

# **Workshop Aplikasi Mikroprosesor & Antarmuka**

**PROGRAM STUDI  
TEKNIK TELKOMUNIKASI**

**Akuwan Saleh, MT**

# PENILAIAN

- ⇒ Laporan + Tugas + Presentasi = **60%**
  - eval-1(Lap.1-5) = 20%
  - eval-2(Lap.6-10) = 20%
  - eval-3(Lap.11-15 & (PPT+ presentasi)) = 20%
- ⇒ TPS = **40%**
  - eval-4 (TPS = Tugas Proyek Semester)

# REFERENSI

- Rui Santos & Sara Santos, “ESP32 Web Server With Arduino IDE.pdf: Step By Step Project Guide”, <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>, juni 2020.
- Julien Bayle, “C Programming for Arduino”, Packt Publishing Ltd, Birmingham, May 2013
- Fabian Winkler, “Arduino/Processing Communication Workshop”, Fall, 2013.
- Jack Purdum, “Beginning C for Arduino, Learn C Programming for the Arduino and Compatible Microcontrollers”, Apress, 2012.
- John-David Warren, Josh Adams, and Harald Molle, “Arduino Robotics”, Springer, New York, 2011.
- Casey Reas and Ben Fry, “Getting Started with Processing”, O’Reilly Media, Inc., June 2010.
- Joshua Noble, “Programming Interactivity: A Designer’s Guide to Processing, Arduino, and openFrameworks”, O’Reilly Media, Inc., July 2009.
- Peter Hoddie, Lizzie Prader, “IoT Development for ESP32 and ESP8266 with JavaScript; A Practical Guide to XS and Moddable SDK”, Apress, Menlo Park, CA,USA, 2020.

# **MATERI**

## **PENDAHULUAN**

- 1. KOMUNIKASI MIKROKONTROLER DENGAN SOFTWARE PROCESSING**
- 2. ANALOG INPUT DAN AUDIO PROCESSING**
- 3. KONTROL MULTI LED MENGGUNAKAN ARDUINO DAN PROCESSING**
- 4. PENCAMPUR WARNA VIRTUAL MENGGUNAKAN ARDUINO DAN PROCESSING**
- 5. MONITORING SUHU DENGAN ARDUINO DAN PROCESSING**
- 6. MONITORING INTENSITAS CAHAYA DENGAN ARDUINO DAN PROCESSING**
- 7. KONTROL MOTOR DC MENGGUNAKAN ARDUINO DAN PROCESSING**
- 8. APLIKASI SENSOR ULTRASONIC MENGGUNAKAN ARDUINO DAN PROCESSING**

# MATERI

9. KONTROL LAMPU AC 220 V BERBASIS ARDUINO DAN PROCESSING
10. **MODUL WiFi ESP32 DENGAN ARDUINO IDE**
11. KOMUNIKASI NIRKABEL MENGGUNAKAN MODUL RF 434 MHz DAN PROCESSING
12. ESP32 WEB SERVER UNTUK KONTROL LED DAN MENAMPILKAN GAMBAR
13. ANTARMUKA MODUL GPS DENGAN MIKROKONTROLER DAN PROCESSING
14. ESP32 WEB SERVER UNTUK PENGUKURAN SUHU DAN KELEMBABAN
15. KOMUNIKASI DATA BERBASIS BLUETOOTH DAN HP

# **10. MODUL WiFi ESP32 DENGAN ARDUINO IDE**

# TUJUAN

- Menambahkan modul WiFi ESP32 pada IDE arduino
- Mendeteksi adanya perangkat WiFi lain disekitar ESP32.
- Mengontrol nyala LED yang terhubung dengan GPIO.

