

# E-BOOK

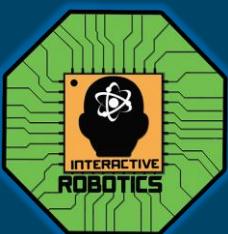
IOT  Blynk

MQTT + Android

WEB PHP + Mysql

ESP-32 DEVKIT

Created By: Iwan Muttaqin | Interactive Robotics





**Iwan Muttaqin**  
08988541995  
[iwancilibur@gmail.com](mailto:iwancilibur@gmail.com)

Community:  
-Interactive Robotics  
-AREI (Asosiasi Robotics Education Indonesia)

Founder at [inteknus.asia](http://inteknus.asia)

**Gaza Haikal Adzandani**  
089515958020  
[gazahaikal.2086@gmail.com](mailto:gazahaikal.2086@gmail.com)

**Social :**  
*IG/TW/FB /Github/Linkedin:*  
*@childlimbo*



Trainer at Interactive Robotics

# Control dan Monitoring Sensor Aktuator dengan Blynk Platform

No	Materi	Pengetahuan	Pencapaian	Metode
1	Pengenalan Aplikasi dan Microcontroller ✓ Arduino ✓ ESP32 Devboard	Memahami macam-macam microcontroller yang dapat digunakan dalam proyek pembuatan alat-alat robotika dan IOT	Dapat menentukan microcontroller yang cocok untuk digunakan pada project yang akan dibangun	Teori
2	ADC (Analog Digital Converter) Pembacaan Nilai Analog dan Digital pada Microcontroller ✓ Button (switch dan On/Off) ✓ Sensor LDR (Pendetksi Cahaya)	Mengerti penggunaan dari macam-macam sensor analog dan digital berikut cara merangkai dan memprogramnya	Dapat memanfaatkan sensor-sensor jenis analog dan digital dalam project elektronika yang ingin dibangun	Teori dan Praktikum
3	Mempelajari sensor-sensor Khusus ✓ DHT 11 (Temperature dan Humidity) ✓ Ultrasonic (Jarak)	Mengerti penggunaan dari macam-macam sensor khusus dan berikut cara merangkai dan memprogramnya.	Dapat memanfaatkan sensor-sensor jenis khusus dalam pembuatan project elektronika yang ingin dibangun	Teori dan Praktikum
4	Menghubungkan ke Aktuator dan Output ✓ Relay ✓ LCD 16 x 2 I2C (Kelas XII)	Mengerti penggunaan aktuator Relay sebagai penggerak objek tertentu seperti (Lampu, Motor dll.)	Dapat memanfaatkan macam-macam aktuator serta dapat menganalisa fungsionalitas dari jenis aktuator pada project yang akan dibangun.	Teori dan Praktikum
5	Komunikasi Jaringan ✓ Jenis-jenis Jaringan (Kabel & Wireless) ✓ Topologi Jaringan ✓ Modem, Access Point	Mengerti komunikasi jaringan dan device-device jaringan yang umum digunakan baik jaringan jenis kabel maupun wireless	Dapat memilih perangkat atau device yang digunakan untuk membangun project berdasarkan jenis jaringan yang akan digunakan.	Teori
6	Menghubungkan Microcontroller dengan Jaringan ✓ ESP32 Devboard (Komunikasi Wireless)	Mengerti cara menghubungkan antara microcontroller dengan jaringan wireless untuk berkomunikasi dengan server	Dapat menghubungkan device microcontroller dengan server menggunakan jenis jaringan wireless	Teori dan Praktikum
7	Mengirim dan Menerima data dari Microcontroller ke Server Blynk ✓ Blynk Publish dan Subscribe ✓ Logika Untuk Control Device Microcontroller ✓ Membangun Dashboard IOT di Blynk	Mengerti cara mengirim data dan menerima data dari server, serta mempelajari cara membuat dashboard IOT dengan Platform Blynk	Dapat memanfaatkan server sebagai penerima dan pengirim data antara device dan server serta menampilkan datanya di dashboard IOT	Teori dan Praktikum
8	Evaluasi			



# LANJUTAN

No	Materi	Pengetahuan	Pencapaian	Metode
1	<b>Mengirim dan Menerima data dari Microcontroller ke Server MQTT Broker</b> <ul style="list-style-type: none"> <li>✓ MQTT Publish dan Subscribe</li> <li>✓ Logika Untuk Control Device Microcontroller</li> <li>✓ Membangun Dashboard IOT di Android Kodular</li> <li>✓ Node RED (Dashboard)</li> </ul>	<ul style="list-style-type: none"> <li>✓ Mengerti cara mengirim data dan menerima data dari server, serta mempelajari cara membuat dashboard IOT dengan Android Kodular</li> </ul>	Dapat memanfaatkan server sebagai penerima dan pengirim data antara device dan server serta menampilkan datanya di dashboard IOT Android Kodular	Teori dan Praktikum
2	<b>Membuat Server Local</b> <ul style="list-style-type: none"> <li>✓ HTTP (XAMPP)</li> </ul>	<ul style="list-style-type: none"> <li>✓ Mengerti Aplikasi untuk membangun Server local yang umum digunakan pada pembuatan <i>project</i> alat-alat IOT</li> </ul>	Dapat membangun server local dan mengkonfigurasi agar server dapat digunakan sebagai pusat mengirim data dan mengambil data	Teori dan Praktikum
3	<b>Mengirim dan Menerima data dari Microcontroller ke Server Local</b> <ul style="list-style-type: none"> <li>✓ HTTP GET</li> <li>✓ Database Mysql dan PHP</li> </ul>	<ul style="list-style-type: none"> <li>✓ Mengerti cara mengirim data dan menerima data dari server, serta mempelajari database Mysql dan pemrogramman PHP</li> </ul>	Dapat memanfaatkan server sebagai penerima dan pengirim data antara device dan server serta menyimpan data ke database mysql dengan pemrogramman PHP	Teori dan Praktikum

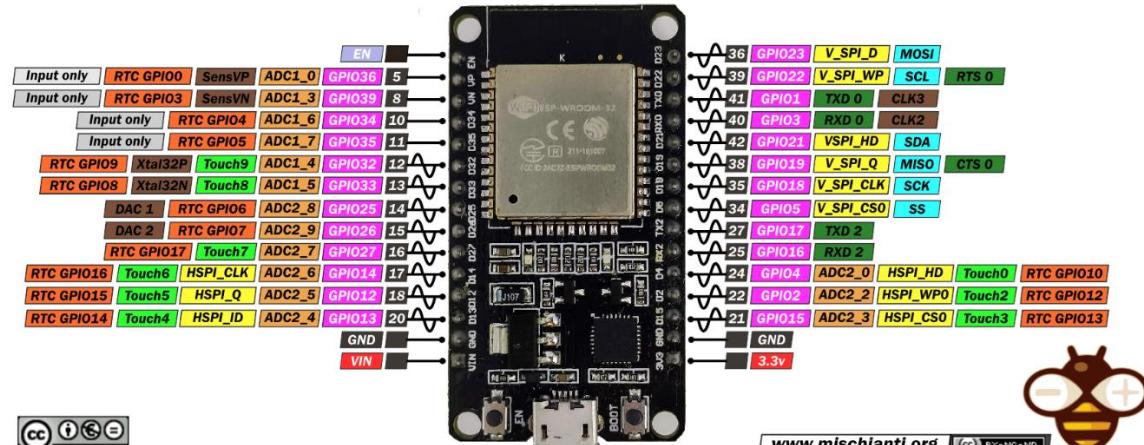
# Pengenalan Aplikasi dan Microcontroller

- Arduino
- ESP32 Devboard



Arduino sendiri merupakan perangkat yang bersifat **open source** dan sering digunakan untuk merancang serta membuat perangkat elektronik. Ditambah lagi **Software-nya** yang mudah digunakan dan membantu pekerjaan pengguna. Ia memiliki beberapa kegunaan, salah satunya adalah untuk mengembangkan perangkat yang dapat bekerja secara otomatis.

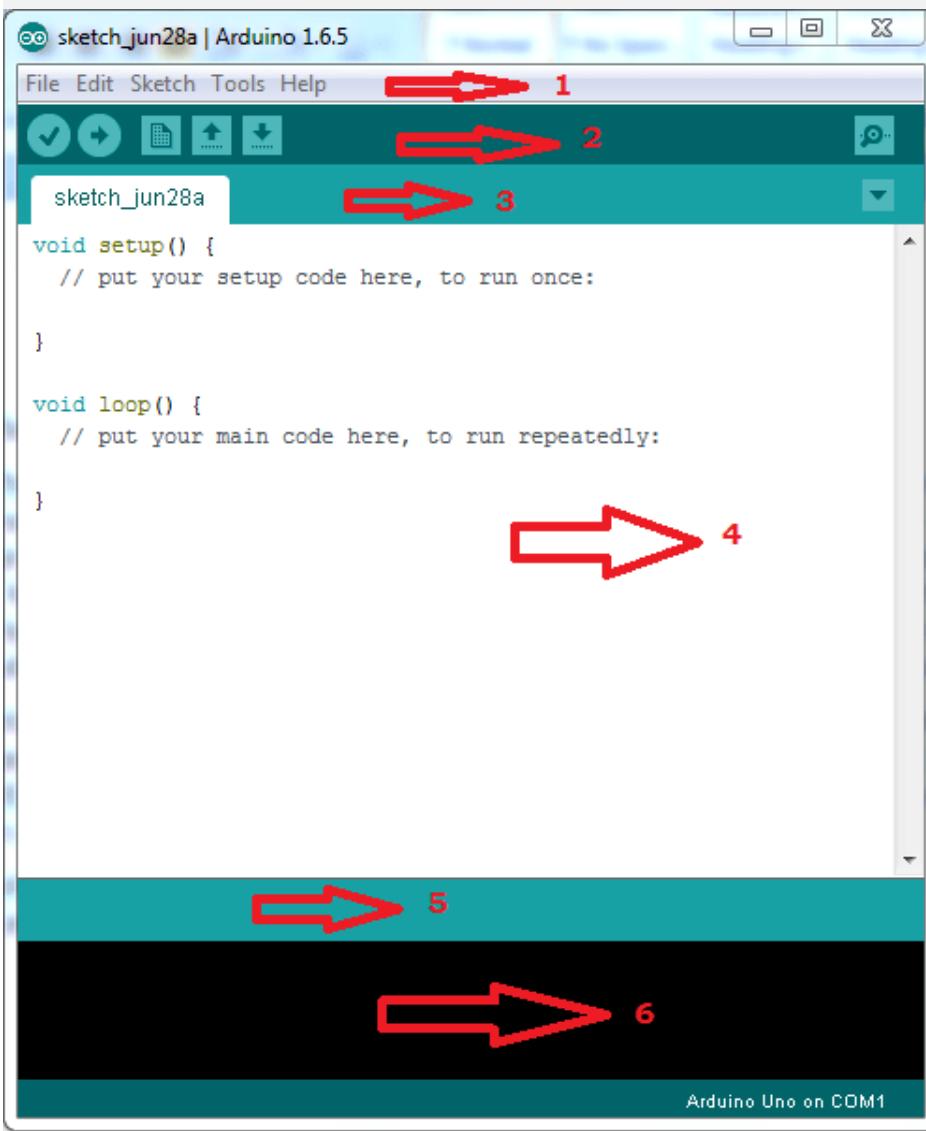
**ESP32 DEV KIT V1** | **PINOUT**



# Pengenalan Software Arduino

1

Download Software



Arduino IDE adalah software yang digunakan untuk membuat sketch pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman microcontroller.

**Berikut adalah penjelasan bagian-bagian IDE Arduino sesuai gambar di atas :**

**No 1 :** Menu IDE Arduino. Setidaknya ada 5 menu yang akan membantu Anda melakukan programming dengan aplikasi ini, yaitu '**File**', '**Edit**', '**Sketch**', '**Tools**' dan '**Help**'. Menu '**File**' digunakan untuk berinteraksi dengan file \*.ino (**new, open, save, print, contoh program, dll**). Menu '**Edit**' digunakan untuk mengedit program yang sedang ditulis di sketch editor, Menu '**Sketch**' untuk **mem-verify, meng-upload sketch**, Menu '**Tools**' untuk memanggil tools pendukung software Arduino ini dan Menu '**Help**' berisi bantuan/ catatan/ keterangan yang mungkin Anda butuhkan jika ada pertanyaan. Tiap menu ini memiliki submenu-submenu sendiri. Silahkan dieksplor untuk lebih detail masing-masing fungsinya.

**No 2 :** Kumpulan shortcut yang sering digunakan dalam proses pemrograman. Dari kiri ke kanan, shortcut tersebut adalah : Verify – Upload – New Sketch, Open, Save. Dan shortcut paling kanan adalah shortcut 'Serial Monitor', yakni jendela yang digunakan untuk memonitor data yang keluar masuk di port serial Arduino (diemulasi dalam port USB ke arah komputer).

