

**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)
PRAKTIK MENGECEK KONEKSI WIFI PADA HARDWARE ESP32**



Afi Kristiani
Fakultas Vokasi, Universitas Brawijaya
Email: kristianiafi@gmail.com

**Fakultas Vokasi
Universitas Brawijaya
Tahun Ajaran 2025**

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*

1.1. 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

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

2. Metodologi

2.1. Tools & Materials (Alat dan Bahan)

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

2.2. 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 :

```
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
13     Serial.println("Pemindaian Jaringan Wi-Fi Dimulai...");
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(" (");
```

Perhatikan bagian `monitor_speed` , baris tersebut berfungsi untuk melakukan preview serial monitor.



Kemudian klik tombol serial monitor

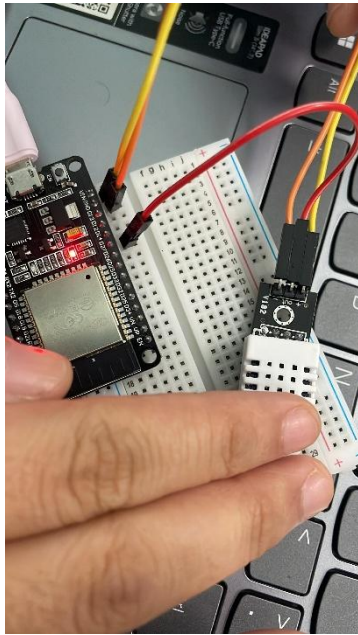
Pastikan tampilan serial monitor menunjukkan nama Access Point WIFI disekitar berikut juga dengan keterangan kekuatan sinyal seperti tampilan berikut ini

```
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.

Showing rows 0 - 7 (8 total. Query took 0.0004 seconds.)

```
SELECT * FROM `transaksi_sensor`
```

☐ Profiling [Edit inline](#) | [Edit](#) | [Explain SQL](#) | [Create PHP code](#) | [Refresh](#)

☐ Show all | Number of rows: 500 | Filter rows: Search this table | Sort by key: None

Extra options

				nama_sensor	nilai1	nilai2	created_at	updated_at
<input type="checkbox"/>				197 Sensor GD	50	27	2025-04-17 07 05 05	2025-04-17 07 05 05
<input type="checkbox"/>				198 Sensor GD	50	27	2025-04-17 07 05 10	2025-04-17 07 05 10
<input type="checkbox"/>				199 Sensor GD	51	27	2025-04-17 07 05 14	2025-04-17 07 05 14
<input type="checkbox"/>				200 Sensor GD	51	26	2025-04-17 07 05 19	2025-04-17 07 05 19
<input type="checkbox"/>				201 Sensor GD	51	26	2025-04-17 07 05 24	2025-04-17 07 05 24
<input type="checkbox"/>				202 Sensor GD	51	26	2025-04-17 07 05 30	2025-04-17 07 05 30
<input type="checkbox"/>				203 Sensor GD	51	26	2025-04-17 07 05 34	2025-04-17 07 05 34
<input type="checkbox"/>				204 Sensor GD	51	26	2025-04-17 07 05 38	2025-04-17 07 05 38

☐ Check all
 With selected:
 Edit
 Copy
 Delete
 Export

☐ Show all | Number of rows: 500 | Filter rows: Search this table | Sort by key: None

3. Hasil dan Pembahasan

3.1 Hasil Eksperimen