# DASAR TEORI

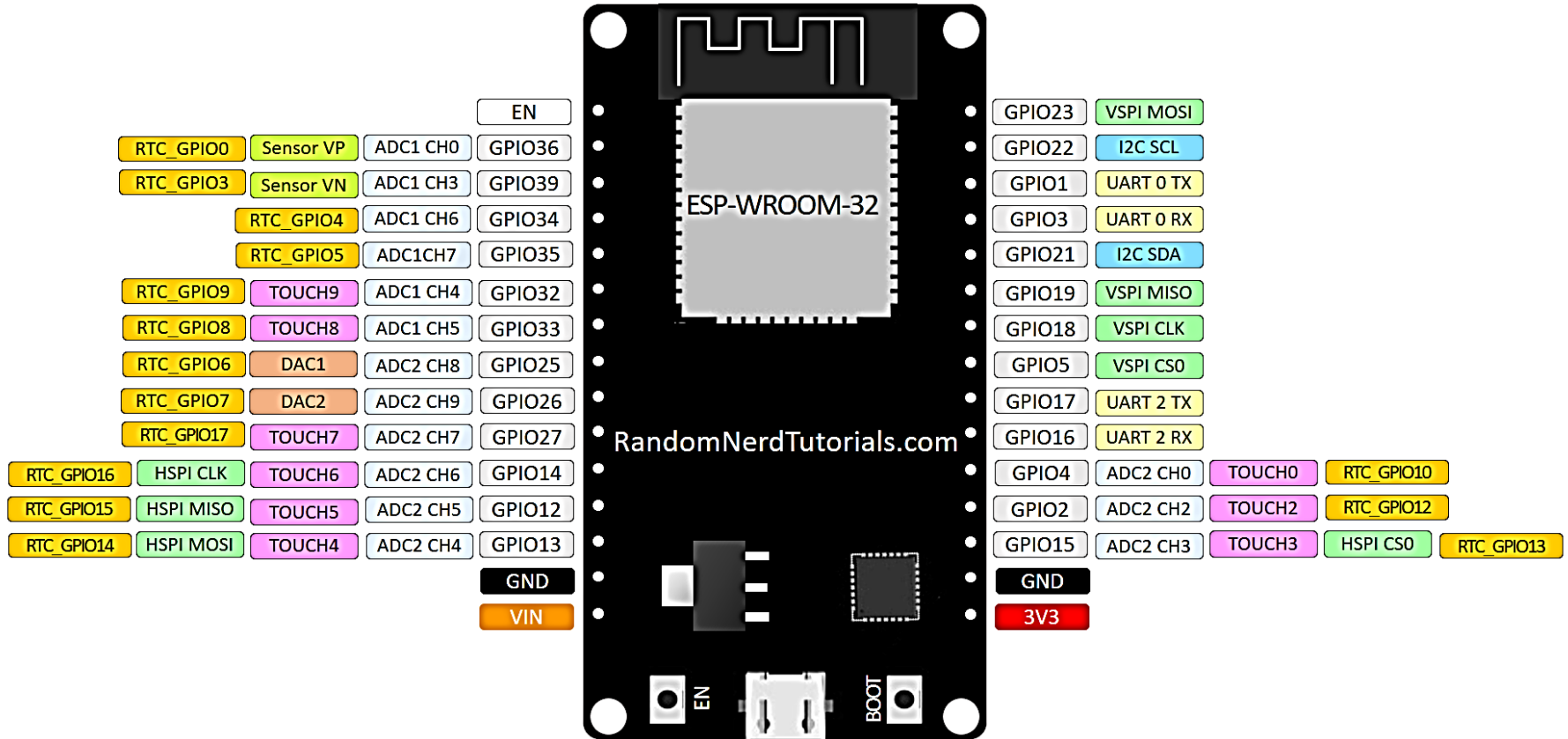
## ➤ ESP32

- ✓ Mikrokontroler ESP32 sebagai penerus dari mikrokontroler ESP8266.
- ✓ Tersedia modul wifi dan bluetooth sehingga sangat mendukung untuk aplikasi Internet of Things (IoT).
- ✓ Memiliki ADC, DAC, Sensor sentuh, jalur antarmuka UART, pin antarmuka I2C, I2S, PWM dan SPI.



# ESP32 DEVKIT V1 - DOIT

version with 30 GPIOs



## Model ESP32 DevKit V1 DOIT

- ✓ **INPUT** : EN, 34, 35, VP(36), VN(39)
- ✓ **INPUT/OUTPUT** : 32, 33, 25, 26, 27, 14, 12, 13, 15, 2, 4, RX2(16), TX2(17), 5, 18, 19, 21, RX0, TX0, 22, 23
- ✓ **ADC1** : VP(36), VN(39), 34, 35, 32, 33
- ✓ **ADC2** : 25, 26, 27, 14, 12, 13, 15, 2, 4
- ✓ **DAC** : 25, 26
- ✓ **TOUCH 0, 2 – 10** : 4, 2, 15, 13, 12, 14, 27, 33, 32

## **Strapping Pin : 2, 4, 5, 12, 15**

- ✓ Strapping pin digunakan untuk memasukan ESP32 ke dalam mode bootloader atau mode flash.
- ✓ Pada saat melakukan pemrograman, pastikan pin 12 dalam keadaan low atau 0 pada saat menyalakan ESP32.

