

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



Nadia Aulia Zahra
Fakultas Vokasi, Universitas Brawijaya
Email: nadiaaulia@student.ub.ac.id

**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. 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

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 :

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

```
void setup() {  
  Serial.begin(115200);
```

```

WiFi.mode(WIFI_STA);
WiFi.disconnect();
delay(100);

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

void loop() {
  int n = WiFi.scanNetworks();
  Serial.println("Pemindaian Selesai");
  if (n == 0) {
    Serial.println("Tidak ada jaringan Wi-Fi yang ditemukan.");
  } else {
    Serial.print(n);
    Serial.println(" jaringan Wi-Fi ditemukan:");
    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("dBm)");
      Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN) ? " " : "*");
      delay(10);
    }
  }
  Serial.println("");

  delay(5000); // Lakukan pemindaian setiap 5 detik
}

```

Lakukan proses **upload**

Ubah kembali file platformio.ini sebagai berikut

```

; PlatformIO Project Configuration File
;
; Build options: build flags, source filter
; Upload options: custom upload port, speed and extra flags
; Library options: dependencies, extra library storages
; Advanced options: extra scripting
;
; Please visit documentation for the other options and examples
; https://docs.platformio.org/page/projectconf.html

```

```
[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
upload_port = COM3
monitor_port = COM3
monitor_speed = 115200
```

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

```
▼ TERMINAL
2: WiFi-UB.x (-66dBm)*
3: eduroam (-66dBm)*
4: WiFi-UB.x (-74dBm)*
5: eduroam (-74dBm)*
6: WiFi-UB.x (-81dBm)*

Pemindaian Selesai
8 jaringan Wi-Fi ditemukan:
1: Lab IT (-57dBm)*
2: WiFi-UB.x (-65dBm)*
3: eduroam (-65dBm)*
4: WiFi-UB.x (-73dBm)*
5: eduroam (-74dBm)*
6: eduroam (-82dBm)*
7: WiFi-UB.x (-83dBm)*
8: WiFi-UB.x (-85dBm)*
```

Pada langkah ini, hardware ESP32 telah berhasil melakukan scanning WIFI disekitar. Langkah berikutnya adalah implementasi Internet of Things dengan menghubungkan sensor suhu dan kelembaban ke sistem API dan database yang telah dibuat. Proses ini sudah pernah dilakukan pada bab 13, namun diperlukan beberapa penyesuaian dan implementasi koding ke hardware ESP32 nyata (bukan simulator WOKWI)

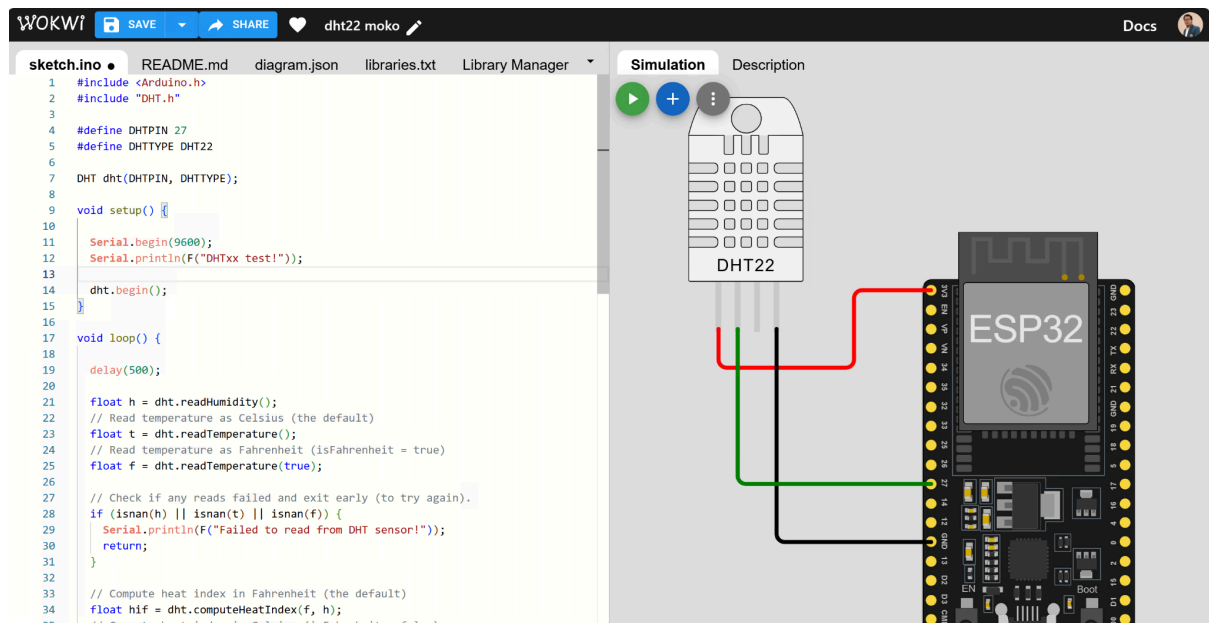
Jalankan API laravel kembali dengan perintah

```
php artisan serve --host=0.0.0.0 --port=8080
```

Kemudian jalankan NGROK

```
ngrok http --scheme=http 8080
```

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

```
#include <Arduino.h>
```