**No 3 :** nama sketch

**No 4 :** Sketch editor. Di sinilah tempat Anda menulis program/sketch Arduino dalam bahasa C.

**No 5 :** Jendela informasi/notifikasi jika melakukan sesuatu di IDE ini

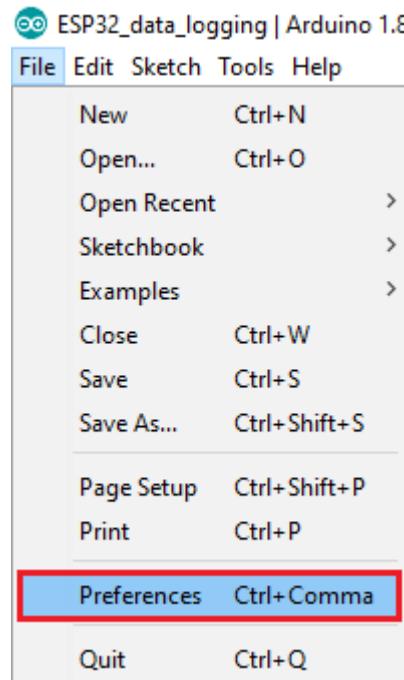
**No 6 :** Jendela debug. Jika ada kesalahan programming maka kesalahan tersebut akan muncul di jendela ini. Ini akan memudahkan Anda dalam mencari bug dalam program Anda



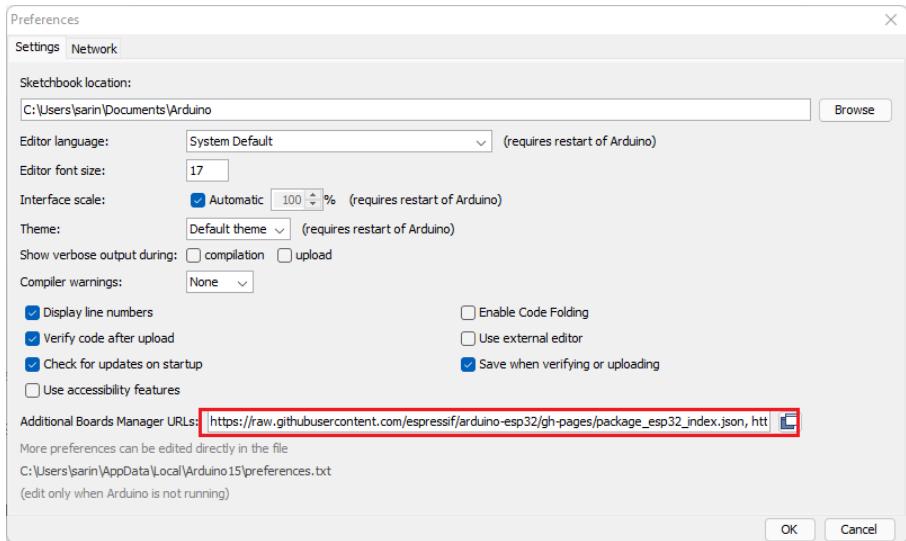
# Install ESP32 on Arduino IDE

1

Langkah 1



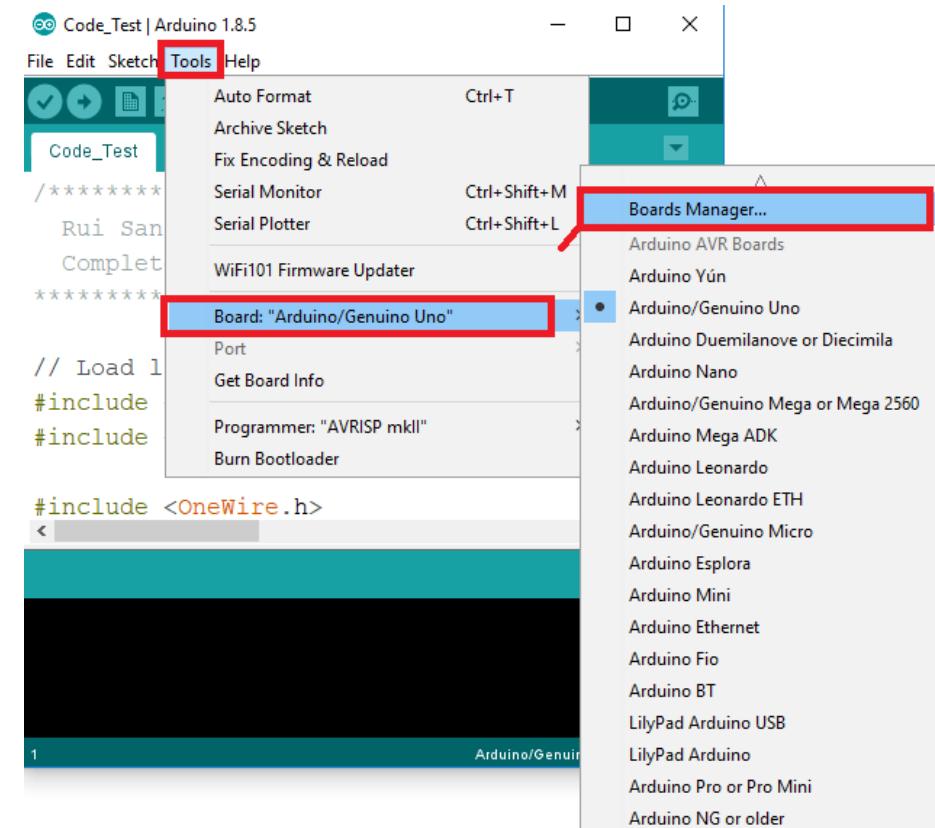
Langkah 2



Link:

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

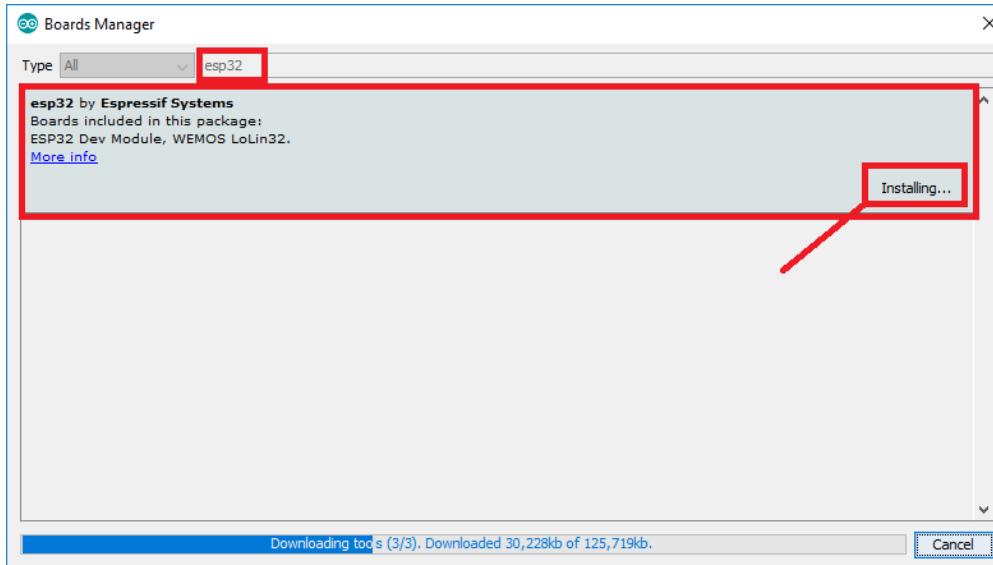
Langkah 3



# Install ESP32 on Arduino IDE

1

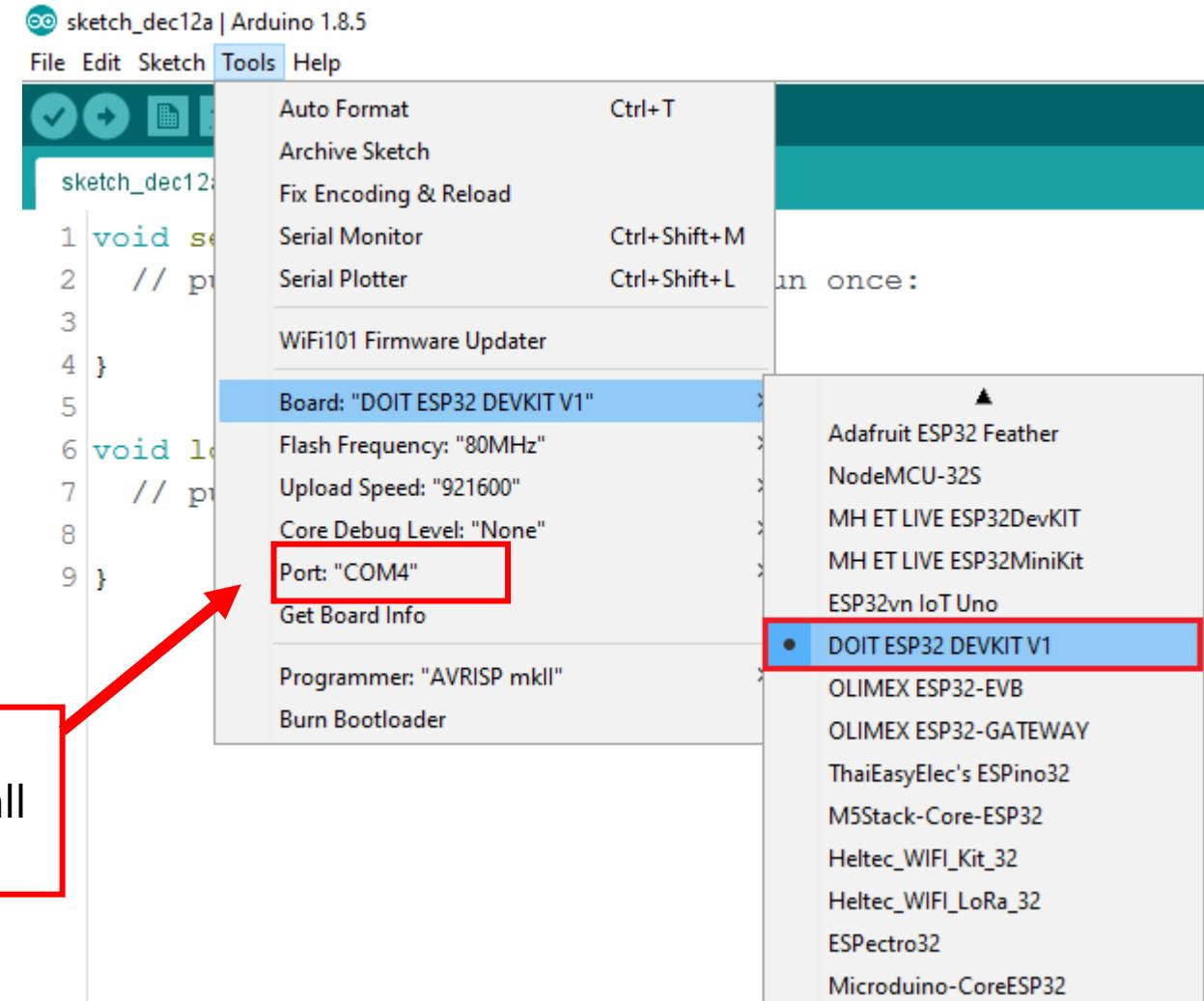
## Langkah 4



Untuk **COM4** disesuaikan dengan **COM** yang terdeteksi, Apabila tidak terdeteksi silahkan install Drivernya Terlebih dahulu

Download Driver

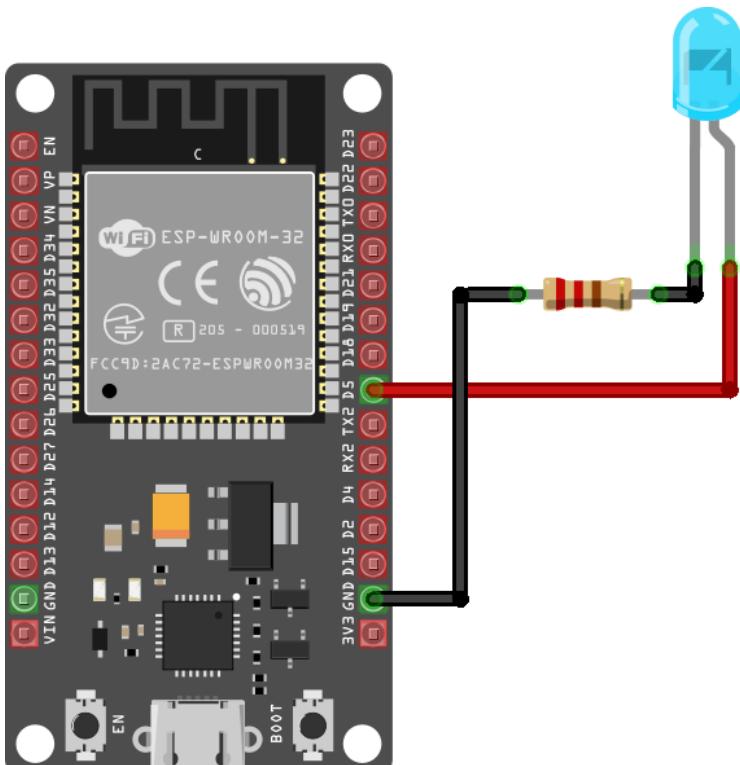
## Langkah 5



# "Hello Word / Blinking LED" pada ESP32 1



Apa itu Blinking LED, Membuat rangkaian LED agar LED dapat menyala selama 1 detik dan padam dalam 1 detik secara berulang, Untuk menampilkan Tulisan klik pada menu Serial Monitor.



```
#define LED 5

void setup() {
    Serial.begin(9600);
    pinMode(LED, OUTPUT);
}

void loop() {
    digitalWrite(LED, HIGH);
    delay(1000);
    digitalWrite(LED, LOW);
    delay(1000);
}
```

Done Saving.  
Sketch names must start with a letter or number, followed by numbers, dashes, dots and underscores. Maximum length

```
#define LED 5

void setup() {
    Serial.begin(9600);
    pinMode(LED, OUTPUT);
}

void loop() {
    Serial.println("LED NYALA");
    digitalWrite(LED, HIGH);
    delay(1000);
    Serial.println("LED MATI");
    digitalWrite(LED, LOW);
    delay(1000);
}
```

Invalid library found in C:\Users\iwan  
Invalid version '' for library in: C:\

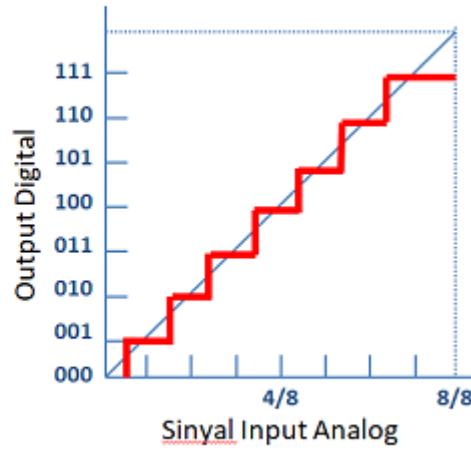
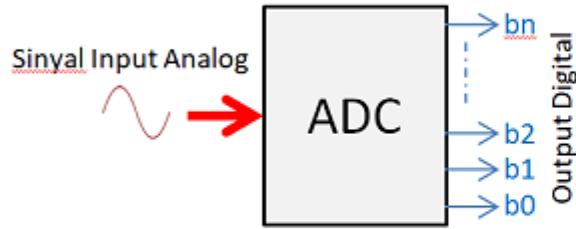
7 DOIT ESP32 DEVKIT V1, 80MHz, 921600, None, Disabled on COM12

