dibuat dengan sebuah sensor *pyroelectric sensor* (seperti yang terlihat pada gambar disamping) yang dapat mendeteksi tingkat radiasi infrared. Segala sesuatu mengeluarkan radiasi dalam jumlah sedikit, tapi semakin panas benda/mahluk tersebut maka tingkat radiasi yang dikeluarkan akan semakin besar. Sensor ini dibagi menjadi dua bagian agar dapat mendeteksi pergerakan **bukan** rata-rata dari tingkat infrared. Dua bagian ini terhubung satu sama lain sehingga jika keduanya mendeteksi tingkat infrared yang sama maka kondisinya akan LOW namun jika kedua bagian ini mendeteksi tingkat infrared yang berbeda (terdapat pergerakan) maka akan memiliki output HIGH dan LOW secara bergantian.

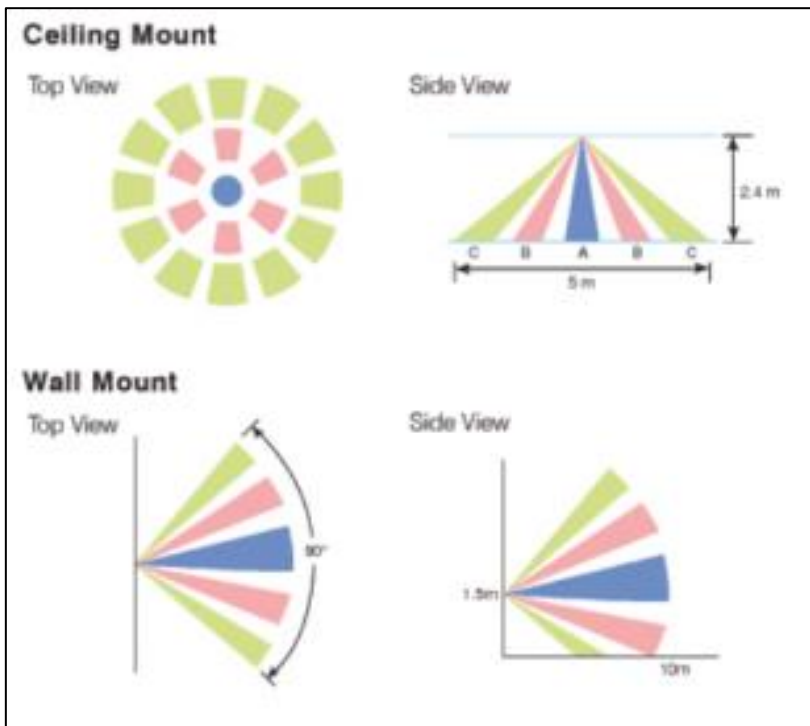




 Counter-Clockwise or Left Increases Sensitivity. Fully left and the range will be approximately 7 meters.	 Clockwise or Right Decreases Sensitivity. Fully right and the range will be approximately 3 meters.
---	---



	
<p>Counter-Clockwise or Left</p>	<p>Clockwise or Right</p>
<p>Decreases Delay Time. Fully left and the delay will be approximately 5 seconds.</p>	<p>Increases Delay Time. Fully right and the delay will be approximately 5 minutes.</p>

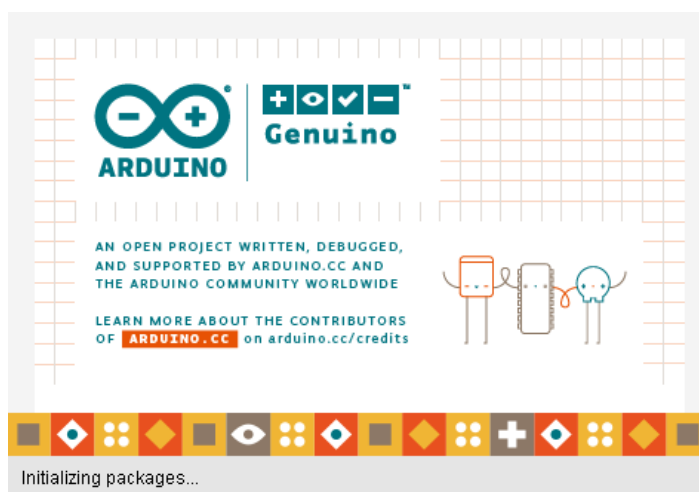


2. PENGENALAN PERANGKAT LUNAK ARDUINO IDE, APLIKASI BLYNK, DAN THINGSPEAK

2.1 Perangkat Lunak Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah software yang di gunakan untuk memprogram di arduino, dengan kata lain Arduino IDE sebagai media untuk memprogram *board* Arduino. Arduino IDE bisa di download secara gratis di website resmi Arduino IDE.

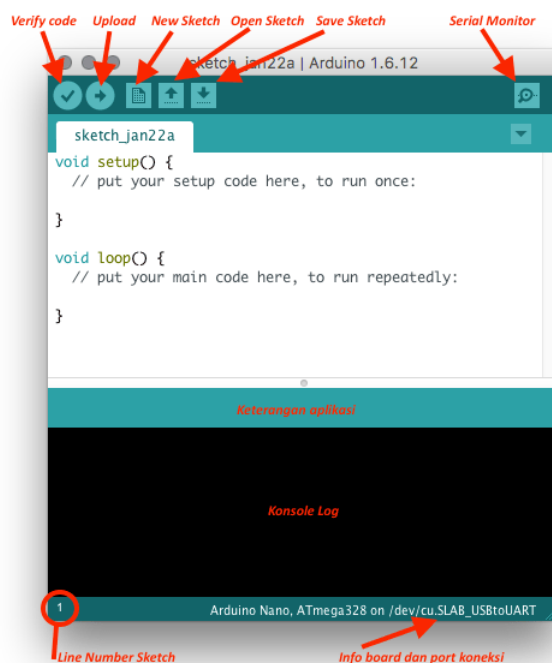
Arduino IDE ini berguna sebagai text editor untuk membuat, mengedit, dan juga mevalidasi kode program. bisa juga digunakan untuk meng-upload ke board Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino “sketch” atau disebut juga source code arduino, dengan ekstensi file source code .ino



Bagian-bagian Arduino IDE

Editor Programming pada umumnya memiliki fitur untuk cut / paste dan untuk find / replace teks, demikian juga pada Arduino IDE. Pada bagian *keterangan aplikasi* memberikan pesan balik saat menyimpan dan mengekspor serta sebagai tempat menampilkan kesalahan. **Konsol log** menampilkan teks log dari aktifitas Arduino IDE, termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pojok kanan bawah menampilkan port serial yang di gunakan. Tombol toolbar terdapat ikon tombol pintas untuk memverifikasi dan meng-upload program, membuat, membuka, dan menyimpan sketch, dan membuka monitor serial.

- **Verify** pada versi sebelumnya dikenal dengan istilah Compile. Sebelum aplikasi di-upload ke board Arduino, biasakan untuk memverifikasi terlebih dahulu sketch yang dibuat. Jika ada



kesalahan pada sketch, nanti akan muncul error. Proses Verify / Compile mengubah sketch ke binary code untuk di-upload ke mikrokontroler.

- **Upload** tombol ini berfungsi untuk mengupload sketch ke board Arduino. Walaupun kita tidak mengklik tombol verify, maka sketch akan di-compile, kemudian langsung diupload ke board. Berbeda dengan tombol verify yang hanya berfungsi untuk memverifikasi source code saja.
- **New Sketch** Membuka window dan membuat sketch baru.
- **Open Sketch** Membuka sketch yang sudah pernah dibuat. Sketch yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file .ino
- **Save Sketch** menyimpan sketch, tapi tidak disertai dengan mengcompile.
- **Serial Monitor** Membuka interface untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.
- **Keterangan Aplikasi** pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal **Compiling** dan **Done Uploading** ketika kita mengcompile dan mengupload sketch ke board Arduino
- **Konsol log** Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang sketch akan muncul pada bagian ini. Misal, ketika aplikasi mengcompile atau ketika ada kesalahan pada sketch yang kita buat, maka informasi error dan baris akan diinformasikan di bagian ini.
- **Baris Sketch** bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada sketch.
- **Informasi Board dan Port** Bagian ini menginformasikan port yang dipakai oleh board Arduino.

Sketch Arduino

Pada arduino bahasa pemrograman yang digunakan adalah bahasa C/C++. Program pada Arduino terbagi menjadi tiga bagian utama yaitu **Structure**, **Values** (berisi variable dan konstantata) dan yang terakhir **function**.

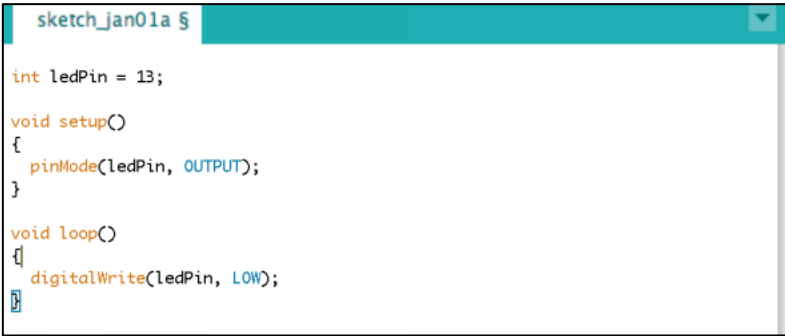
1. **Structure.** struktur kode pada arduino yaitu berisi fungsi `setup()` dan `loop()`.

a. Setup()

fungsi ini dipanggil pertama kali ketika menjalankan sketch. digunakan sebagai tempat inialisai *variable*, *pin mode*, penggunaan *library* dan lainnya. fungsi ini dijalankan sekali ketika *board* dinyalakan atau di reset.

b. loop()

Setelah membuat fungsi `setup()` sebagai tempat inialisai variabel dan menetapkan nilai maka selanjutnya fungsi `loop()` seperti namanya fungsi ini akan melakukan perulangan berturut-turut, memungkina program untuk mengubah dan menanggapi. digunakan untuk mengontrol *board* Arduino.



```

sketch_jan01a §
int ledPin = 13;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, LOW);
}

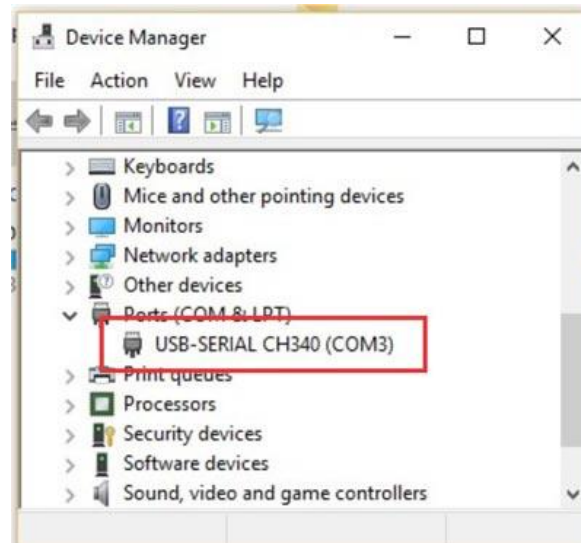
```

2. **Values.** Berisi variable atau konstanta sesuai dengan type data yang didukung oleh Arduino.
3. **Function.** Segmentasi kode ke fungsi memungkinkan programmer untuk membuat potongan-potongan modular kode yang melakukan tugas yang terdefinisi dan kemudian kembali ke asal

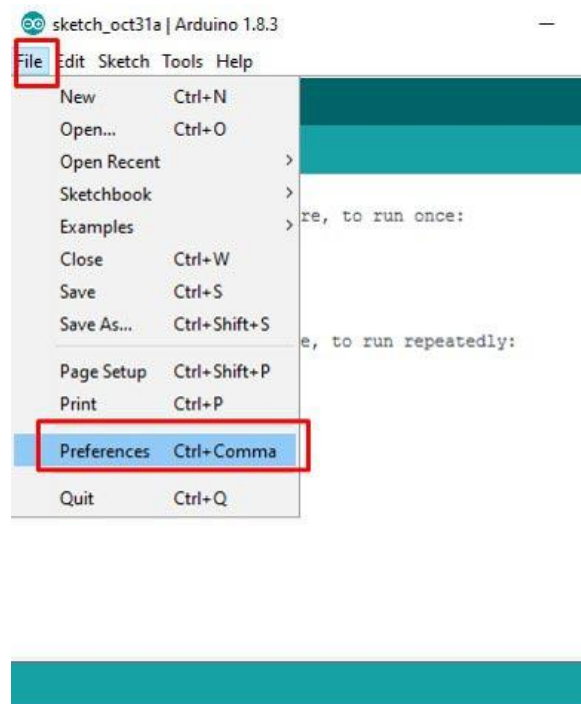
kode dari mana fungsi itu “dipanggil”. Umumnya menggunakan fungsi adalah ketika salah satu kebutuhan untuk melakukan tindakan yang sama beberapa kali dalam sebuah program.

Agar proyek praktikum dalam kegiatan ini dapat diselesaikan dengan baik maka perlu dilakukan langkah-langkah berikut ini :

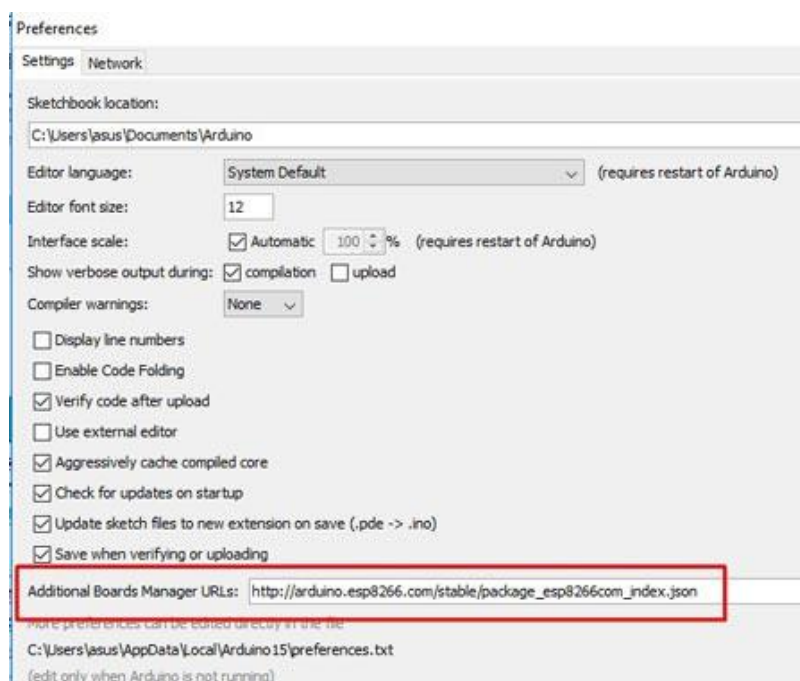
1. Siapkan kabel USB micro, kemudian sambungkan dengan port USB NodeMCU dan port USB komputer/laptop. Pastikan komputer /laptop terhubung dengan internet.
2. Cek apakah sudah terdeteksi di komputer. Buka *Device manager* dan terlihat merupakan USB-SERIAL CH340 karena memang driver yang digunakan NodeMCU V3 ini CH340. Mengenai nomor com dapat berbeda tiap komputer.



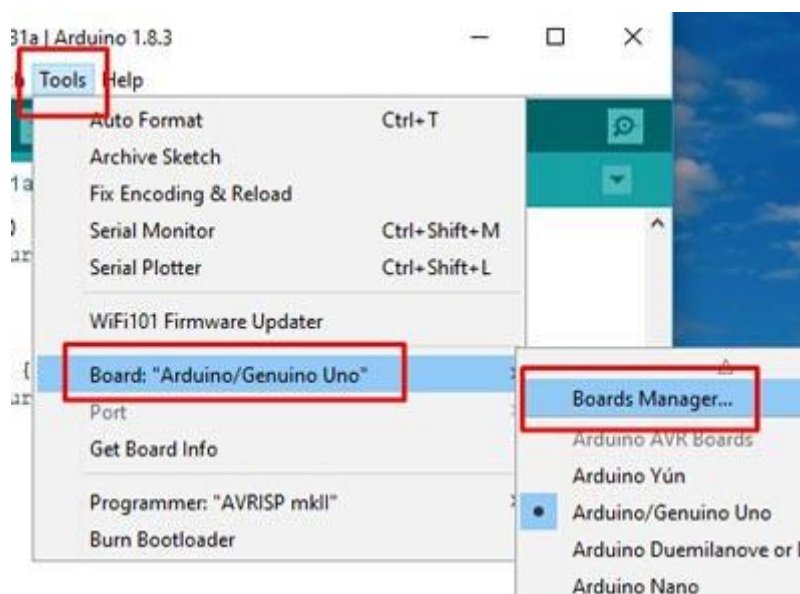
3. Sekarang buka/jalankan Arduino IDE. Klik menu *File – Preference*.



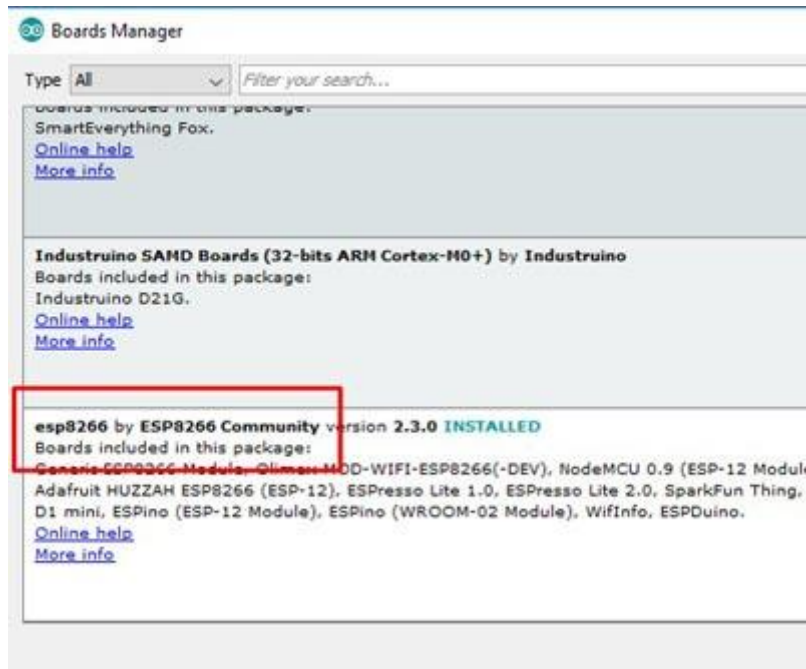
- Di bagian bawah terdapat kolom "Additional Board Manager URLs", isikan: http://arduino.esp8266.com/stable/package_esp8266com_index.json



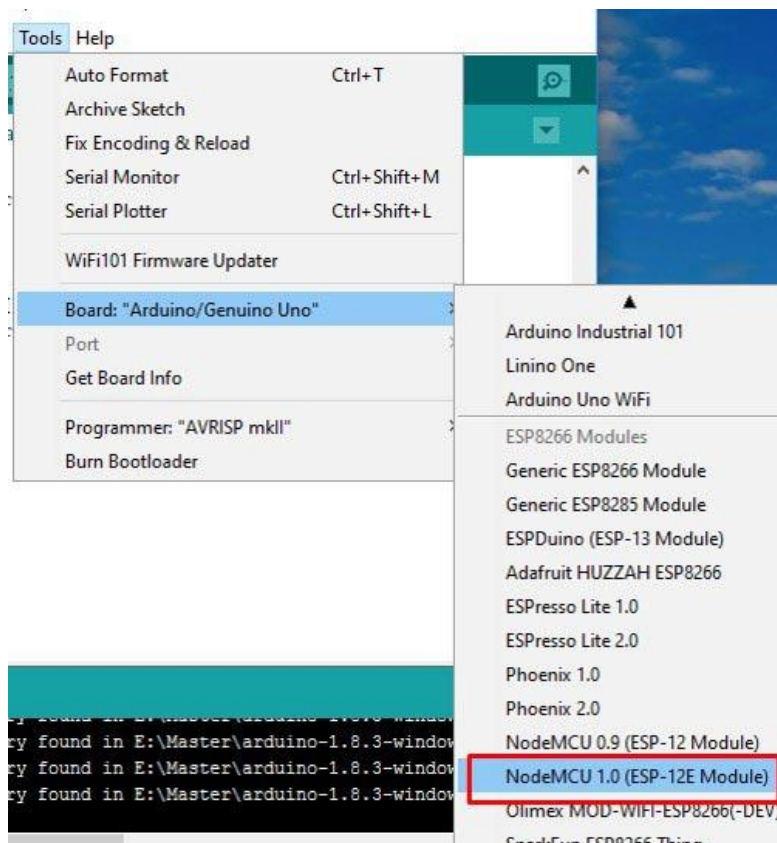
- Klik "OK". Selanjutnya dari menu **Tools – Board – Boards Manager...**



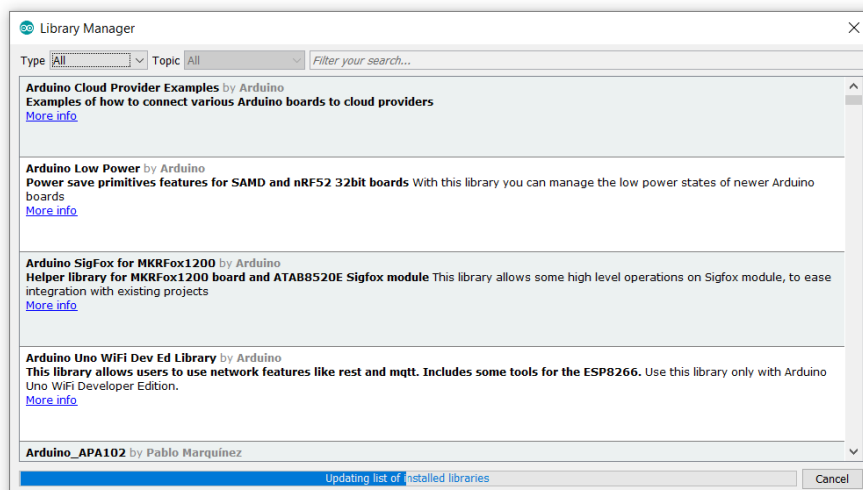
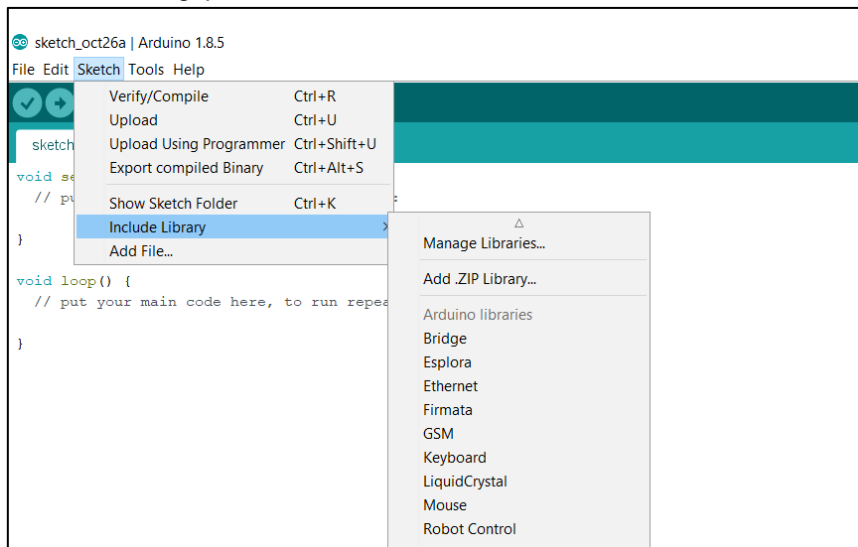
6. Pada kolom "file your search.." ketikkan **esp8266** hingga muncul "**esp8266 by esp8266 community**" kemudian klik **Install** disebelah kanan-nya. Tunggu hingga proses instalasi selesai.



