

MODUL WORKSHOP ANTARES



ANTARES

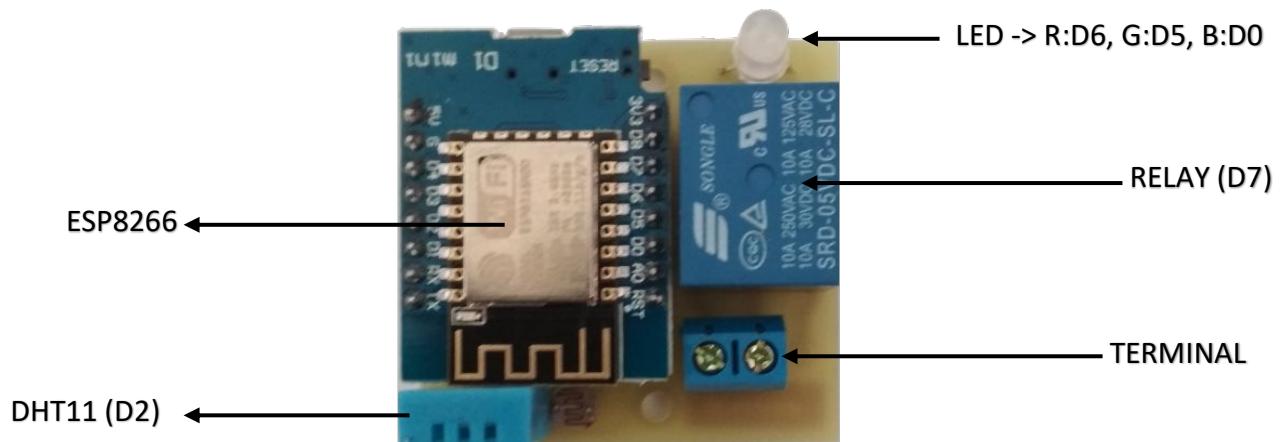
NAMA : _____

WORKSHOP ANTARES

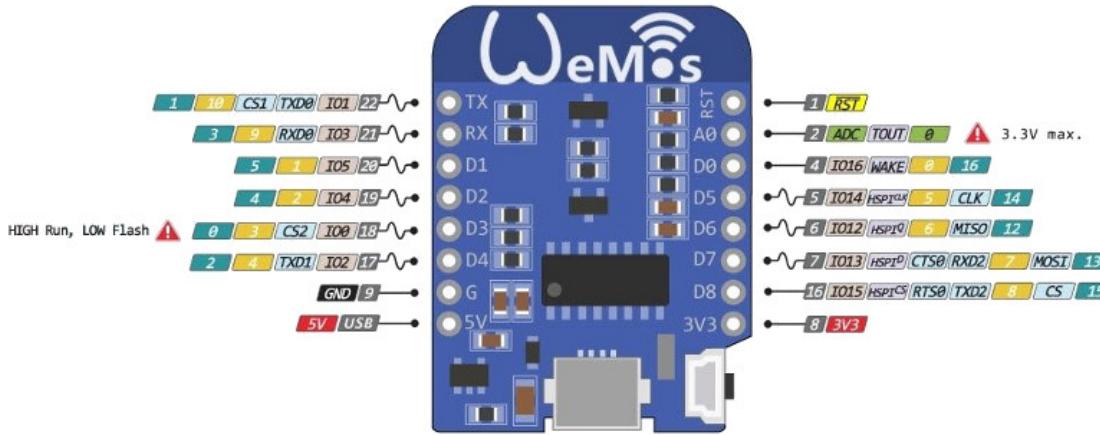
Hardware

Kelengkapan Workshop :

1. ESP8266 (Wemos)
2. DHT11 (Temperature + Humidity Sensor)
3. Relay
4. Kabel Data
5. LED
6. Casing



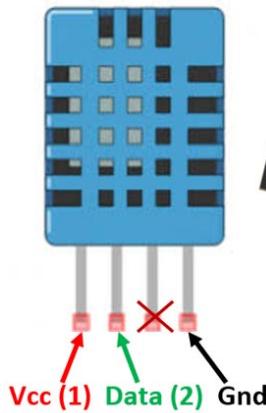
ESP8266 (WEMOS)



Gambar 1. WEMOS D1 Mini

WEMOS D1 Mini ESP8266 adalah sebuah development board seperti Arduino and dan dikembangkan khusus untuk Internet of Things aplikasi dan solusi. Jenis WEMOS ini sangat cocok digunakan untuk pembelajaran dan hobby project Internet Of Things. Dengan sebuah arsitektur 32 bit yang lebih baik diandingkan Arduino dan WEMOS ini memiliki konektivitas WiFi. WEMOS ini dapat deprogram menggunakan Arduino IDE dan Bahasa Lua.

DHT11



Gambar 2. Sensor Kelembapan dan Temperature DHT11

DHT11 merupakan sensor kelembapan dan temperature yang memiliki spesifikasi seperti berikut :

- Ultra low cost
- 3 to 5V power and I/O

- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings $\pm 2^\circ\text{C}$ accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

Relay



Gambar 4. Modul Relay

Relay adalah suatu peranti yang menggunakan elektromagnet untuk mengoperasikan seperangkat kontak sakelar. Relay terdiri dari kumparan kawat penghantar yang dililit pada inti besi. Bila kumparan ini diberikan energi, medan magnet yang terbentuk menarik armatur berporos yang digunakan sebagai pengungkit mekanisme sakelar magnet.

Software

Perangkat lunak (Software) yang dibutuhkan :

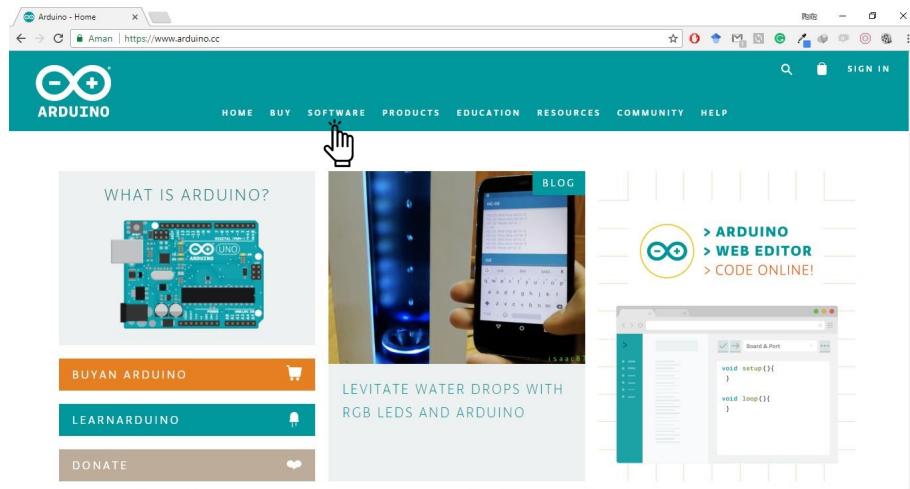
1. Arduino IDE
2. Library ESP8266
3. Library Pendukung (DHT, MQTT, dll)
4. Library Antares
5. Android Studio