12 DOIT ESP32 DEVKIT V1, 80MHz, 921600, None, Disabled on COM12



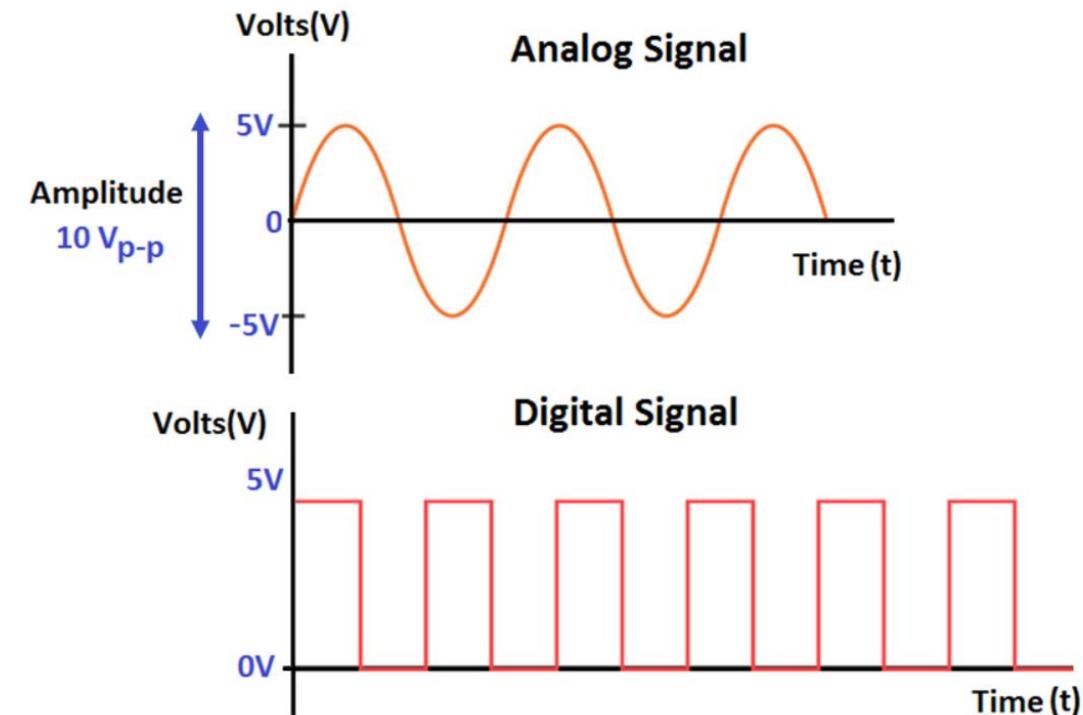
# ADC (Analog Digital Converter) Pembacaan Nilai Analog dan Digital pada Microcontroller

- Button (switch dan On/Off)
- Sensor LDR (Pendetksi Cahaya)



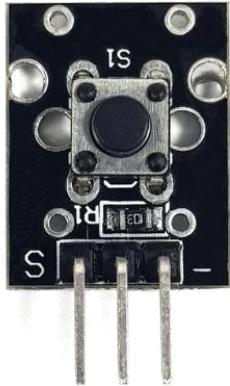
Apa yang dimaksud Analog To Digital Converter?

**ADC** adalah kepanjangan dari Analog To Digital Converter yang berfungsi untuk mengubah input analog menjadi kode – kode digital. ADC banyak digunakan sebagai Pengatur proses industri, komunikasi digital dan rangkaian pengukuran/atau pengujian.



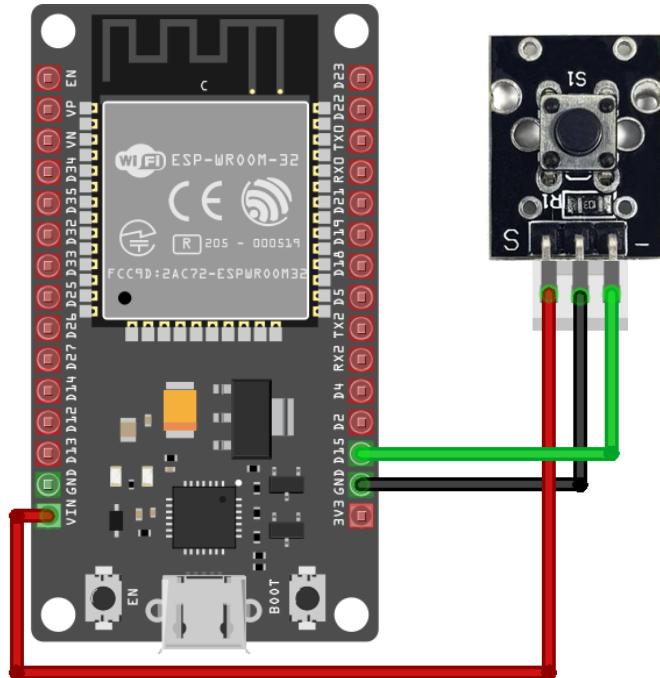
# Button (switch dan On/Off)

2



Pin 1	5V
Pin 2	GND
Pin 3	DATA

Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan unlock (tidak mengunci).



```
2_Button | Arduino 1.8.19
File Edit Sketch Tools Help
2_Button $ 
#define BUTTON 15

void setup() {
  Serial.begin(9600);
  pinMode (BUTTON, INPUT);
}

void loop() {
  int BacaButton = digitalRead(BUTTON);
  Serial.println(BacaButton);
  delay(100);
}
```

```
2_Button | Arduino 1.8.19
File Edit Sketch Tools Help
2_Button 
#define BUTTON 15

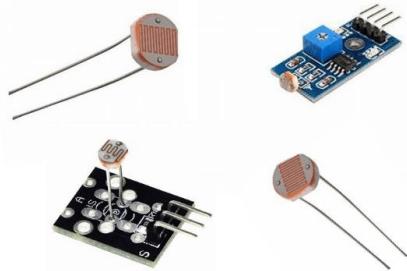
void setup() {
  Serial.begin(9600);
  pinMode (BUTTON, INPUT);
}

void loop() {
  int BacaButton = digitalRead(BUTTON);
  Serial.println(BacaButton);

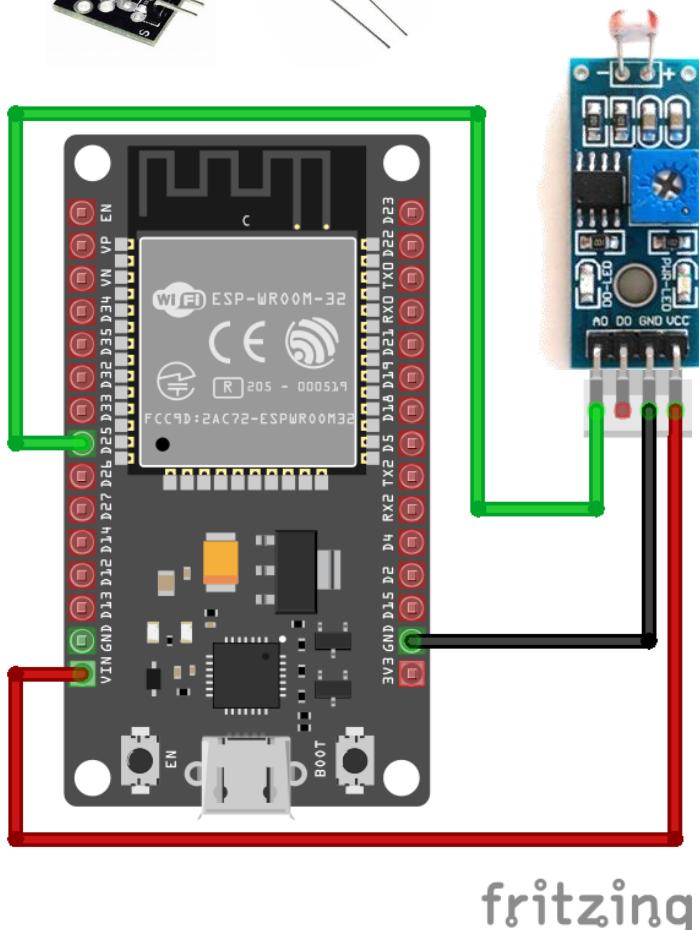
  if(BacaButton == 0){
    Serial.println("Button Ditekan!");
  }else{
    Serial.println("Button Dilepas!");
  }
  delay(100);
}
```

# Sensor LDR (Pendeteksi Cahaya)

2



**Light Dependent Resistor** atau yang biasa disebut LDR adalah jenis resistor yang nilainya berubah seiring intensitas cahaya yang diterima oleh komponen tersebut. Biasa digunakan sebagai detektor cahaya atau pengukur besaran konversi cahaya



```
#define LDR 25

void setup() {
    Serial.begin(9600);
    pinMode (LDR, INPUT);
}

void loop() {
    int BacaLDR = analogRead(LDR);
    Serial.println(BacaLDR);
    delay(100);
}
```

Save Canceled.  
Invalid library found in C:\Users\iwancilil...  
Invalid version '' for library in: C:\User...

## **Chalange:**

Jika Sensor LDR mendeteksi adanya Cahaya maka Lampu LED Mati dan Ketika Gelap LED Menyal



# 3

## Mempelajari sensor-sensor Khusus

- DHT 11 (Temperature dan Humidity)
- Ultrasonic (Jarak)

Apa yang dimaksud dengan sensor khusus, Sensor khusus merupakan sensor yang membutuhkan nilai pasti atau nilai yang sudah ditentukan satunya, Seperti Jarak dengan satuan Centi Meter (cm), Meter (m) dan Kilo Meter (Km), atau yang lainnya seperti Suhu (Celcius) dan masih banyak lainnya, dan biasanya sensor ini sudah dibuatkan library oleh produsen pembuat sensor tersebut.

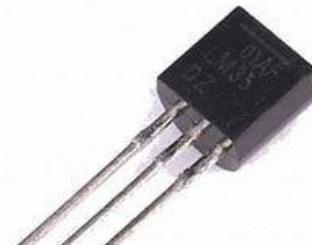


**DHT11**

**Temp + Humidity**



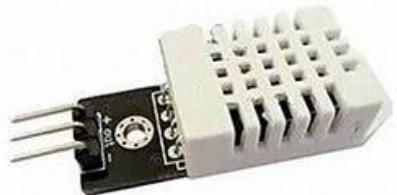
**Thermistors**



**LM35**



**DS18b20 one-wire  
Digital Temperature  
Sensor**



**DHT22/AM2302**



**DS18S20**

**MAX6675 + K type Thermocouple  
Temperature Sensor**

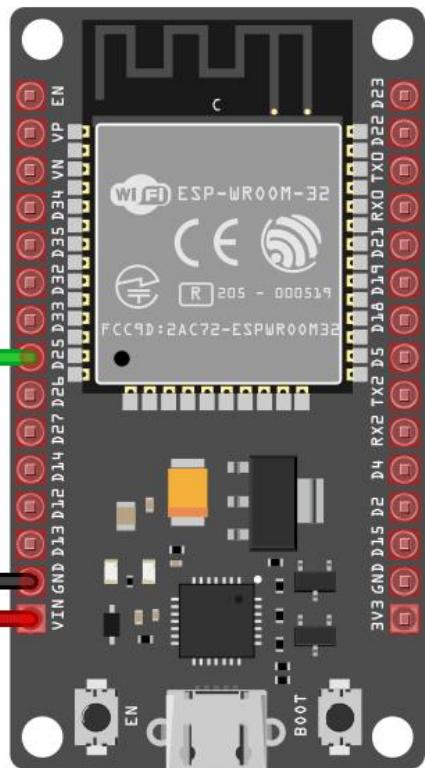
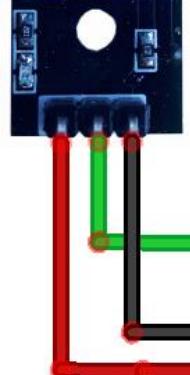
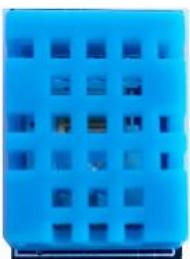
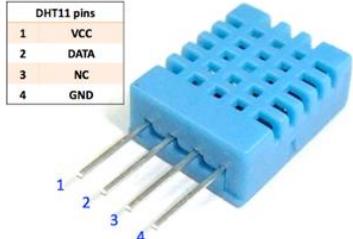


**MLX90614  
Infrared Temperature  
Sensor**



# DHT 11 (Temperature dan Humidity)

3



fritzing

Sensor DHT11 adalah sensor dengan kalibrasi sinyal digital yang mampu memberikan informasi suhu dan kelembaban udara. Sensor ini tergolong komponen yang memiliki tingkat stabilitas yang sangat baik serta fitur kalibrasi yang sangat akurat.

```
4_DHT11 | Arduino 1.8.19
File Edit Sketch Tools Help
4_DHT11
#include <DHT.h>

#define DHTPIN 25
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

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

void loop() {
  float suhu = dht.readTemperature();
  float kelembaban = dht.readHumidity();

  //POST DATA SERIAL
  Serial.print("SUHU      :"); Serial.print(suhu); Se
  Serial.print("KELEMBABAN :"); Serial.print(kelembaban); Se
  Serial.println();
  delay(1000);
}
```

## Instal Library

- DHT
- Adafruit Unified Sensor

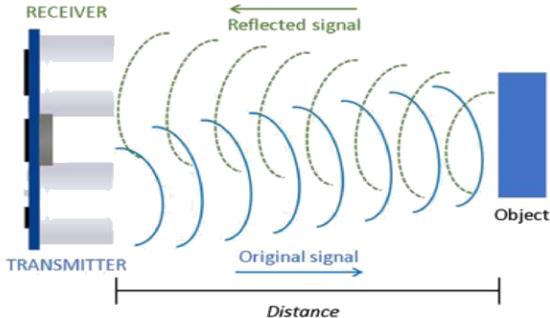
## Chalange

Jika Suhu Lebih dari **31 C**,  
Maka Muncul Pesan "**Suhu Mulai Panas**", Jika kurang  
dari Pesan "**Normal**"