7. Pilih board NodeMCU dari menu **Tools – Board** cari "**NodeMCU 1.0(ESP-12E Module)**".






- Pilih menu Sketch > Include Library > Manage Library. Kemudian pilih Library Manager. Ketik beberapa library yang perlu ditambahkan secara bergantian pada kolom "filter your search..": Blynk, DHT11, dan Thingspeak, kemudian install.



2.2 Aplikasi Blynk

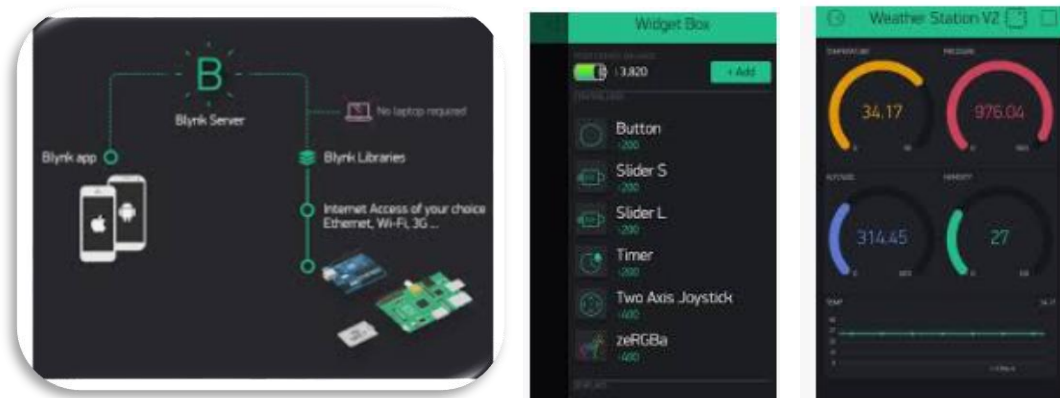
Blynk adalah aplikasi OS Mobile (iOS dan Android) yang bertujuan untuk kendali module *Arduino*, *Raspberry Pi*, *ESP8266*, *WEMOS D1*, dan module sejenisnya melalui Internet. Blynk dapat diinstal melalui Appstore atau Playstore.



 <h3>A Smartphone</h3> <ul style="list-style-type: none"> • Android OS version 4.2+ • iOS version 9+ <p>Blynk doesn't run on Windows Phones, Blackberries and other dead platforms. Sorry.</p> <p>You can also run Blynk on emulators</p>	 <h3>IoT Hardware</h3> <p>Blynk can run on over 400 hardware modules. The most popular are:</p> <ul style="list-style-type: none"> • ESP8266 • ESP32 • NodeMCU • Arduino (any model) • Raspberry Pi (any model) • Particle (any model) <p>Full list of supported hardware →</p>	 <h3>Internet Connection</h3> <p>To connect your hardware to the Internet, you can choose almost any module either built-in, or external shields</p> <p>Supported connectivity</p> <ul style="list-style-type: none"> • WiFi • Ethernet • Cellular (GSM, 2g, 3g, 4g, LTE) • Serial • USB via your PC • Bluetooth (BETA)
--	--	--

Penggunaannya sangat mudah untuk mengatur semuanya dan dapat dikerjakan dalam waktu singkat. Blynk tidak terikat pada papan atau module tertentu. Blynk dapat mengontrol perangkat dari jarak jauh, dimanapun kita berada dan waktu kapanpun selama terhubung melalui koneksi internet.

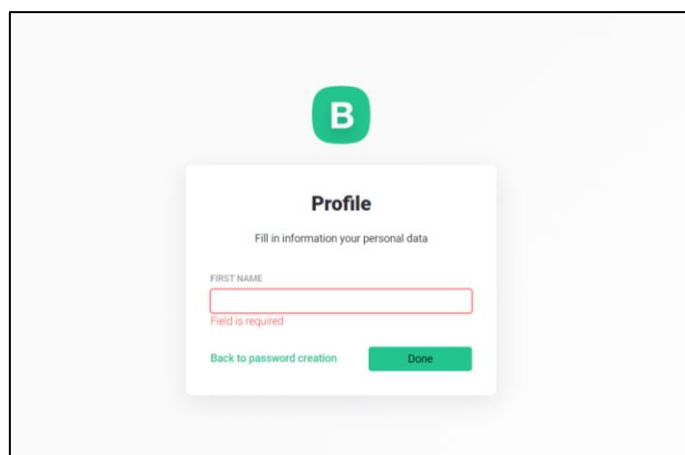
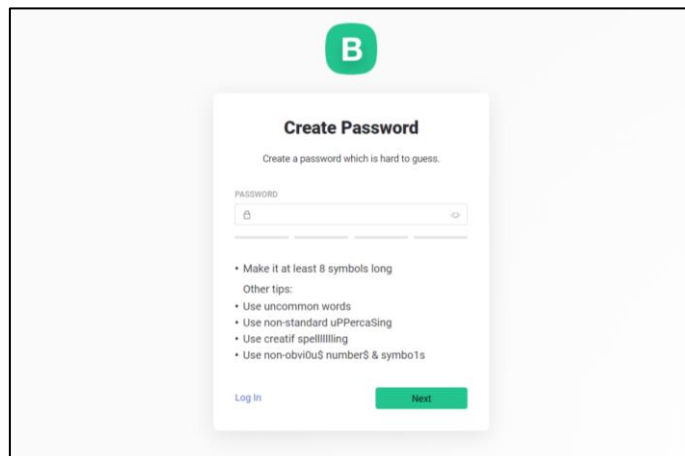
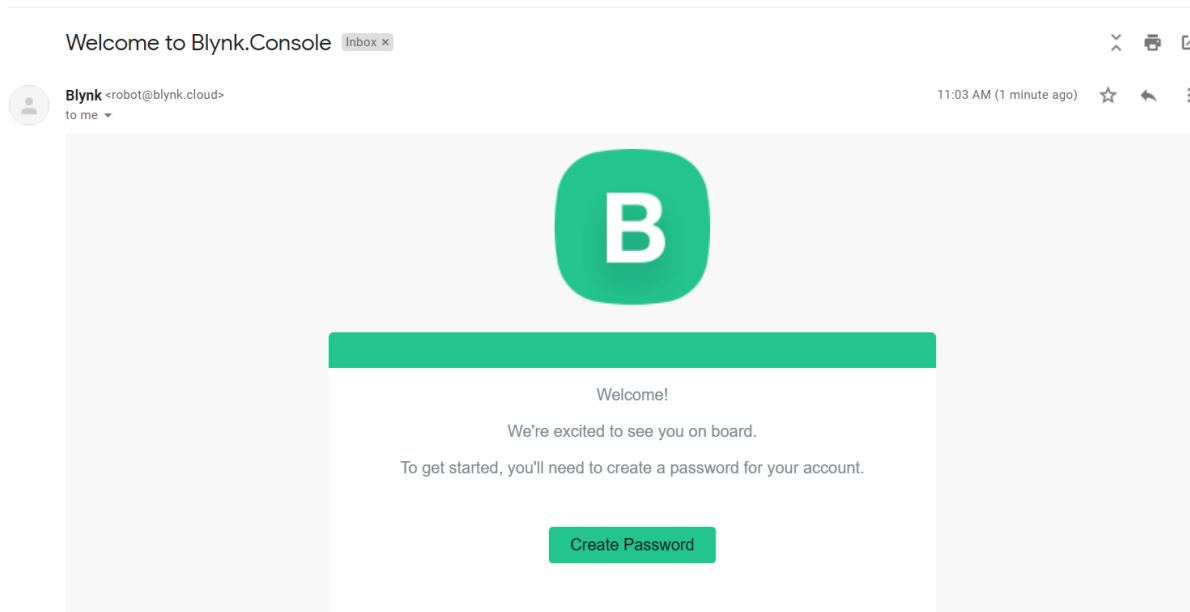
Langkah penggunaan Blynk yaitu :



1. Download dan install aplikasi melalui "PlayStore"
2. Buka aplikasi, dan silahkan sign up new account atau login menggunakan "Facebook"
3. Buat new project, dan pilihlah salah satu module yang akan Anda gunakan maupun aksesoris module yang berfungsi sebagai sarana terhubung ke Internet.
4. Klik Create untuk mengirimkan Token Auth melalui email
5. Atur layout tampilan dashboard IoT dengan cara drag and drop melalui menu Widget Box.
6. Cek inbox email Anda dan temukan Auth Token yang dimana ini akan digunakan sketch/kode program pada Arduino IDE yang akan diuploadkan ke perangkat NodeMCU.

BLYNK IOT (CONSOLE)

Untuk penggunaan Blynk Console (IoT) melalui browser silakan ikuti langkah verifikasi setelah pembuatan akun Blynk berikut ini :



Quickstart

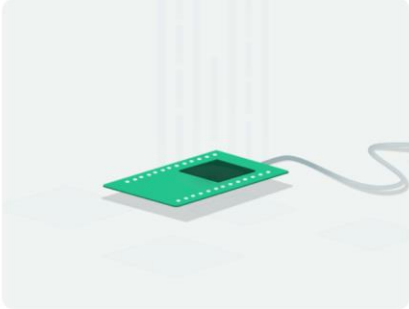
This is a step by step guide to get your first device online and start controlling it from anywhere in the world in **less than 5 minutes**

What you will need:

- Supported hardware. Check the full list of supported hardware [here](#).
- IDE. You can use Arduino IDE or PlatformIO or any other editor.
- Blynk Library
- It will be beneficial if you already know how to upload code to your hardware.

What we will do together:

- Create your first device in Blynk.Cloud
- Prepare firmware code and upload it to your device
- See your device online and control it from Blynk.Console and Blynk.Apps.



Cancel **Let's go!**

Quickstart

1 Hardware — 2 IDE — 3 Blynk Library — 4 Code — 5 Device activation

Which hardware are you using?

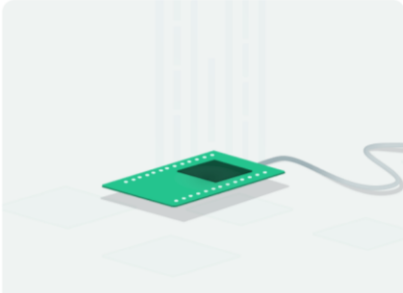
We will help you prepare the code for you board

Choose hardware

What is your device connectivity type

Blynk supports various connection types (BLE is not supported yet).

Choose connectivity



Cancel Next →

Quickstart

1 Hardware — 2 IDE — 3 Blynk Library — 4 Code — 5 Device activation

Which hardware are you using?

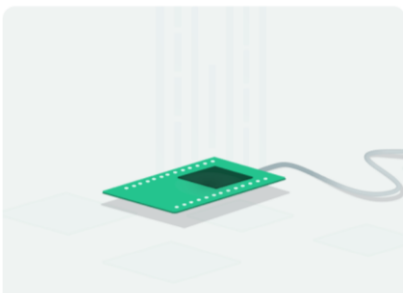
We will help you prepare the code for you board

ESP8266

What is your device connectivity type

Blynk supports various connection types (BLE is not supported yet).

WiFi



Cancel **Next →**

Quickstart


Progress: 1 Hardware (checked) — 2 IDE (active) — 3 Blynk Library — 4 Code — 5 Device activation

Which IDE do you use?

Arduino
Download →

PlatformIO
Download →

Other



← Back Cancel **Next →**

Quickstart

Progress: 1 Hardware (checked) — 2 IDE (checked) — 3 Blynk Library (checked) — 4 Code (active) — 5 Device activation

Here is a code for your device

1. Enter your Wi-Fi network SSID (name) and password to connect your device.
* We never store or send this information anywhere. It's only used to generate the firmware code. You can leave these fields empty and manually add WiFi credentials in your sketch.
2. Copy code from the right panel (or download it as a file).
3. Create a new sketch in your IDE and paste the code.
4. In IDE check that you are using correct board and port settings.

```
..... Copy code Download As File
This is a simple demo of sending and receiving some data.
Be sure to check out other examples!
.....
// Template ID, Device Name and Auth Token are provided by the Blynk Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TNP1LN018ah"
#define BLYNK_DEVICE_NAME "Quickstart Device"
#define BLYNK_AUTH_TOKEN "wXqMgaJFP1weXD-wCPgerK__fkQjYh"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial
```

Cancel **Next →**

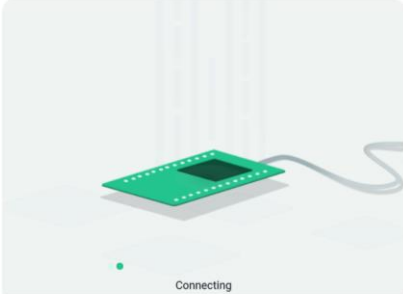
Quickstart

Progress: 1 Hardware (checked) — 2 IDE (checked) — 3 Blynk Library (checked) — 4 Code (checked) — 5 Device activation (active)

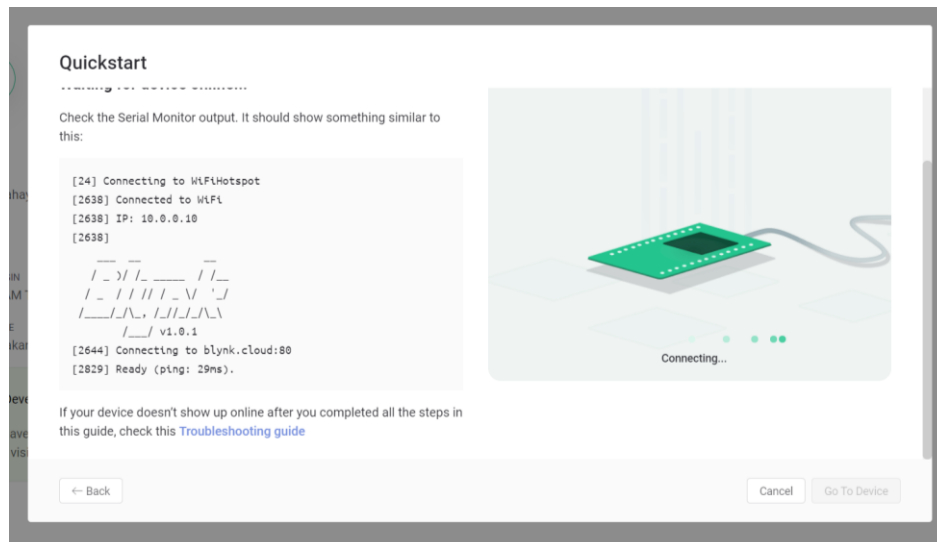
Waiting for device online...

Check the Serial Monitor output. It should show something similar to this:

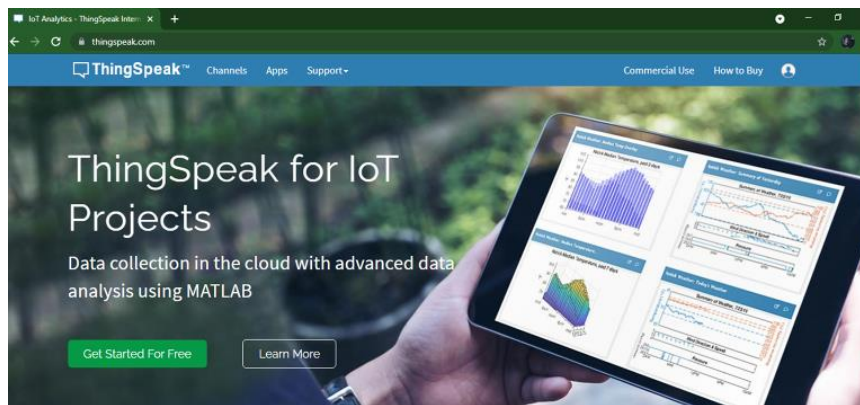
```
[24] Connecting to WiFiHotspot
[2638] Connected to WiFi
[2638] IP: 10.0.0.10
[2638]
  _ _ _ _ _
 / _ \ / _ \ / _ \
/_ _/_/_/_/_/_/_/_/_\
 / _ \ / _ \ / _ \ / _ \
/_ _/_/_/_/_/_/_/_/_\
  / _ \ v1.0.1
[2644] Connecting to blynk.cloud:80
[2644] blynk.cloud:80
```



← Back Cancel **Go To Device**



2.3 Aplikasi Thingspeak



Thingspeak.com merupakan platform Internet of Things dibagian cloud dimana kita dapat mengirim atau menerima suatu data dengan protokol komunikasi HTTP dan juga dapat menampilkan nilai data melalui dashboard gratis yang diberikan.

Thingspeak.com berfungsi sebagai pengumpul data yang berasal dari perangkat node berupa sensor-sensor yang sudah terhubung ke internet dan juga memungkinkan pengambilan data dari perangkat lunak untuk keperluan visualisasi, notifikasi, kontrol dan analisis historis data.

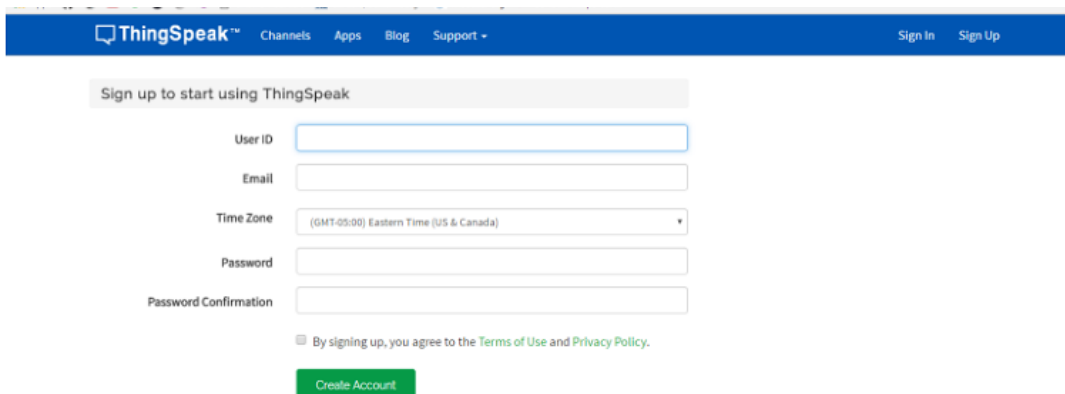
Unsur utama dari ThingSpeak adalah saluran, yang berisi bidang data, bidang lokasi, dan bidang status. Setelah membuat saluran ThingSpeak, kita bisa menulis data ke saluran proses dan melihat hasil data lewat MATLAB. Dari situ dapat dilihat reaksi terhadap data dengan *tweet* dan alert lainnya.

Fitur-fitur Dari Thingspeak.com

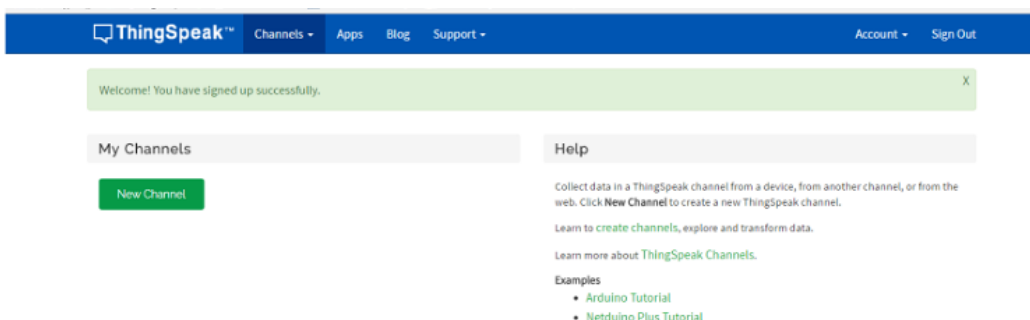
- | | |
|------------------------------|------------------------------|
| a. Open Api | e. Data Visualizations |
| b. Real-time data collection | f. Device status messages |
| c. Geolocation data | g. Plugins |
| d. Data processing | h. Penggunaan Thingspeak.com |

Langkah menggunakan Thingspeak yaitu:

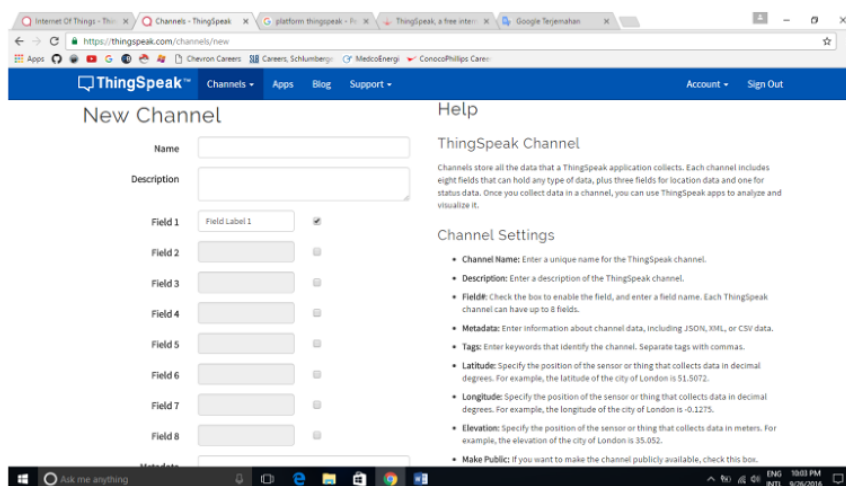
1. Buat akun Thingspeak.com dengan mengisikan User ID, Email dan juga Password. Pastikan email yang digunakan aktif karena digunakan untuk konfirmasi akun dan pemberitahuan lebih lanjut. Pada tahap awal ini juga perlu dibuat akun MathWorks agar terafiliasi dengan akun Thingspeak.