Install Arduino IDE

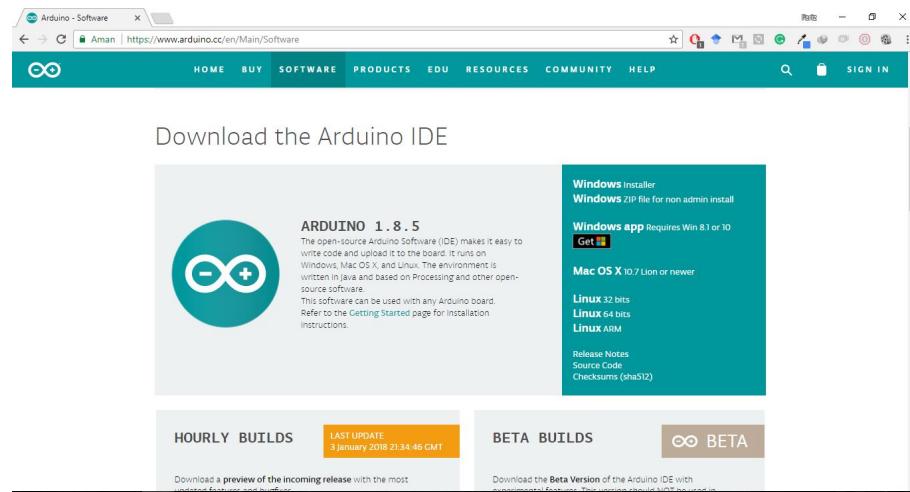
Arduino IDE digunakan untuk membuat/menuliskan sebuah program yang selanjutnya akan di-compile dan di-flash ke sebuah chip microcontroller contohnya Arduino

(atmega, atxmega, dll), WEMOS (esp8266, esp32), dan lain-lain. Untuk melakukan instalasi software Arduino IDE, langkah-langkahnya sebagai berikut :

