

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

PRAKTIK MENGECEK KONEKSI WIFI PADA HARDWARE ESP32

Assyfaul Khoiriyah

Fakultas Vokasi, Universitas Brawijaya

assyfaassyfa0@gmail.com

Abstract (Abstrak)

ESP32 merupakan mikrokontroler modern yang banyak digunakan dalam pengembangan aplikasi Internet of Things (IoT) karena dilengkapi dengan modul WiFi dan Bluetooth bawaan. Salah satu tahapan penting dalam implementasi perangkat berbasis ESP32 adalah memastikan konektivitas terhadap jaringan WiFi berjalan dengan baik. Pada eksperimen ini dilakukan pengujian koneksi WiFi menggunakan board ESP32 dengan memanfaatkan software Arduino IDE. ESP32 diprogram untuk terhubung ke jaringan WiFi dan mencetak status koneksi serta alamat IP ke Serial Monitor. Hasil menunjukkan bahwa ESP32 berhasil terkoneksi ke jaringan dan memperoleh alamat IP dari router, yang menandakan komunikasi antara perangkat dan jaringan berjalan lancar. Eksperimen ini menjadi dasar penting sebelum melanjutkan ke tahap pengembangan sistem IoT yang lebih kompleks.

Kata Kunci: *ESP32, WiFi, Internet of Things, Arduino IDE, Koneksi Jaringan, Mikrokontroler*

1. Introduction (Pendahuluan)

1.1 Latar Belakang

Internet of Things (IoT) merupakan konsep yang memungkinkan perangkat elektronik terhubung dan saling bertukar data melalui jaringan internet. Salah satu komponen penting dalam pengembangan perangkat IoT adalah mikrokontroler dengan kemampuan konektivitas nirkabel, seperti ESP32. ESP32 merupakan mikrokontroler yang dilengkapi dengan modul WiFi dan Bluetooth bawaan, yang menjadikannya pilihan populer dalam berbagai aplikasi IoT. Salah satu langkah awal yang penting dalam pengembangan proyek berbasis ESP32 adalah memastikan bahwa perangkat dapat terkoneksi dengan jaringan WiFi. Oleh karena itu, pada eksperimen ini dilakukan pengecekan koneksi WiFi menggunakan ESP32 untuk memastikan perangkat dapat berkomunikasi melalui jaringan lokal.

1.2 Tujuan Eksperimen

Tujuan dari eksperimen ini adalah untuk:

- Mengetahui cara menghubungkan ESP32 ke jaringan WiFi.
- Mengecek status koneksi WiFi pada ESP32.
- Memastikan bahwa ESP32 berhasil mendapatkan alamat IP dari jaringan.

2. Methodology (Metodologi)

2.1 Tools & Materials (Alat dan Bahan)

- 1 buah Board ESP32
- Kabel USB Type-C atau Micro USB (tergantung tipe board)
- Komputer/Laptop dengan software Arduino IDE
- Koneksi internet (WiFi)
- Library dan Board ESP32 terinstal pada Arduino IDE

2.2 Implementation Steps (Langkah Implementasi)

Pada latihan berikutnya perlu dilakukan upload program untuk mengecek apakah hardware ESP32 dapat terhubung ke Access Point WIFI disekitar. Untuk melakukannya perlu melakukan perubahan koding pada file main.cpp seperti berikut ini :

```
PRAK... [Icons]
> .pio
> .vscode
> include
> lib
▼ src
  main.cpp
  test
  .gitignore
  platformio.ini

OUTLINE
TIMELINE

src > main.cpp > ...
1  #include <WiFi.h>
2
3
4  void setup() {
5      Serial.begin(115200);
6
7
8      WiFi.mode(WIFI_STA);
9      WiFi.disconnect();
10     delay(100);
11
12     Serial.println("Pemindaian Jaringan Wi-Fi Dimulai...");
13 }
14
15
16
17 void loop() {
18     int n = WiFi.scanNetworks();
19     Serial.println("Pemindaian Selesai");
20     if (n == 0) {
21         Serial.println("Tidak ada jaringan Wi-Fi yang ditemukan.");
22     } else {
23         Serial.print(n);
24         Serial.println(" jaringan Wi-Fi ditemukan:");
25         for (int i = 0; i < n; ++i) {
26             Serial.print(i + 1);
27             Serial.print(": ");
28             Serial.print(WiFi.SSID(i));
29             Serial.print(" (");
```