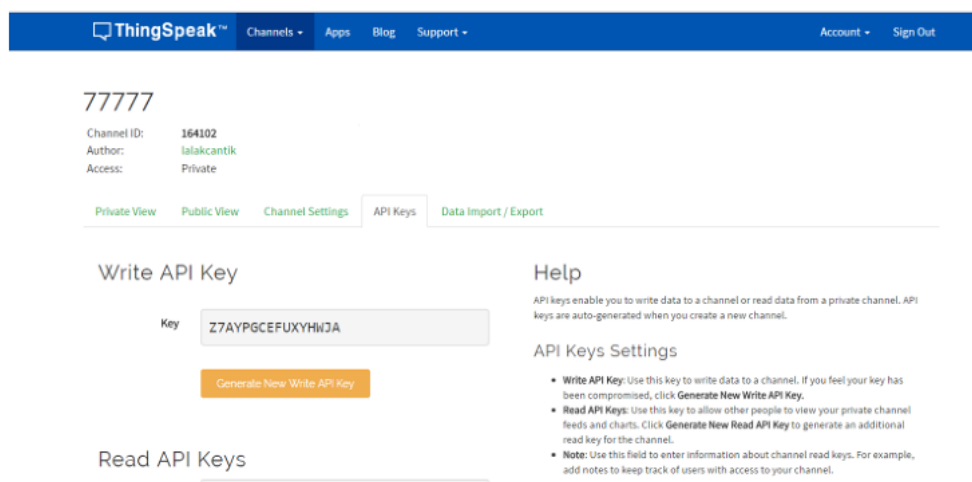
2. Buat Channel baru dengan cara menekan tombol "New Channel" yang akan digunakan untuk menampilkan data yang dikirim dari perangkat-perangkat sensor yang terhubung ke internet.



3. Berikutnya mengatur channel. Nama saluran bebas dalam pelabelannya dan ini sekaligus menjadi nama data tersimpan atau menjadi proyek kerja. Silakan kosongkan beberapa field jika memang tidak digunakan dalam penyimpanan datanya.



4. Terakhir gunakan API Key yaitu API Write yang berguna untuk keperluan mengirim data ke thingspeak.com dan juga API Read yang digunakan untuk mengambil data dari thingspeak.com untuk keperluan analisis data maupun kontrol.



The screenshot shows the Thingspeak website interface for channel 77777. The top navigation bar includes the Thingspeak logo, 'Channels', 'Apps', 'Blog', 'Support', 'Account', and 'Sign Out'. The channel information is displayed as follows:

Channel ID:	164102
Author:	lalakantik
Access:	Private

Navigation tabs include 'Private View', 'Public View', 'Channel Settings', 'API Keys', and 'Data Import / Export'. The 'API Keys' tab is active, showing a 'Write API Key' section with a text input field containing the key 'Z7AYPGCEFUXYHNJA' and a 'Generate New Write API Key' button. Below this is a 'Read API Keys' section. To the right, a 'Help' section explains that API keys enable writing or reading data from a channel and are auto-generated. Below the help is an 'API Keys Settings' section with a bulleted list:

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

3. PRAKTIKUM

3.1 Praktikum 1 : LED berkedip

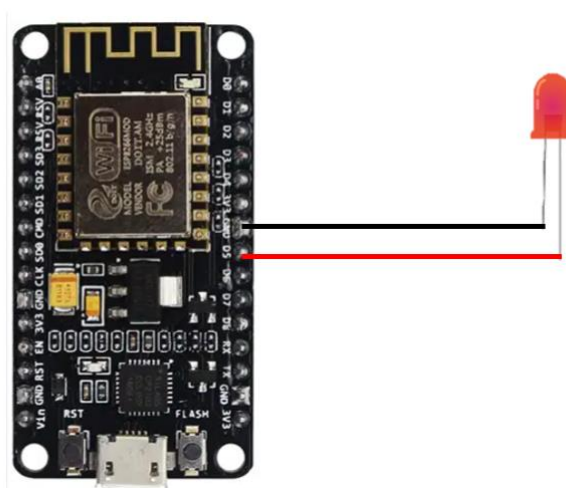
a. Prosedur

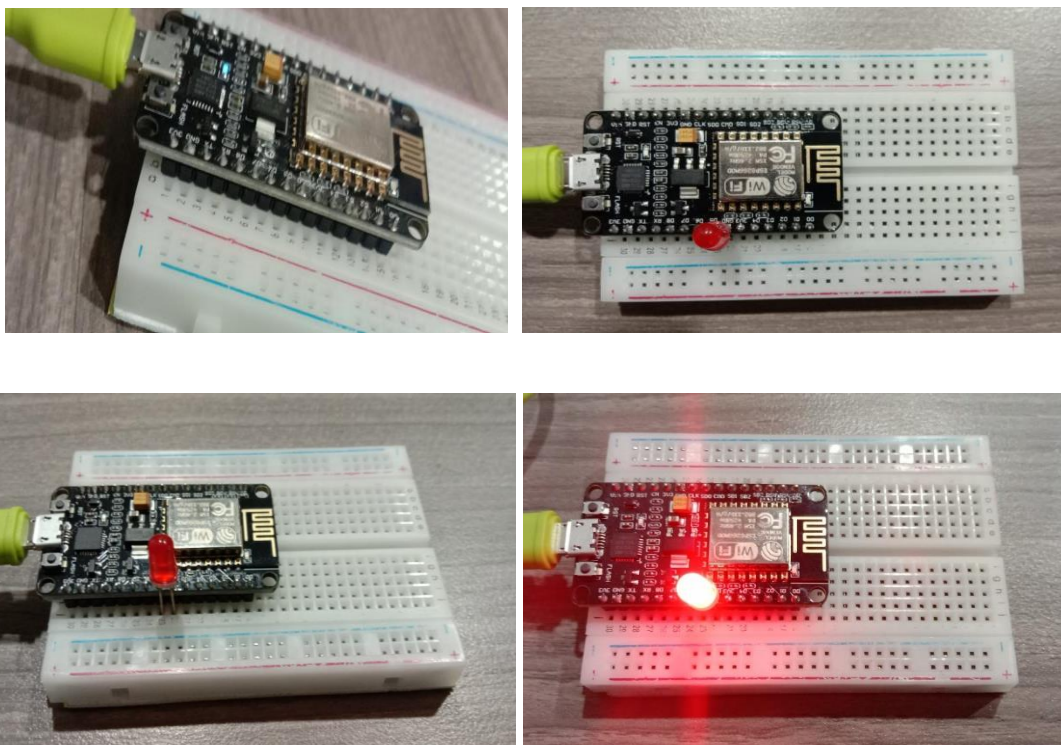
1. Pasang perangkat sesuai dengan skematik rangkaian.
2. Pasang kabel mikroUSB antara NodeMCU dan Laptop/PC.
3. Ketik sketch/kode program pada Arduino IDE.
4. Upload sketch tersebut ke NodeMCU.
5. Setelah proses upload selesai maka LED akan menyala berkedip.

b. Aplikasi dan Komponen

No	Komponen	Jumlah
1	Arduino IDE	1
2	LED	1
3	NodeMCU	1
4	Kabel USB	1
5	Breadboard	1
6	Resistor 220 ohm (optional)	1

c. Skema rangkaian





d. Sketch/Kode program

```
#define LED D5

// Led in NodeMCU at pin GPIO14 (D5).
void setup() {
  pinMode(LED, OUTPUT);

  // LED pin as output.
}

void loop() {
  digitalWrite(LED, HIGH); // turn the LED off.(Note that LOW is the voltage
                           // level but actually

//the LED is on; this is because it is acive low on the ESP8266.
```




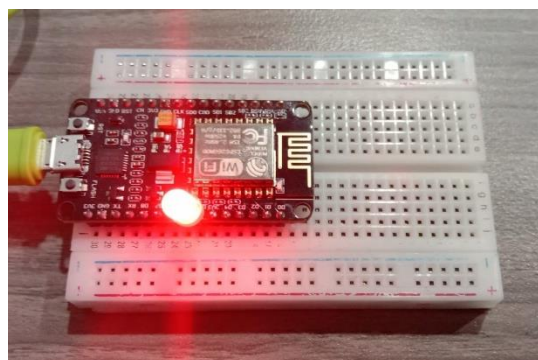
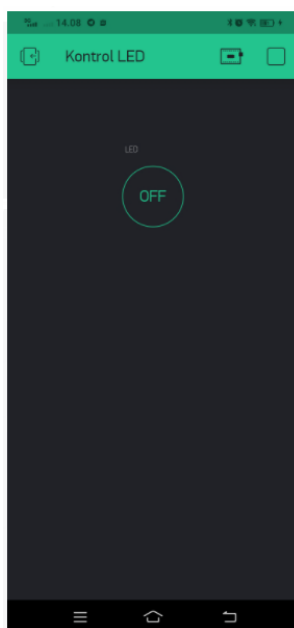
```
1_CloTP_LED_NodeMCU | Arduino 1.8.5
File Edit Sketch Tools Help
1_CloTP_LED_NodeMCU
#define LED D5 // Led in NodeMCU at pin GPIO14 (D5).
void setup() {
  pinMode(LED, OUTPUT); // LED pin as output.
}
void loop() {
  digitalWrite(LED, HIGH); // turn the LED off. (Note that LOW is the voltage level but actually
                           // the LED is on; this is because it is active low on the ESP8266.
  delay(1000); // wait for 1 second.
  digitalWrite(LED, LOW); // turn the LED on.
  delay(1000); // wait for 1 second.
}
```

(new can return nullptr), All SSL ciphers (most compatible), 4MB (FS:2MB OTA~1019KB), 2 v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM6

3.2 Praktikum 2 : Kontrol LED menggunakan Blynk

a. Prosedur

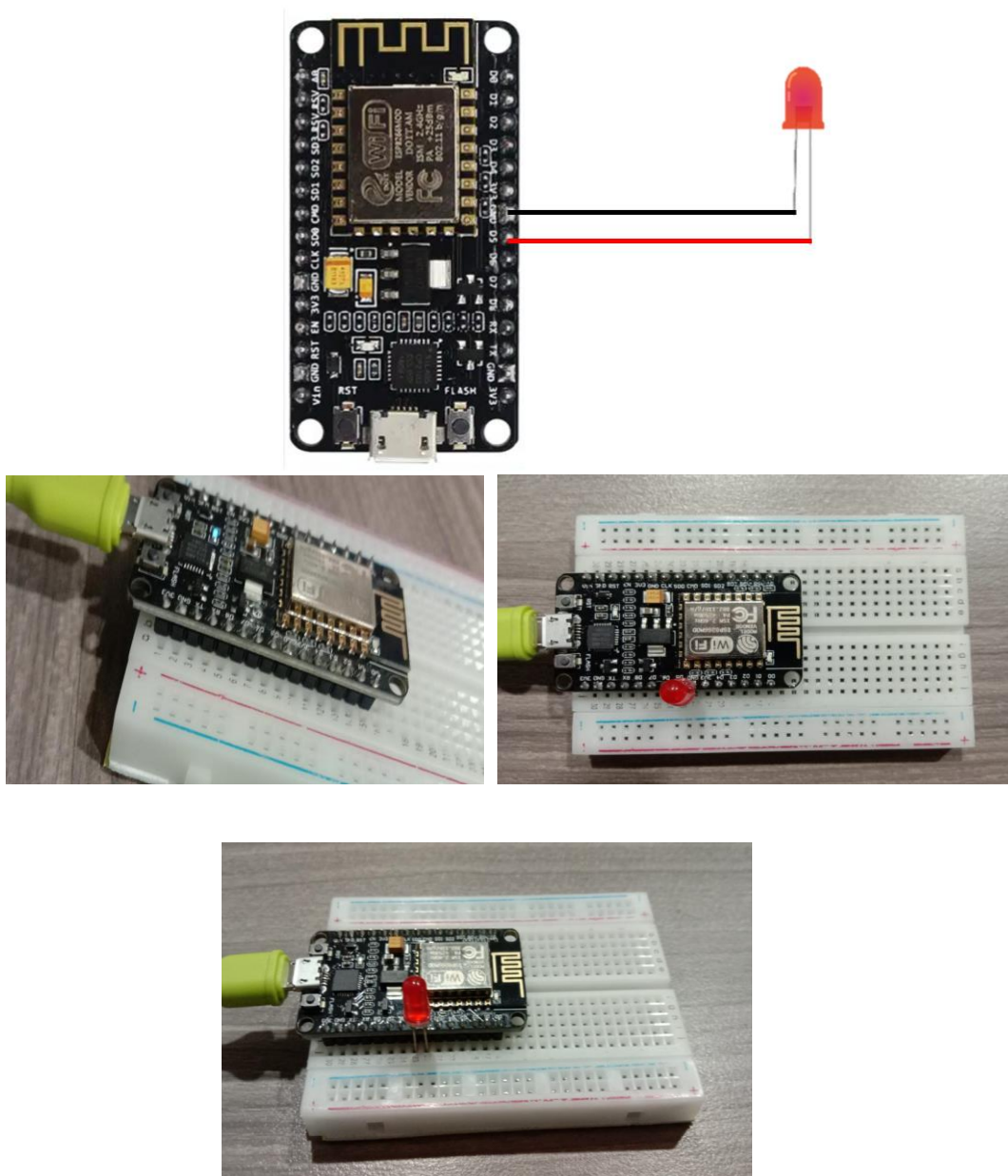
1. Pasang perangkat sesuai dengan skematik rangkaian.
2. Lakukan pengaturan project pada aplikasi Blynk.
3. Pasang kabel mikroUSB antara NodeMCU dan Laptop/PC.
4. Ketik sketch/kode program pada Arduino IDE.
5. Upload sketch tersebut ke NodeMCU.
6. Jalankan project Blynk dengan cara klik .
7. Kontrol lampu LED pada perangkat dengan cara klik button pada dashboard Blynk.



b. Komponen

No	Komponen	Jumlah
1	Arduino IDE	1
2	Aplikasi Blynk	1
3	LED	1
4	NodeMCU	1
5	Kabel USB	1
6	Breadboard	1
7	Resistor 220 ohm (optional)	1

c. Skema rangkaian



d. Sketch/Kode program

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

```
// You should get Auth Token in the Blynk App.
```

```
// Go to the Project Settings (nut icon).
```

```
char auth[] = "isi_kode_authentifikasi";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "wifi ssid";
char pass[] = "wifi password";

BlynkTimer timer;

// This function sends Arduino's up time every second to Virtual Pin
(5).
// In the app, Widget's reading frequency should be set to PUSH. This
means
// that you define how often to send data to Blynk App.
void myTimerEvent()
{
    // You can send any value at any time.
    // Please don't send more that 10 values per second.
    Blynk.virtualWrite(V5, millis() / 1000);
}
void setup()
{
    // Debug console
    Serial.begin(9600);
    Blynk.begin(auth, ssid, pass);
    // You can also specify server:
    //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
    //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
    // Setup a function to be called every second
    timer.setInterval(1000L, myTimerEvent);
}
void loop()
{
```

```

Blynk.run();

timer.run(); // Initiates BlynkTimer

}

```



```

CloTP_LED_NodeMCU_Blynk- | Arduino 1.8.5
File Edit Sketch Tools Help
CloTP_LED_NodeMCU_Blynk-
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "isi_kode_authentifikasi";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "wifi ssid";
char pass[] = "wifi password";

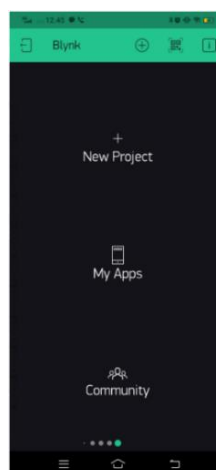
BlynkTimer timer;

// This function sends Arduino's up time every second to Virtual Pin (5).
// In the app, Widget's reading frequency should be set to PUSH. This means
// that you define how often to send data to Blynk App.
void myTimerEvent()
{
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V5, millis() / 1000);
}
void setup()
{
  // Debug console
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
  // Setup a function to be called every second
  timer.setInterval(1000L, myTimerEvent);
}
Done Saving
leaving...
hard resetting via RST pin...
2# SSL ciphers (most compatible), 4MB (FS:2MB OTA~1019KB), 2, V2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM6

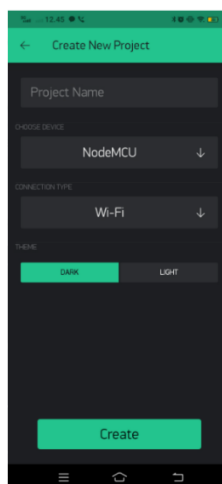
```

e. Pengaturan Project Blynk

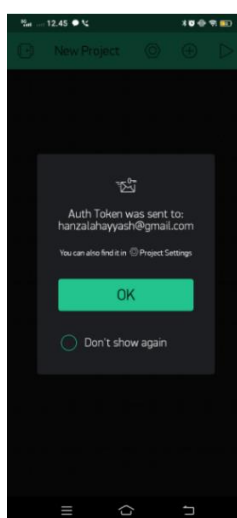
1. Buka aplikasi Blynk. Klik New Project.




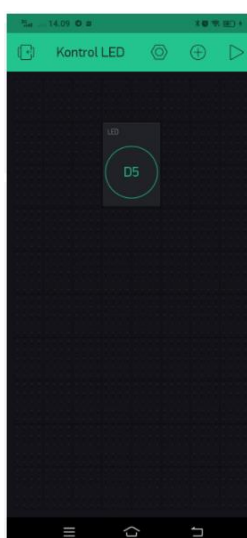
2. Beri nama project pada Project Name. Pilih NodeMCU sebagai device dan WIFI sebagai connection type. Klik Create.



3. Klik OK kemudian periksa email yang dikirimkan oleh Blynk untuk menyalin kode autentifikasi ke dalam Sketch.

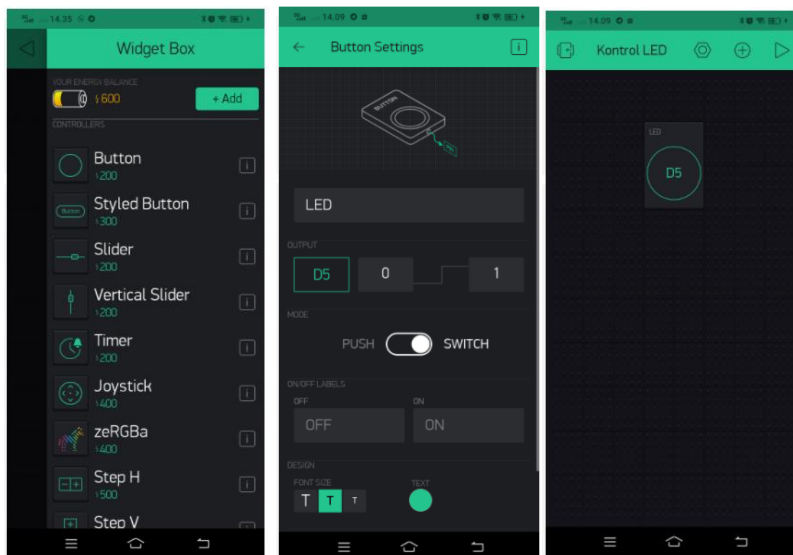


4. Klik ikon  untuk membuat tampilan dashboard Blynk.



5. Pilih 1 buah widget pada Widget Box yaitu Button. Lakukan pengaturan button sebagai berikut :

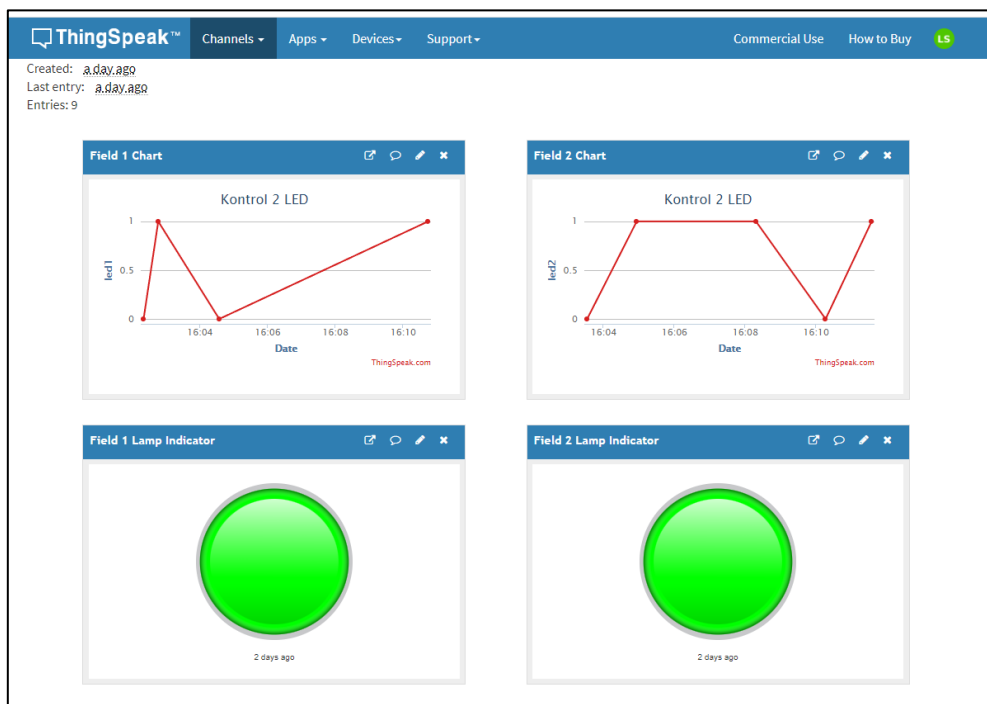
- Name
- : LED
- PIN
- : Digital D5
- Style
- : Switch