1. Download software di Arduino.cc



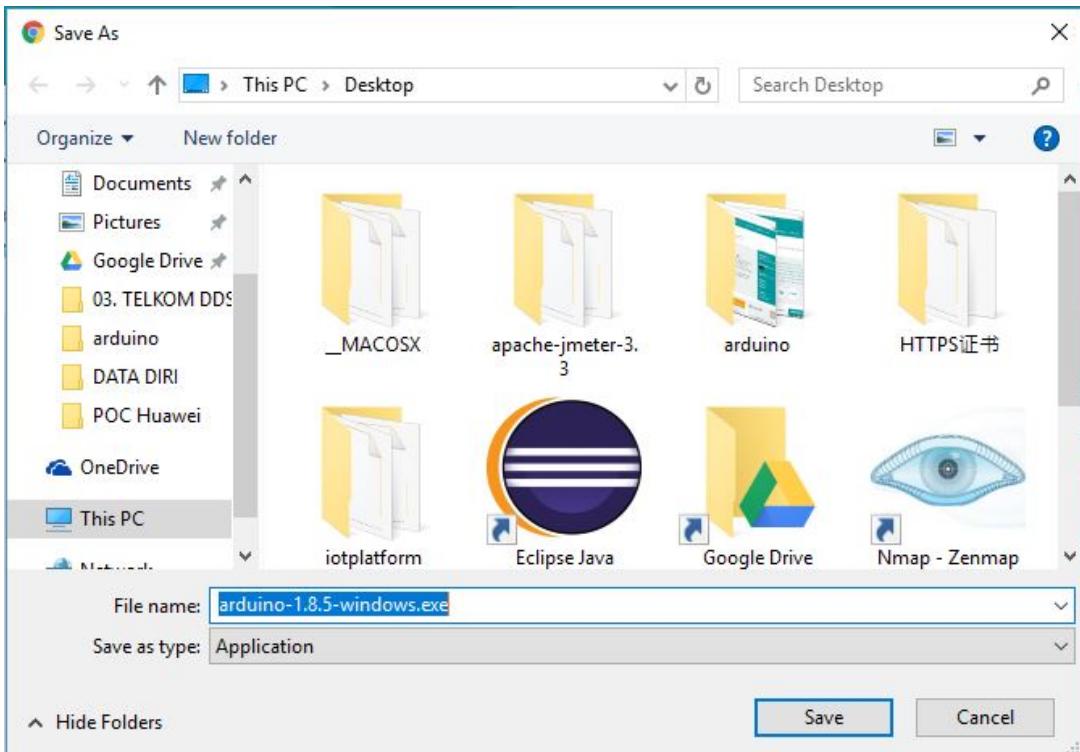
2. Pilih Menu Software



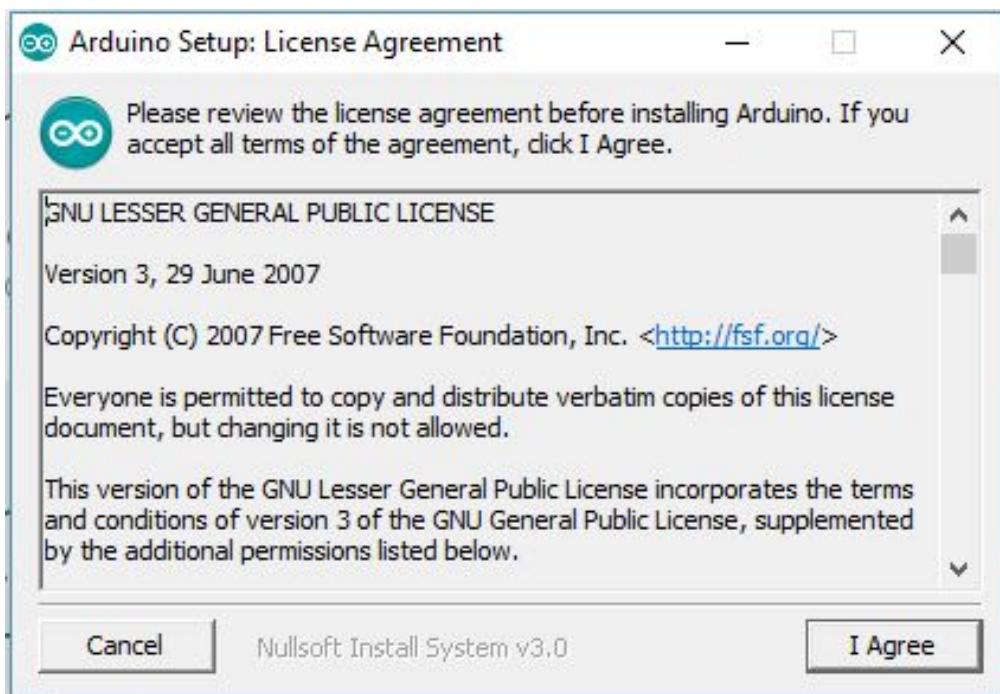
3. Pilih system operasi yang sesuai



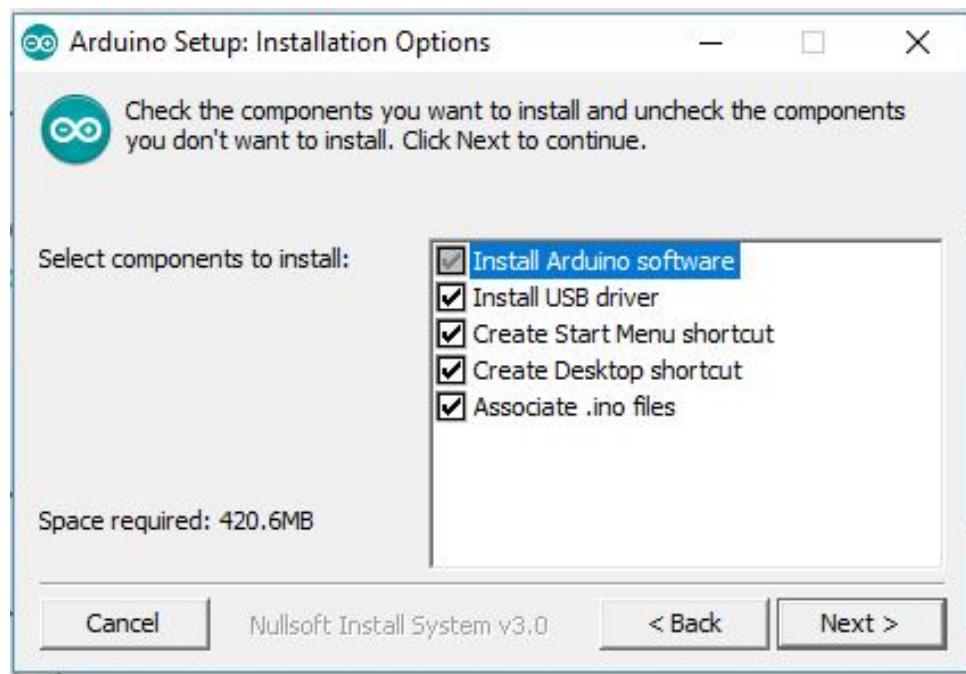
4. Pilih direktori penyimpanan



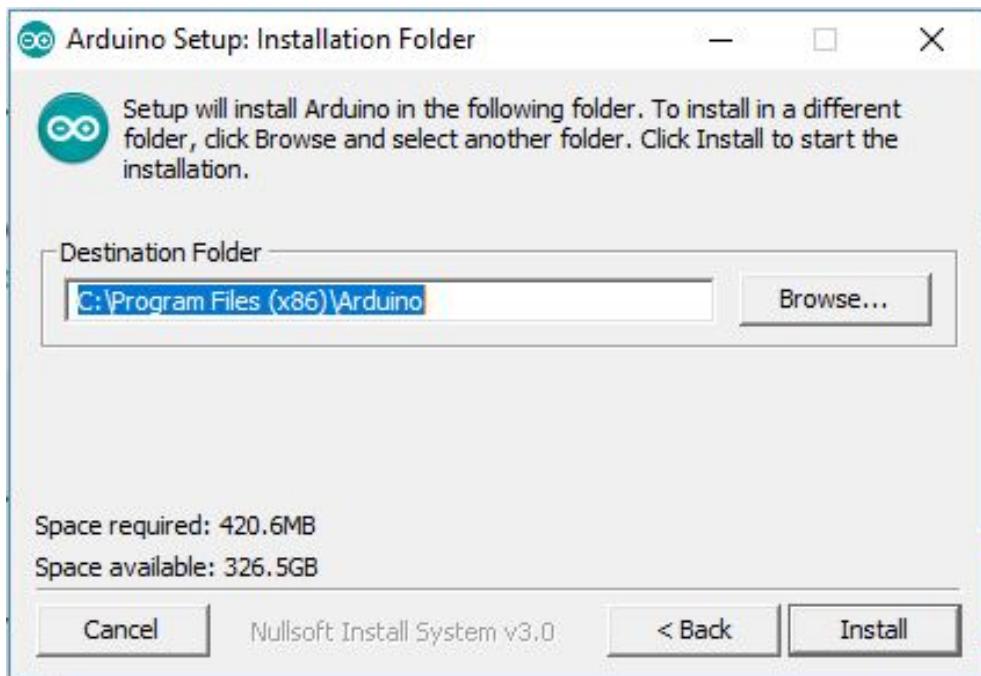
5. Seletah download selesai, Lakukan instalasi dan pilih I agree