## **Pins High at Boot : 5, 14, 15**

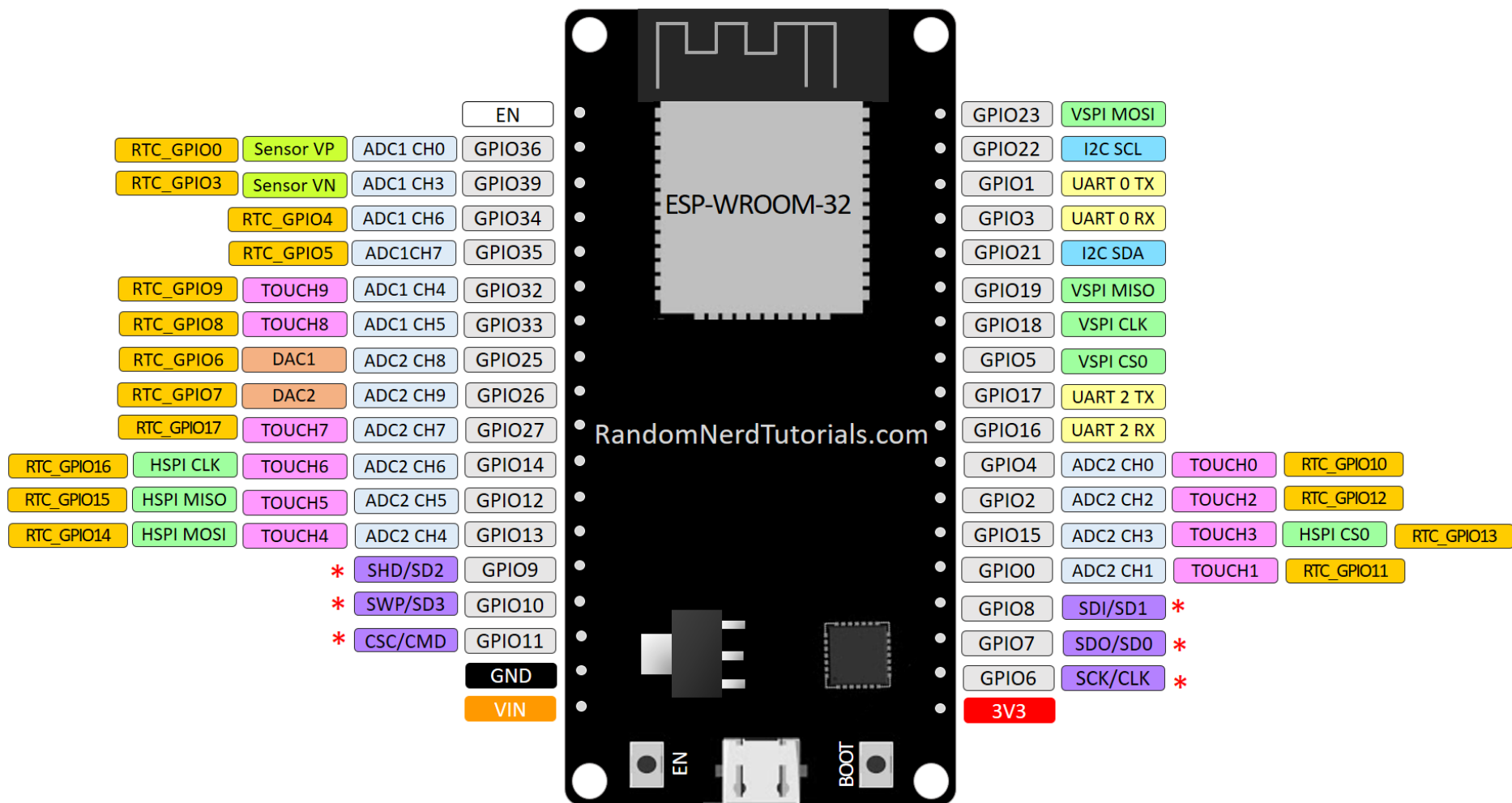
- ✓ Jika ESP32 di reset atau baru dinyalakan, ada beberapa pin yang mengeluarkan logic HIGH.
- ✓ Jika ada perangkat yang terhubung mungkin dapat mempengaruhi.

## **ADC2 :**

- ✓ pin dengan ADC2 tidak dapat digunakan jika mengaktifkan WiFi.
- ✓ jika ingin menggunakan WiFi maka gunakan pin-pin ADC1.
- ✓ Pin ADC ini mempunyai resolusi 12bit, jadi data analog yang akan terbaca bernilai 0 sampai 4095, dimana 0 adalah 0V, dan 4095 adalah 3.3V.

## ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs



\* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

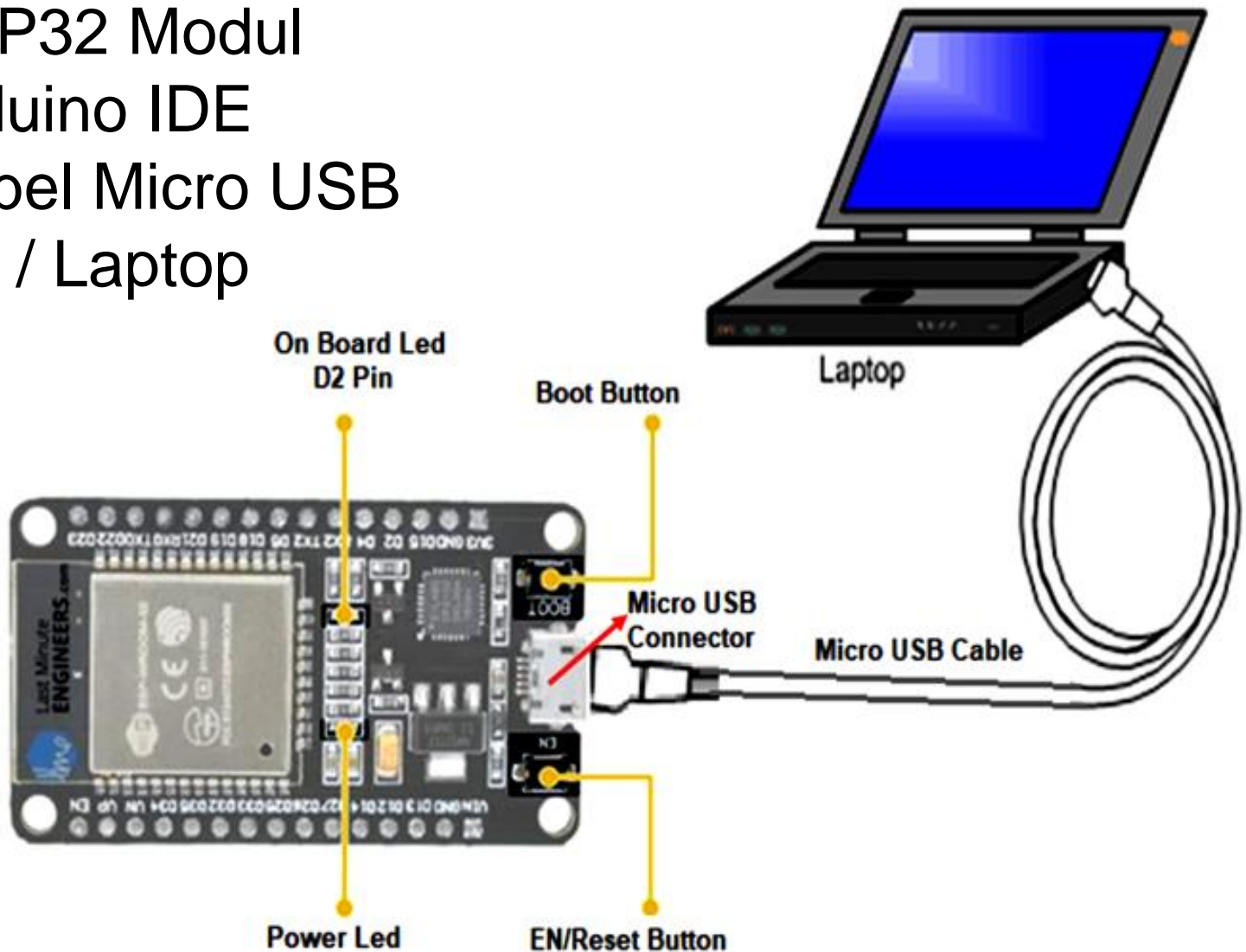
# ➤ Perbedaan ESP8266 Vs ESP32

VARIANS	ESP8266	ESP32
PICTURE MODULE	 <p>The image shows an ESP8266 module, a small black PCB with gold-plated pins. It features a white label with 'AI' and 'WiFi' logos, and text including 'MODEL ESP-14 MOD', 'VENDOR AI-THINKER', 'FC', 'STM8S003 INSIDE', and '802.11b/g/n'.</p>	 <p>The image shows an ESP32 module, a small black PCB with gold-plated pins. It features a white label with 'ESP-WROOM-32', 'CE 1313', '211-161007', and 'FCC ID: 2AC7Z-ESPWROOM02'.</p>
MCU	Xtensa Single-core 32-bit L106	Xtensa Dual-Core 32-bit LX6 with 600 DMIPS
Wi-Fi	802.11 b/g/n tipe HT20	802.11 b/g/n tipe HT40
Bluetooth	Tidak ada	tipe 4.2 dan BLE
Typical Frequency	80 MHz	160 MHz
SRAM	Tidak ada	ada
Total GPIO	17	36
Total SPI-UART-I2C-I2S	2-2-1-2	4-2-2-2
Resolusi ADC	10 bit	12 bit
Suhu operasional Kerja	-40°C to 125°C	-40°C to 125°C
Sensor di dalam module	Tidak ada	touch sensor, temperature sensor, hall effect sensor