3.3 Praktikum 3 : Kontrol LED menggunakan Thingspeak

a. Prosedur

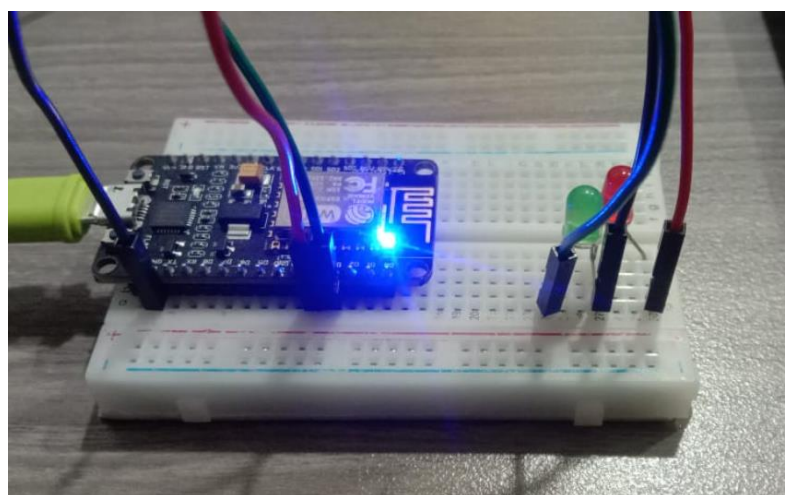
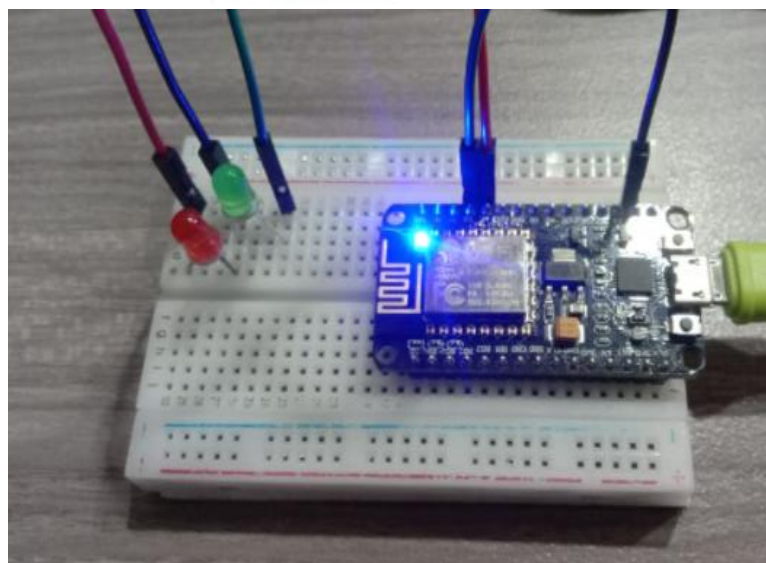
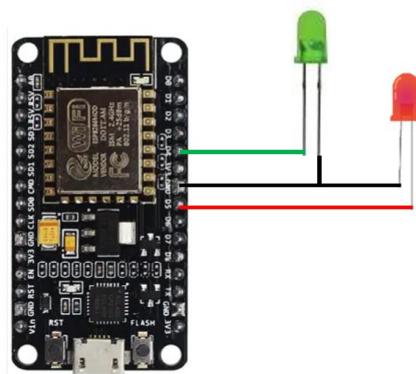
1. Pasang perangkat sesuai dengan skematik rangkaian.
2. Pasang kabel mikroUSB antara NodeMCU dan Laptop/PC.
3. Ketik sketch/kode program pada Arduino IDE.
4. Upload sketch tersebut ke NodeMCU.
5. Lakukan pembuatan akun dan channel Thingspeak
6. Lakukan pembuatan dan hosting website
7. Jalankan project dengan cara menekan tombol ON atau OFF pada website yang sudah dibuat.
8. Perhatikan respon output LED pada perangkat dan dashboard Thingspeak.



b. Komponen

No	Komponen	Jumlah
1	Arduino IDE	1
2	Platform Thingspeak	1
3	LED	2
4	NodeMCU	1
5	Kabel USB	1
6	Breadboard	1
7	Resistor 220 ohm (optional)	1

c. Skema rangkaian



d. Sketch/Kode program

```
#include <ThingSpeak.h> //library platform thingspeak
#include <ESP8266WiFi.h> //library platform thingspeak
```

```
const char* ssid    = "isi nama ssid wifi";
//setting nama wifi router

const char* password = "isi password wifi";

//password wifi router

const char * myCounterWriteAPIKey = "isi Write APIKey";

// Write API Key

unsigned long channel = isi channel id;

//ID channel project pada Thingspeak

//1,2 merupakan channel yang kita gunakan pada channel Thingspeak

unsigned int led1 = 1;
unsigned int led2 = 2;

WiFiClient  client;

void setup() {
    Serial.begin(9600);

    //memulai komunikasi serial dengan baud rate 115200

    delay(100);

    //waktu tunda 0.1 detik

    pinMode(D4, OUTPUT);

    //inisialisasi pin D4 sebagai output
    pinMode(D5, OUTPUT);

    //inisialisasi pin D5 sebagai output
    digitalWrite(D4, 0);

    //memberikan output LOW atau tegangan 0 volt ke pin D4
    digitalWrite(D5, 0);

    //memberikan output LOW atau tegangan 0 volt ke pin D5
    // We start by connecting to a WiFi network

    Serial.println();
    Serial.println();
```

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

WiFi.begin(ssid, password);

//menghubungkan dengan wifi router

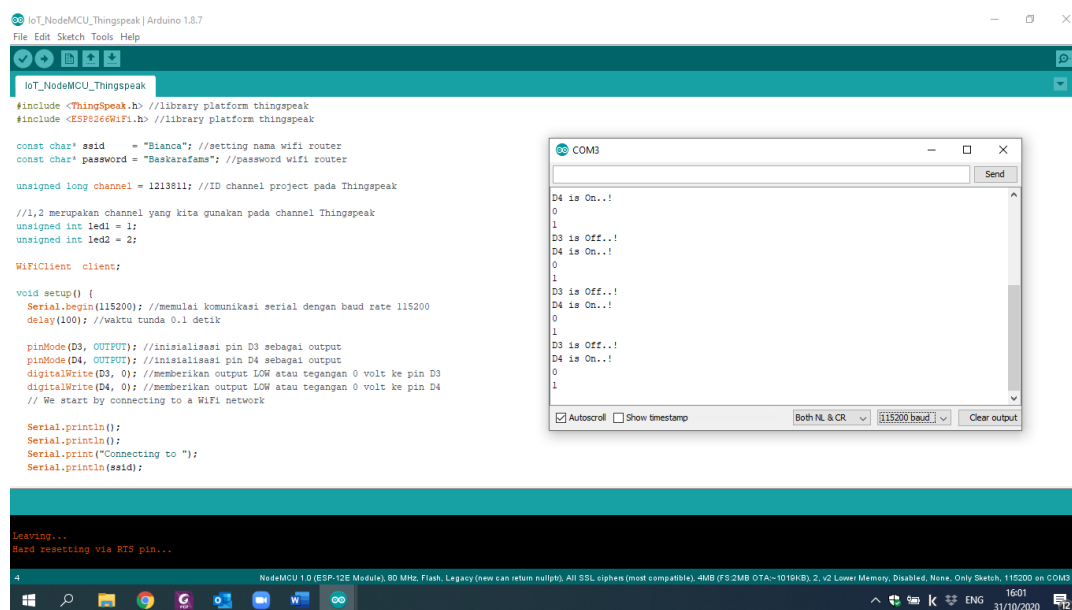
while (WiFi.status() != WL_CONNECTED) {
//Cek koneksi Wifi.
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
Serial.print("Netmask: ");
Serial.println(WiFi.subnetMask());
Serial.print("Gateway: ");
Serial.println(WiFi.gatewayIP());
ThingSpeak.begin(client);
}

void loop(){
    //update data terakhir dari field Thingspeak
    int led_1 = ThingSpeak.readFloatField(channel, led1,
myCounterWriteAPIKey);

    int led_2 = ThingSpeak.readFloatField(channel, led2,
myCounterWriteAPIKey);

    if(led_1 == 1){
//Jika data yang dibaca dari Field1 bernilai 1, maka LED 1 menyala
        digitalWrite(D4, 1);
        Serial.println("D4 is On..!");
    }
}
```

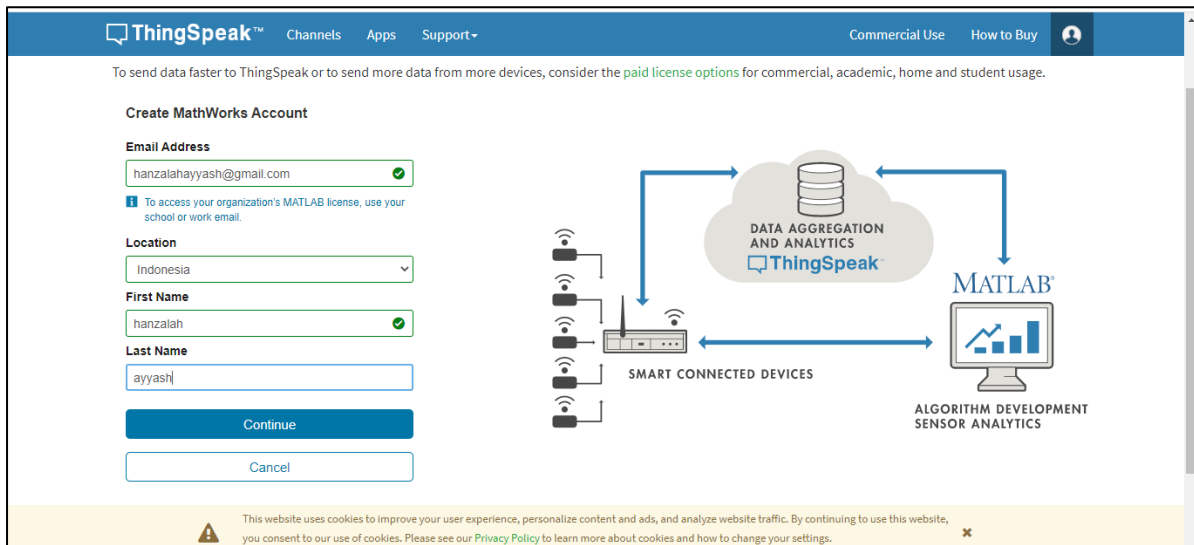
```
if(led_1 == 0){  
  
//Jika data yang dibaca dari Field1 bernilai 0, maka LED 1 padam  
  
    digitalWrite(D4, 0);  
  
    Serial.println("D4 is Off..!");  
  
}  
  
if(led_2 == 1){  
  
//Jika data yang dibaca dari Field1 bernilai 1, maka LED 2 menyala  
  
    digitalWrite(D5, 1);  
  
    Serial.println("D5 is On..!");  
  
}  
  
if(led_2 == 0){  
  
//Jika data yang dibaca dari Field1 bernilai 0, maka LED 2 padam  
  
    digitalWrite(D5, 0);  
  
    Serial.println("D5 is Off..!");  
  
}  
  
Serial.println(led_1);  
  
Serial.println(led_2);  
  
delay(500);  
  
}
```



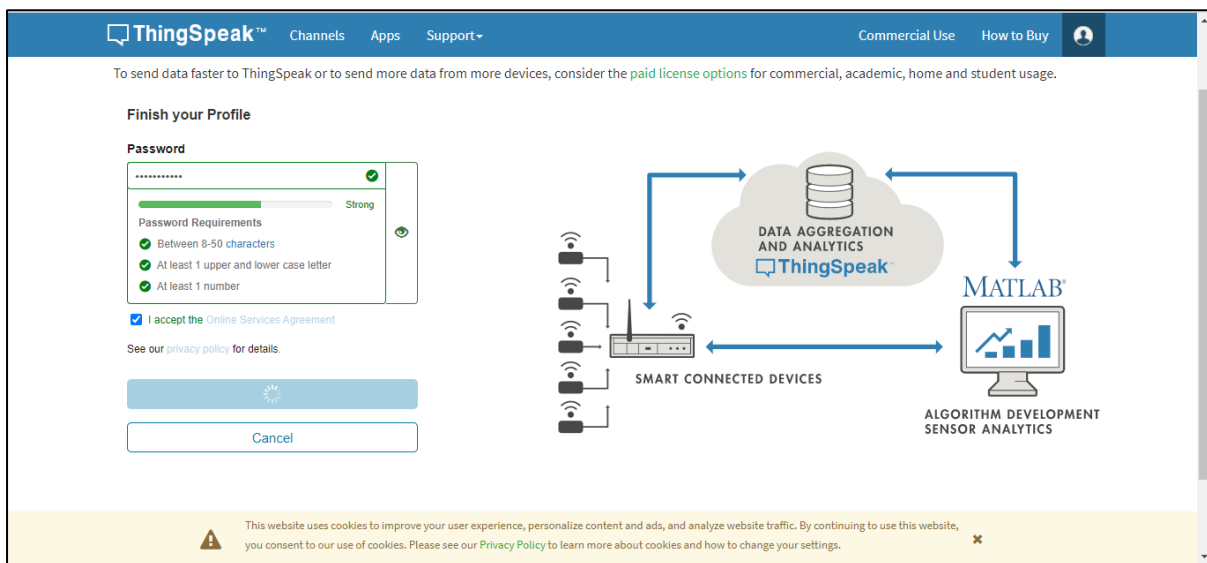
```
IoT_NodeMCU_Thingspeak | Arduino 1.8.7  
File Edit Sketch Tools Help  
  
IoT_NodeMCU_Thingspeak  
#include <ThingSpeak.h> //library platform thingspeak  
#include <ESP8266WiFi.h> //library platform thingspeak  
  
const char* ssid = "Blanca"; //setting nama wifi router  
const char* password = "Baskarafans"; //password wifi router  
  
unsigned long channel = 1213811; //ID channel project pada Thingspeak  
  
//1,2 merupakan channel yang kita gunakan pada channel Thingspeak  
unsigned int led1 = 1;  
unsigned int led2 = 2;  
  
WiFiClient client;  
  
void setup() {  
    Serial.begin(115200); //memulai komunikasi serial dengan baud rate 115200  
    delay(100); //waktu tunda 0.1 detik  
  
    pinMode(D3, OUTPUT); //inisialisasi pin D3 sebagai output  
    pinMode(D4, OUTPUT); //inisialisasi pin D4 sebagai output  
    digitalWrite(D3, 0); //memberikan output LOW atau tegangan 0 volt ke pin D3  
    digitalWrite(D4, 0); //memberikan output LOW atau tegangan 0 volt ke pin D4  
    // We start by connecting to a WiFi network  
  
    Serial.println();  
    Serial.println();  
    Serial.print("Connecting to ");  
    Serial.println(ssid);  
  
    Leaving...  
    Hard resetting via RTS pin...  
  