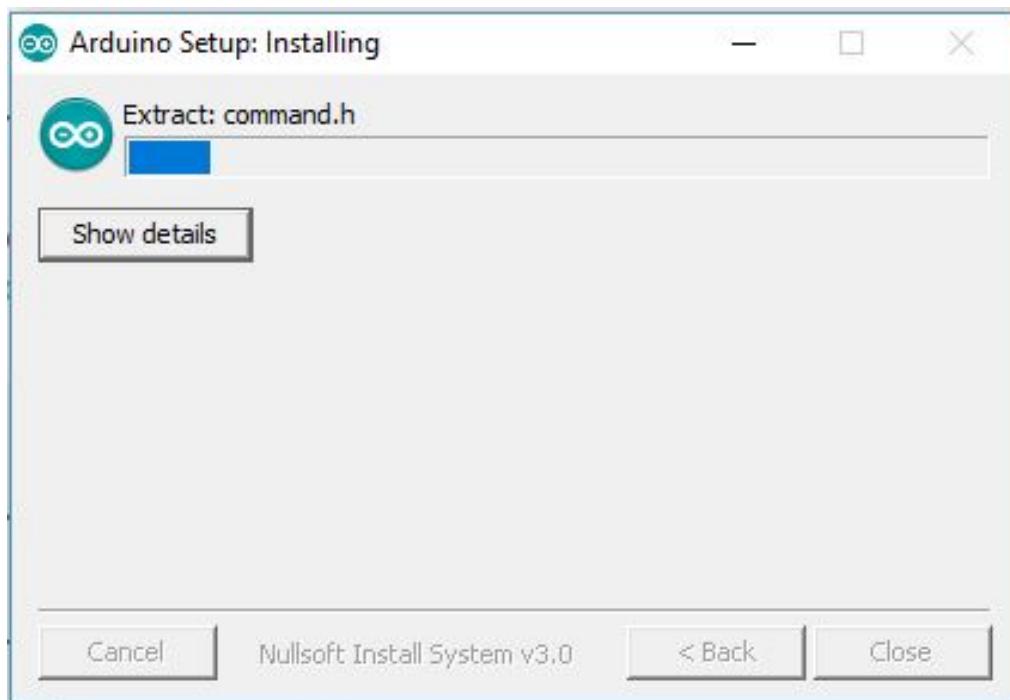
6. Ceklis semua pilihan, lalu Next



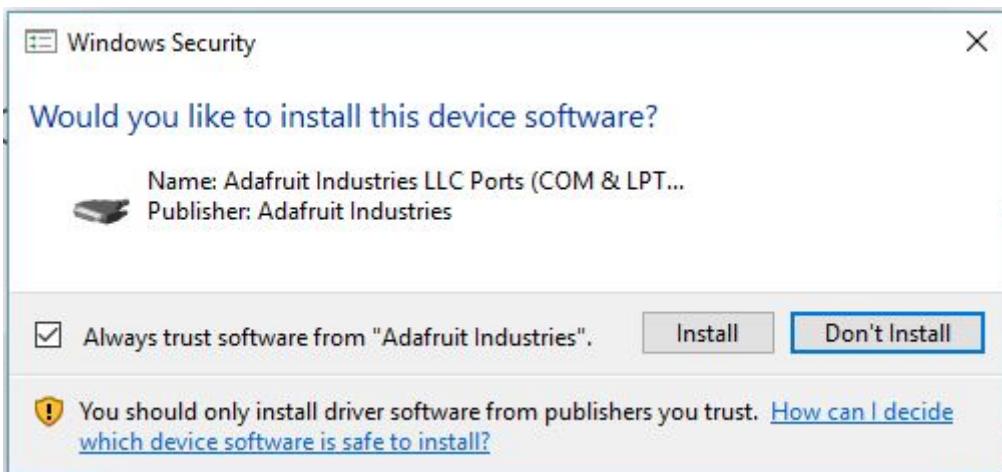
7. Pilih destinasi penyimpanan



8. Tunggu hingga selesai instalasi



9. Pilih Install apabila muncul kotak dialog seperti gambar dibawah.

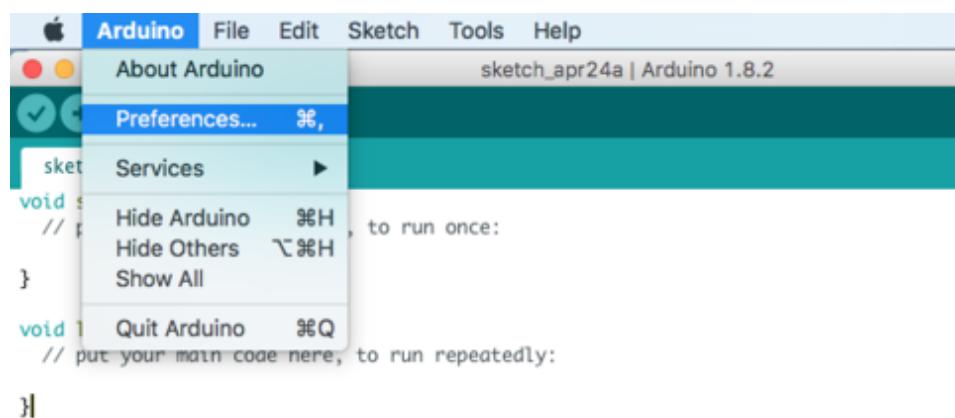


Arduino IDE telah terinstall dan siap untuk digunakan untuk membuat project.

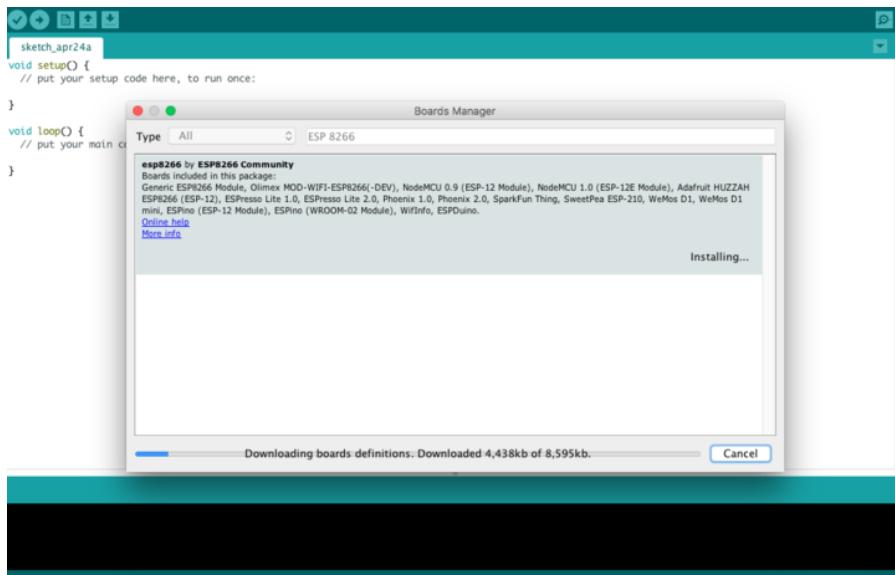
Install Library ESP8266

Untuk memulai membuat program pada WEMOS, hal yang perlu dilakukan yaitu melakukan instalasi library esp8266 pada Arduino IDE. Instalasi library dilakukan dengan langkah sebagai berikut :

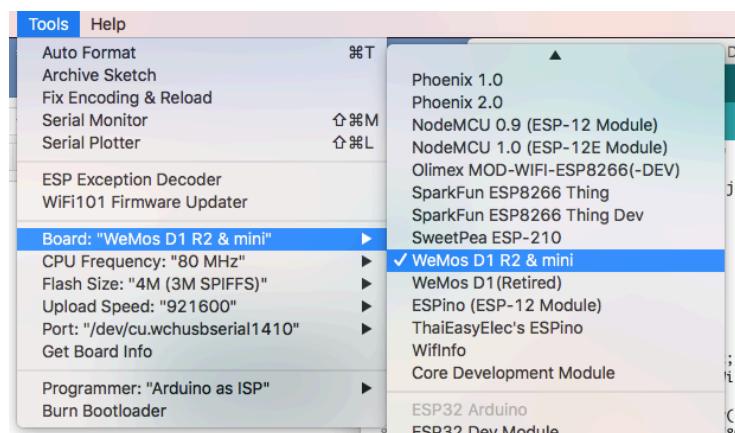
1. Buka aplikasi Arduino IDE
2. Pilih Files -> Preferences dan masukan URL
http://arduino.esp8266.com/stable/package_esp8266com_index.json



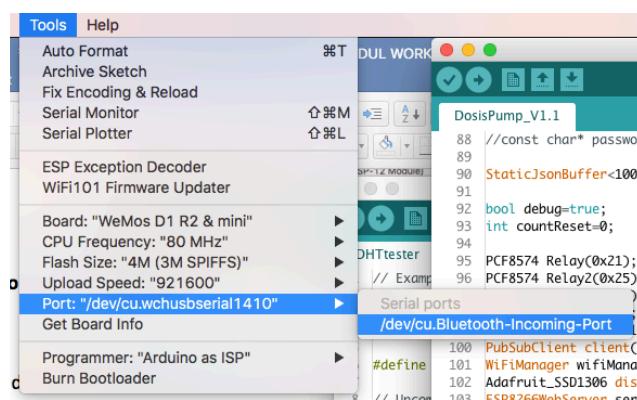
3. Pilih Tools -> Boards -> Boards Manager dan install esp8266