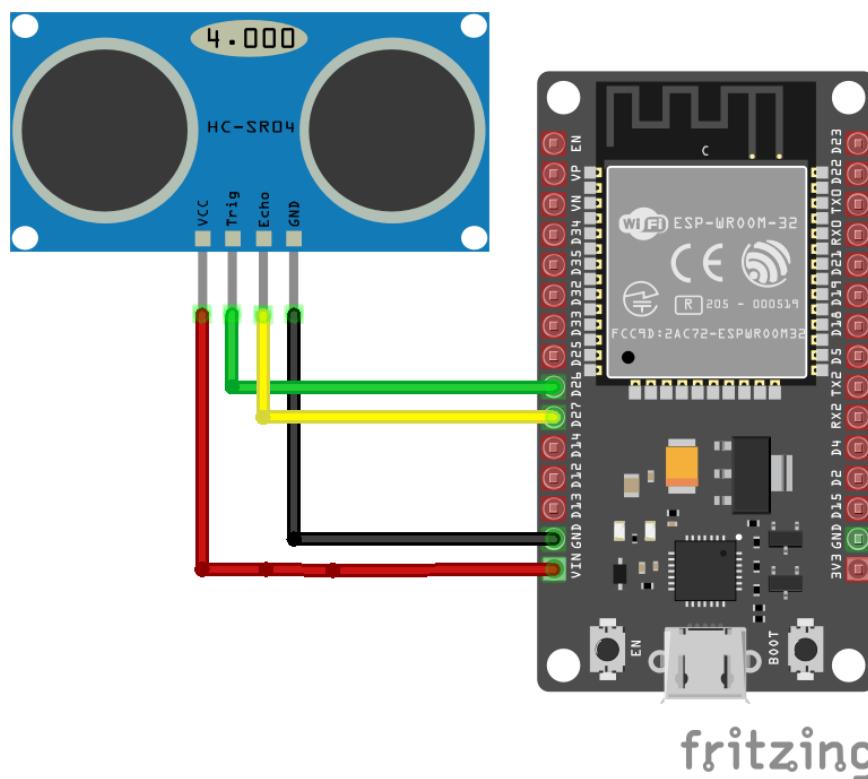


# Ultrasonic HCSR-04 (Jarak)

3



Sensor ultrasonik adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu.



```
5_Ultrasonic | Arduino 1.8.19
File Edit Sketch Tools Help
5_Ultrasonic §
#include <NewPing.h>

#define TRIGGER_PIN 26
#define ECHO_PIN    27
#define MAX_DISTANCE 100 // Maximum distance 400-500cm.

NewPing cm(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

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

void loop() {
  int getjarak=cm.ping_cm();
  Serial.print("Jarak :"); Serial.print(getjarak); Serial.print(" cm");
  Serial.println();
  delay(200);
}
```

**Instal Library**  
-NewPing

## Chalange

Jarak >30 maka "AMAN!"  
Jarak >10 dan <30 "Hati-Hati!"  
Jarak <10 maka "Berhenti!"



# Menghubungkan ke Output Display dan Aktuator

-LCD 16 x 2 I2C (Kelas XII)  
-Relay



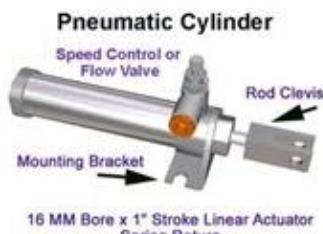
Relay



Motor DC  
listrik



piston



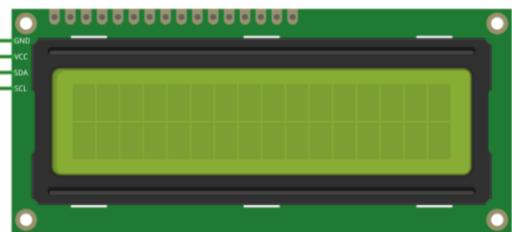
Silinder  
pneumatik



solenoid



motor  
stepper



Aktuator adalah sebuah peralatan mekanis untuk menggerakkan atau mengontrol sebuah mekanisme atau sistem. Aktuator diaktifkan dengan menggunakan mekanisme dan elektrikal biasanya digerakkan oleh signal pada tegangan 0-5 Volt, yang dikendalikan oleh media pengontrol otomatis yang terprogram di antaranya mikrokontroler.

## Contoh Akuator :

Relay, SSR (Solid State Relay), Mosfet, Motor Listrik dll.

## Contoh Output :

LCD 16 x 2

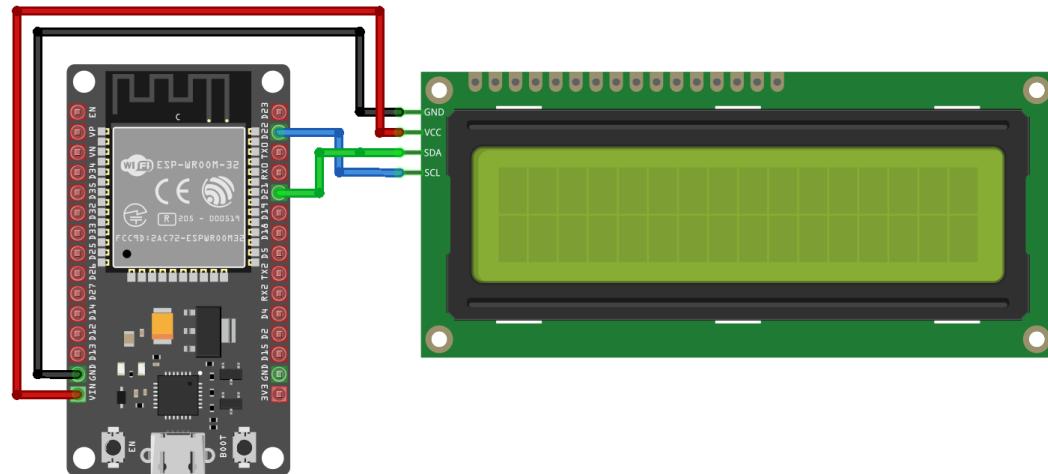
OLED

Thermal Printer dll.

# LCD 16 x 2 I2C

4

LCD adalah media tampilan yang paling mudah untuk diamati karena menghasilkan tampilan karakter yang baik dan cukup banyak. Pada LCD 16x2 dapat ditampilkan 32 karakter, 16 karakter pada baris atas dan 16 karakter pada baris bawah



## Instal Library

-LiquidCrystal\_I2C

## Chalange

-Munculkan Nama anda di LCD

-Munculkan Data sensor Suhu dan Kelambaban

```
6_LCD_16x2_I2c | Arduino 1.8.19
File Edit Sketch Tools Help
6_LCD_16x2_I2c
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

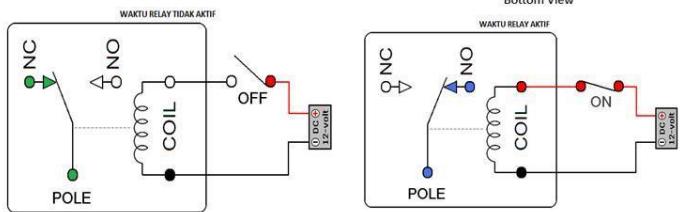
void setup() {
  Serial.begin(115200);
  lcd.begin();
}

void loop() {
  lcd.clear();
  lcd.setCursor(0,0); lcd.print("HALLO ^_^");
  lcd.setCursor(0,1);lcd.print("IWAN CILIBUR");
  delay(1000);

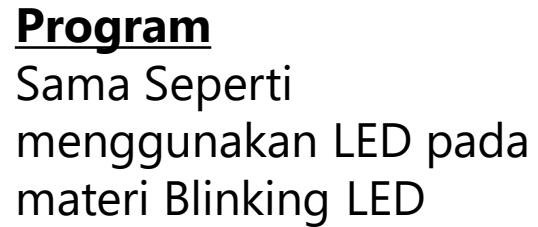
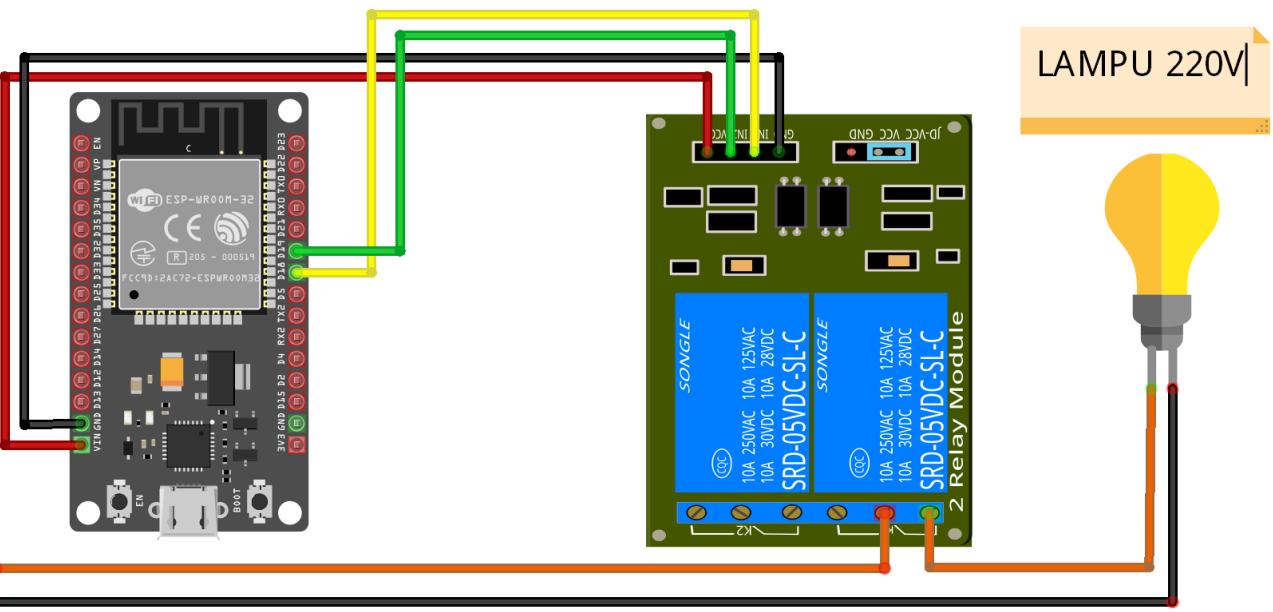
  lcd.clear();
  lcd.setCursor(0,0); lcd.print("HALLO ^_^");
  lcd.setCursor(0,1);lcd.print("ROBOTIKA");
  delay(1000);
}
```

# Relay dan Lampu 220 v

4

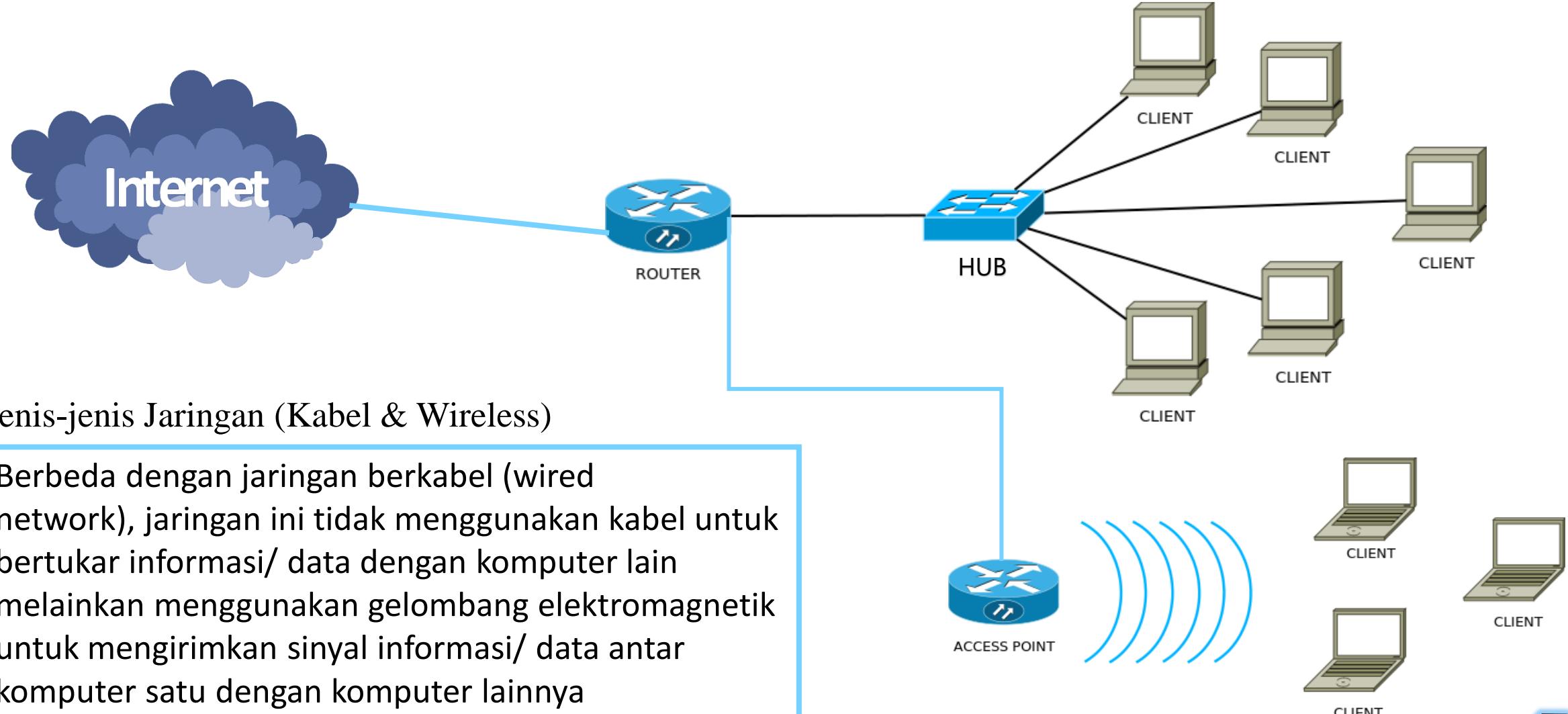


Relay biasanya digunakan untuk menggerakkan arus/tegangan yang besar (misalnya peralatan listrik 4-10 A/AC 220V) dengan memakai arus/tegangan yang kecil (misalnya 0.1 A/5 volt DC). Relay adalah komponen listrik yang bekerja berdasarkan prinsip induksi medan elektromagnetik.