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Legacy (new can return nullptr), All SSL ciphers (most compatible), 4MB (FS 2MB OTA~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3  
4  
1601  
31/10/2020
```

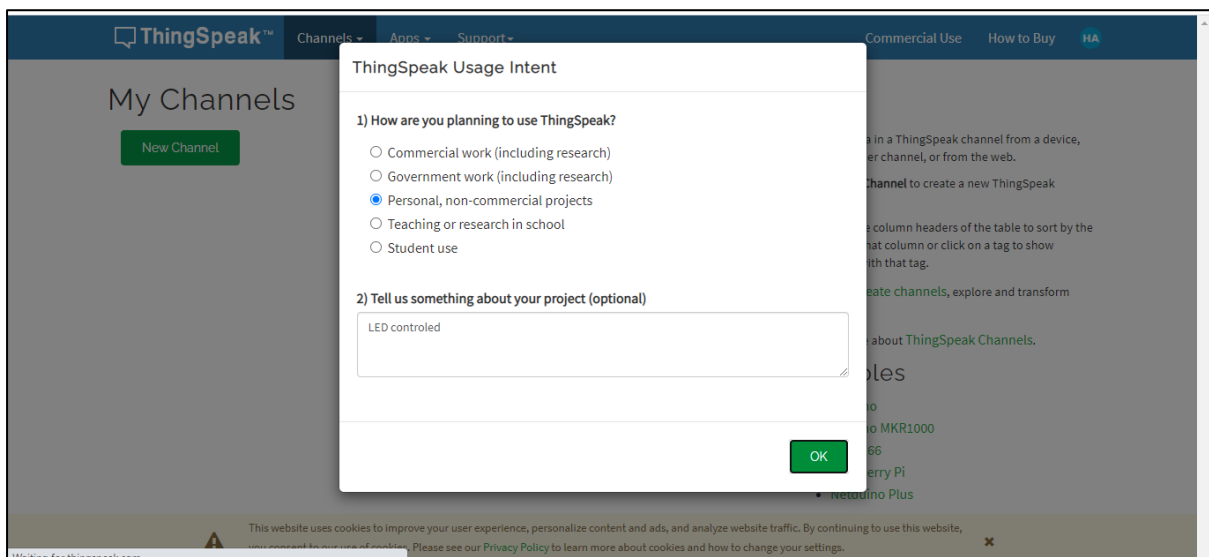
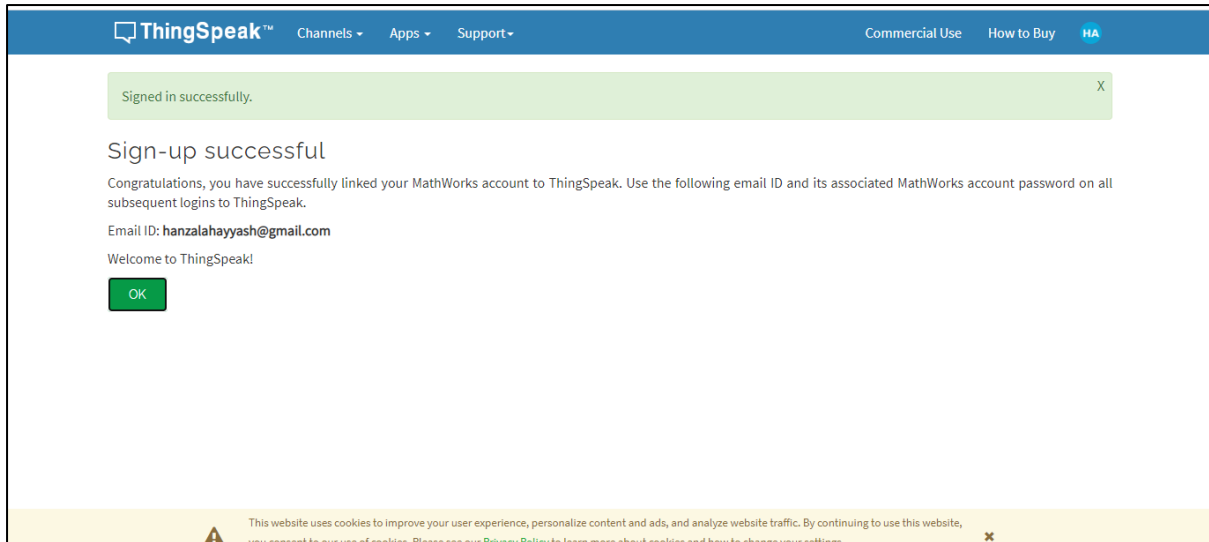
e. Pengaturan channel Thingspeak

1. Buka www.thingspeak.com

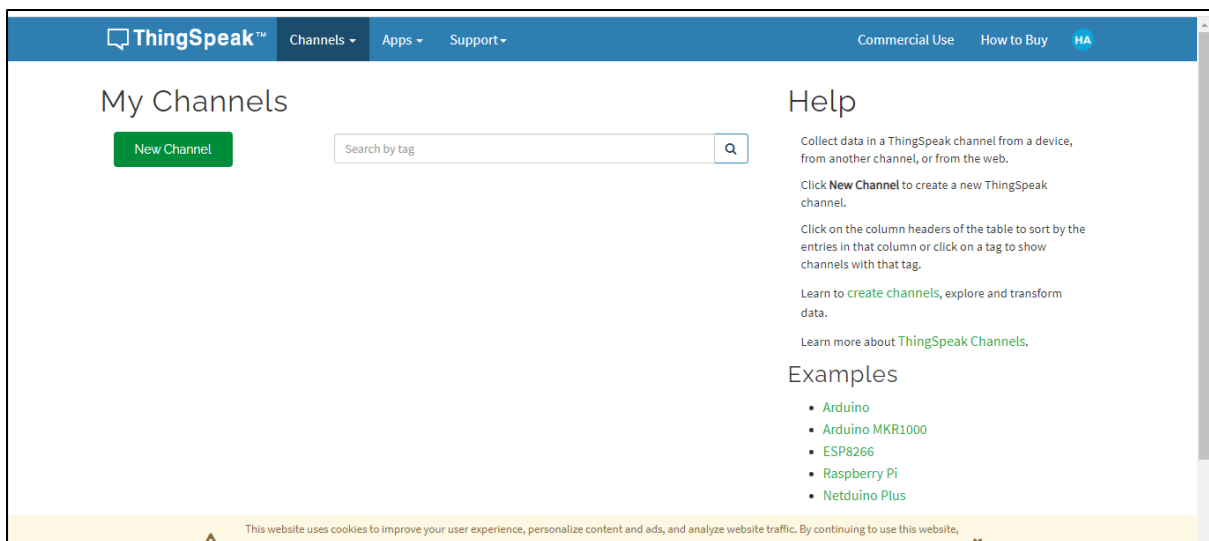


2. Verifikasi di email
3. Pilih MathWorks US website





4. Klik New Channel



ThingSpeak™ Channels Apps Support Commercial Use How to Buy HA

New Channel

Name: Kontrol LED

Description: Tes kontrol lampu LED

Field 1: LED1

Field 2: LED2

Field 3:

Field 4:

Field 5:

Field 6:

Field 7:

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.

This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to use this website, you consent to our use of cookies. Please see our [Privacy Policy](#) to learn more about cookies and how to change your settings.

ThingSpeak™ Channels Apps Support Commercial Use How to Buy HA

Elevation:

Show Channel Location:

Latitude: 0.0

Longitude: 0.0

Show Video:

YouTube

Vimeo

Video URL: http://

Show Status:

Save Channel

You can get data into a channel from a device, website, or another ThingsSpeak channel. You can then visualize data and transform it using ThingSpeak Apps.

See [Get Started with ThingSpeak™](#) for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

[Learn More](#)

This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to use this website, you consent to our use of cookies. Please see our [Privacy Policy](#) to learn more about cookies and how to change your settings.

ThingSpeak™ Channels Apps Support Commercial Use How to Buy HA

Kontrol LED

Channel ID: 1213811 | Tes kontrol lampu LED

Author: mwa000020175542

Access: Private

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Add Visualizations **Add Widgets** **Export recent data** **MATLAB Analysis** **MATLAB Visualization**

Channel Stats

Created: less than a minute ago

Entries: 0

Field 1 Chart Kontrol LED

Field 2 Chart Kontrol LED

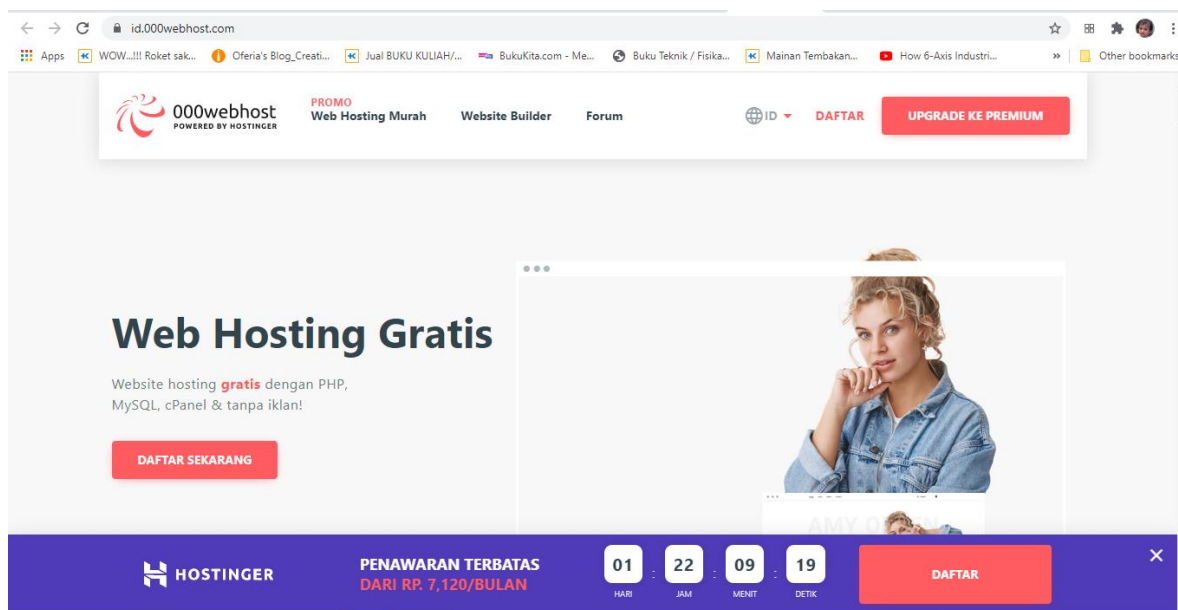
This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to use this website, you consent to our use of cookies. Please see our [Privacy Policy](#) to learn more about cookies and how to change your settings.

The screenshot shows the 'Channel Sharing Settings' page for a ThingSpeak channel named 'Kontrol LED'. The channel ID is 1213811, the author is mwa0000020175542, and the access is currently set to 'Private'. The 'Sharing' tab is selected, showing three options: 'Keep channel view private' (unselected), 'Share channel view with everyone' (selected), and 'Share channel view only with the following users' (unselected). There is an 'Add User' button and an input field for email addresses. A help section explains that sharing data requires an API key. A cookie notice is visible at the bottom.

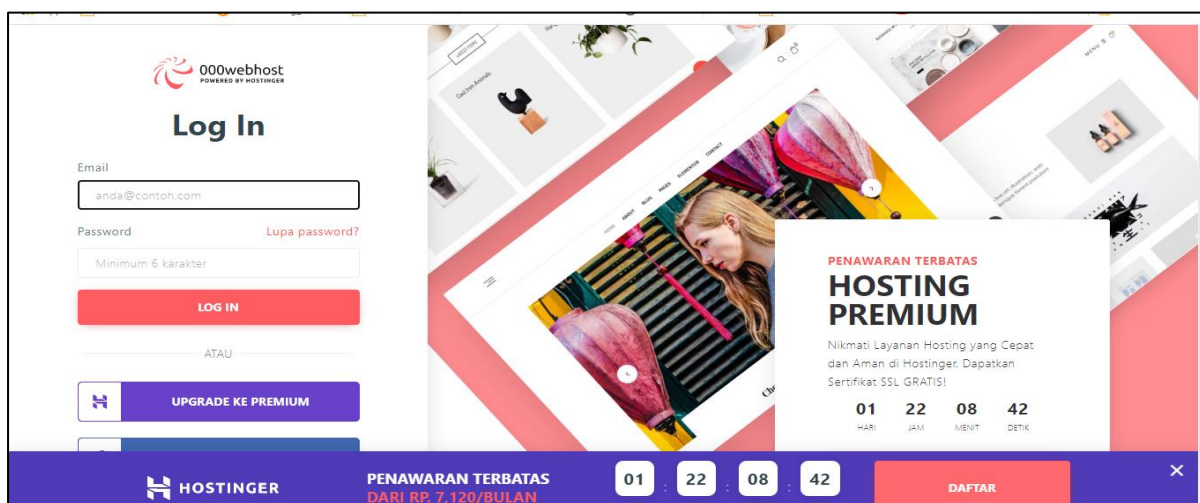
The screenshot shows the 'API Keys' page for the same 'Kontrol LED' channel. The access is now set to 'Public'. The 'API Keys' tab is selected, displaying a 'Write API Key' section with a text input field containing the key 'N4I1KVVAJN7UL0ZZ' and a 'Generate New Write API Key' button. A 'Read API Keys' section is also present. The help section explains that API keys are used for writing and reading data. A cookie notice is visible at the bottom.

f. Pengaturan Website

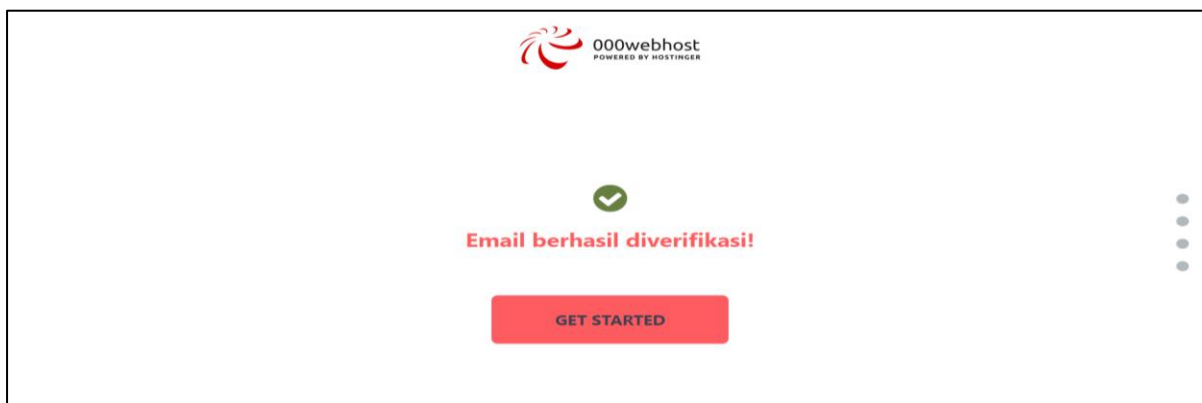
1. Buka **id.000webhost.com** pada address bar browser. Klik **Daftar**.



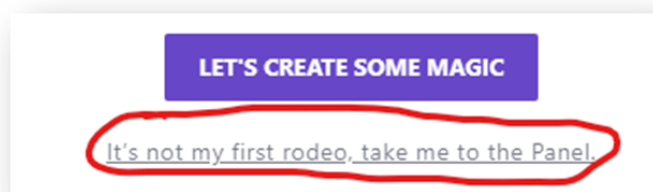
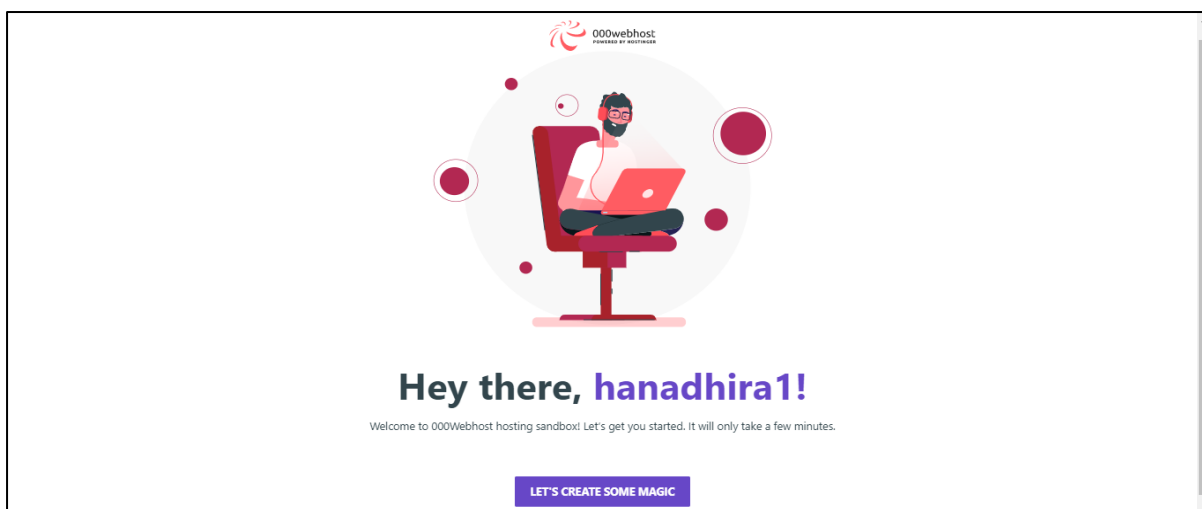
2. Tutup pop-up **penawaran** hosting. Klik **Daftar**. Lakukan pendaftaran.



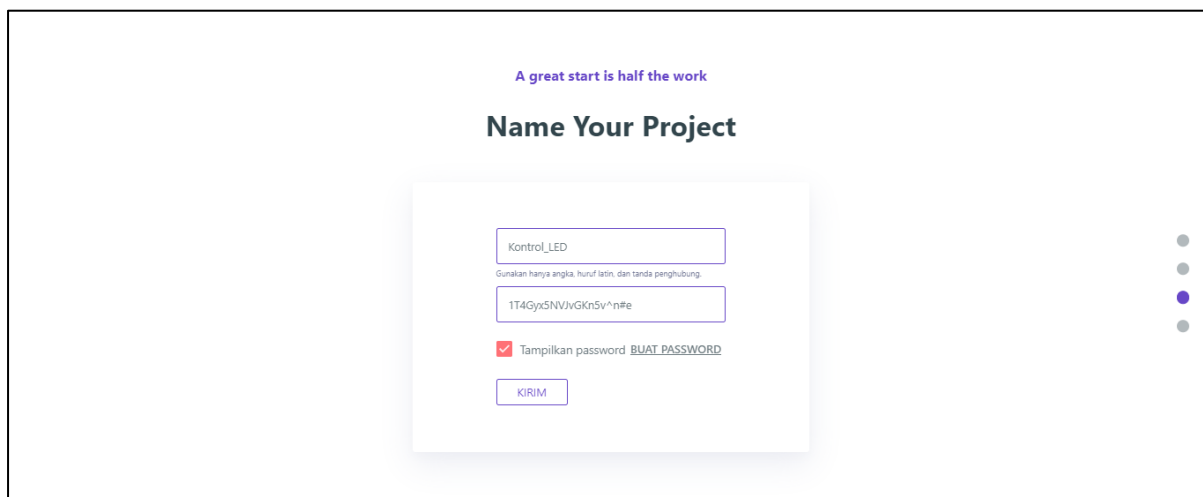
3. Verifikasi email **pada** kotak masuk email Anda.



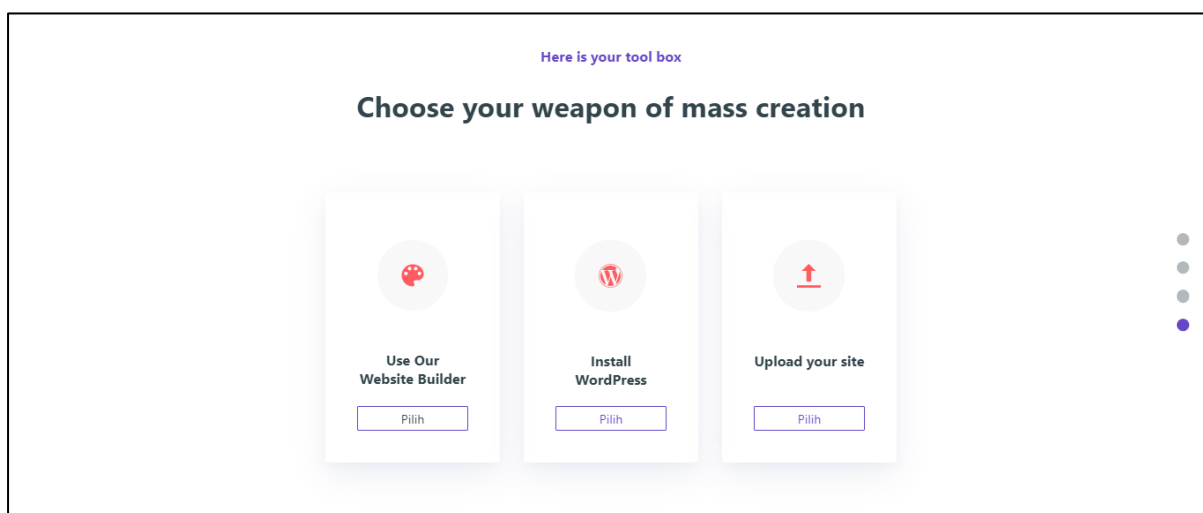
4. Klik "It's not my **first** rodeo, take me to the panel"



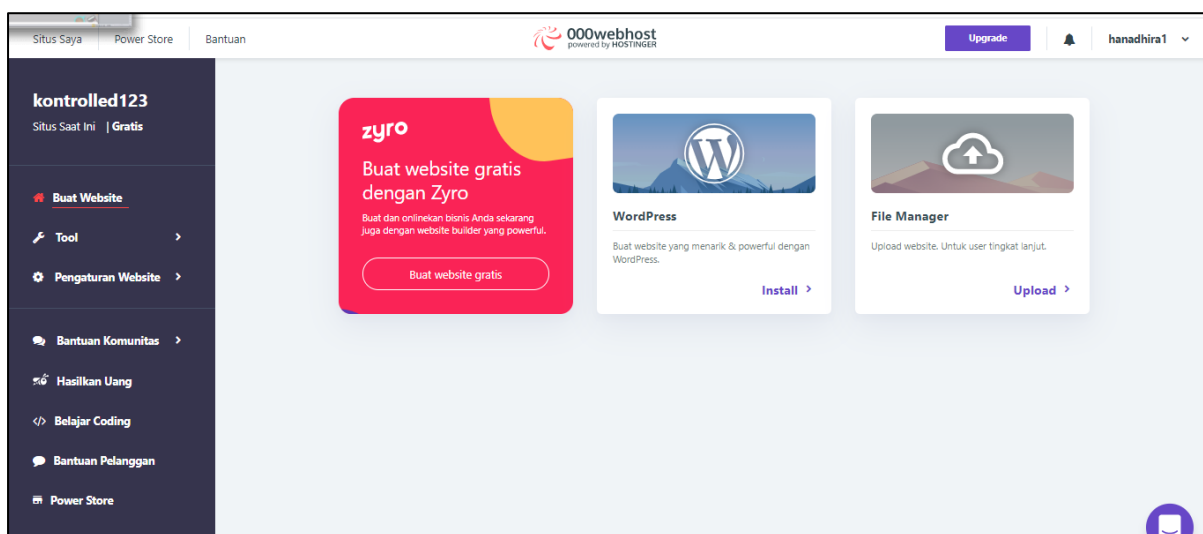
5. Beri nama proyek **website** (opsional) ; dapat langsung klik “Kirim”.



6. Klik “Upload your site”



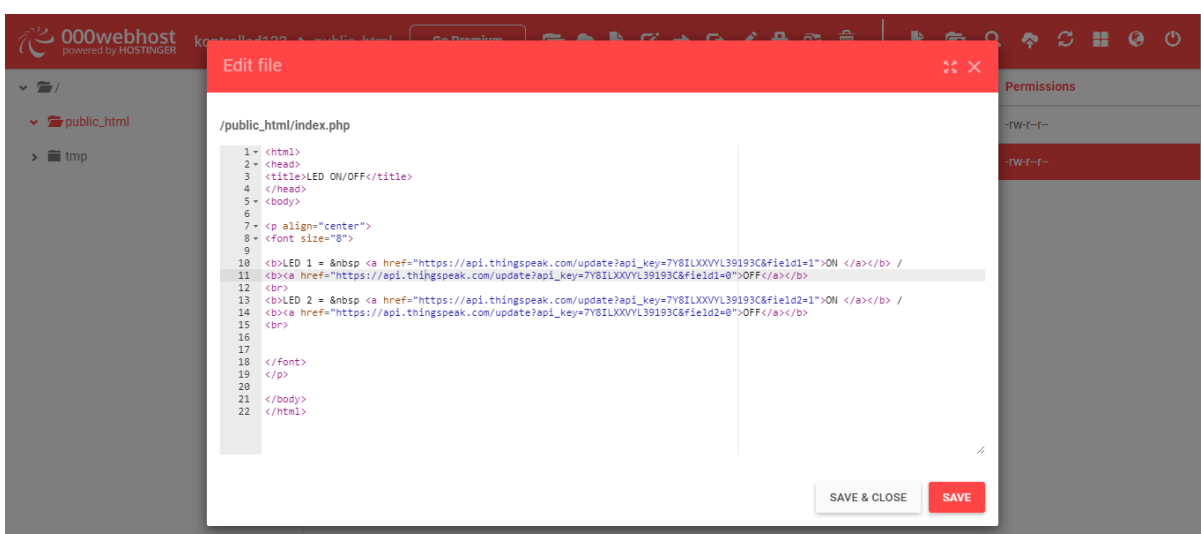
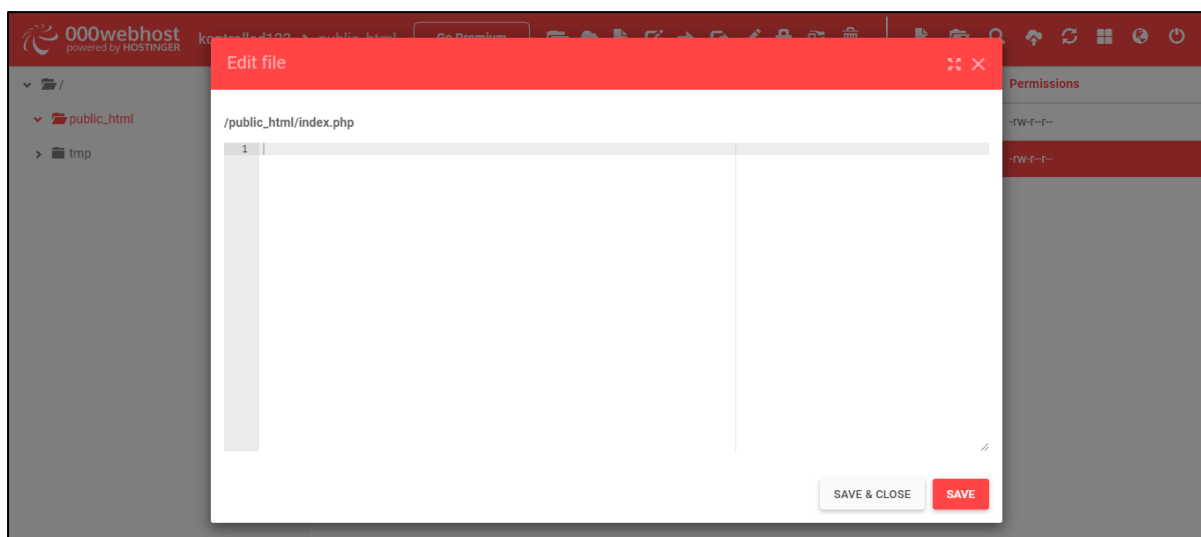
7. Buat Website. Klik “Upload” File Manager.