4. Pilih WEMOS D1 mini di menu Tools -> Board



5. Pilih Tools > Port > COMX.

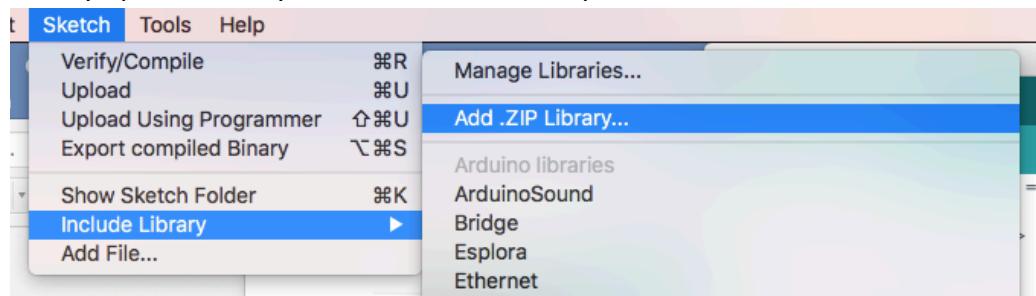


6. Tutup dan buka kembali Arduino IDE

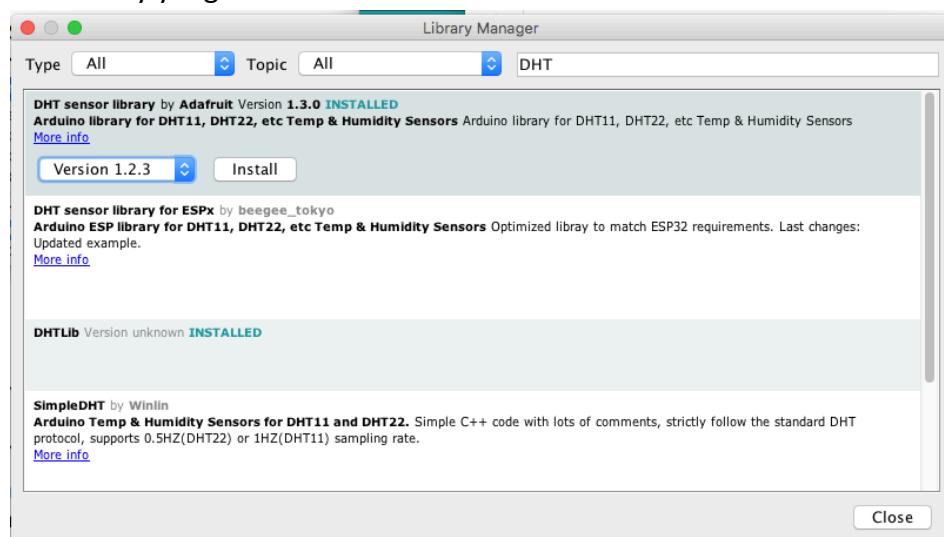
Install Library Pendukung

Library pendukung ini digunakan untuk mempermudah dalam membuat sebuah program. Di dalam library tersebut terdapat beberapa fungsi yang dapat digunakan untuk melakukan akses terhadap hardware atau software tertentu. Untuk menambahkan library pendukung ke Arduino IDE, dapat dilakukan dengan cara :

1. Pilih Sketch -> Include Library -> Manage Libraries, Atau dapat pilih Add .ZIP Library apabila library sudah ada dalam komputer

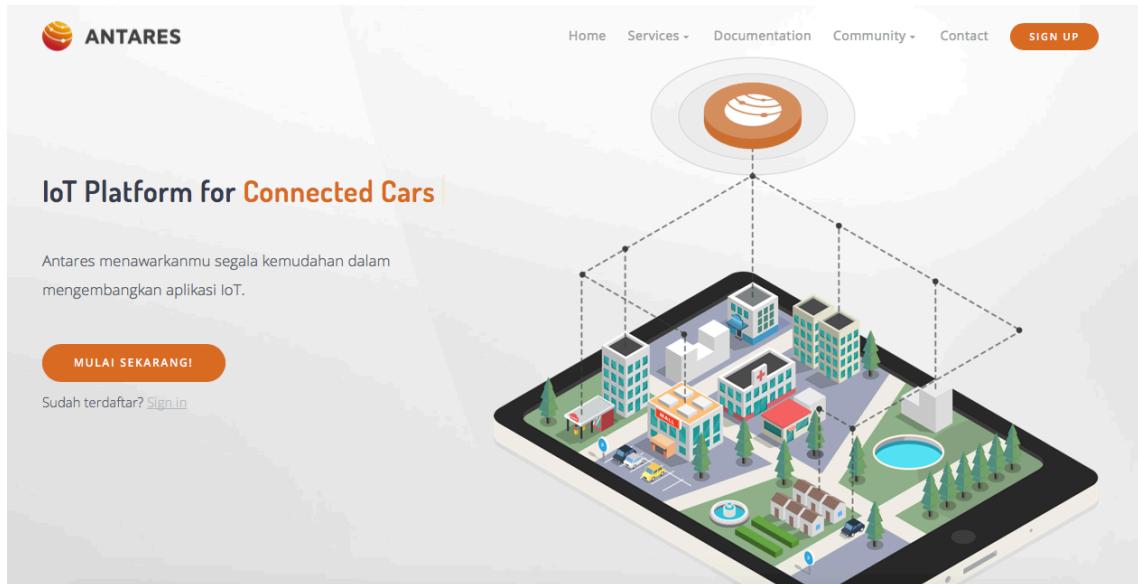


2. Cari library yang akan di install



Install Library Antares For ESP8266

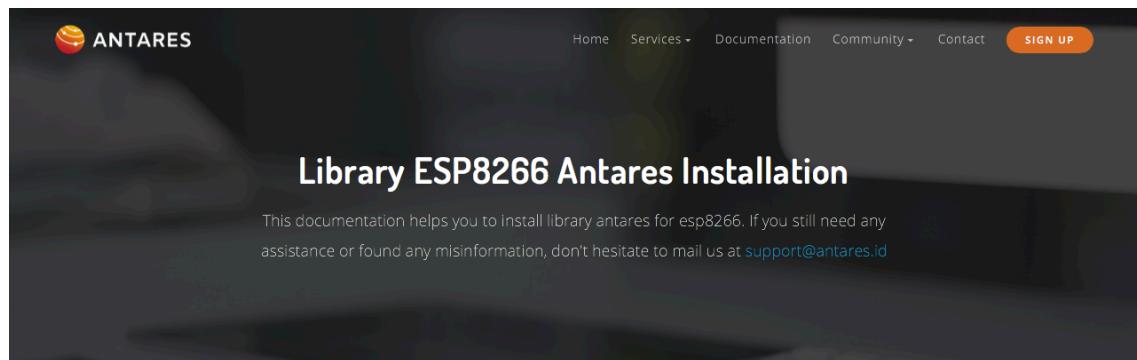
1. Masuk Ke Antares.id



2. Pilih Documentation

The image shows the 'Documentation' section of the Antares website. On the left, a sidebar contains a navigation menu with sections like 'Getting Started', 'Examples & Library', 'Data Model', 'Port Configuration', and 'API | MQTT'. Under 'Examples & Library', there are sub-options for 'ESP 8266', 'Arduino Uno', 'Android', 'Raspberry Pi', and 'Data Model'. To the right, there are four main documentation cards: 'ESP 8266' showing an ESP8266 module, 'ARDUINO UNO' showing an Arduino Uno board, 'ANDROID' showing the Android logo, and 'RASPBERRY PI' showing a Raspberry Pi board. Each card has a 'GO TO TUTORIAL' button.

3. Klik Go To Tutorial ESP8266



Steps

1. Install the [Arduino IDE](#), if you do not already have it.

4. Ikuti Instruksi hingga selesai.

Membuat Program Sederhana

Untuk membuat program langkah-langkah yang dilakukan yaitu :

- a. Buka aplikasi Arduino IDE
- b. Pilih File -> New
- c. Jendela editor akan terbuka

A screenshot of the Arduino IDE interface. The title bar says "sketch_mar20a | Arduino 1.8.1". The main area is a code editor with the following text:

```
sketch_mar20a
1void setup() {
2  // put your setup code here, to run once:
3
4}
5
6void loop() {
7  // put your main code here, to run repeatedly:
8
9}
```

The status bar at the bottom shows the text "WeMos D1 R2 & mini, 80 MHz, 921600, 4M (3M SPIFFS) on /dev/cu.wchusbserial1410".

