



VERSI 2.0  
19 JULI, 2023

# PIRANTI CERDAS

## MODUL 3 : WIFI CONNECTION

**TIM PENYUSUN :**  
ZAMAH SARI, ST., MT.  
CLARISSA SANINDITA REIKISYIFA  
CHINTYA TRIA DIANA OKTAVIANI

PRESENTED BY: LAB. INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH MALANG

## PIRANTI CERDAS

---

### PERSIAPAN MATERI

Praktikan mempelajari tentang koneksi antara perangkat ke jaringan WIFI menggunakan modul ESP32 dan Arduino IDE

---

### TUJUAN

1. Mahasiswa mampu memahami cara mengatur koneksi WiFi pada perangkat cerdas menggunakan modul ESP32.
  2. Mahasiswa mampu memahami langkah-langkah untuk menghubungkan perangkat cerdas ke jaringan WiFi yang tersedia.
  3. Mahasiswa mampu memahami pentingnya konfigurasi koneksi WiFi dalam pengembangan piranti cerdas.
  4. Mahasiswa mampu memindai jaringan WiFi menggunakan modul ESP32 dan mengakses informasi yang relevan.
- 

### KEBUTUHAN HARDWARE DAN SOFTWARE

1. HARDWARE
    - NodeMCU ESP32
    - Kabel USB
    - Breadboard
    - Kabel Jumper
    - MPU6050
    - Buzzer
    - Resistor 220 ohm
  2. SOFTWARE
    - Arduino IDE
- 

### MATERI POKOK

#### A. WIFI CONNECTION

Untuk menghubungkan perangkat keras ESP32 dengan jaringan Wi-Fi, dapat digunakan *library* <WiFi.h> dari Arduino. Pada *library* <WiFi.h> ada empat metode yang akan sering digunakan :

- **WiFi.status()** untuk melihat koneksi Wi-Fi
- **WiFi.localIP()** untuk melihat alamat IP dari router yang terhubung
- **WiFi.RSSI()** untuk memperoleh nilai kuat sinyal Wi-Fi
- **WiFi.begin()** untuk memulai hubungan dengan jaringan Wi-Fi.

## 1. Menghubungkan ke Jaringan WiFi

Langkah penting dalam mengintegrasikan modul ESP32 dengan jaringan WiFi adalah melalui fungsi `WiFi.begin()`. Proses ini memungkinkan modul untuk menginisialisasi koneksi ke jaringan WiFi yang diinginkan. Konsep dasar dari fungsi ini melibatkan penggunaan SSID (Service Set Identifier) atau nama jaringan WiFi dan juga password WiFi sebagai informasi autentikasi.

Konsep dasar fungsi `WiFi.begin()` :

- **SSID (Service Set Identifier):** Ini adalah nama unik yang mengidentifikasi jaringan WiFi tertentu. SSID diperlukan agar perangkat tahu ke jaringan mana harus terhubung.
- **Password WiFi:** Ini adalah kata sandi yang diperlukan untuk mengautentikasi perangkat ke jaringan WiFi yang aman. Hanya perangkat dengan password yang tepat yang diizinkan untuk terhubung.

Contoh dalam program di Arduino IDE :

```
1  #include <WiFi.h> //import modul
2
3  const char* ssid = "Nama_Jaringan";    // Ganti dengan SSID jaringan Anda
4  const char* password = "Password_Jaringan"; // Ganti dengan password jaringan Anda
5
6  void setup() {
7      Serial.begin(9600);
8      WiFi.begin(ssid, password); // Menghubungkan ke jaringan WiFi
```

## 2. Memeriksa Status Koneksi WiFi pada ESP32

Proses memeriksa status koneksi WiFi pada modul ESP32 sangat penting dalam pengembangan piranti cerdas. Status ini memberikan informasi tentang apakah modul terhubung ke jaringan WiFi atau tidak. Fungsi yang umum digunakan untuk memeriksa status ini adalah `WiFi.status()`. Pada `WiFi.status()` ada beberapa macam status dari koneksi Wi-Fi (return values) :

- **WL\_IDLE\_STATUS (0):** Modul dalam keadaan bebas atau idle, tidak terhubung ke jaringan.
- **WL\_NO\_SSID\_AVAIL (1):** Tidak ada jaringan WiFi dengan SSID yang dicari.
- **WL\_SCAN\_COMPLETED (2):** Pencarian jaringan WiFi di sekitar modul telah selesai.
- **WL\_CONNECTED (3):** Modul terhubung ke jaringan WiFi dengan sukses.

- **WL\_CONNECT\_FAILED (4):** Gagal menghubungkan ke jaringan WiFi.
- **WL\_CONNECTION\_LOST (5):** Koneksi WiFi terputus.
- **WL\_DISCONNECTED (6):** Modul terputus dari jaringan WiFi.
- **WL\_NO\_SHIELD (255):** Tidak ada router Wi-Fi yang tersedia di sekitar perangkat.

## B. MPU6050



Sensor MPU6050 adalah sensor mampu membaca kemiringan sudut berdasarkan data dari sensor accelerometer dan sensor gyroscope. Modul MPU6050 adalah papan elektronik yang mengintegrasikan kedua elemen tersebut untuk memungkinkan kita dapat mengukur perubahan posisi elemen sehingga menghasilkan suatu reaksi misalnya seperti, ketika suatu objek bergerak, LED menyala, atau hal-hal lain yang jauh lebih rumit.