```
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 void setup() {
10   // WiFi
11   WiFi.begin("SSID", "PASSWORD");
12   while (WiFi.status() != WL_CONNECTED) {
13     delay(5000);
14     Serial.print("Connecting to WiFi... ");
15   }
16   Serial.println("Connected to WiFi!");
17   // DHT
18   DHT dht(DHTPIN, DHTTYPE);
19 }
20
21 void loop() {
22   // DHT
23   float h = dht.readHumidity();
24   float t = dht.readTemperature();
25   // HTTP
26   HTTPClient http;
27   http.begin("http://192.168.1.100:8080/api/sensor");
28   http.POST("{\"nama_sensor\":\"Sensor GD\", \"nilai1\":54.00, \"nilai2\":27.00}");
29   int httpCode = http.GET();
30   if (httpCode == HTTP_200) {
31     String payload = http.getString();
32     Serial.println(payload);
33   }
34   http.end();
35   delay(1000);
36 }
```

--- More details at <https://bit.ly/gpio-monitor-filters>
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H
Menghubungkan ke WiFi. Terhubung!
{\"nama_sensor\":\"Sensor GD\", \"nilai1\":54.00, \"nilai2\":27.00}
Kode respons HTTP: 201
Responses dari server:
{\"data\":{\"id\":\"139\", \"nama_sensor\":\"Sensor GD\", \"nilai1\":54, \"nilai2\":27}}
{\"nama_sensor\":\"Sensor GD\", \"nilai1\":54.00, \"nilai2\":27.00}
Kode respons HTTP: 201
Responses dari server:
{\"data\":{\"id\":\"139\", \"nama_sensor\":\"Sensor GD\", \"nilai1\":54, \"nilai2\":27}}
{\"nama_sensor\":\"Sensor GD\", \"nilai1\":54.00, \"nilai2\":27.00}
Kode respons HTTP: 201
Responses dari server:
{\"data\":{\"id\":\"140\", \"nama_sensor\":\"Sensor GD\", \"nilai1\":54, \"nilai2\":27}}
{\"nama_sensor\":\"Sensor GD\", \"nilai1\":54.00, \"nilai2\":27.00}
Kode respons HTTP: 201
Responses dari server:
{\"data\":{\"id\":\"141\", \"nama_sensor\":\"Sensor GD\", \"nilai1\":54, \"nilai2\":27}}
{\"nama_sensor\":\"Sensor GD\", \"nilai1\":53.00, \"nilai2\":27.00}
Kode respons HTTP: 201
Responses dari server:
Showing rows 0 - 7 (8 total, Query took 0.0004 seconds)
SELECT * FROM \"transaksi_sensor\"
[Profiling | Edit inline | Edit | Explain SQL | Create PHP code | Refresh]
[Show all | Number of rows: 500 | Filter rows: Search this table | Sort by key: None]
Extra options
+ T +
id nama_sensor nilai1 nilai2 created_at updated_at
[Edit] [Copy] [Delete] 197 Sensor GD 50 27 2025-04-17 07:05:05 2025-04-17 07:05:05
[Edit] [Copy] [Delete] 198 Sensor GD 50 27 2025-04-17 07:05:10 2025-04-17 07:05:10
[Edit] [Copy] [Delete] 199 Sensor GD 51 27 2025-04-17 07:05:14 2025-04-17 07:05:14
[Edit] [Copy] [Delete] 200 Sensor GD 51 26 2025-04-17 07:05:19 2025-04-17 07:05:19
[Edit] [Copy] [Delete] 201 Sensor GD 51 26 2025-04-17 07:05:24 2025-04-17 07:05:24
[Edit] [Copy] [Delete] 202 Sensor GD 51 26 2025-04-17 07:05:30 2025-04-17 07:05:30
[Edit] [Copy] [Delete] 203 Sensor GD 51 26 2025-04-17 07:05:34 2025-04-17 07:05:34
[Edit] [Copy] [Delete] 204 Sensor GD 51 26 2025-04-17 07:05:38 2025-04-17 07:05:38
[Check all] [With selected:] [Edit] [Copy] [Delete] [Export]
[Show all | Number of rows: 500 | Filter rows: Search this table | Sort by key: None]

3.2 Kesimpulan

ESP32 dapat terhubung ke jaringan WiFi dengan baik apabila SSID dan password dimasukkan dengan benar. Hal ini ditandai dengan munculnya status “WiFi terhubung” dan alamat IP pada *Serial Monitor*. Keberhasilan koneksi ini menunjukkan bahwa ESP32 siap digunakan dalam proyek IoT yang membutuhkan konektivitas internet.