# Peralatan:

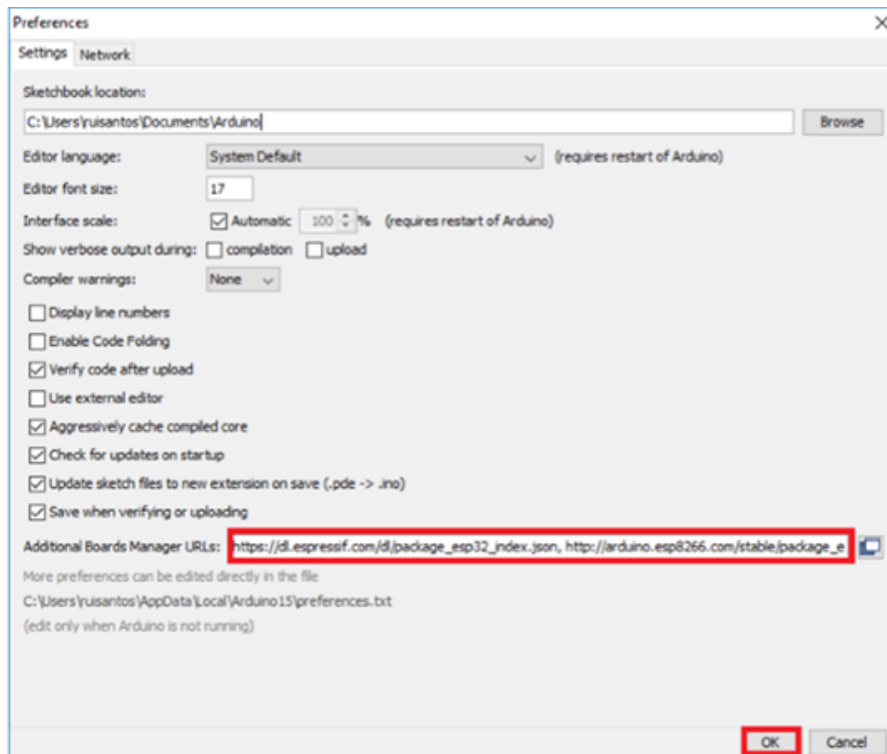
1. ESP32 Modul
2. Arduino IDE
3. Kabel Micro USB
4. PC / Laptop

# Rangkaian:



# *Instalasi* Board ESP32 :

- ✓ Open the preferences window from the Arduino IDE. Go to **File > Preferences**.
- ✓ Enter [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) into the “**Additional Board Manager URLs**” field as shown in the figure below. Then, click the “**OK**” button.



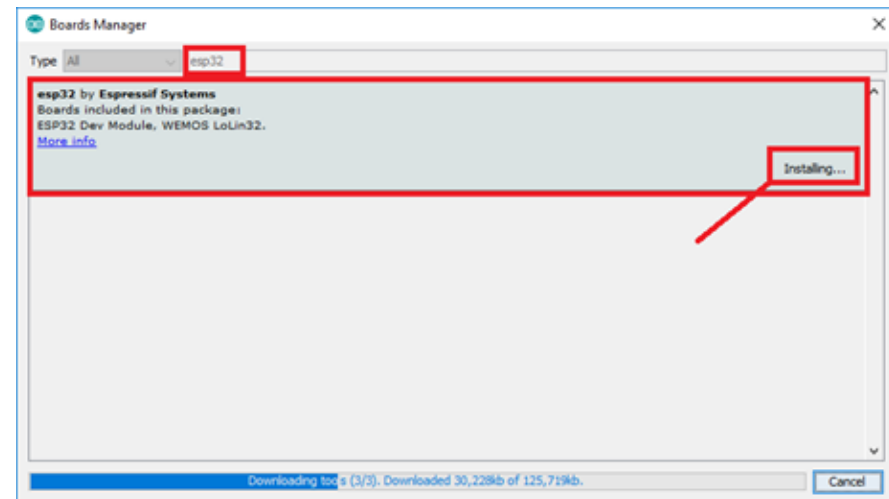
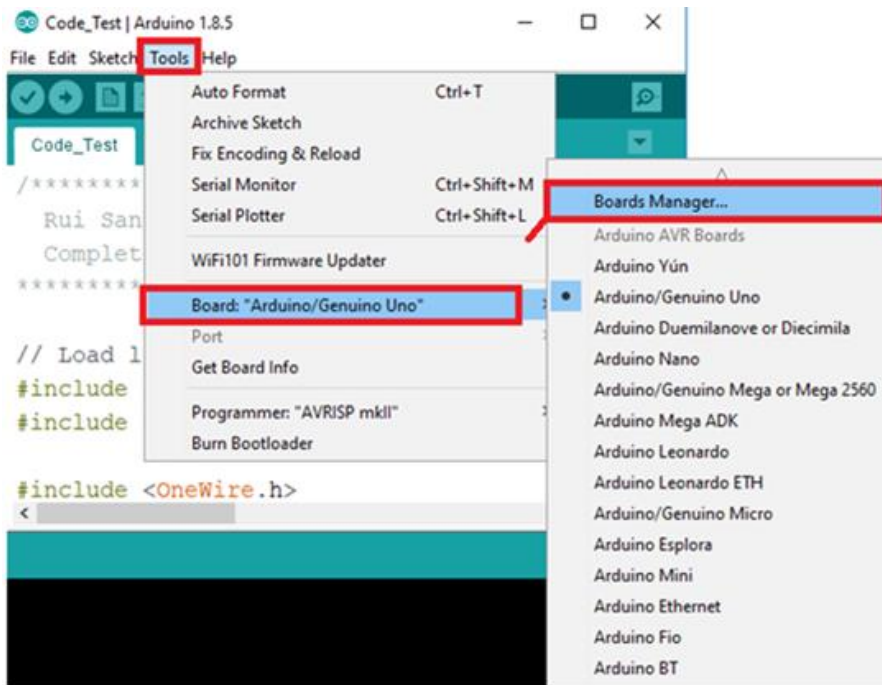
## Note:

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) ,  
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



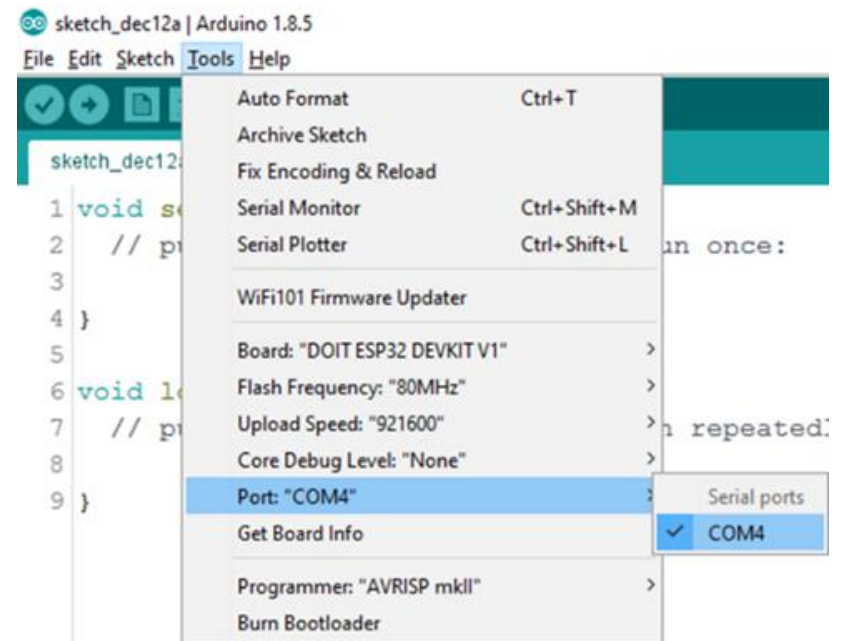
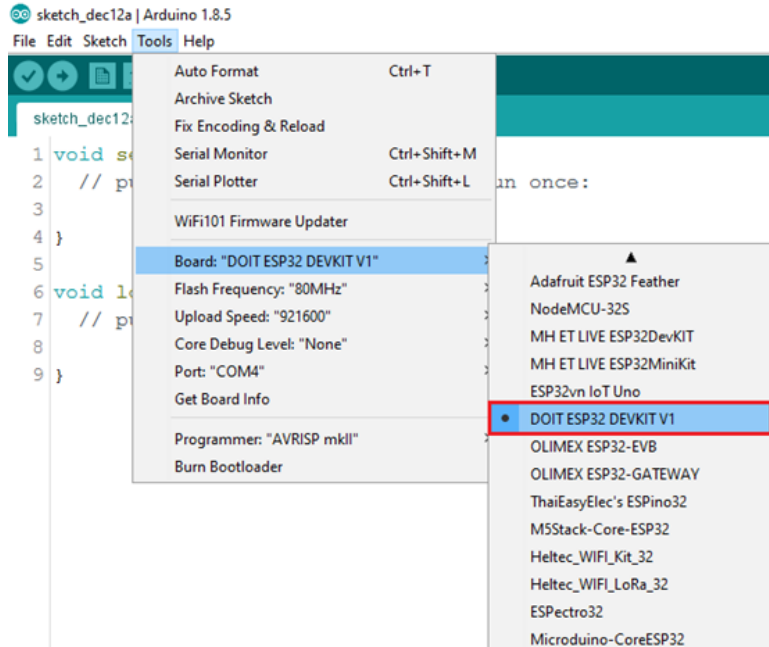
✓ Open boards manager.  
Go to **Tools > Board > Boards Manager...**

✓ Search for ESP32 and press install button for the “ESP32 by Espressif Systems”.