# Komunikasi Jaringan

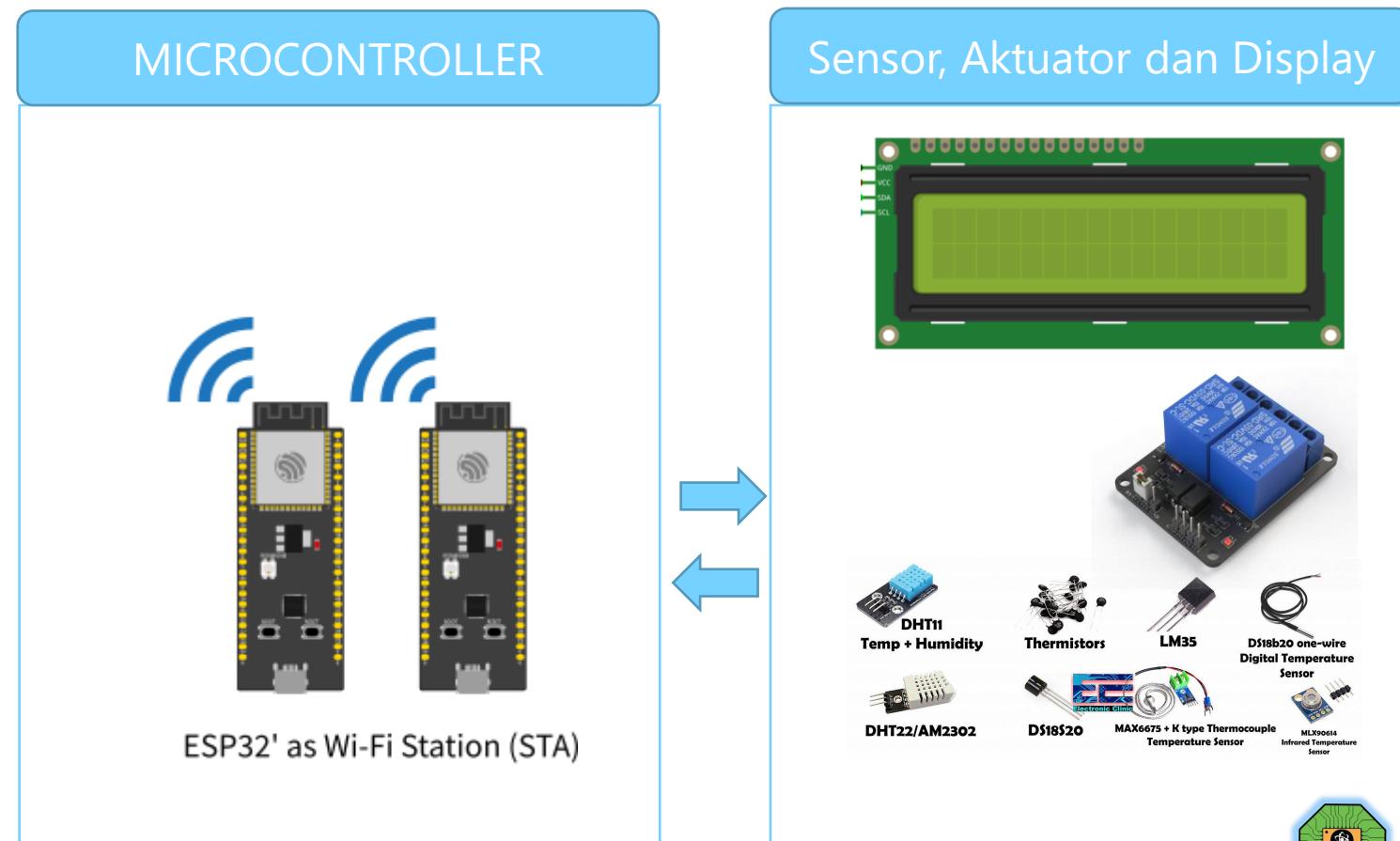
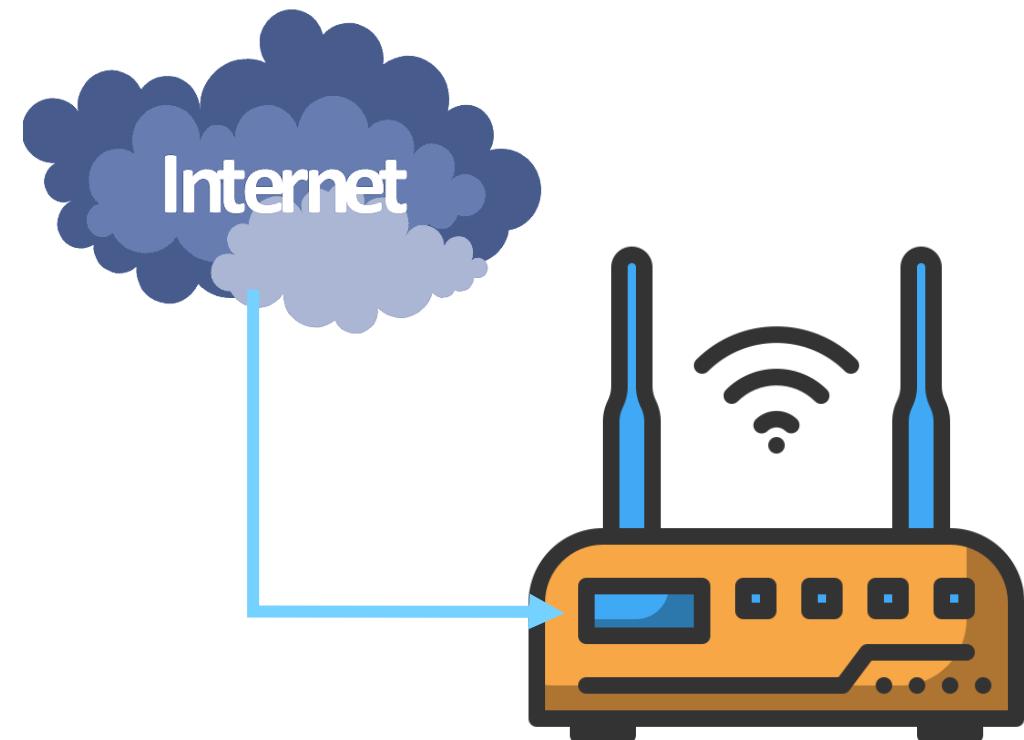
- Jenis-jenis Jaringan (Kabel & Wireless)
- Topologi Jaringan
- Modem, Access Point



# Topologi Jaringan Modem dan Acces Point 5



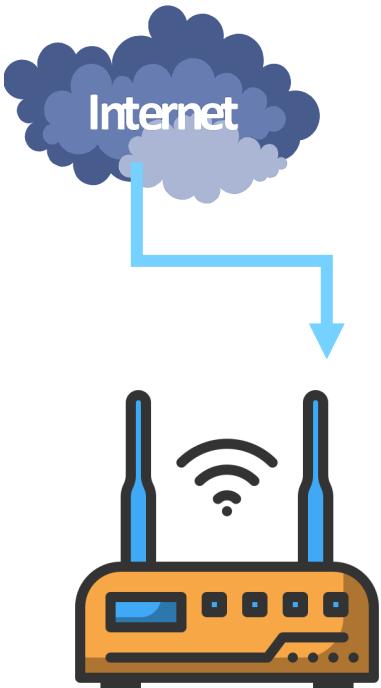
Apa itu Blinking LED, Membuat rangkaian LED agar LED dapat menyala selama 1 detik dan padam dalam 1 detik secara berulang.



# 6

## Menghubungkan Microcontroller dengan Jaringan

-ESP32 Devboard (Komunikasi Wireless)



A screenshot of the Arduino IDE interface. The title bar says "7\_Test\_Koneksi\_WIFI\_to\_Web | Arduino 1.8.19". The code window contains the following C++ code:

```
#include <WiFi.h>
#include <HTTPClient.h>

// WiFi Parameters
const char* ssid = "Mamangcoffee";
const char* password = "mamangkopi2021";

void setup() {
  Serial.begin(9600);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting...");
  }
  Serial.println("Connected!");
}

void loop() {
  // Check WiFi Status
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http; //Object of class HTTPClient
    http.begin("https://test.inteknus.asia/index.php");
    int httpCode = http.GET();
  }
}
```

### Program

Silahkan Buka **Program 7**  
***Test Koneksi Wifi to Web***

Apabila Balasanya seperti dibawah berarti  
sudah terhubung dengan Jaringan!

### Note:

**SSID** : isi dengan SSID Access Point

**PASS**: isi dengan Password Access Point

A screenshot of the Serial Monitor window titled "COM12". The window shows a series of text messages in a monospaced font. Each message consists of a timestamp followed by a right-pointing arrow, the text "Jawaban: Web Interactive Robotics by inteknus.asia", and a timestamp at the end of the line. There are five such entries, spaced evenly apart.

Timestamp	Message	Timestamp
13:36:46.412	>> Jawaban: Web Interactive Robotics by inteknus.asia	Time: 2024-02-22 13:36:45
13:36:50.647	>> Jawaban: Web Interactive Robotics by inteknus.asia	Time: 2024-02-22 13:36:49
13:36:52.878	>> Jawaban: Web Interactive Robotics by inteknus.asia	Time: 2024-02-22 13:36:52
13:36:55.030	>> Jawaban: Web Interactive Robotics by inteknus.asia	Time: 2024-02-22 13:36:54
13:36:57.229	>> Jawaban: Web Interactive Robotics by inteknus.asia	Time: 2024-02-22 13:36:56

## Mengirim dan Menerima data dari Microcontroller ke Server Blynk

- Blynk Publish dan Subscribe
- Logika Untuk Control Device Microcontroller
- Membangun Dashboard IOT di Blynk

Blynk merupakan sebuah platform Internet of Things (IoT) yang dapat digunakan untuk menghubungkan perangkat keras IoT dengan sebuah platform IoT. Dengan menggunakan platform ini kita dapat mengontrol dan memonitor perangkat keras dari jarak jauh.

The screenshot shows the Blynk website homepage. At the top, there is a navigation bar with links: Blynk (highlighted in green), FEATURES ▾, PRICING, BUSINESSES, DEVELOPERS, INDUSTRIES ▾, and CASE STUDIES. To the right of the navigation bar is the URL <https://blynk.io>. The main headline reads "We make internet of things simple for your company". Below the headline, there is a descriptive text: "Build and maintain smart products with a unified IoT platform — secure cloud, firmware libraries, and an intuitive app builder, all from one central console." At the bottom left, there is a red-outlined button labeled "START FREE". Next to it is another button labeled "BUSINESS SOLUTION". In the bottom right corner, there is a small logo for "INTERACTIVE ROBOTICS".

# Blynk Publish dan Subscribe

7

## Register dan Aktivasi

1 

### Sign Up

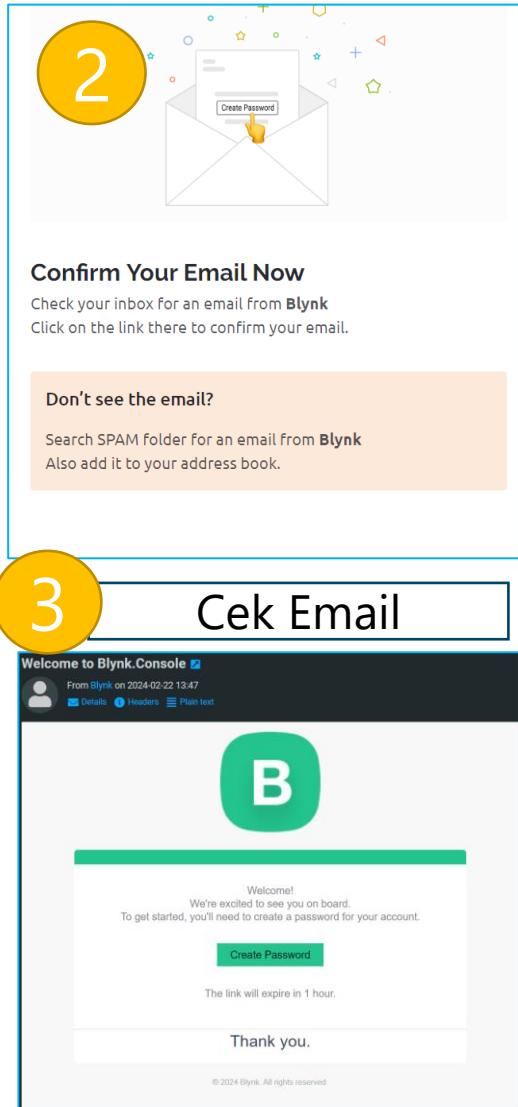
Welcome! Fill in your email address and we will send an account activation link.

EMAIL

I agree to [Terms and Conditions](#) and accept [Privacy Policy](#)

**Sign Up**

[Back to Login](#)



4 

### Create Password

Create a password which is hard to guess.

PASSWORD  

Good

- Make it at least 8 symbols long
- Other tips:
  - Use uncommon words
  - Use non-standard uPPercasing
  - Use creative spelling
  - Use non-obvious numbers & symbols

**Log In** **Next**

5 

### Profile

Fill in your personal data

FIRST NAME

[Back to password creation](#) **Done**

Isikan Nama Profile dengan nama Kamu!



# Blynk Publish dan Subscribe

7

## Konfigurasi Device Board ESP32 dan *Install Library Blynk*

1

Quickstart

1 Hardware    2 IDE    3 Blynk

Which hardware are you using?

We will help you prepare the code for your board

ESP32

What is your device connectivity?

Blynk supports various connection types

WiFi

Quickstart

1 Hardware    2 IDE    3 Blynk

Which IDE do you use?

Arduino    PlatformIO    Other

Download →    Download →   

2

Quickstart

1 Hardware    2 IDE    3 Blynk

Install Blynk Library for Arduino

1. Go to Arduino -Tools - Manage Libraries...  
2. Search for Blynk there.  
3. Choose the latest version and press Install.