- d. Setelah menuliskan baris program, langkah selanjutnya save program dan pilih icon 

1. Hardware Test Program

```
#include "DHT.h"

#define DHTTYPE    DHT11
#define SENSOR_DHT D2
#define LED_RED    D5
#define LED_BLUE   D0
#define LED_GREEN  D5
#define RELAY      D7

DHT dht(SENSOR_DHT, DHTTYPE);

void ledBlueOn()
{
    digitalWrite(LED_GREEN,HIGH);
    digitalWrite(LED_BLUE,LOW);
}

void ledBlueOff()
{
    digitalWrite(LED_GREEN,HIGH);
    digitalWrite(LED_BLUE,HIGH);
}

void ledGreenOn()
{
    digitalWrite(LED_BLUE,HIGH);
    digitalWrite(LED_GREEN,LOW);
}

void ledGreenOff()
{
    digitalWrite(LED_GREEN,HIGH);
    digitalWrite(LED_BLUE,HIGH);
}

void relayOn()
{
    digitalWrite(RELAY,HIGH);
```

```
}

void relayOff()
{
    digitalWrite(RELAY,LOW);
}

float getTemperature()
{
    float t = dht.readTemperature();
    if (isnan(t)) return 0;
    return t;
}

float getHumidity()
{
    float h = dht.readHumidity();
    if (isnan(h)) return 0;
    return h;
}

void setup() {
    Serial.begin(115200);
    pinMode(LED_RED,OUTPUT);
    pinMode(LED_GREEN,OUTPUT);
    pinMode(LED_BLUE,OUTPUT);

    digitalWrite(LED_RED,HIGH);
    digitalWrite(LED_GREEN,HIGH);
    digitalWrite(LED_BLUE,HIGH);

    pinMode(RELAY,OUTPUT);

    dht.begin();

    Serial.println("Test Hardware Workshop Antares ...");
}

void loop() {
    Serial.println("=====TESTING=====TESTING=====");
    Serial.println("Temperature : " + (String)getTemperature()+" *C");
    Serial.println("Humidity   : " + (String)getHumidity()+" %");
```

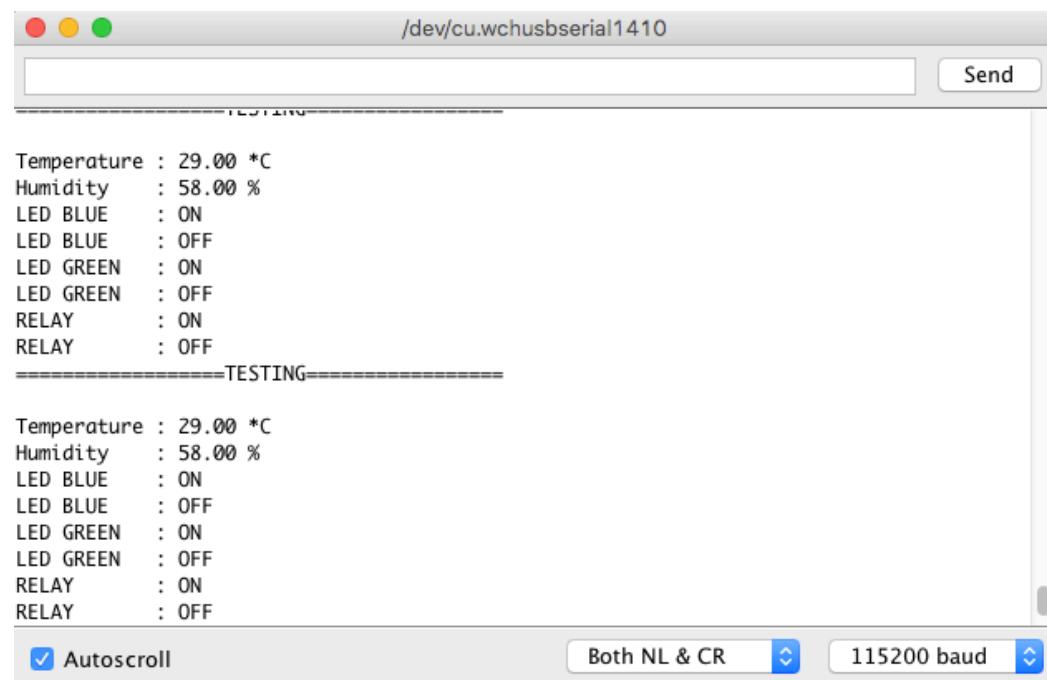
```

Serial.println("LED BLUE : ON");
ledBlueOn();
delay(1000);
Serial.println("LED BLUE : OFF");
ledBlueOff();
delay(1000);

Serial.println("LED GREEN : ON");
ledGreenOn();
delay(1000);
Serial.println("LED GREEN : OFF");
ledGreenOff();

Serial.println("RELAY : ON");
relayOn();
delay(1000);
Serial.println("RELAY : OFF\n");
relayOff();
delay(2000);
}

```



The screenshot shows a terminal window with the title bar reading "/dev/cu.wchusbserial1410". The window contains two sets of data output. The first set is environmental sensor data:

```

Temperature : 29.00 *C
Humidity    : 58.00 %
LED BLUE    : ON
LED BLUE    : OFF
LED GREEN   : ON
LED GREEN   : OFF
RELAY       : ON
RELAY       : OFF
=====TESTING=====

```

The second set is relay control output:

```

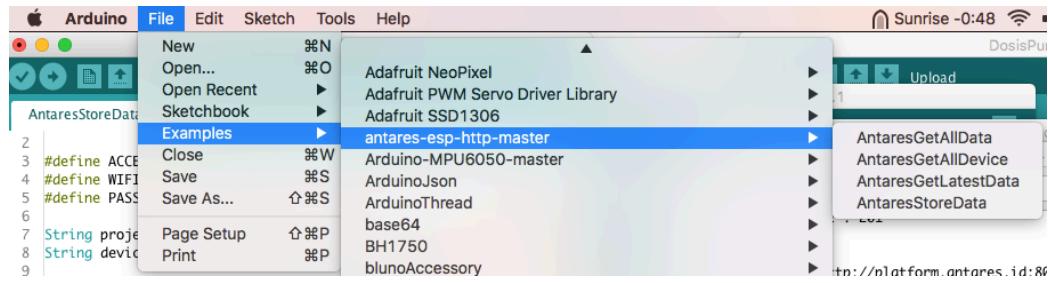
Temperature : 29.00 *C
Humidity    : 58.00 %
LED BLUE    : ON
LED BLUE    : OFF
LED GREEN   : ON
LED GREEN   : OFF
RELAY       : ON
RELAY       : OFF

```

At the bottom of the terminal window, there are several status indicators: a checked "Autoscroll" checkbox, a "Both NL & CR" dropdown, and a "115200 baud" dropdown.