# Pengujian Modul WiFi ESP32 :

- ✓ Open Arduino IDE.
- ✓ Select your Board in Tools → **Board menu** (in our case it's the DOIT ESP32 DEVKIT V1)
- ✓ Select the Port (COM Port in your Arduino IDE)



# PROGRAM:

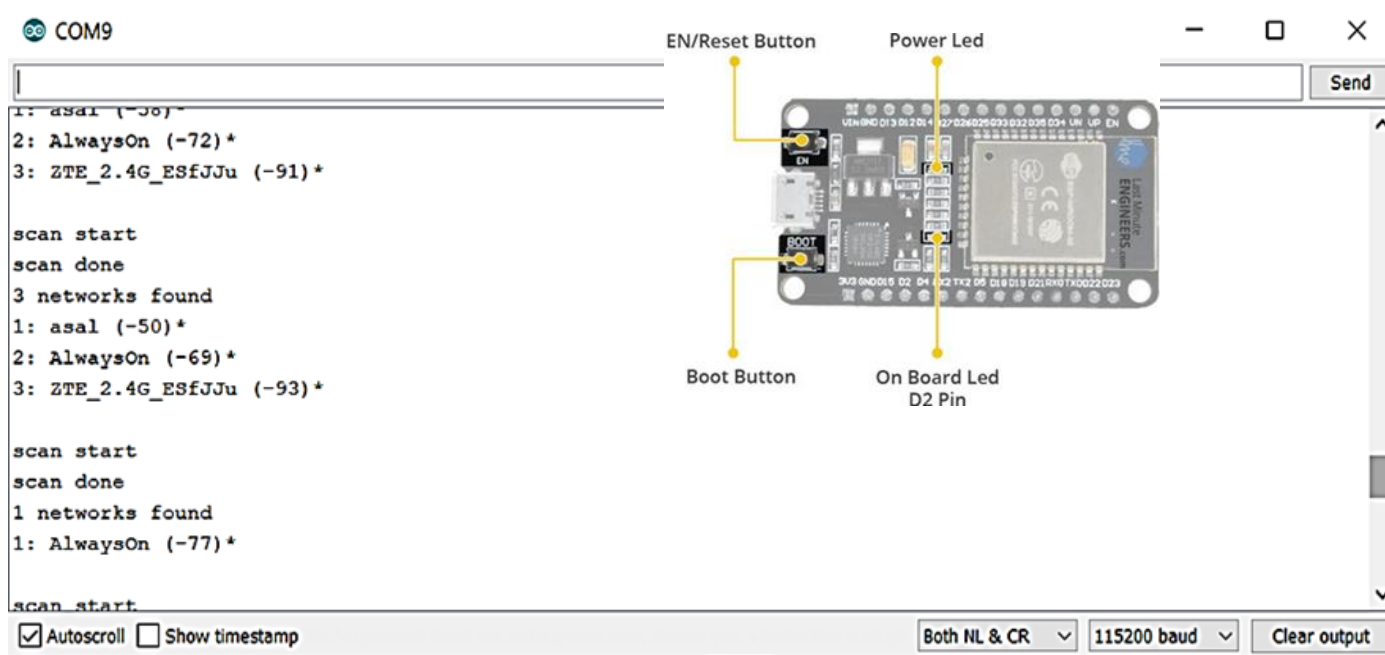
Arduino IDE

```
#include "WiFi.h"

void setup()
{
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
  Serial.println("Setup done");
}

void loop()
{
  Serial.println("scan start");
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
  if (n == 0) {
    Serial.println("no networks found");
  } else {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i) {
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.print(")");
      Serial.println((WiFi.encryptionType(i) ==
        WIFI_AUTH_OPEN)?" ":"*");
      delay(10);
    }
  }
  Serial.println("");
  delay(5000);}
}
```

- ✓ Press the **Upload button** in the Arduino IDE. Wait a few seconds while the code compiles and uploads to your board.
- ✓ Open the Arduino IDE Serial Monitor at a **baud rate of 115200**
- ✓ Press the ESP32 on-board **Enable button** and you should see the networks available near your ESP32:



## ***Hasil :***

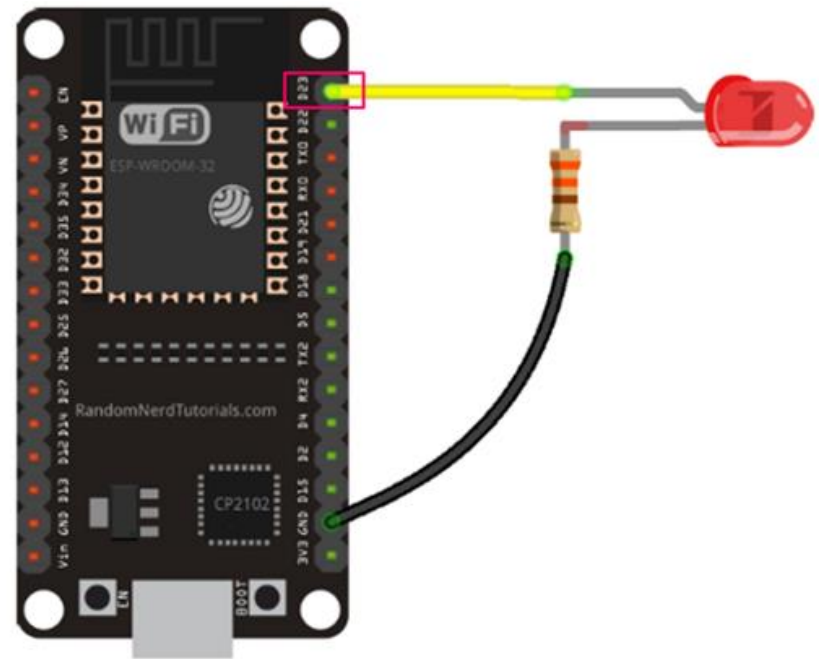
- Amati pada IDE Serial Monitor, catat WiFi yang aktif dan terdeteksi oleh modul ESP32