```

<b><a
href="https://api.thingspeak.com/update?api_key=0Y8IAGFRL39193C&field2=0">O
FF</a></b>
<br>
</font>
</p>
</body>
</html>

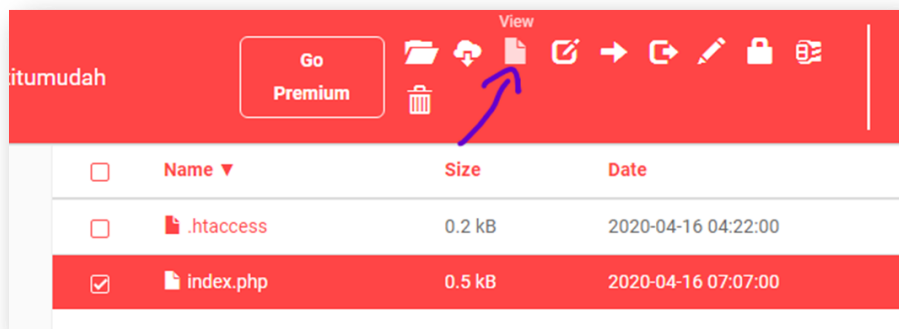
```



Keterangan :

- Ganti WriteAPIKey berlatar warna kuning dengan WriteAPIKey yang Anda miliki (setiap channel berbeda WriteAPIKey).
- Field1 = led1 di Pin D4
- Field2 = led2 di Pin D5

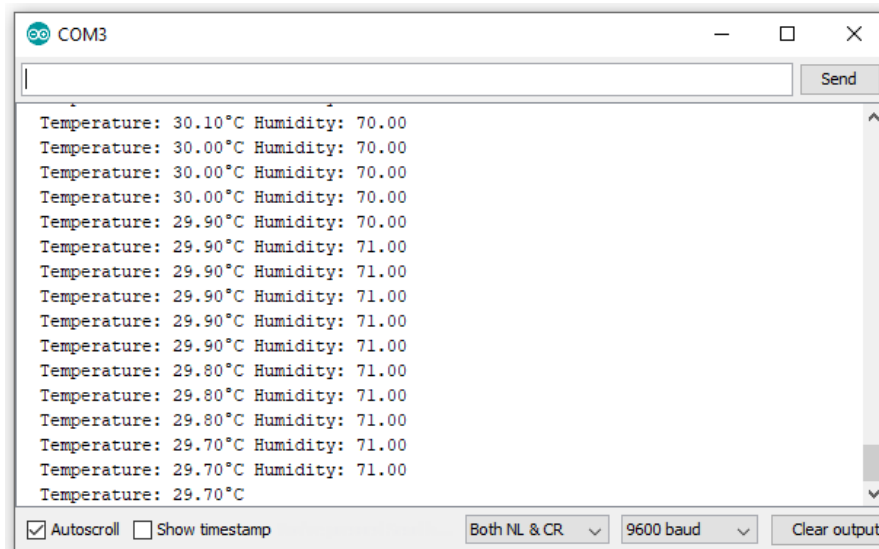
11.Klik View



3.4 Praktikum 4 : Pengukuran suhu dan kelembapan menggunakan DHT11

a. Prosedur

1. Pasang perangkat sesuai dengan skematik rangkaian.
2. Pasang kabel mikroUSB antara NodeMCU dan Laptop/PC.
3. Ketik sketch/kode program pada Arduino IDE.
4. Upload sketch tersebut ke NodeMCU.
5. Setelah proses upload selesai kemudian buka serial monitor pada Arduino IDE untuk menampilkan nilai suhu dan kelembapan.



The screenshot shows the Serial Monitor window for COM3. The output displays a series of temperature and humidity readings. The temperature values range from 30.10°C down to 29.70°C, and the humidity values are consistently 70.00 or 71.00. The window includes a 'Send' button, a scroll bar, and settings for 'Autoscroll', 'Show timestamp', 'Both NL & CR', '9600 baud', and 'Clear output'.

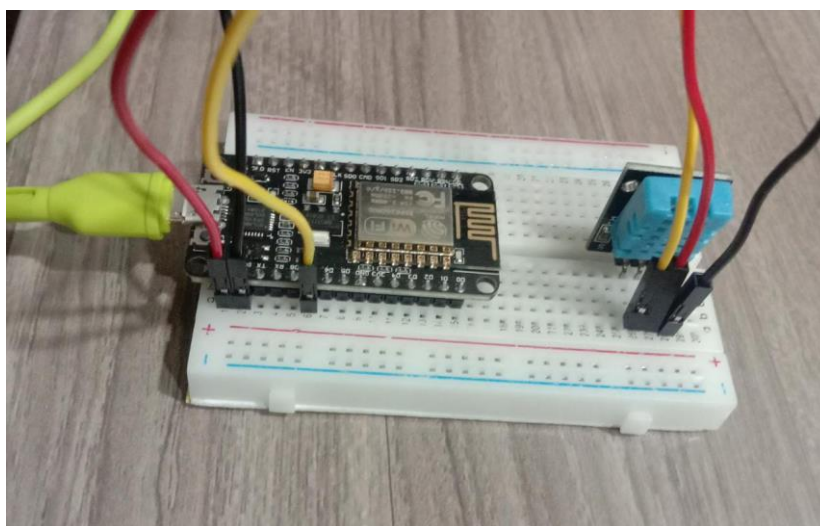
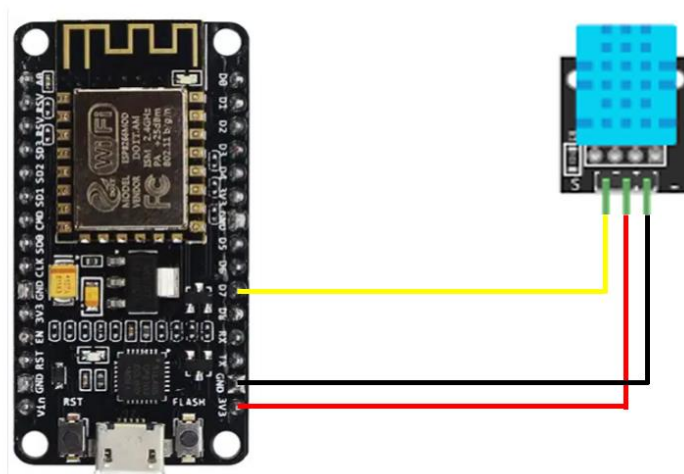
```

COM3
Temperature: 30.10°C Humidity: 70.00
Temperature: 30.00°C Humidity: 70.00
Temperature: 30.00°C Humidity: 70.00
Temperature: 30.00°C Humidity: 70.00
Temperature: 29.90°C Humidity: 70.00
Temperature: 29.90°C Humidity: 71.00
Temperature: 29.90°C Humidity: 71.00
Temperature: 29.90°C Humidity: 71.00
Temperature: 29.90°C Humidity: 71.00
Temperature: 29.90°C Humidity: 71.00
Temperature: 29.80°C Humidity: 71.00
Temperature: 29.80°C Humidity: 71.00
Temperature: 29.80°C Humidity: 71.00
Temperature: 29.70°C Humidity: 71.00
Temperature: 29.70°C Humidity: 71.00
Temperature: 29.70°C
  
```

b. Komponen

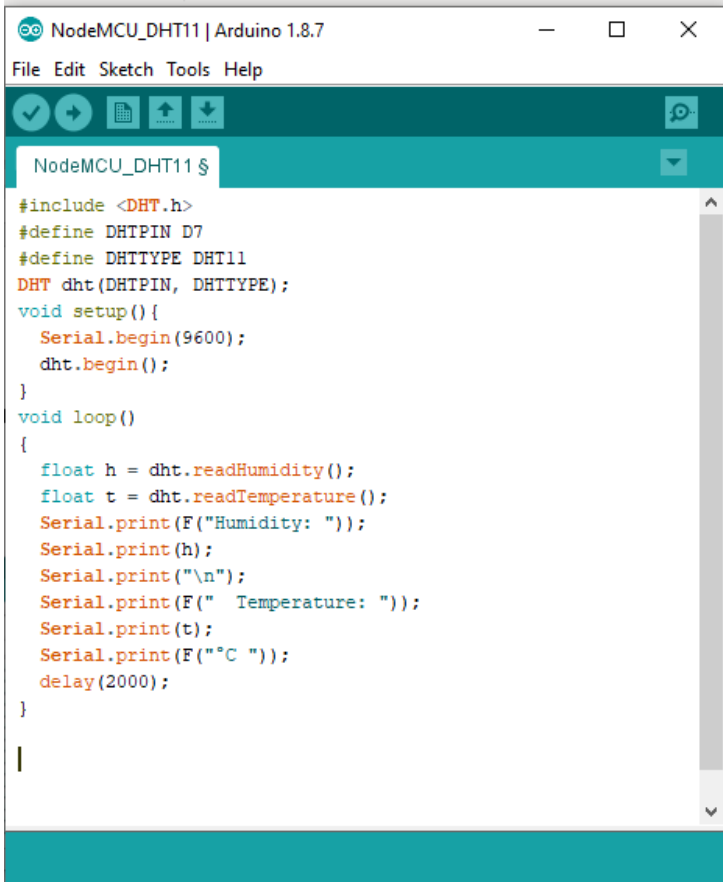
No	Komponen	Jumlah
1	Arduino IDE	1
2	Sensor suhu dan kelembapan – DHT11	1
3	NodeMCU	1
4	Kabel USB	1
5	Breadboard	1
6	Kabel jumper male to male	3

c. Skema rangkaian



d. Sketch/Kode program


```
#include <DHT.h>
#define DHTPIN D7
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup(){
  Serial.begin(9600);
  dht.begin();
}
void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print("\n");
  Serial.print(F("  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  delay(2000);
}
```

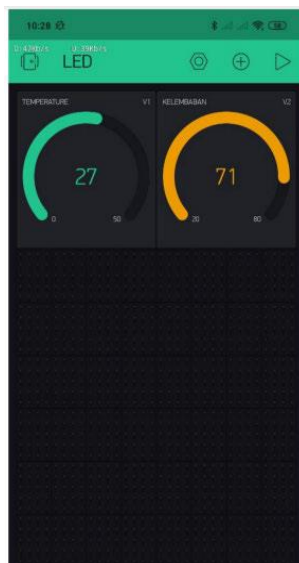


```
NodeMCU_DHT11 | Arduino 1.8.7
File Edit Sketch Tools Help
NodeMCU_DHT11 $
#include <DHT.h>
#define DHTPIN D7
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  dht.begin();
}
void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print("\n");
  Serial.print(F(" Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  delay(2000);
}
```

3.5 Praktikum 5 : Pengukuran suhu dan kelembapan menggunakan DHT11 & Blynk

a. Prosedur

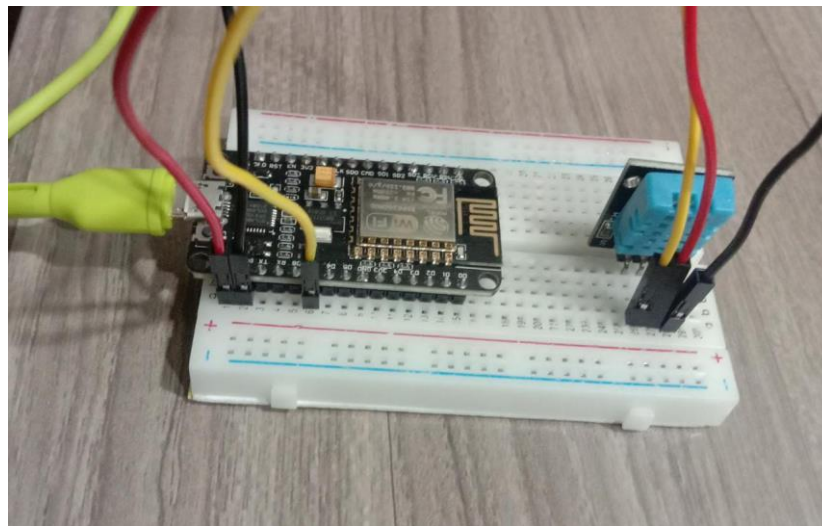
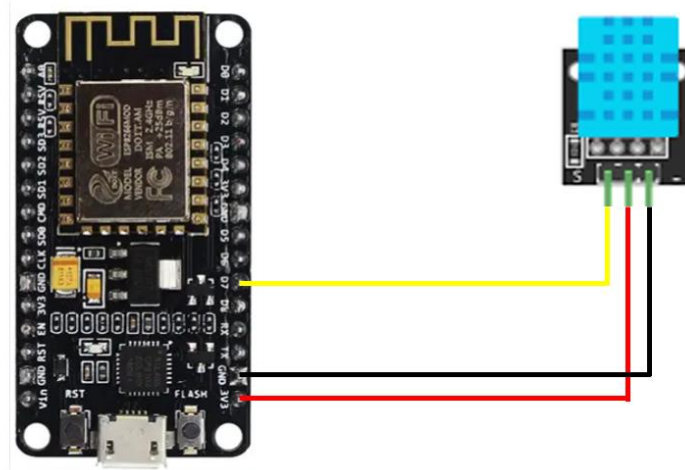
1. Pasang perangkat sesuai dengan skematik rangkaian.
2. Lakukan pengaturan project pada aplikasi Blynk.
3. Pasang kabel mikroUSB antara NodeMCU dan Laptop/PC.
4. Ketik sketch/kode program pada Arduino IDE.
5. Upload sketch tersebut ke NodeMCU.
6. Jalankan project Blynk dengan cara klik .
7. Hasil pengukuran suhu dan kelembapan akan terlihat pada dashboard Blynk.



b. Komponen

No	Komponen	Jumlah
1	Arduino IDE	1
2	Blynk	1
3	Sensor suhu & kelembapan – DHT11	1
4	NodeMCU	1
5	Kabel USB	1
6	Breadboard	1
7	Kabel jumper male to male	5

c. Skema rangkaian



d. Sketch/Kode program

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

//auth blynk
char auth[] = "isi_auth_token";

//wifi
char ssid[] = "ssid_wifi_Anda";
char pass[] = "password_wifi_Anda";

#define DHTPIN D7
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
```

```
BlynkTimer timer;

void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Gagal membaca sensor!");
    return;
  }
  Blynk.virtualWrite(V0, t);
  Blynk.virtualWrite(V1, h);
}

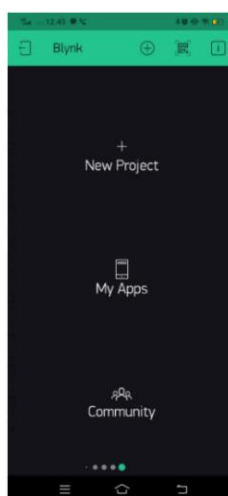
void setup()
{
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);

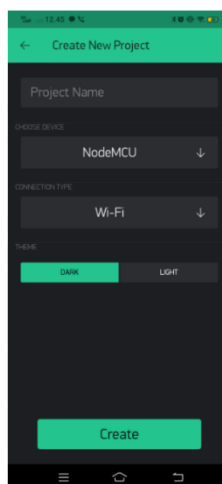
  dht.begin();
  timer.setInterval(1000L, sendSensor);
}

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

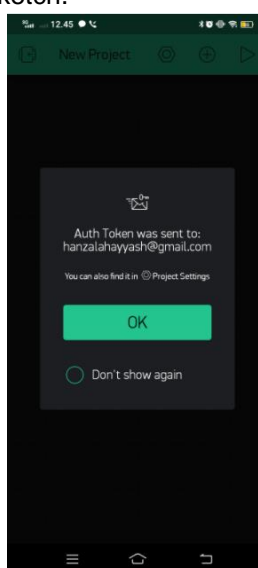
- e. Pengaturan Project Blynk
- Buka aplikasi Blynk. Klik New Project.




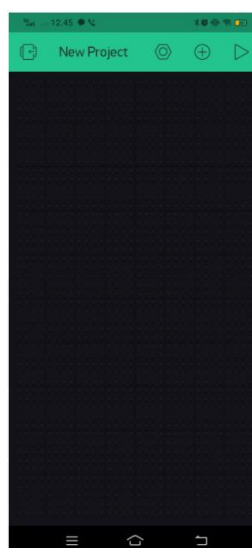
- Beri nama project pada Project Name. Pilih NodeMCU sebagai device dan WIFI sebagai connection type. Klik Create.



- Klik OK kemudian periksa email yang dikirimkan oleh Blynk untuk menyalin kode autentifikasi ke dalam Sketch.

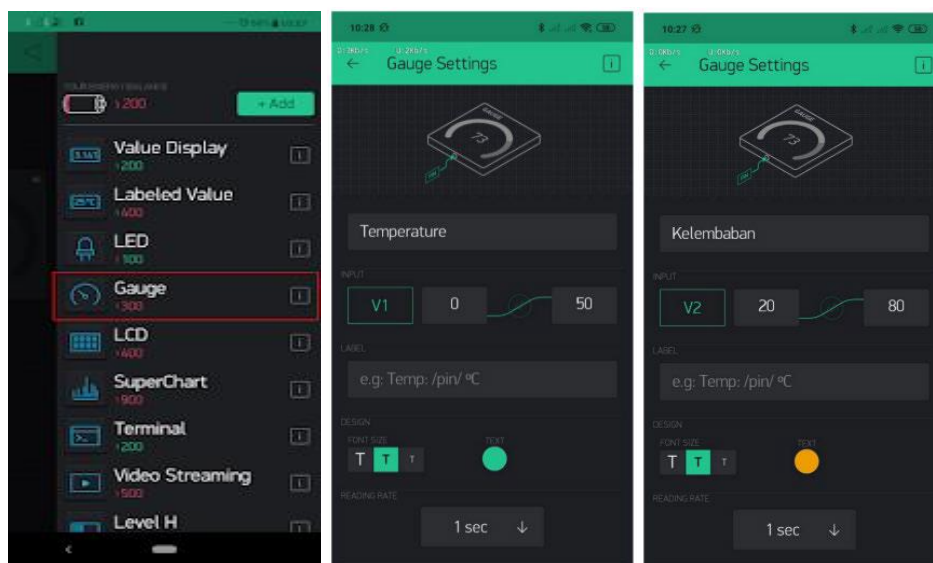


- Klik ikon  untuk membuat tampilan dashboard Blynk.



- Pilih 2 buah widget pada Widget Box yaitu Gauge. Klik gauge pertama dengan nama "Temperature" kemudian setting PIN menjadi Virtual 1 (V1) kemudian klik

gauge pertama dengan nama “Kelembaban” kemudian setting PIN menjadi Virtual 0 (V0).

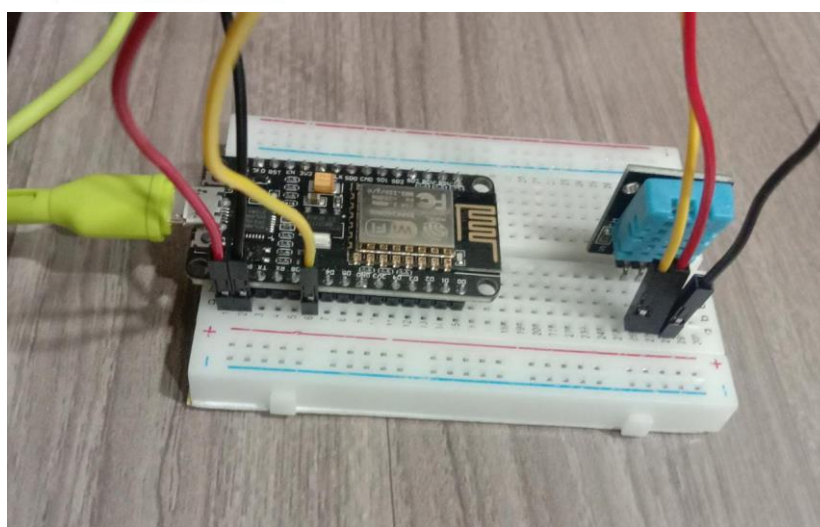
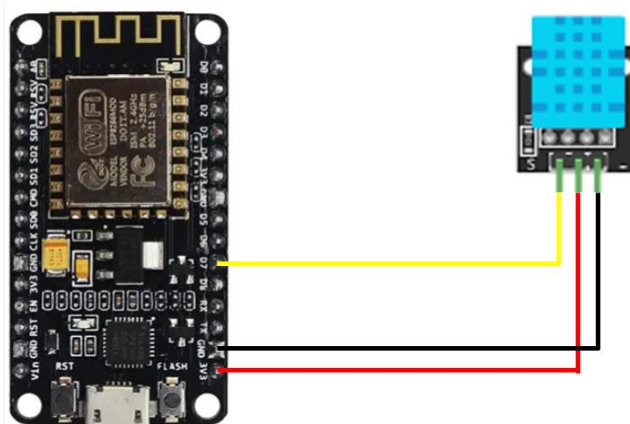


3.6 Praktikum 6 : Pengukuran suhu dan kelembapan menggunakan DHT11 & Thingspeak

a. Komponen

No	Komponen	Jumlah
1	Arduino IDE	1
2	Thingspeak	1
3	Sensor suhu & kelembapan – DHT11	1
4	NodeMCU	1
5	Kabel USB	1
6	Breadboard	1
7	Kabel jumper male to male	5

b. Skema rangkaian



c. Sketch/Kode program

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>;
#include <ThingSpeak.h>;
String apiKey = "isi_auth_token_Anda";
const char *ssid = "ssid_wifi_Anda";
const char *pass = "password_wifi_Anda";
const char* server = "api.thingspeak.com";
#define DHTPIN D7
DHT dht(DHTPIN, DHT11);
WiFiClient client;
unsigned long myChannelNumber = id_channel_Anda;
const char * myWriteAPIKey = "isi_auth_token_Anda";
void setup()
{
    Serial.begin(9600);
    delay(10);
    dht.begin();
    Serial.println("Connecting ");
    Serial.println(ssid);
    WiFi.begin(ssid, pass);
    ThingSpeak.begin(client);
}

void loop()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    if (isnan(h) || isnan(t))
    {
```

```

        Serial.println("Sensor Tidak Dapat
Mendeteksi");

        return;

    }

    ThingSpeak.writeField(myChannelNumber, 1,t,
myWriteAPIKey);

    ThingSpeak.writeField(myChannelNumber, 2,h,
myWriteAPIKey);

    delay(100);

    //ThingSpeak.read();

}

```



```

DHT_ThingSpeak | Arduino 1.8.7
File Edit Sketch Tools Help
DHT_ThingSpeak$
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>;
#include <ThingSpeak.h>;
String apiKey = "isi_APIKEY_Anda";
const char *ssid = "ssid_wifi_Anda";
const char *pass = "password_wifi_Anda";
const char *server = "api.thingspeak.com";
#define DHTPIN D3
DHT dht(DHTPIN, DHT11);
WiFiClient client;
unsigned long myChannelNumber = isi_nomor_channel_Anda;
const char * myWriteAPIKey = " isi_APIKEY_Anda";

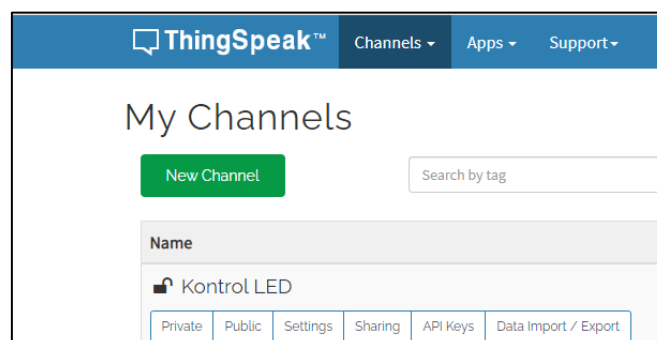
void setup()
{
    Serial.begin(9600);
    delay(10);
    dht.begin();
    Serial.println("Connecting ");
    Serial.println(ssid);
    WiFi.begin(ssid, pass);
    ThingSpeak.begin(client);
}

void loop()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();
}