2. Mengirim Data Ke Antares

- Buka source example di File -> Examples -> Antares-esp-http-master -> Antares Store Data



2. Isi AccessKey, SSID, Password, projectName, deviceName sesuai dengan user account Antares.

```
AntaresStoreData | Arduino 1.8.1

AntaresStoreData
1 #include "AntaresESPHTTP.h"
2
3 #define ACCESSKEY "your-access-key"
4 #define WIFISSID "your-wifi-ssid"
5 #define PASSWORD "your-wifi-password"
6
7 String projectName = "your-project-name";
8 String deviceName = "your-device-name";
9
10 Antares antares(ACCESSKEY);
11
12 void setup() {
13     Serial.begin(115200);
14     antares.setDebug(true);
15     antares.wifiConnection(WIFISSID,PASSWORD);
16 }
17
18 void loop() {
19     int sizeData = 2;
20     String dataName[sizeData],dataValue[sizeData];
21     dataName[0]= "Temperature";
22     dataName[1]= "Humidity";
23
24     dataValue[0]= (String)random(50,100);
25     dataValue[1]= (String)random(50,100);
26     Serial.println(antares.storeData(projectName, deviceName, dataName, dataValue,sizeData));
27     delay(10000);
28 }
29
30
31 }
```

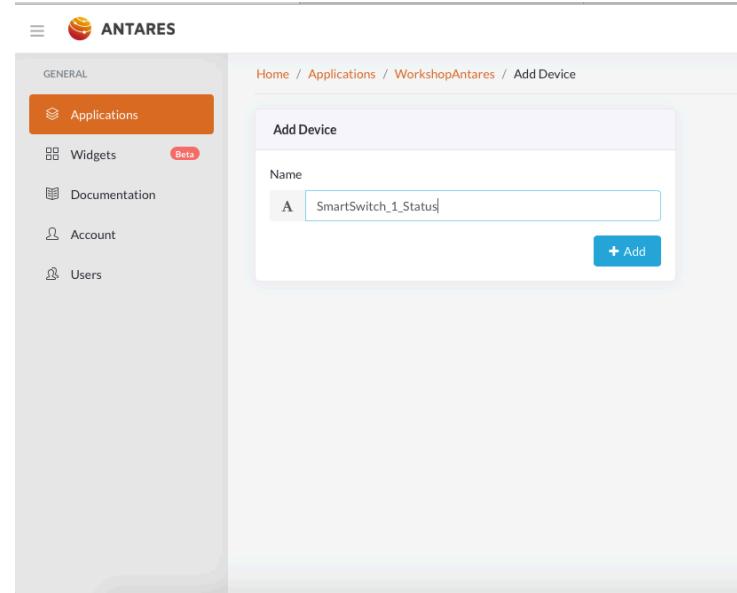
3. Mendapatkan AccessKey, Login di console.antares.id -> Account

The screenshot shows the Antares account management interface. On the left, a sidebar lists 'GENERAL' sections: Applications (selected), Widgets, Documentation, Account (highlighted in orange), and Users. The main content area shows a user profile picture of a man with glasses and a black shirt. Below the profile is a section titled 'Access Key (ID-Password)' containing a 'Key' field with the value 'e7e349fc2216941a:9d0cf82c25277bdd'. To the right is a 'Personal Data' form with fields for Name ('Antares Demo'), Email ('demo@antares.id'), Phone Number ('+62'), and Address ('Jl.'), with a 'Save' button at the bottom.

4. Membuat Application/Project

The screenshot shows the Antares application management interface. The left sidebar has 'GENERAL' sections: Applications (selected and highlighted in orange), Widgets (Beta), Documentation, Account, and Users. The main area is titled 'Add Application' and contains fields for 'Application Name' ('workshopAntares') and 'Application ID' ('workshopAntares1'). Below these are 'Labels' fields for 'Location' ('Indonesia') and a 'Key-value pairs' section ('Type - Sensor, Location - Home'). A '+ Add' button is at the bottom right. The top right corner shows the user profile 'Antares Demo'.

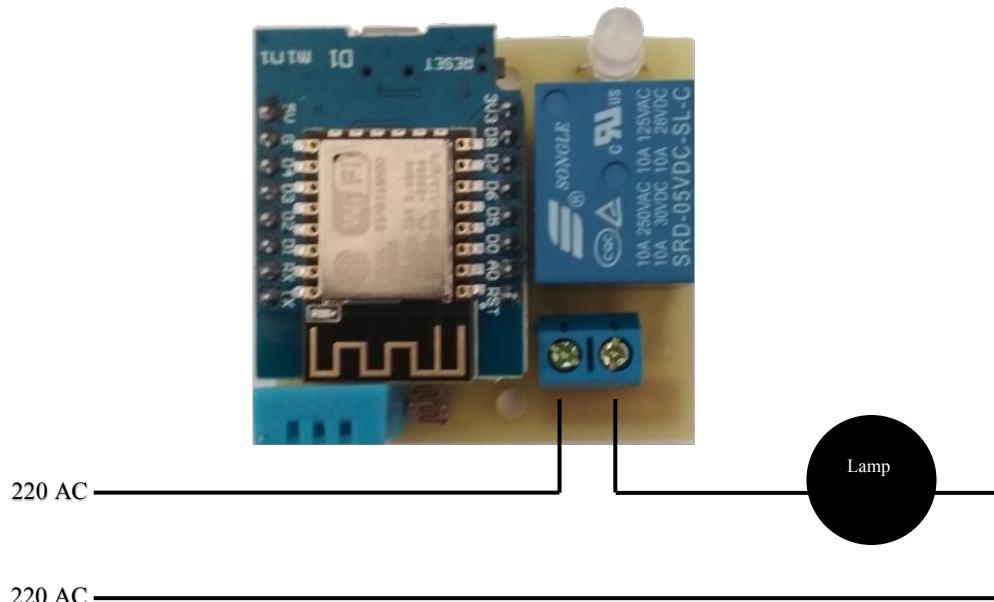
5. Membuat Device



The screenshot shows the 'WorkshopAntares' application interface. At the top, there is a header with the application name, a 'Subscribe' button, a 'User Access' button, and a trash icon. Below the header, there is a section titled 'Manage your application here. You can add/remove devices, subscribe to this application, manage user access key, etc.' On the right side, there are sections for 'Hierarchical URI' (with the URL 'https://platform.antares.id:8443/~/antares-cse/antares-id/WorkshopAntares') and 'Non-hierarchical URI' (with the URL 'https://platform.antares.id:8443/~/antares-cse/CAE814973361'). There are also icons for 'Type' (Sensor/Actuator), 'Category' (Temperature, Humidity, Intensity), and 'Location' (JDV). Below these sections, there is a link to 'How to Create Device via API'. At the bottom, there is a search bar with the placeholder 'Search Devices...' and a blue 'Add Device' button. Two device cards are visible: 'SmartSwitch_1_Status' and 'SmartSwitch_1_Control'.

3. Membuat Project SmartSwitch

Setelah membuat device **SmartSwitch_1_Status** dan **SmartSwitch_1_Control** di console Antares. Maka selanjutnya membuat wiring dan program di hardware.