3

Cancel    Next →

# Blynk Publish dan Subscribe

7

## Mengambil Coding dari Blynk untuk Microcontroller ESP32

### Quickstart

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.

Mamangcoffee  
mamangkopi2021

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

5. Upload sketch to your board.

You can find other examples for your board [here](#).

Quickstart

Hardware    IDE    Blynk Library    4 Code    5 Device activation

**Here is a code for your device**

```
/*
 * This is a simple demo of sending and receiving some data.
 * Be sure to check out other examples!
 */

/* Fill-in information from Blynk Device Info here */
#define BLYNK_TEMPLATE_ID          "TMPL6A7LVJJv7"
#define BLYNK_TEMPLATE_NAME         "Quickstart Template"
#define BLYNK_AUTH_TOKEN            "COT86q6M_hNAtIR18uC5inYZcuCmgU1P"

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

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Mamangcoffee";
char pass[] = "mamangkopi2021";
```

Copy    Download

← Back    Cancel    Next →

Silahkan isikan SSID dan Passwrodnnya, Setelah itu copy Coding yang ada di Dashboard Tersebut!



# Blynk Publish dan Subscribe

7

## Menghubungkan Microcontroller Ke Blynk

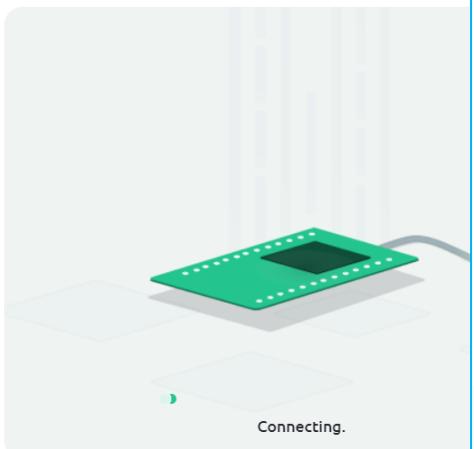
Silahkan Upload Program dan Tunggu  
Sampai muncul pesan **Congrats!**

## Quickstart



**Waiting for device online...**

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



If your device doesn't show up online after you completed all the steps in this guide, check this [Troubleshooting guide](#)

[← Back](#)

Back

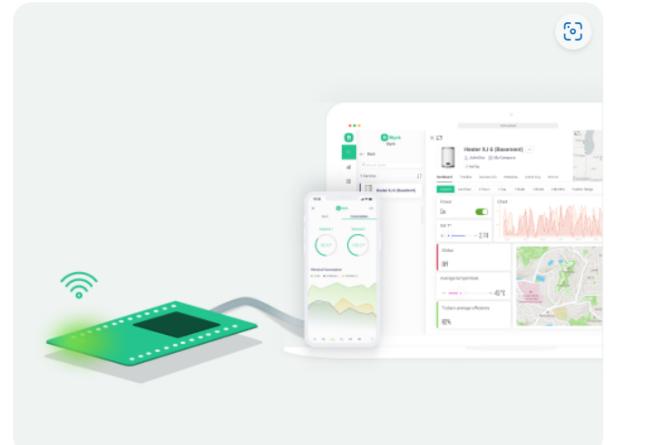
Cancel

## Go To Device

↑ Back

Cancelling

Go To Device



# Tambahan Code

```
void setup()
{
    // Debug console
    Serial.begin(9600);
    dht.begin();
    pinMode (RELAY1,OUTPUT);
    pinMode (RELAY2,OUTPUT);
    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
    // ...
}
```

```
// This function sends Arduino's uptime every second to Virtual
#include <DHT.h>
#define DHTPIN 25
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
void myTimerEvent()
{
    // You can send any value at any time.
    // Please don't send more than 10 values per second.
    float suhu = dht.readTemperature();
    float kelembaban = dht.readHumidity();

    Blynk.virtualWrite(V5, suhu);
    Blynk.virtualWrite(V6, kelembaban);
}
```

```
#define RELAY1 18
#define RELAY2 19
BLYNK_WRITE(V0)
{
    // Set incoming value from pin V0 to a variable
    int value = param.asInt();

    // Update state
    Blynk.virtualWrite(V1, value);
    if(value == 1){
        digitalWrite(RELAY1,HIGH);
    }else{
        digitalWrite(RELAY1,LOW);
    }
}

BLYNK_WRITE(V4)
{
    // Set incoming value from pin V0 to a variable
    int value = param.asInt();

    // Update state
    Blynk.virtualWrite(V4, value);
    if(value == 1){
        digitalWrite(RELAY2,HIGH);
    }else{
        digitalWrite(RELAY2,LOW);
    }
}
```

# Blynk Publish dan Subscribe

7

ESP32 Online ...

inteknus ESP32

Add Tag

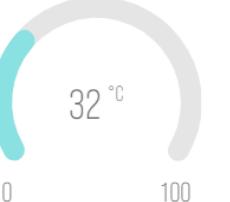
Dashboard Timeline Device Info Metadata Actions Log

Last Hour

Latest 6 Hours 1 Day 1 Week 1 Month 3 Months 6 Months 1 Year Custom

RELAY 1  RELAY 1 

RELAY 2  RELAY 2 

SUHU  KELEMBAPAN 

GRAFIK SUHU



3:25 PM 3:30 PM 3:35 PM 3:40 PM 3:45 PM 3:50 PM 3:55 PM 4:00 PM 4:05 PM 4:10 PM 4:15 PM 4:20 PM

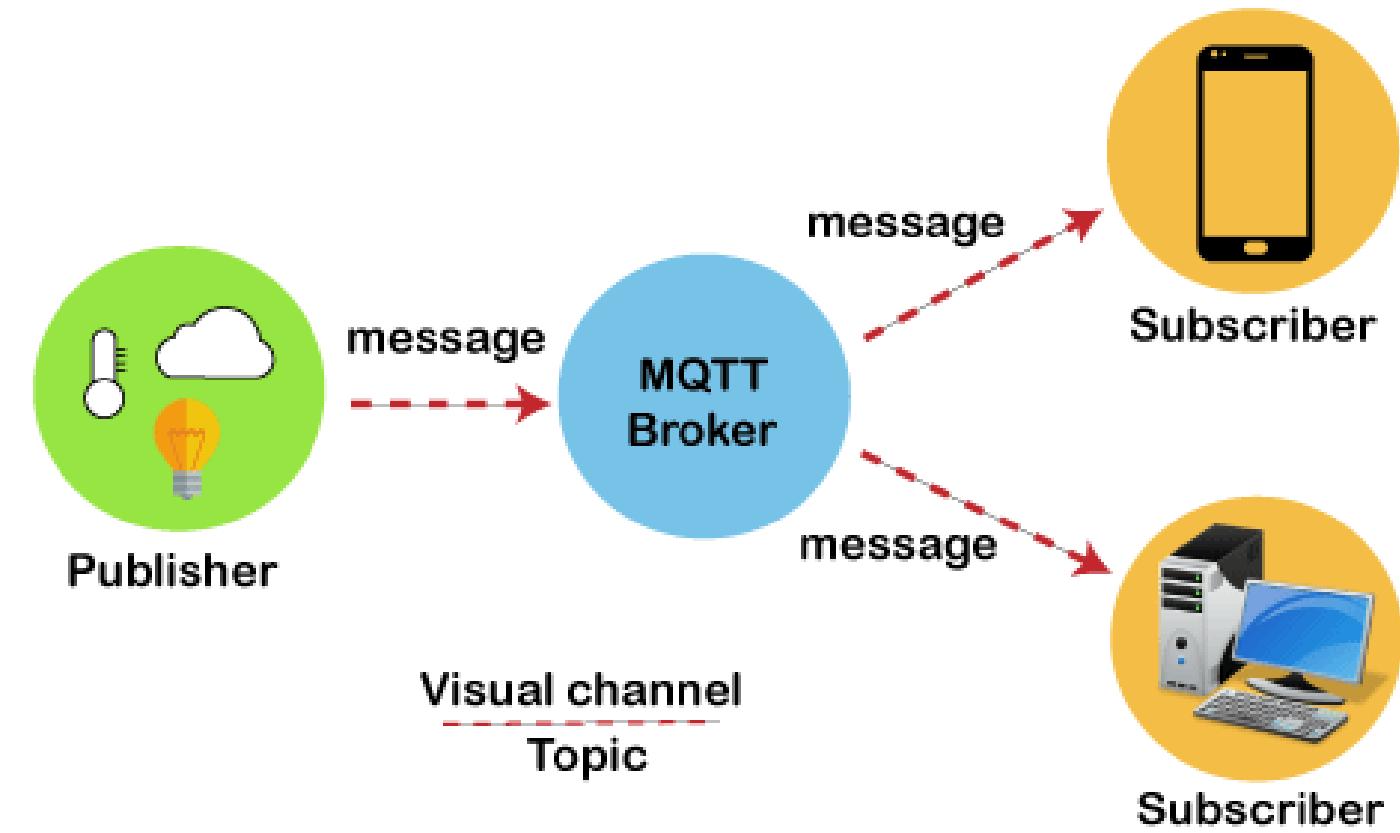
Region: sgp1 Privacy Policy

Tampilan Blynk Dashboard



- MQTT Publish dan Subscribe
- Logika Untuk Control Device Microcontroller
- Membangun Dashboard IOT di Android Kodular
- Node RED (Dashboard)

## MQTT Architecture



**MQTT** (Message Queuing Telemetry Transport) adalah protokol komunikasi ringan yang digunakan untuk mentransmisikan data antar perangkat, terutama dalam konteks ***Internet of Things (IoT)***. Dalam konteks MQTT, terdapat dua istilah utama yang perlu dipahami: "**Topic**" dan "**Payload**".

# MQTT Publish dan Subscribe

8

## Topic:

**Definisi:** Topic adalah cara untuk mengorganisir pesan-pesan dalam sistem MQTT. Ini adalah string yang digunakan untuk memberi identifikasi pada pesan atau menunjukkan ke topik tertentu.

**Fungsi:** Topik memungkinkan pihak-pihak yang terlibat dalam pertukaran pesan untuk menentukan area atau subjek tertentu dari data yang diinginkan atau diterima.

**Contoh:** Jika Anda memiliki perangkat IoT yang mengukur suhu, Anda dapat menggunakan topik seperti "**iwan/sensorsuhu/kamar1**" untuk menyampaikan data suhu dari sensor di kamar 1.

## Payload:

**Definisi:** Payload adalah data aktual atau informasi yang dikirimkan dalam pesan MQTT setelah topik ditentukan. Ini bisa berupa teks, angka, atau format data lainnya sesuai kebutuhan aplikasi.

**Fungsi:** Payload berisi informasi yang ingin dikomunikasikan antar perangkat. Misalnya, dalam pesan yang membahas suhu di kamar 1, payload mungkin berisi nilai suhu aktual.

## Contoh Pesan MQTT:

**Topic** : **iwan/sensorsuhu/kamar1**

**Payload** : **25.5**

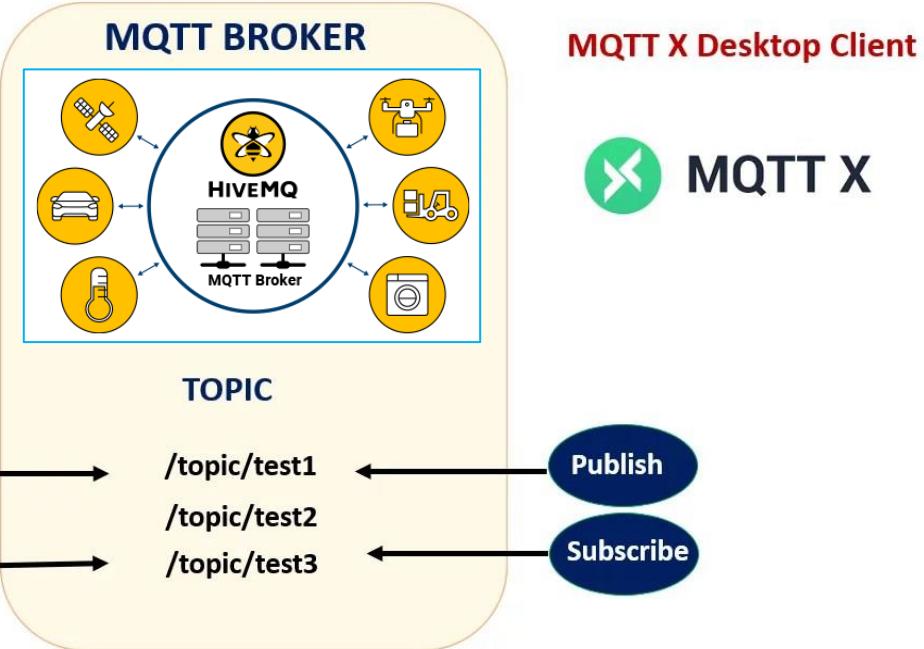
Dalam contoh ini, "**iwan/sensorsuhu/kamar1**" adalah **topik** yang memberi tahu penerima bahwa pesan berisi data suhu di kamar 1, sementara "**25.5**" adalah **payload** yang merupakan nilai suhu aktual yang dikirim.



# MQTT Publish dan Subscribe

8

ESP32 MQTT Client  
Publisher & Subscriber



WEB MQTT BROKER  
<https://www.hivemq.com/mqtt/public-mqtt-broker/>

## Free Public MQTT Broker

Looking for a full-featured MQTT platform?  
Try HiveMQ.

Try HiveMQ as a managed service or self-hosted.

Start free

### Free Public MQTT Broker by HiveMQ

Our [Public HiveMQ MQTT broker](#) is open for anyone to use. Feel free to write an MQTT client that connects with this broker. We also keep a list of [MQTT client libraries](#) that can be used to connect to HiveMQ.

You can access the MQTT broker securely at:

Broker: [broker.hivemq.com](http://broker.hivemq.com)

TCP Port: 1883

WebSocket Port: 8000

TLS TCP Port: 8883

TLS WebSocket Port: 8884

[View public broker](#)

# MQTT Publish dan Subscribe

8

## Free Public MQTT Broker by HiveMQ

Our [Public HiveMQ MQTT broker](#) is open for anyone to use. Feel free to write an MQTT client that connects with this broker. We also keep a list of [MQTT client libraries](#) that can be used to connect to HiveMQ.