```

d. Pengaturan Channel Thingspeak

1. Buka www.thingspeak.com
2. Buat akun baru atau tambah channel baru bila sudah memiliki akun Thingspeak.



ThingSpeak™ Channels Apps Support

New Channel

Name:

Description:

Field 1:

Field 2:

Field 3:

Field 4:

Field 5:

Field 6:

Field 7:

Field 8:

ThingSpeak™ Channels Apps Support

Elevation:

Show Channel Location:

Latitude:

Longitude:

Show Video:

YouTube

Vimeo

Video URL:

Show Status:

ThingSpeak™ Channels Apps Support

My Channels

Name

Kontrol LED

Monitoring Suhu & Kelembapan

3. Tambahkan widget untuk visualisasi data

ThingSpeak™ Channels Apps Support Commercial Use How to Buy HA

Channel 2 of 2 < >

Channel Stats

Created: 29 minutes ago
 Last entry: less than a minute ago
 Entries: 10

Field 1 Chart

Monitoring Suhu & Kelembapan

Suhu

Date	Suhu
11:04:30	26.9
11:05:00	26.8
11:05:30	26.8
11:06:00	26.9
11:06:30	27.0

ThingSpeak.com

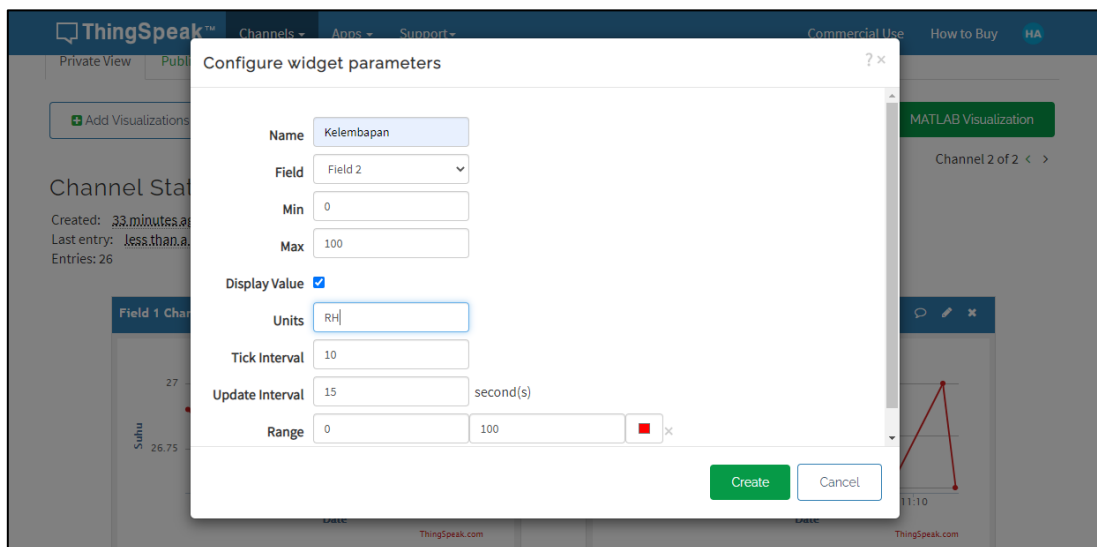
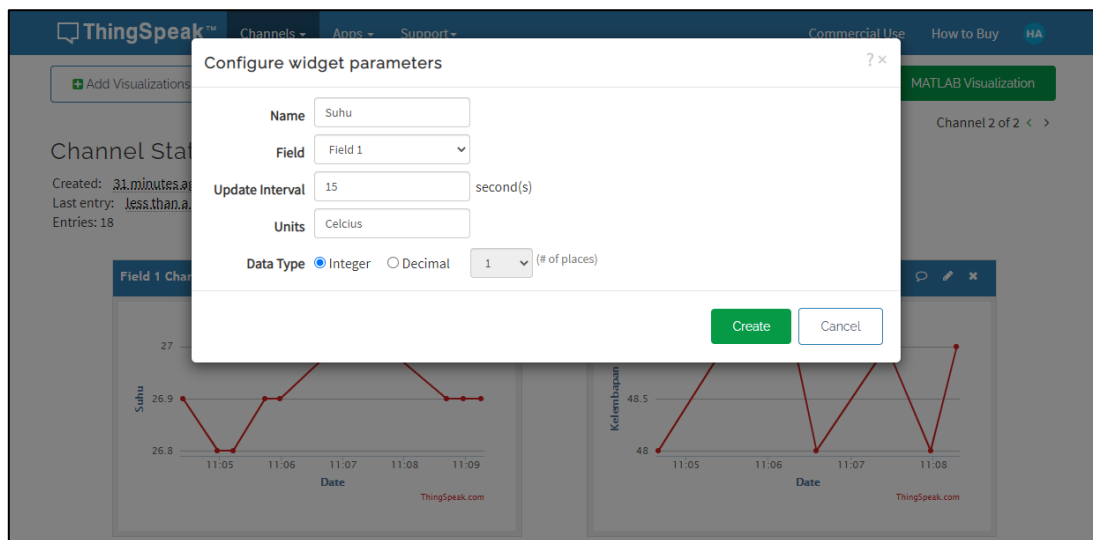
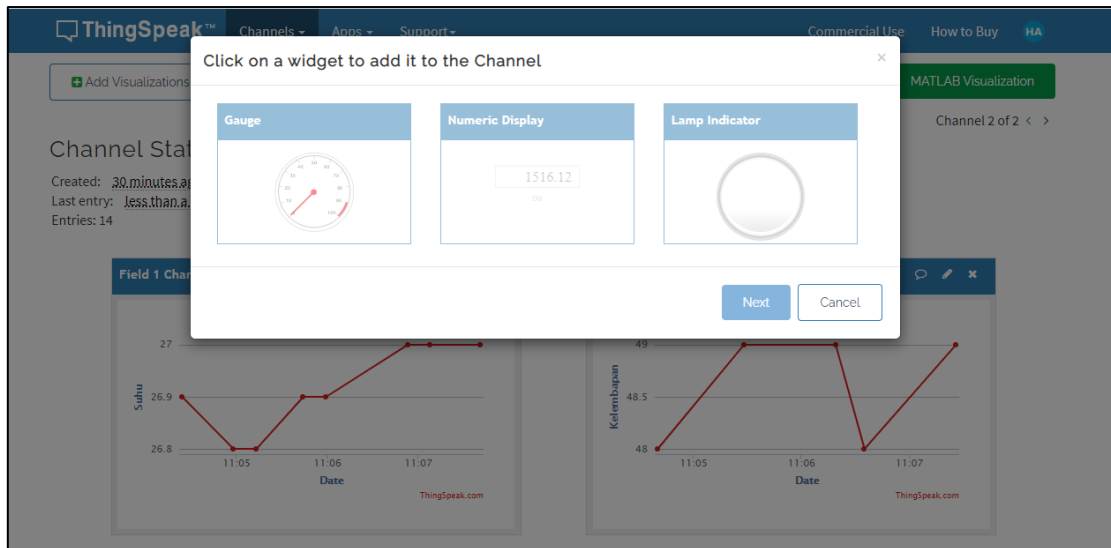
Field 2 Chart

Monitoring Suhu & Kelembapan

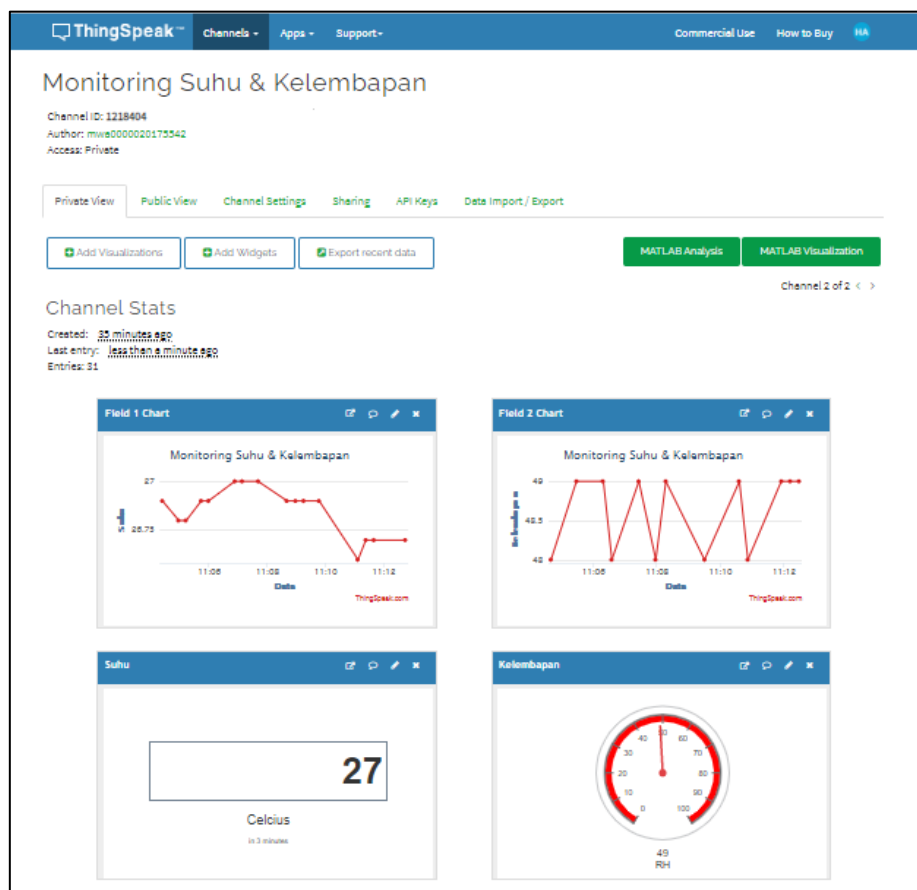
Kelembapan

Date	Kelembapan
11:05:00	48.0
11:05:30	49.0
11:06:00	49.0
11:06:30	48.0

ThingSpeak.com



- e. Pengaturan Channel Thingspeak
 1. Pasang perangkat sesuai dengan skematik rangkaian.
 2. Lakukan pengaturan project pada platform Thingspeak.
 3. Pasang kabel mikroUSB antara NodeMCU dan Laptop/PC.
 4. Ketik sketch/kode program pada Arduino IDE.
 5. Upload sketch tersebut ke NodeMCU.
 6. Buka tampilan private pada Thingspeak.



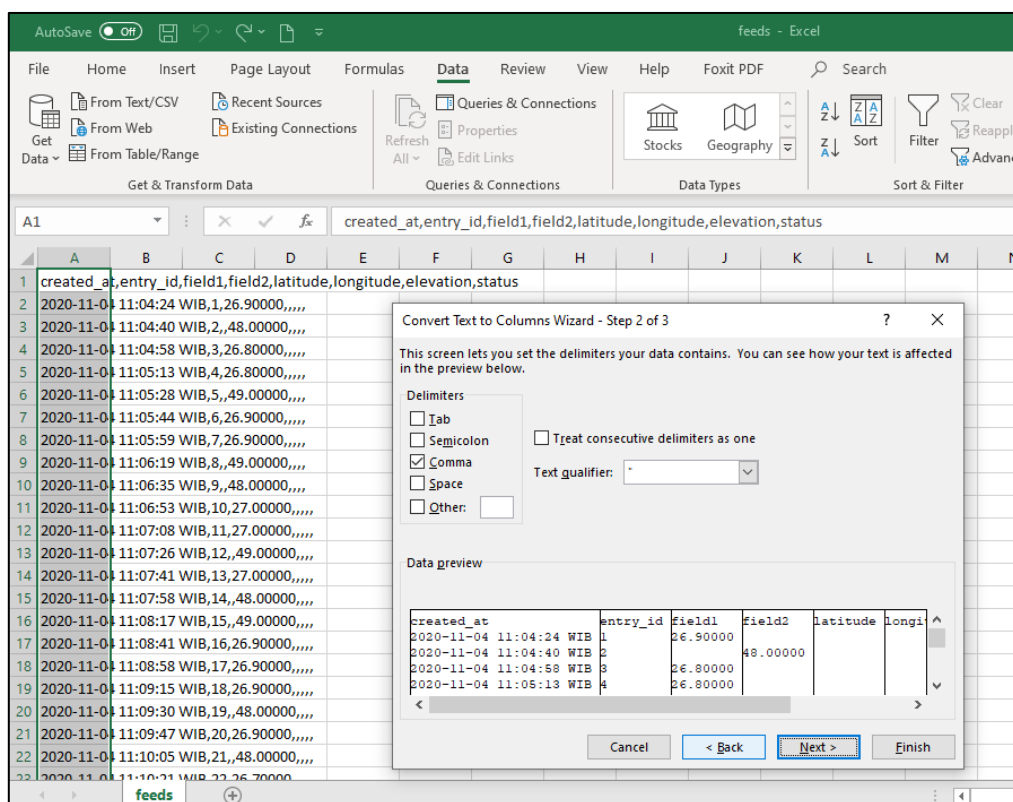
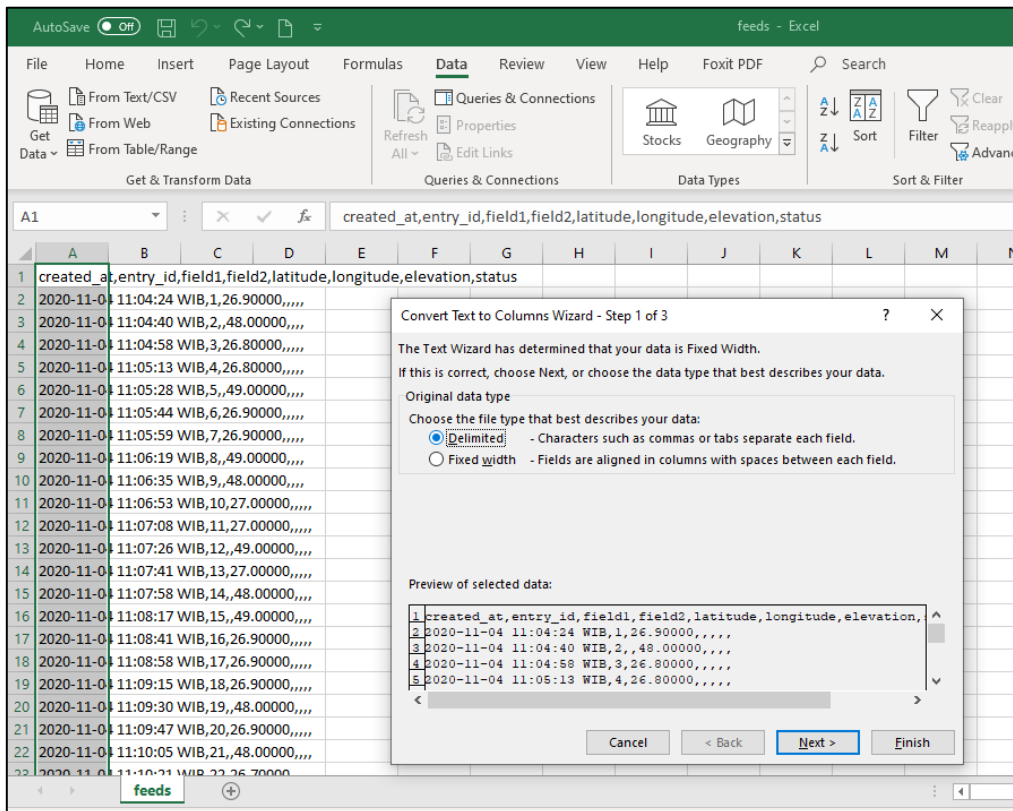
7. Unduh data hasil monitoring

The screenshot shows the ThingSpeak interface for a channel named "Monitoring Suhu & Kelembapan". The channel ID is 1218404, the author is mwa0000020175542, and the access is private. The "Data Import / Export" tab is selected. Under the "Import" section, there is a "File" field with a "Choose File" button, a "Time Zone" dropdown set to "(GMT+00:00) UTC", and an "Upload" button. Under the "Export" section, there is a "Time Zone" dropdown set to "(GMT+07:00) Jakarta" and a "Download" button. A "Help" section on the right explains the CSV import format, showing a sample row: "datetime, field1, field2, ..." with a timestamp "2019-01-01T10:11:12-04:00".

The screenshot shows the Microsoft Excel interface with the "Data" tab selected. The "Get & Transform Data" group contains options like "From Text/CSV", "From Web", and "From Table/Range". The "Queries & Connections" group contains "Refresh All" and "Edit Links". The data table is displayed in the following format:

	A	B	C	D	E	F	G	H
1	created_at,	entry_id,	field1,	field2,	latitude,	longitude,	elevation,	status
2	2020-11-04 11:04:24	WIB,1,	26.90000,	,,,,,				
3	2020-11-04 11:04:40	WIB,2,,	48.00000,	,,,,,				
4	2020-11-04 11:04:58	WIB,3,	26.80000,	,,,,,				
5	2020-11-04 11:05:13	WIB,4,	26.80000,	,,,,,				
6	2020-11-04 11:05:28	WIB,5,,	49.00000,	,,,,,				
7	2020-11-04 11:05:44	WIB,6,	26.90000,	,,,,,				
8	2020-11-04 11:05:59	WIB,7,	26.90000,	,,,,,				
9	2020-11-04 11:06:19	WIB,8,,	49.00000,	,,,,,				
10	2020-11-04 11:06:35	WIB,9,,	48.00000,	,,,,,				
11	2020-11-04 11:06:53	WIB,10,	27.00000,	,,,,,				
12	2020-11-04 11:07:08	WIB,11,	27.00000,	,,,,,				

8. Seleksi semua data. Pilih menu Data>Text to Column



The screenshot shows the 'Convert Text to Columns Wizard - Step 3 of 3' dialog box in Microsoft Excel. The dialog is titled 'Convert Text to Columns Wizard - Step 3 of 3' and has a question mark icon in the top right corner. The main text reads: 'This screen lets you select each column and set the Data Format.' Under 'Column data format', the 'General' radio button is selected. Below it, there are options for 'Text', 'Date: DMY', and 'Do not import column (skip)'. An 'Advanced...' button is to the right. The 'Destination' field is set to 'SAS1'. A 'Data preview' window shows a table with columns: 'General', 'General', 'General', 'General', 'General'. The data rows are: 'created_at', 'entry_id', 'field1', 'field2', 'latitude', 'longitude', 'elevation', 'status'. The first row of data is: '2020-11-04 11:04:24 WIB,1,26.90000,,,,,'. The dialog has 'Cancel', '< Back', 'Next >', and 'Finish' buttons at the bottom.

The screenshot shows the result of the text-to-columns conversion in Microsoft Excel. The spreadsheet now has columns A through H with headers: 'created_at', 'entry_id', 'field1', 'field2', 'latitude', 'longitude', 'elevation', 'status'. The data is neatly separated into these columns. The first few rows of data are:

created_at	entry_id	field1	field2	latitude	longitude	elevation	status
2020-11-04	1	2.690.000					
2020-11-04	2		4.800.000				
2020-11-04	3	2.680.000					
2020-11-04	4	2.680.000					
2020-11-04	5		4.900.000				
2020-11-04	6	2.690.000					
2020-11-04	7	2.690.000					
2020-11-04	8		4.900.000				
2020-11-04	9		4.800.000				
2020-11-04	10	2.700.000					
2020-11-04	11	2.700.000					
2020-11-04	12		4.900.000				
2020-11-04	13	2.700.000					
2020-11-04	14		4.800.000				
2020-11-04	15		4.900.000				
2020-11-04	16	2.690.000					
2020-11-04	17	2.690.000					
2020-11-04	18	2.690.000					
2020-11-04	19	2.690.000					
2020-11-04	20		4.800.000				
2020-11-04	21	2.690.000					
2020-11-04	22		4.800.000				

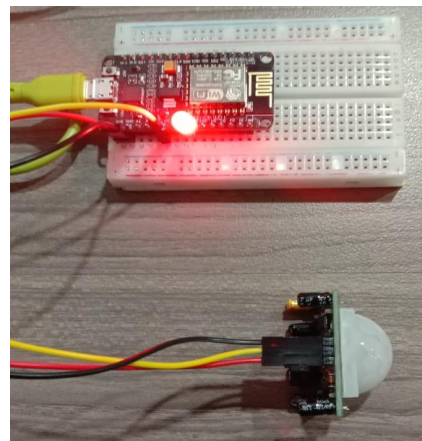
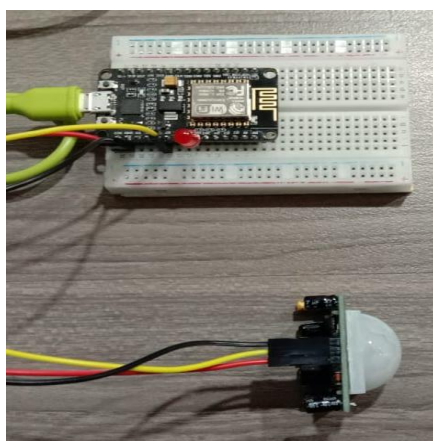
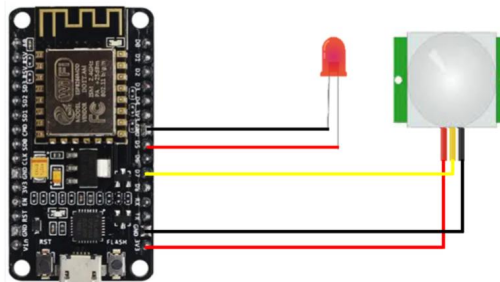
3.7 Praktikum 7 : Deteksi gerak menggunakan sensor PIR

- a. Prosedur
1. Pasang perangkat sesuai dengan skematik rangkaian.
 2. Pasang kabel mikroUSB antara NodeMCU dan Laptop/PC.
 3. Ketik sketch/kode program pada Arduino IDE.
 4. Upload sketch tersebut ke NodeMCU.
 5. Setelah proses upload selesai maka sensor PIR akan mendeteksi adanya gerakan. Bila terdeteksi gerakan maka LED akan menyala.
 6. Buka serial monitor pada Arduino IDE untuk menampilkan hasil deteksi gerakan.