## C. BUZZER



Buzzer adalah komponen elektronika yang dapat menghasilkan getaran suara dalam bentuk gelombang bunyi. Buzzer lebih sering digunakan karena ukuran penggunaan dayanya yang minim. Prinsip kerja buzzer adalah sangat sederhana, Ketika suatu aliran listrik mengalir ke rangkaian buzzer, maka terjadi pergerakan mekanis pada buzzer tersebut. Akibatnya terjadi perubahan energi dari energi listrik menjadi energi suara yang dapat didengar oleh manusia.

---

## LATIHAN PRAKTIKUM

Latihan praktikum wajib dikerjakan setiap orang dengan menggunakan laptop masing-masing. Perangkat fisik yang digunakan dalam latihan dapat bergantian dalam 1 kelompok yang sama.

### A. Koneksi WiFi Otomatis ke SSID yang Open Akses

1. Sebelum memulai pastikan disekitar lingkunganmu terdapat WiFi yang open akses.
2. Tulis kode program berikut pada Arduino IDE.

```

1  #include <WiFi.h>
2
3  const char* ssid = "Nama_SSID";
4
5  void setup() {
6      Serial.begin(9600);
7      WiFi.begin(ssid);
8
9      while (WiFi.status() != WL_CONNECTED) {
10         delay(1000);
11         Serial.println("Connecting to WiFi...");
12     }
13
14     Serial.println("Connected to WiFi");
15 }
16
17 void loop() {
18     // put your main code here, to run repeatedly:
19
20 }

```

3. Ganti "Nama\_SSID" dengan nama SSID jaringan WiFi open akses yang ingin anda sambungkan.
4. Compile dan upload program.
5. Kemudian buka serial monitor dan sesuaikan kecepatan penerimaan data yang telah ditentukan pada program sebelumnya. Jika telah sesuai, serial monitor akan menampilkan data yang dikirim dari board NodeMCU ESP32
6. Perangkat akan mencoba terhubung ke jaringan WiFi yang open. Pesan "Connected to WiFi" akan muncul jika berhasil terhubung.

### B. Koneksi WiFi otomatis ke SSID Tertentu dengan Password

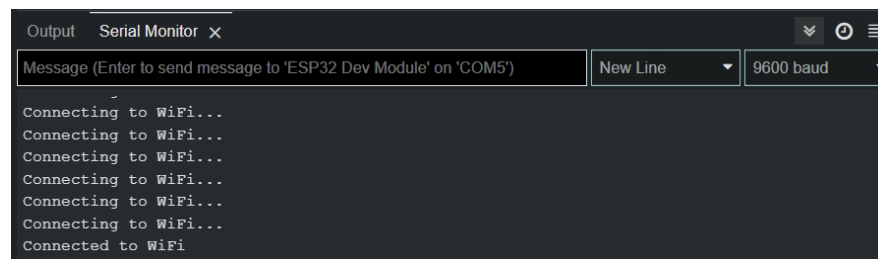
1. Sebelum memulai pastikan disekitar lingkunganmu terdapat WiFi yang memiliki password
2. Tulis kode program berikut pada Arduino IDE.

```

1  #include <WiFi.h>
2
3  const char* ssid = "SSID";
4  const char* password = "Password";
5
6  void setup() {
7      Serial.begin(9600);
8      WiFi.begin(ssid, password);
9
10     while (WiFi.status() != WL_CONNECTED) {
11         delay(1000);
12         Serial.println("Connecting to WiFi...");
13     }
14
15     Serial.println("Connected to WiFi");
16 }
17
18 void loop() {
19     // put your main code here, to run repeatedly:
20
21 }

```

3. Ganti "SSID" dengan nama SSID jaringan WiFi dan ganti "Password" dengan password yang sudah ditentukan pada WiFi yang ingin anda sambungkan.
4. Compile dan upload program.
5. Kemudian buka serial monitor dan sesuaikan kecepatan penerimaan data yang telah ditentukan pada program sebelumnya. Jika telah sesuai, serial monitor akan menampilkan data yang dikirim dari board NodeMCU ESP32
6. Perangkat akan mencoba terhubung ke jaringan WiFi dengan SSID dan password yang ditentukan. Pesan "Connected to WiFi" akan muncul jika berhasil terhubung.



## LEMBAR KERJA

### Kegiatan 1

- Buatlah program yang dapat menghubungkan ESP32 ke jaringan WiFi dengan SSID dan Password yang sudah ditentukan. Jika program terhubung maka akan muncul informasi terhubung ke jaringan dan nama SSID.

- Jika gagal maka akan terkoneksi ke jaringan WiFi dengan SSID manapun yang memiliki keamanan terbuka (open network). Tampilkan juga informasi terhubung ke jaringan dan nama SSID.

## Kegiatan 2

- Buatlah program yang menggunakan sensor MPU 6050 dan Buzzer dimana jika koneksi Wifi terhubung maka sensor akan mulai membaca. Jika sensor mengalami perubahan sudut melebihi ambang tertentu, buzzer akan berbunyi sebagai peringatan. Tiap sudut (kanan, kiri, depan, belakang) akan mendapat bunyi yang berbeda dan akan menampilkan informasi di monitor.

**PERINGATAN: Terdapat Kemiripan Source Code Yang Signifikan Dengan Praktikan Lain, Nilai Maksimal D**

### RUBRIK PENILAIAN PRAKTIKUM

Detail	Bobot Nilai
Pemahaman Materi	40%
Ketepatan Jawaban	30%
Program dapat berjalan tanpa error	15%
Tugas Pekan Materi	15%