You can access the MQTT broker securely at:

Broker: [broker.hivemq.com](http://broker.hivemq.com)

TCP Port: 1883

Websocket Port: 8000

TLS TCP Port: 8883

TLS Websocket Port: 8884

[View public broker](#)



The screenshot shows the Arduino IDE interface with a sketch named "9.\_MQTT\_Client". The code implements an MQTT client using the PubSubClient library. It defines WiFi credentials and initializes the WiFiClient and PubSubClient. The setup function disconnects from WiFi, begins serial communication at 9600 bps, sets up WiFi, and configures the MQTT client to connect to the broker.hivemq.com server on port 1883. A callback function is registered for incoming messages. The code also includes a helper function for WiFi setup and prints to the serial monitor.

```
#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid      = "Mamangcoffee";
const char* password  = "mamangkopi2021";
const char* mqtt_server = "broker.hivemq.com";
const char* client_id  = "iwan/esp32/001";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void setup() {
  WiFi.disconnect();
  Serial.begin(9600);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

void setup_wifi() {
  delay(10);
  Serial.println();
}
```

## Note:

-Install Library **PubSubClient**

-Sesuaikan WIFI SSID dan Password.

-**Client\_id** Harus **Unix** tidak boleh sama dengan orang lain atau Device lain.

-**Topic** Juga harus **Unix** tidak boleh sama dengan orang lain atau Device lain.

# Logika Control Device ESP32

8

```
void reconnect() {  
    // Loop until we're reconnected  
    while (!client.connected()) {  
        Serial.print("Attempting MQTT connection...");  
        // Attempt to connect  
        if (client.connect(client_id)) {  
            Serial.println("connected");  
            // Subscribe  
            client.subscribe("iwan/relay1"); //Masukan Sesuai dengan Data  
        } else {  
            Serial.print("failed, rc=");  
            Serial.print(client.state());  
            Serial.println(" try again in 5 seconds");  
            // Wait 5 seconds before retrying  
            delay(5000);  
        }  
    }  
}
```

```
void callback(char* topic, byte* message, unsigned int length) {  
    Serial.print("Terima Data: ");  
    Serial.print(topic);  
    Serial.print(". isi Pesan: ");  
    String messageTemp;  
  
    for (int i = 0; i < length; i++) {  
        Serial.print((char)message[i]);  
        messageTemp += (char)message[i];  
    }  
    Serial.println();  
  
    if (String(topic) == "iwan/relay1") {  
        Serial.print("status : ");  
        if(messageTemp == "on"){  
            Serial.println("LAMPU ON");  
        }  
        else if(messageTemp == "off"){  
            Serial.println("LAMPU OFF");  
        }  
    }  
}
```

Topic

Payload

## Note:

-Topic untuk **Subscribe** Harus **Unix** dan tidak boleh sama dengan orang lain

# Dashboard IOT di Android Kodular

8

Kodular merupakan aplikasi berbasis web based yang dapat membantu para pengembang dalam membangun aplikasi berbasis android. Dengan menggunakan konsep 'drag and drop', Kodular menjadi aplikasi yang sangat populer digunakan oleh pengembang dalam membangun aplikasi berbasis android.

<https://www.kodular.io>

## KODULAR

Much more than a modern app creator without coding

 CREATE APPS!

+ SEE MORE

# Dashboard IOT di Android Kodular

8

## Membuat Akun Kodular

1



Sign In  
Use your Kodular Account

  Close

Create Account

Sign In

Unable to login? Reset your password

Or sign in with



4

Hi inteknus,  
Welcome to Kodular!

To complete your signup, please verify your account by clicking on the following link.

Confirm Account

2



Sign Up  
Create your Kodular Account

  
inteknus  
module@inteknus.asia

Already have an account?  
Sign in instead

Next

Or sign up with



3



Almost done  
Secure your account with a password

  
.....  
  
.....  


I agree to the Terms of Service

←

Create account

Or sign up with



5



Sign In  
Use your Kodular Account

  
module@inteknus.asia  
.....  


Create Account Sign In

Unable to login? Reset your password

Or sign in with



Konfirmasi  
Email



# Dashboard IOT di Android Kodular

8

## Aktivasi Akun Kodular

**1**

Account verified!

Sign In

Use your Kodular Account

Enter your email\* module@inteknus.asia

Enter your password\* .....

I'm not a robot reCAPTCHA Privacy - Terms

Create Account

Unable to login? Reset your password

Or sign in with

**2**

Authorize Kodular Creator?

module@inteknus.asia

Kodular Creator will be able to

View your login information

Cancel

**3**

**Kodular Creator**

Last updated January 01, 2020

**Terms of Use and Privacy Policy**

Welcome to Kodular Creator website (the "Site"). The Site runs on Google's App Engine service. You must read and agree to these Terms of Service and Privacy Policy (collectively, the "Terms") prior to using any portion of this Site. These Terms are an agreement between you and Kodular. If you do not understand or do not agree to be bound by these Terms, please immediately exit this Site.

Kodular reserves the right to modify these Terms at any time and will publish notice of any such modifications online on this page for a reasonable period of time following such modifications, and by changing the effective date of these Terms. By continuing to access the Site after notice of such changes have been posted, you signify your agreement to be bound by them. Be sure to return to this page periodically to ensure familiarity with the most current version of these Terms.

**Description of Kodular Creator**

From this Site you can access Kodular Creator, which lets you develop applications for Android devices using a web browser and either a connected phone or emulator. You can also use the Site to store your work and keep track of your projects. Kodular Creator was originally developed from MIT App Inventor. The Site also includes documentation and educational content.

**Account Required for Use of Kodular Creator**

In order to log in to Kodular Creator, you need to use a Google account or provide a valid email. Your use of that account is subject to Google's Terms of Service for accounts, and the information you provide to Google is governed by Google's Privacy Policy. Kodular has no access to your Google account or the information you provide for it other than the account email address, which we may use to contact you.

Information about you

I ACCEPT THE TERMS OF SERVICE



# Dashboard IOT di Android Kodular

8

The image is a collage of four screenshots from the Kodular platform, numbered 1 through 4, illustrating the process of creating an IoT dashboard for an Android app.

- Screenshot 1:** Shows the main project list screen. A yellow circle labeled "1" is positioned above the "Create project" button, which is highlighted with a red border. Other buttons include "Import project" and "Sort by: Date modified (latest first)".
- Screenshot 2:** Shows the "Create new project" screen. A yellow circle labeled "2" is positioned above the "Give your new project a name" input field. The input field contains the text "MQTT\_IWAN".
- Screenshot 3:** Shows the "Configure your project" screen. A yellow circle labeled "3" is positioned above the "App Name" input field, which contains "MQTT\_IWAN". Other configuration options shown include "Theme: Default", "Minimum SDK Level: Android 5.0 - 5.0.2 (API 21)", "Package Name: inteknus.asia", and "Primary Color: #3F51B5FF".
- Screenshot 4:** Shows the "Creator" interface for "Screen1". A yellow circle labeled "4" is positioned above the "Screen1" preview. The interface includes a palette of components like Button, Checkbox, Circular Progress, etc., and a properties panel on the right.

**Welcome to Kodular!**

Create project Import project Sort by: Date modified (latest first)

Create new project

Give your new project a name

MQTT\_IWAN

Next

Back

Cancel

App Name: MQTT\_IWAN

Theme: Default

Minimum SDK Level: Android 5.0 - 5.0.2 (API 21)