b. Komponen

No	Komponen	Jumlah
1	Arduino IDE	1
2	Sensor Gerak - PIR	1
3	NodeMCU	1
4	LED	
5	Kabel USB	1
6	Breadboard	1
7	Kabel jumper male to male	5

c. Skema rangkaian



d. Sketch/Kode program

```
int sensor = D7;

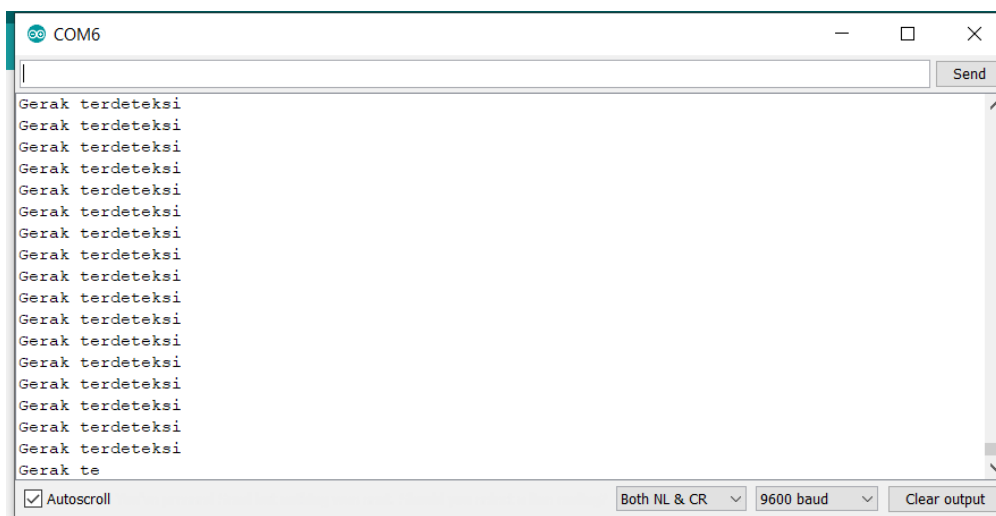
int led = D5;

void setup() {
  // put your setup code here, to run once:
  Serial.begin (9600);
  pinMode (sensor,INPUT);
  pinMode (led,OUTPUT);
}

void loop() {

  // put your main code here, to run repeatedly
  int nilaiSensor = digitalRead (sensor);


  if (nilaiSensor == HIGH){
    digitalWrite (led, HIGH);
    Serial.println("Gerak terdeteksi");
  }
  else
  {
    digitalWrite (led, LOW);
    Serial.println("Tidak ada gerakan");
  }
}
```

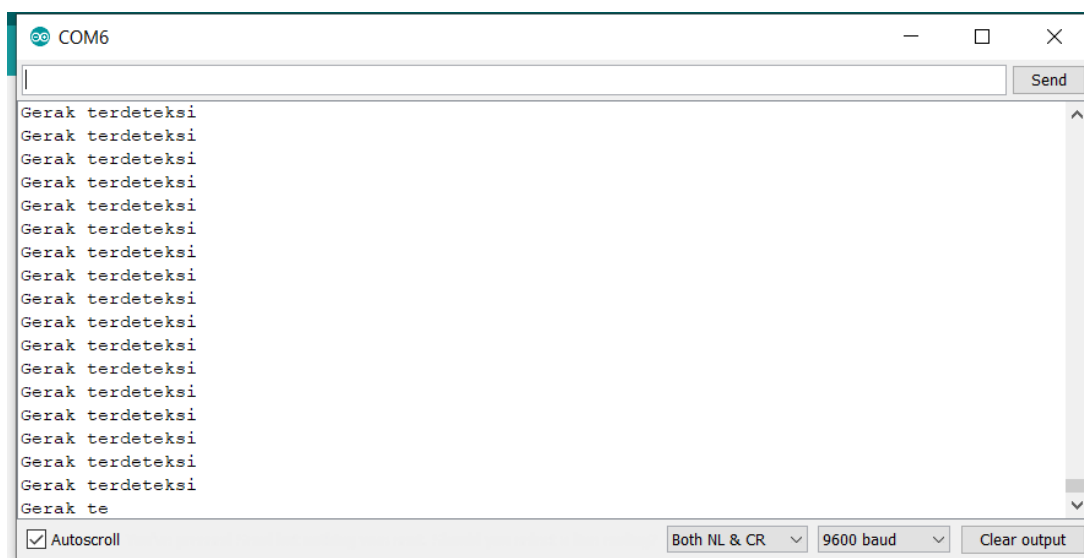


```
COM6
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak terdeteksi
Gerak te
Autoscroll Both NL & CR 9600 baud Clear output
```

3.8 Praktikum 8 : Deteksi gerak menggunakan sensor PIR & Blynk

a. Prosedur

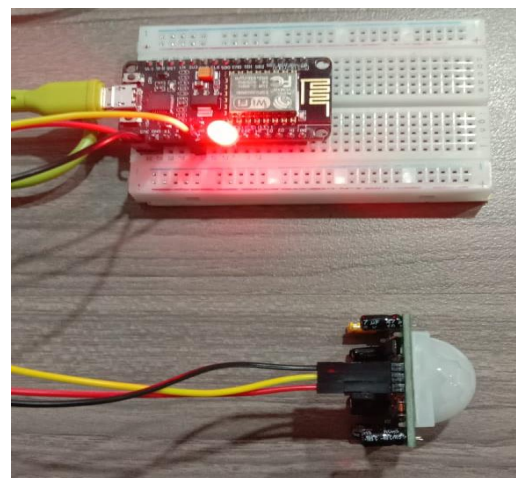
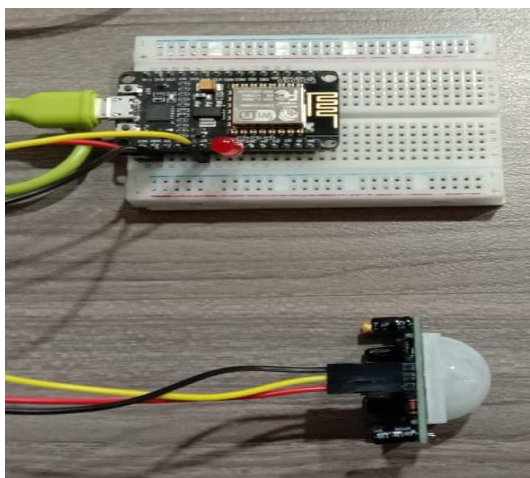
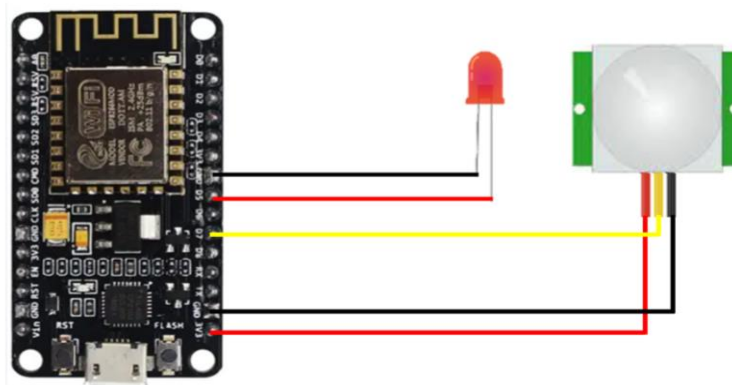
1. Pasang perangkat sesuai dengan skematik rangkaian.
2. Lakukan pengaturan project pada aplikasi Blynk.
3. Pasang kabel mikroUSB antara NodeMCU dan Laptop/PC.
4. Ketik sketch/kode program pada Arduino IDE.
5. Upload sketch tersebut ke NodeMCU.
6. Jalankan project Blynk dengan cara klik .
7. Klik button pada aplikasi Blynk untuk mengaktifkan sensor PIR.
8. Sensor PIR siap mendeteksi gerakan. Gerakan yang terdeteksi akan ditampilkan pada aplikasi Blynk dan serial monitor.



b. Aplikasi dan Komponen

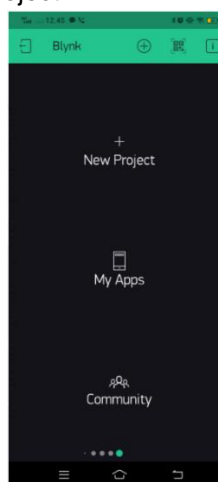
No	Komponen	Jumlah
1	Arduino IDE	1
2	Blynk	1
3	Sensor Gerak - PIR	1
4	NodeMCU	1
5	LED	1
6	Kabel USB	1
7	Breadboard	1
8	Kabel jumper male to male	5

c. Skema rangkaian

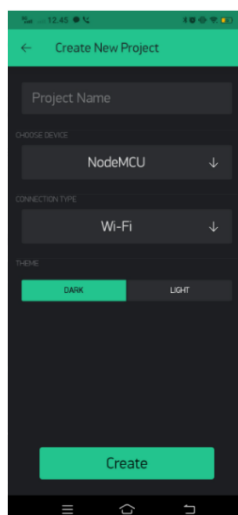


d. Pengaturan Project Blynk

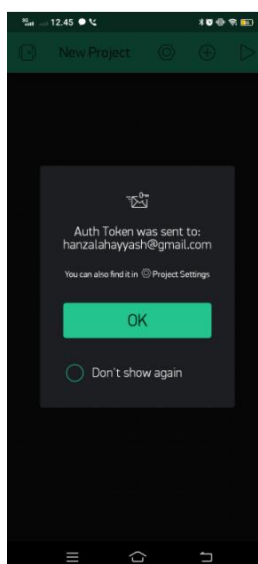
1. Buka aplikasi Blynk. Klik New Project.




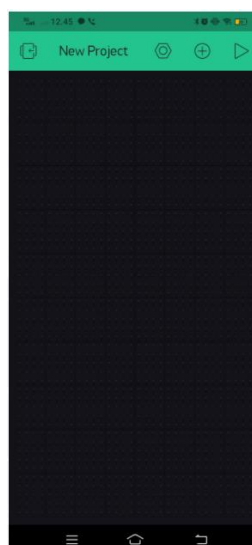
2. Beri nama project pada Project Name. Pilih NodeMCU sebagai device dan WIFI sebagai connection type. Klik Create.



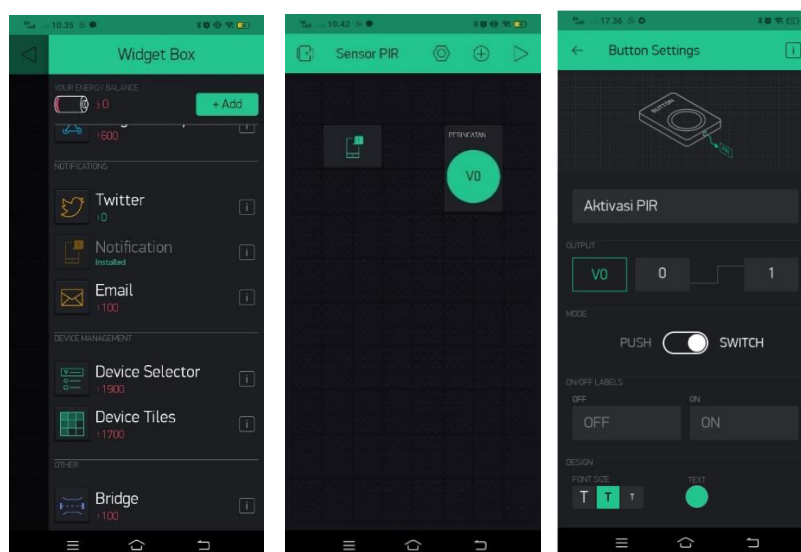
3. Klik OK kemudian periksa email yang dikirimkan oleh Blynk untuk menyalin kode autentifikasi ke dalam Sketch.



4. Klik ikon  untuk membuat tampilan dashboard Blynk.



5. Pilih 2 buah widget pada Widget Box yaitu Notification dan Button. Klik button kemudian setting PIN menjadi Virtual 0 (V0).



e. Sketch/Kode program

```
#include <ESP8266WiFi.h>

#define BLYNK_PRINT Serial

#include <BlynkSimpleEsp8266.h>
char auth[] = "isi kode auth";

/* WiFi credentials */
char ssid[] = "isi nama WIFI";
char pass[] = "isi Password WIFI";

/* HC-SR501 Motion Detector */
#define pirPin D7 // Input for HC-S501
int pirValue; // Place to store read PIR Value
int pinValue; //Variable to read virtual pin

BLYNK_WRITE(V0)
{
  pinValue = param.asInt();
}
```

```
void setup()
{
  Serial.begin(115200);
  delay(10);
  Blynk.begin(auth, ssid, pass);
  pinMode(pirPin, INPUT);
}

void loop()
{
  if (pinValue == HIGH)
  {
    getPirValue();
  }
  Blynk.run();
}

void getPirValue(void)      //Get PIR Data
{
  pirValue = digitalRead(pirPin);
  if (pirValue)
  {
    Serial.println("Gerak terdeteksi");
    Blynk.notify("Gerak terdeteksi");
  }
}
```

```
8_PIR_NodeMCU_Blynk | Arduino 1.8.5
File Edit Sketch Tools Help
8_PIR_NodeMCU_Blynk
#include <ESP8266WiFi.h>
#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>
char auth[] = "isi kode auth";

/* WiFi credentials */
char ssid[] = "isi nama WIFI";
char pass[] = "isi Password WIFI";

/* HC-SR501 Motion Detector */
#define pirPin 4 // Input for HC-S501
int pirValue; // Place to store read PIR Value
int pinValue; //Variable to read virtual pin

BLYNK_WRITE(V0)
{
  pinValue = param.asInt();
}

void setup()
{
  Serial.begin(115200);
  delay(10);
  Blynk.begin(auth, ssid, pass);
  pinMode(pirPin, INPUT);
}

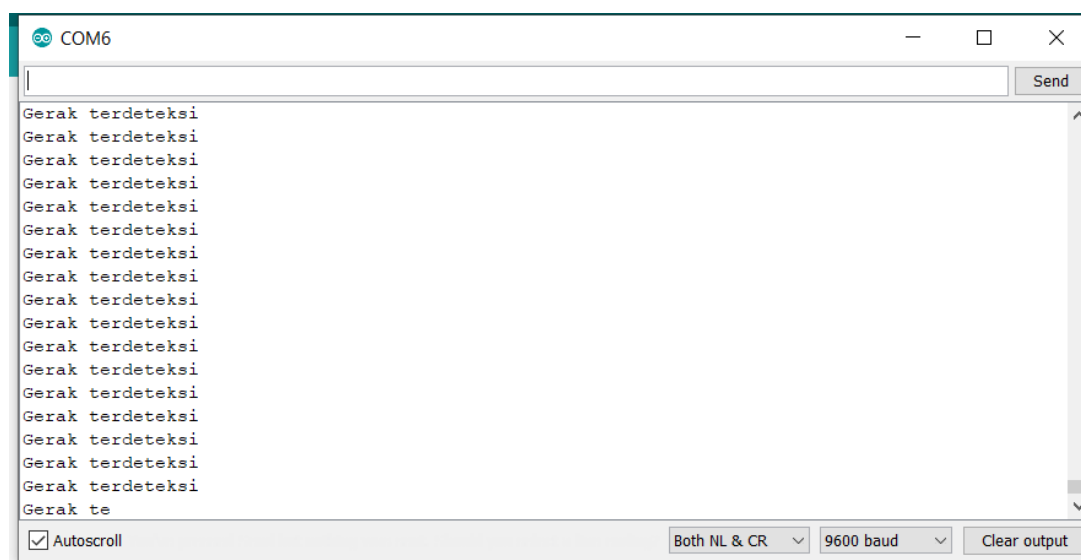
void loop()
{
  if (pinValue == HIGH)
```

Done compiling

Sketch uses 279520 bytes (26%) of program storage space. Maximum is 1044464 bytes
Global variables use 29092 bytes (35%) of dynamic memory, leaving 52828 bytes

3.9 Praktikum 9 : Deteksi gerak menggunakan sensor PIR & Thingspeak

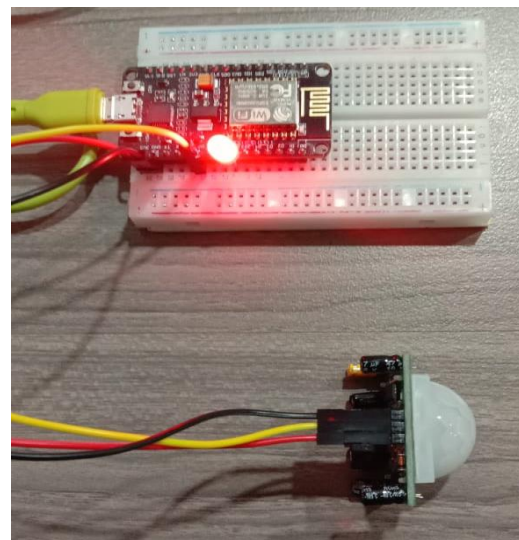
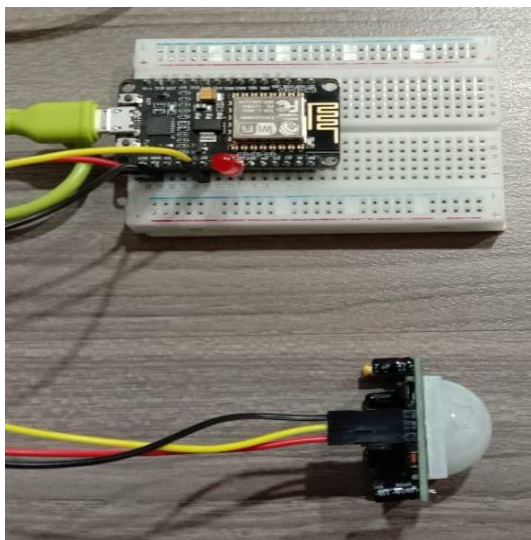
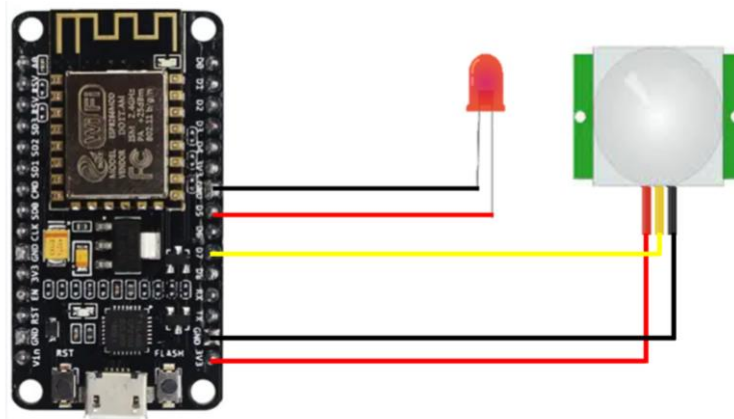
a. Prosedur



b. Aplikasi dan Komponen

No	Komponen	Jumlah
1	Arduino IDE	1
2	Thingspeak	1
3	Sensor Gerak - PIR	1
4	NodeMCU	1
5	LED	1
6	Kabel USB	1
7	Breadboard	1
8	Kabel jumper male to male	5

c. Skema rangkaian



d. Sketch/Kode program

```

#include "ThingSpeak.h" // Include ThingSpeak Library
#include <ESP8266WiFi.h> // Include ESP8266wifi Library

char ssid[] = "isi nama ssid wifi"; // Enter your WiFi SSID (router
    name)
char pass[] = "isi password wifi"; // Enter your WiFi router password
unsigned long myChannelNumber = isi channel_id; // Enter your
    Thingspeak Channel Number
const char * myWriteAPIKey = "isi Write API Key";

int ledPin = D5; // Choose the pin for the LED (Pin D5, GPIO
    14)

```

```
int inputPin = D7; // Choose the input pin for motion sensor
    (Pin D6, GPIO 12)

int motionState = 0; // We start, assuming no motion detected

int val = 0; // Variable for reading the pin status

unsigned long currentMillis = millis();
unsigned long previousMillis = 0;
unsigned long resetMillis = 0;

const long interval = 90000; // Time delay interval (90 seconds) to
    prevent false triggers when setting up device

int status = WL_IDLE_STATUS;

WiFiClient client;

void setup()
{
    WiFi.begin(ssid, pass); // Start WiFi connection
    ThingSpeak.begin(client); // Start ThingSpeak connection
    Serial.begin(115200); // Serial Baud Rate
    pinMode(ledPin, OUTPUT); // Set the LED pin as an Output
    pinMode(inputPin, INPUT); // Set the Motion Sensor pin as an
        Input

    Serial.println();
    Serial.print("Connecting to ");
    Serial.print(ssid);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    Serial.println();
    Serial.println("WiFi connected");
    Serial.print("IP address: ");
```

```
Serial.println(WiFi.localIP());

Serial.println("Motion Sensor Alarm");

ThingSpeak.writeField(myChannelNumber, 1, val, myWriteAPIKey);    // Update
    ThingSpeak channel the PIR sensor value '0'

}

void loop()
{
//***** Read motion sensor and upload to ThingSpeak
    *****

currentMillis = millis();    // Set the currentMillis equal to
    the millis function

val = digitalRead(inputPin);    // Read motion sensor

if (val == HIGH)    // Check if the input pin is HIGH
{
    if ((motionState == 0) && (currentMillis >= interval))
    {
        digitalWrite(ledPin, HIGH);    // Turn LED ON
        Serial.println("Motion detected!");    // Motion Detected
        motionState = 1;    // We only want to update on the
            change, not state
        ThingSpeak.writeField(myChannelNumber, 1, val, myWriteAPIKey);    // Update
            ThingSpeak channel the PIR sensor value '0'
        Serial.println("Data Sent to ThingSpeak!");
        delay(500);
    }
} else {
    digitalWrite(ledPin, LOW);    // Turn LED OFF

    if (motionState == 1)
    {
        // Motion stopped
    }
}
```

```
Serial.println("Motion ended!");

motionState = 0; // We only want to print on the
output change, not state

ThingSpeak.writeField(myChannelNumber, 1, val, myWriteAPIKey); // Update
ThingSpeak channel the PIR sensor value '0'

Serial.println("Data Sent to ThingSpeak!");

delay(500);

}

}

if (currentMillis - previousMillis >= 10000) // Update Serial Monitor
    every ten seconds (10000 milliseconds)
{ // as to how long the program has been
    running
    Serial.print("Program running for ");
    Serial.print((currentMillis/1000)/60);
    Serial.println(" Minutes");
    previousMillis = currentMillis;
    yield(); // Yield to prevent the WDT from
    crashing the NodeMCU
}

}
```