Gambar Wiring Smart Switch Untuk Lampu

```

#include "AntaresESPHTTP.h"
#include <ArduinoJson.h>
#include "DHT.h"

#define DHTPIN D2
#define DHTTYPE DHT11
#define RELAY D7
#define LED_RED D6
#define LED_BLUE D0
#define LED_GREEN D5

DHT dht(DHTPIN, DHTTYPE);

#define ACCESSKEY "your-access-key"
#define WIFISSID "your-wifi-ssid"
#define PASSWORD "your-wifi-password"

#define URI_PROJECT_CONTROL "your-Non-hierarchical URI"

StaticJsonBuffer<10000> jsonBuffer;

String projectName = "WorkshopAntares";
String deviceNameStatus = "SmartSwitch_1_Status";
String deviceNameControl = "SmartSwitch_1_Control";

```

```
String Value,Unit,Label;

unsigned long intervalSend=10000; // the time we need to wait
unsigned long previousMillis=0;
int statusSwitch=0;

Antares antares(ACCESSKEY);

void ledBlueOn()
{
    digitalWrite(LED_GREEN,HIGH);
    digitalWrite(LED_BLUE,LOW);
    digitalWrite(LED_RED,HIGH);
}

void ledBlueOff()
{
    digitalWrite(LED_BLUE,HIGH);
}

void ledRedOn()
{
    digitalWrite(LED_GREEN,HIGH);
    digitalWrite(LED_BLUE,HIGH);
    digitalWrite(LED_RED,LOW);
}

void ledRedOff()
{
    digitalWrite(LED_RED,HIGH);
}

void ledGreenOn()
{
    digitalWrite(LED_BLUE,HIGH);
    digitalWrite(LED_GREEN,LOW);
    digitalWrite(LED_RED,HIGH);
}

void ledGreenOff()
{
    digitalWrite(LED_GREEN,HIGH);
}
```

```
void setup() {
    Serial.begin(115200);
    antares.setDebug(true);
    antares.wifiConnection(WIFISSID,PASSWORD);
    pinMode(RELAY,OUTPUT);
    pinMode(LED_BLUE,OUTPUT);
    pinMode(LED_RED,OUTPUT);
    pinMode(LED_GREEN,OUTPUT);
    dht.begin();
}

void loop() {
    unsigned long currentMillis = millis();
    if (antares.checkWifiConnection())
    {
        if ((unsigned long)(currentMillis - previousMillis) >= intervalSend) {
            float humidity = dht.readHumidity();
            float temperature = dht.readTemperature();

            int sizeData = 3;
            String dataName[sizeData],dataValue[sizeData];
            dataName[0]= "temperature";
            dataName[1]= "humidity";
            dataName[2]= "status";

            dataValue[0]= (String)humidity;
            dataValue[1]= (String)temperature;
            dataValue[2]= (String)statusSwitch;

            String responseCode = antares.storeData(projectName,
deviceNameStatus, dataName, dataValue,sizeData);

            if (responseCode=="201")
            {
                Serial.println("SENDING SUCCESS");
                ledGreenOn();
                delay(1000);
                ledGreenOff();
            }
        }
    }
}
```

```

        else
        {
            Serial.println("SENDING FAILED");
            ledRedOn();
            delay(1000);
            ledRedOff();
        }
        previousMillis = millis();
    }

    String dataFromAntares =
    antares.retrieveLatestData(projectName,deviceNameControl);

    if (dataFromAntares != "")
    {
        dataFromAntares.replace(" ","");
        dataFromAntares.replace("\n","");
        dataFromAntares.replace("\r","");
        dataFromAntares.replace("\\\\"," ");
        dataFromAntares.replace("[","");
        dataFromAntares.replace("]","");
        //Serial.println(dataFromAntares);
        JsonObject& root = jsonBuffer.parseObject(dataFromAntares);
        JsonObject& root4 =
        jsonBuffer.parseObject(root["m2m:cin"].as<String>());
        Label = root4["pi"].as<String>();

        String con = root4["con"].as<String>();
        con.replace(" ","\\\"");

        JsonObject& contentAntares = jsonBuffer.parseObject(con);
        String statusSwitchs = contentAntares["status"].as<String>();

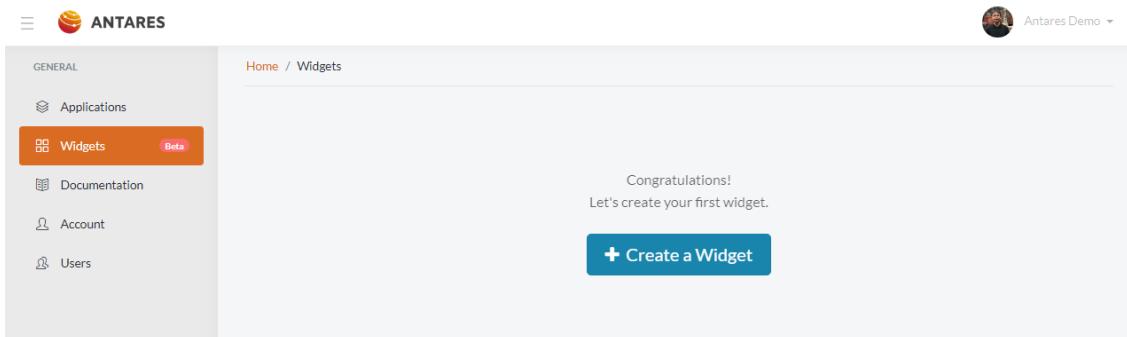
        if (Label == URI_PROJECT_CONTROL)
        {
            if (statusSwitchs == "1")
            {
                digitalWrite(RELAY,LOW);
                statusSwitch=0;
            }
            else
            {

```

```
    digitalWrite(RELAY,HIGH);
    statusSwitch=1;
}
}
jsonBuffer.clear();
dataFromAntares="";
}
}
delay(1000);
}
```

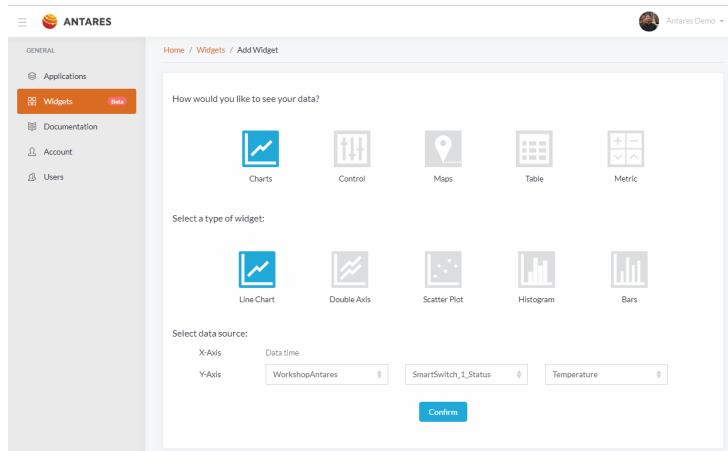
Membuat Widget

1. Pilih Menu Widget -> Create a Widget



The screenshot shows the ANTARES application interface. On the left, there's a sidebar with 'GENERAL' and 'Applications' sections. Under 'Applications', 'Widgets' is highlighted with a blue background and has a 'Beta' badge. Other options include 'Documentation', 'Account', and 'Users'. The main content area is titled 'Widgets' and shows the URL 'Home / Widgets'. It features a message 'Congratulations! Let's create your first widget.' and a large blue button with a plus sign and the text '+ Create a Widget'.

2. Pilih Jenis Widget



This screenshot shows the 'Add Widget' page. It has a sidebar with 'GENERAL' and 'Applications' sections, with 'Widgets' selected. The main area asks 'How would you like to see your data?' and lists five categories: Charts, Control, Maps, Table, and Metric. Below that, it says 'Select a type of widget:' and shows five sub-options: Line Chart, Double Axis, Scatter Plot, Histogram, and Bars. At the bottom, it says 'Select data source:' and shows dropdown menus for X-Axis (set to 'Data time'), Y-Axis (set to 'WorkshopAntares'), and three other fields ('SmartSwitch_1_Status' and 'Temperature'). A 'Confirm' button is at the bottom right.

3. Data akan tampil pada widget