Kemudian klik tombol serial monitor

Pastikan tampilan serial monitor menunjukkan nama Access Point WIFI disekitar berikut juga dengan keterangan kekuatan sinyal seperti tampilan berikut ini

```
PRAK... [Icons]
> .pio
> .vscode
> include
> lib
▼ src
  main.cpp
  test
  .gitignore
  platformio.ini

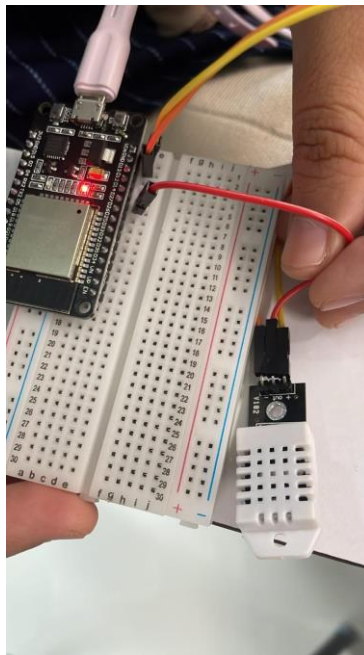
OUTLINE
TIMELINE
VS CODE PETS

src > main.cpp > ...
41

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Pemindaian Selesai
12 jaringan Wi-Fi ditemukan:
1: Oi (~45dBm)*
2: WiFi-UB.x (~50dBm)*
3: eduroam (~50dBm)*
4: iPhone (~65dBm)*
5: WiFi-UB.x (~71dBm)*
6: eduroam (~71dBm)*
7: Lab IT (~73dBm)*
8: vivo Y33T (~75dBm)*
9: WiFi-UB.x (~81dBm)*
10: eduroam (~81dBm)*
11: eduroam (~85dBm)*
12: WiFi-UB.x (~86dBm)*
```

Lakukan proses wiring cable sesuai dengan diagram yang telah dibuat pada bab sebelumnya



Setelah memastikan wiring cable benar, langkah berikutnya adalah implementasi kode main.cpp di hardware ESP32. Lakukan modifikasi file main.cpp sebagai berikut

```
main.cpp x platformio.ini
src > main.cpp > loop()
1  #include <Arduino.h>
2  #include <WiFi.h>
3  #include <HttpClient.h>
4  #include "DHT.h"
5
6  #define DHTPIN 27
7  #define DHTTYPE DHT22
8
9  DHT dht(DHTPIN, DHTTYPE);
10
11 // Ganti dengan kredensial WiFi Anda
12 const char* ssid = "Oi";
13 const char* password = "iiiiiii";
14
15 unsigned long previousMillis = 0;
16 const long interval = 5000; // Interval 5 detik (5000 ms)
17
18 void setup() {
19     Serial.begin(115200);
20
21     // Hubungkan ke WiFi
22     WiFi.begin(ssid, password);
23     Serial.print("Menghubungkan ke WiFi");
24     while (WiFi.status() != WL_CONNECTED) {
25         delay(500);
```

Lakukan proses upload. Kemudian jalankan simulasi. Pastikan data yang dikirim dari hardware ESP32 dapat masuk ke database.

3. Results and Discussion (Hasil dan Pembahasan)

3.1 Experimental Results (Hasil Eksperimen)

3. Results and Discussion (Hasil dan Pembahasan)

3.1 Experimental Results (Hasil Eksperimen)