Package Name: inteknus.asia

Primary Color: #3F51B5FF

Cancel

Finish

Creator

Project Test Export Help

MQTT\_IWAN Screen1 Add Screen Copy Screen Remove Screen

Palette Viewer

Screen1 Properties

Common properties

About Screen

About Screen Background Color: #444444FF

About Screen Light Theme

About Screen Title

About this application

Align Horizontal: Left : 1

Align Vertical: Top : 1

Background Color: #FFFFFF

Background Image

Close Screen Animation

Navigation Bar C: #000000FF

Open Screen Animation

Screen Orientation

Screen1

Screen1

Screen1 Properties

Common properties

About Screen

About Screen Background Color: #444444FF

About Screen Light Theme

About Screen Title

About this application

Align Horizontal: Left : 1

Align Vertical: Top : 1

Background Color: #FFFFFF

Background Image

Close Screen Animation

Navigation Bar C: #000000FF

Open Screen Animation

Screen Orientation

Screen1

Screen1 Properties

Common properties

About Screen

About Screen Background Color: #444444FF

About Screen Light Theme

About Screen Title

About this application

Align Horizontal: Left : 1

Align Vertical: Top : 1

Background Color: #FFFFFF

Background Image

Close Screen Animation

Navigation Bar C: #000000FF

Open Screen Animation

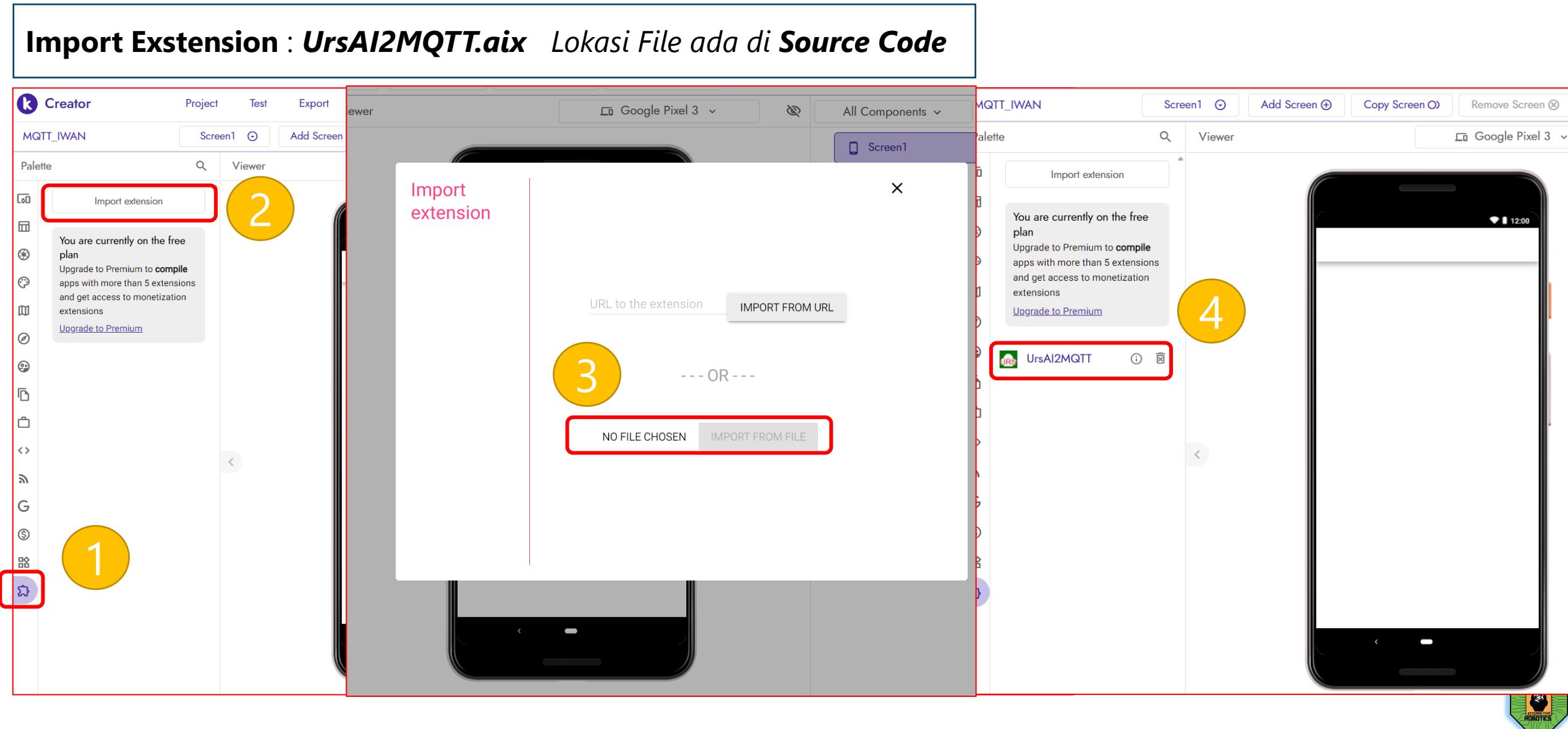
Screen Orientation

Screen1

# Dashboard IOT di Android Kodular

8

**Import Exstension : UrsAI2MQTT.aix Lokasi File ada di Source Code**

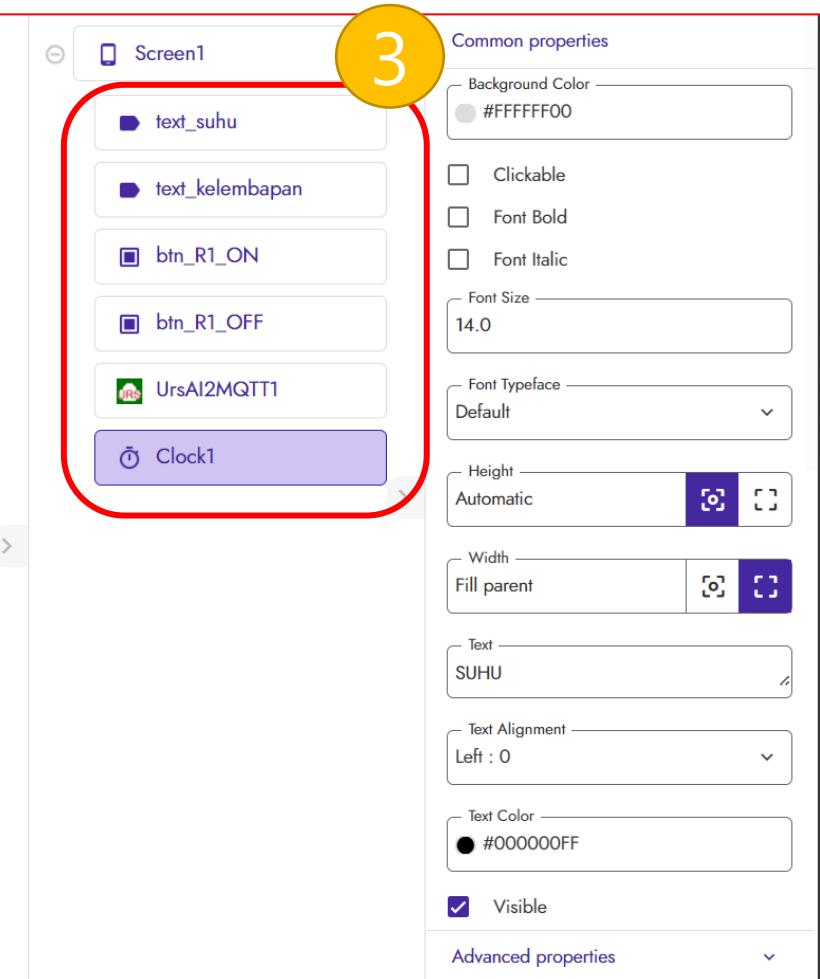
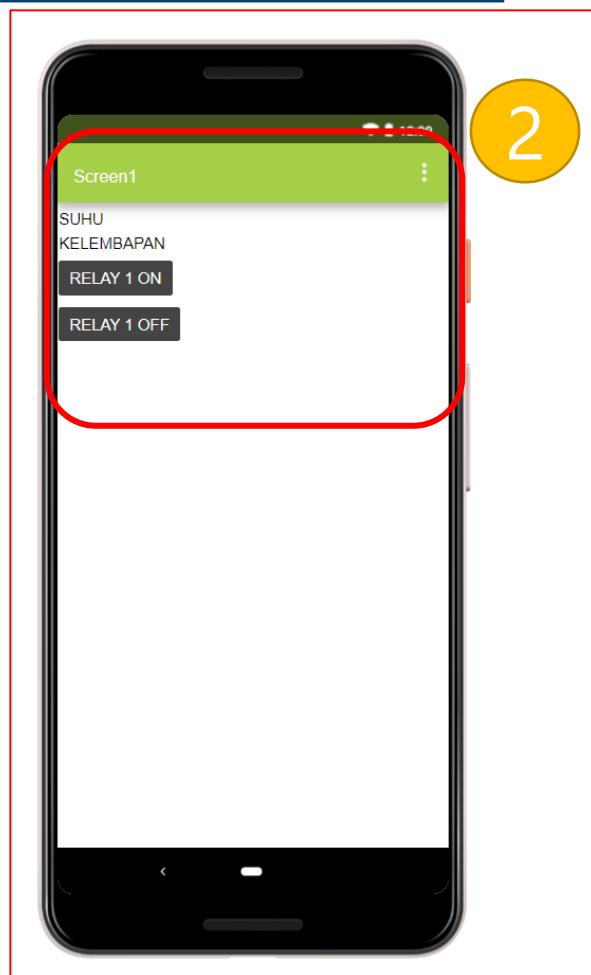
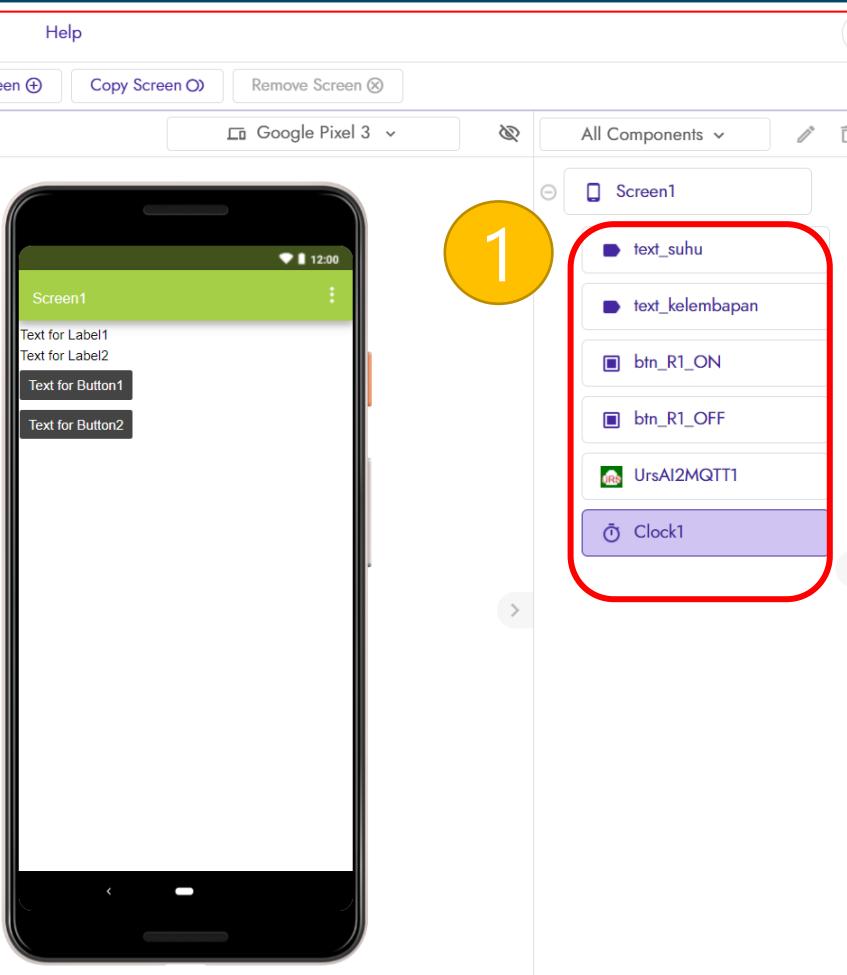


# Dashboard IOT di Android Kodular

8

Masukan **Label**, **Button** dan Extension **UrsAi2MQTT1**

Rubahlah **Nama** dan **Text** dari masing-masing Komponen



# Dashboard IOT di Android Kodular

8

1

**Block Code**

**Creator** Project Test Export Help

MQTT\_IWAN Screen1 Add Screen Copy Screen Remove Screen

Free Designer Blocks

**Blocks**

**Viewer**

**2**

```
when Clock1 > .Timer
do
    set UrsAI2MQTT1 .Broker to "broker.hivemq.com"
    set UrsAI2MQTT1 .Port to 1883
    set UrsAI2MQTT1 .Client ID to "iwan/android/001"
    call UrsAI2MQTT1 .Connect
        CleanSession true
    call UrsAI2MQTT1 .Subscribe
        Topic "iwan/suhu"
        QoS 0
    call UrsAI2MQTT1 .Subscribe
        Topic "iwan/kelembapan"
        QoS 0

when UrsAI2MQTT1 PublishedReceived
Topic Payload Message RetainFlag DupFlag
do
    if compare texts get Topic = "iwan/suhu"
    then set text_suhu .Text to get Message
    if compare texts get Topic = "iwan/kelembapan"
    then set text_kelembapan .Text to get Message

when btn_R1_ON Click
do
    call UrsAI2MQTT1 .Publish
        Topic "iwan/relay1"
        Message "on"

when btn_R1_OFF Click
do
    call UrsAI2MQTT1 .Publish
        Topic "iwan/relay1"
        Message "off"
```

**Blocks**

- Logic
- Math
- Text
- Lists
- Dictionaries
- Colors
- Variables
- Procedures
- Screen1
- text\_suhu
- text\_kelembapan
- btn\_R1\_ON
- btn\_R1\_OFF
- UrsAI2MQTT1
- Clock1

Show Warnings

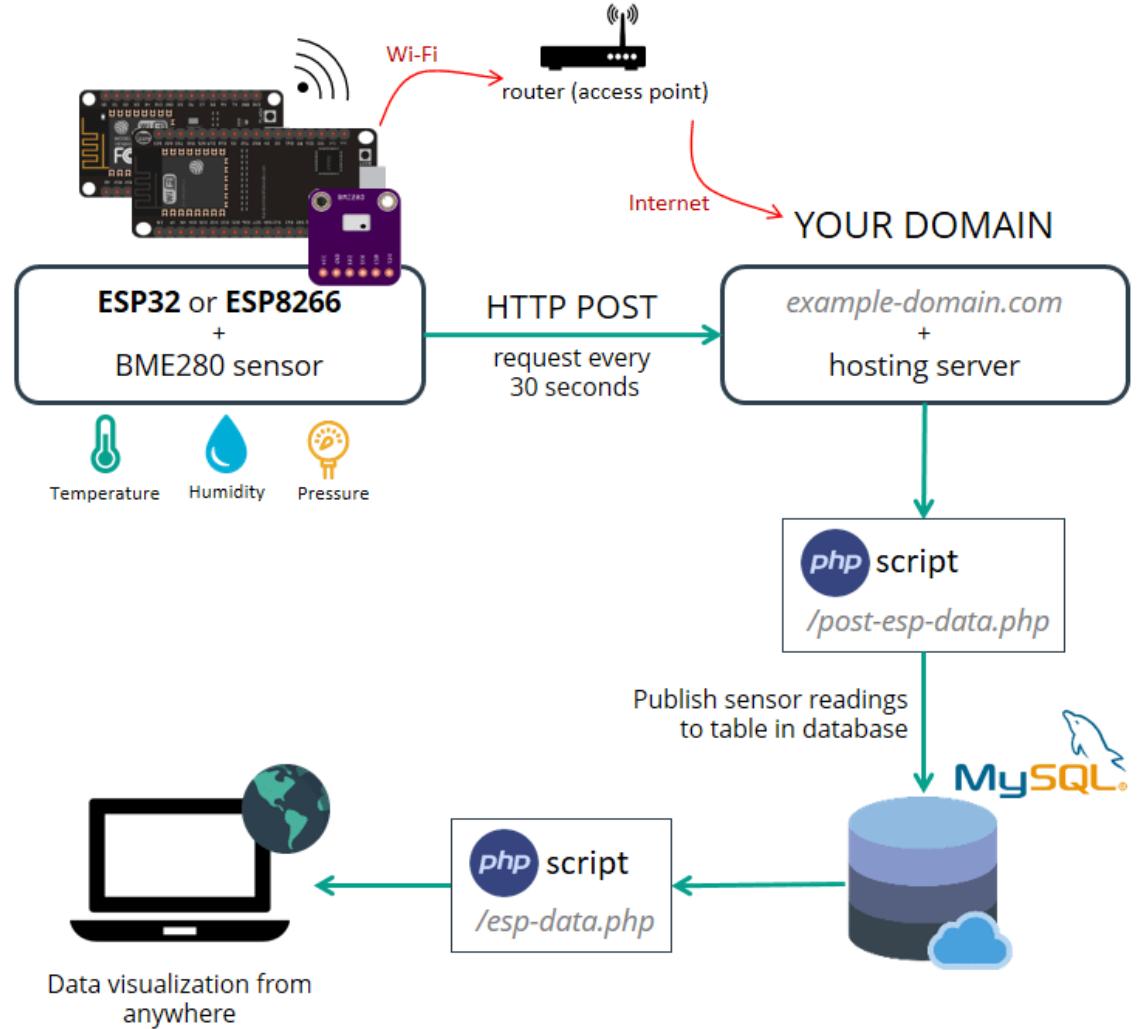
Langkah Selanjutnya silahkan Download Aplikasi **Companion** Kodular di **Playstore** Untuk mengujicoba aplikasi yang sudah dibuat.

Setelah selesai bisa berkreasi untuk menambahkan Button dan Mempercantik tampilan UI pada desain Aplikasi Android.



# Mengirim dan Menerima data dari Microcontroller ke Server Local (XAMPP)

- HTTP GET
- Database Mysql dan PHP
- Bootstrap (Responsive)



**Web server** adalah komputer yang menyimpan, memproses, dan mengirim file website ke web browser.

**Web server** terdiri dari hardware dan software yang menggunakan **HTTP (Hypertext Transfer Protocol)** untuk merespons permintaan pengguna web dari World Wide Web

# Server Local (XAMPP)

9



XAMPP

xampp-windows-x64-7.4.27-1-VC15-installer.exe

2021-12-24

167.8 MB

20



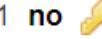
Download Xampp

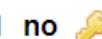


## ESP32 PHP & Mysql

- ✓ Menginstall Web server Xampp
- ✓ Membuat Database "irciot" di **phpmyadmin**
- ✓ Masukan File Project (web) yang sudah diberikan kedalam folder **C://xampp/htdocs** "*Disesuaikan dengan folder pada saat install xampp*"
- ✓ Install Library **AnduinJson** pada Arduino IDE
- ✓ Upload Program **ESP32 POST** yang diberikan dan sesuaikan terlebih dulu **Host** dengan Server / IP Address masing-masing.

# Merancang Database

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 no 	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 waktu	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()
<input type="checkbox"/>	3 namadevice	varchar(100)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	4 suhu	int(30)			No	None		
<input type="checkbox"/>	5 kelembaban	int(30)			No	None		
<input type="checkbox"/>	6 relay1	int(11)			No	None		
<input type="checkbox"/>	7 relay2	int(11)			No	None		

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 no 	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 waktu	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()
<input type="checkbox"/>	3 namadevice	varchar(100)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	4 suhu	int(30)			No	None		
<input type="checkbox"/>	5 kelembaban	int(30)			No	None		

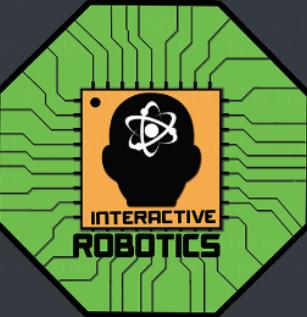
tb\_dashboard

tb\_save



# HTTP GET

9



## REALTIME DASHBOARD MONITORING

2024-02-25 18:12:55

[BUKA LOG DATA](#)

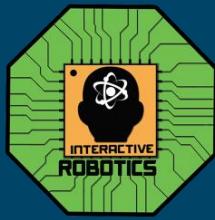
[Relay 1 ON](#) [Relay 2 ON](#)

Nama Device	Suhu	Kelembaban
iwancilibur	26 °C	76 %

Copyright ©2020 iwan cilbur | Interactive Robotics







# E-BOOK IOT BLYNK ESP-32 DEVKIT

Created By | Interactive Robotics

THANK YOU