# • Percobaan Led Blink

## PERALATAN :

1. Breadboard(Optional)
2. ESP32 Modul
3. Arduino IDE
4. 1 LED
5. 1 resistor 330  $\Omega$
6. Kabel Micro USB

## Rangkaian:



# ***PROGRAM:*** *Led blink*

Arduino IDE

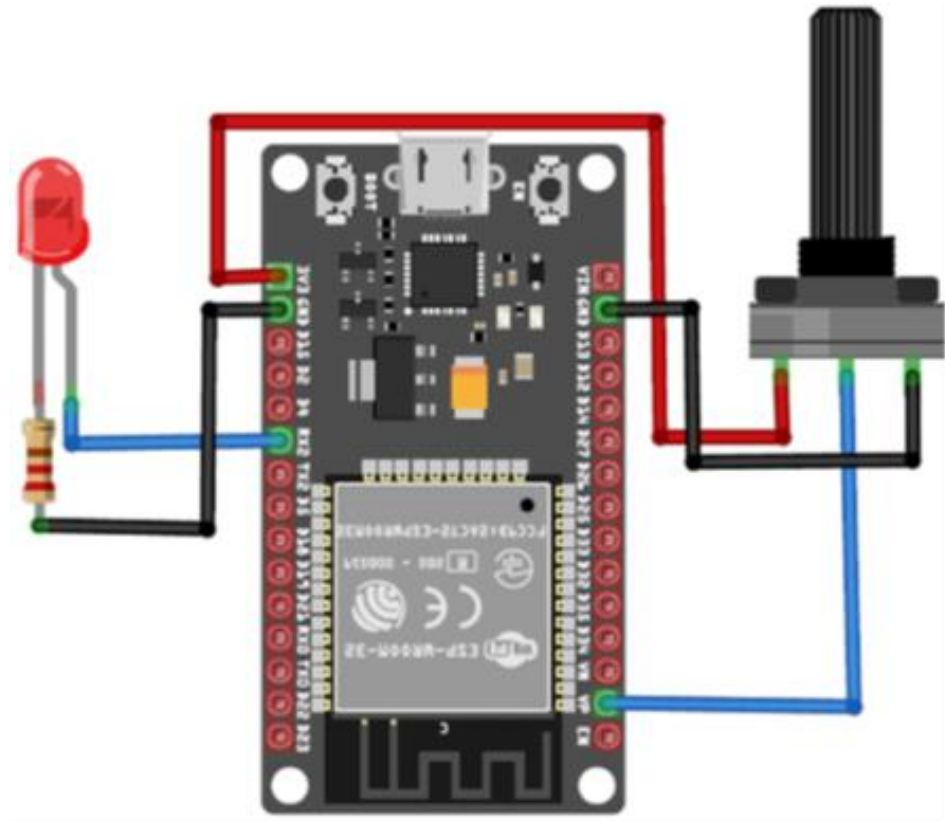
```
const int ledPin = 23; // ledPin refers to ESP32 GPIO 23
void setup() {
  pinMode(ledPin, OUTPUT); // initialize digital pin
  ledPin as an output.
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

# • PWM ESP32 Kontrol Brightness LED

## PERALATAN :

1. Breadboard(Optional)
2. ESP32 Modul
3. Arduino IDE
4. 1 Potensiometer
5. 1 LED
6. 1 resistor 330  $\Omega$
7. Kabel Micro USB

## Rangkaian:





# ***PROGRAM:*** Brightness LED

Arduino IDE

```
const int ledPin = 16;
const int freq = 15000; // setting PWM properties
const int ledChannel = 0;
const int resolution = 13;
void setup(){
  Serial.begin(9600);
  ledcSetup(ledChannel, freq, resolution);
  ledcAttachPin(ledPin, ledChannel); }
void loop(){
  dutyCycle = analogRead(A0);
  Serial.print(dutyCycle);
  ledcWrite(ledChannel, dutyCycle
  delay(15);
}
```

## ***Hasil :***

- Amati nyala LED catat perubahan nyala LED tersebut.

## ***Latihan :***

1. Buatlah eksperimen menggunakan modul WiFi ESP32 untuk melakukan komunikasi dengan salah satu perangkat WiFi lain yang telah terbaca/terdeteksi oleh program WiFiScan.
2. Buatlah eksperimen menggunakan modul WiFi ESP32 untuk mengontrol brightness Led dengan tampilan menggunakan software processing.