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

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

```
#include "DHT.h"
```

```
#define DHTPIN 27
```

```
#define DHTTYPE DHT22
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
// Ganti dengan kredensial WiFi Anda
```

```
const char* ssid = "Wokwi-GUEST";
```

```
const char* password = "";
```

```
unsigned long previousMillis = 0;
```

```
const long interval = 5000; // Interval 5 detik (5000 ms)
```

```
void setup() {  
  Serial.begin(115200);  
  
  // Hubungkan ke WiFi  
  WiFi.begin(ssid, password);  
  Serial.print("Menghubungkan ke WiFi");  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
  Serial.println(" Terhubung!");  
  
  dht.begin();  
  
  // Tunggu sebentar agar koneksi stabil  
  delay(1000);  
}  
  
void loop() {  
  unsigned long currentMillis = millis();  
  
  // Lakukan POST setiap interval yang telah ditentukan  
  if (currentMillis - previousMillis >= interval) {  
    previousMillis = currentMillis;  
  
    float h = round(dht.readHumidity());  
    // Read temperature as Celsius (the default)  
    float t = round(dht.readTemperature());  
  
    // Check if any reads failed and exit early (to try again).  
    if (isnan(h) || isnan(t)) {  
      Serial.println(F("Failed to read from DHT sensor!"));  
      return;  
    }  
  }
```

```
// Compute heat index in Celsius (isFahreheit = false)
```

```
float hic = dht.computeHeatIndex(t, h, false);
```

```
// Inisialisasi HTTPClient
```

```
HttpClient http;
```

```
String url = "http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app/api/posts"; // Ganti  
dengan URL ngrok yang benar
```

```
http.begin(url); // Menggunakan HTTP, bukan HTTPS
```

```
http.addHeader("Content-Type", "application/json");
```

```
String payload = "{\"nama_sensor\":\"Sensor GD\", \"nilai1\": \"\" + String(h) + \"\", \"nilai2\": \"\" + String(t) + \"\"}";
```

```
Serial.println(payload); // Untuk melihat apakah payload sudah terbentuk dengan benar
```

```
// Kirim POST request
```

```
int httpResponseCode = http.POST(payload);
```

```
// Tampilkan kode respons HTTP
```

```
Serial.print("Kode respons HTTP: ");
```

```
Serial.println(httpResponseCode);
```

```
// Tampilkan respons dari server jika request berhasil
```

```
if (httpResponseCode == 200 || httpResponseCode == 201) {
```

```
String response = http.getString();
```

```
Serial.println("Respons dari server:");
```

```
Serial.println(response);
```

```
} else {
```

```
Serial.println("Gagal mengirim data");
```

```
}
```



```
// Tutup koneksi HTTP
http.end();
}
}
```

Ubah bagian ini

```
// Ganti dengan kredensial WiFi Anda
const char* ssid = "Wokwi-GUEST";
const char* password = "";
```

Sesuaikan dengan WIFI access point yang akan dihubungkan. *(Gunakan WIFI tethering dari smartphone)*

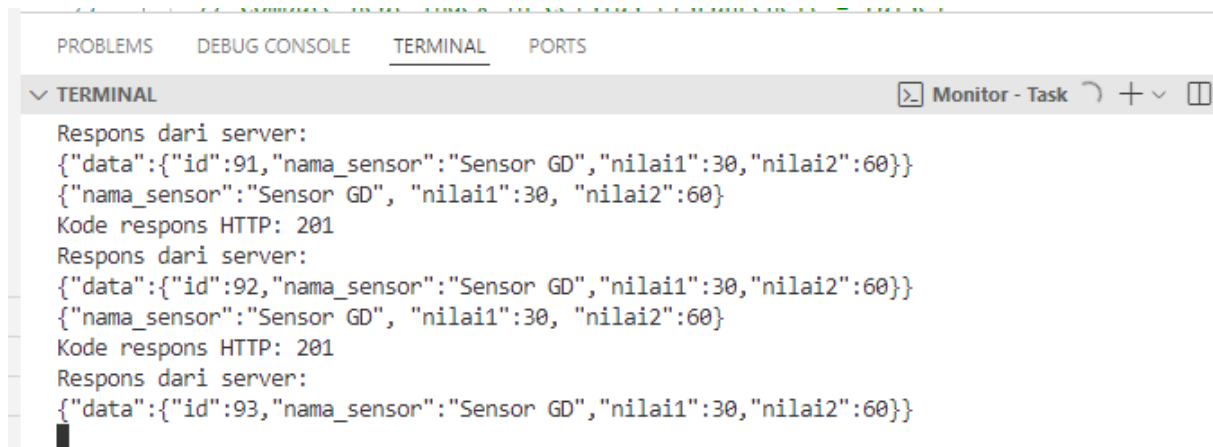
```
String url = "http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app/api/posts"; // Ganti
dengan URL ngrok yang benar
```

Ubah dan sesuaikan dengan alamat URL ngrok Anda. **pastikan HTTP bukan HTTPS.**

Kemudian lakukan modifikasi kembali pada file platformio.ini dengan tambahan baris sebagai berikut

```
[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
upload_port = COM3
monitor_port = COM3
monitor_speed = 115200
lib_deps =
  adafruit/DHT sensor library@^1.4.4
  adafruit/Adafruit Unified Sensor@^1.1.14
```

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



```
PROBLEMS  DEBUG CONSOLE  TERMINAL  PORTS
▼ TERMINAL  Monitor - Task  + - □
Respons dari server:
{"data":{"id":91,"nama_sensor":"Sensor GD","nilai1":30,"nilai2":60}}
{"nama_sensor":"Sensor GD", "nilai1":30, "nilai2":60}
Kode respons HTTP: 201
Respons dari server:
{"data":{"id":92,"nama_sensor":"Sensor GD","nilai1":30,"nilai2":60}}
{"nama_sensor":"Sensor GD", "nilai1":30, "nilai2":60}
Kode respons HTTP: 201
Respons dari server:
{"data":{"id":93,"nama_sensor":"Sensor GD","nilai1":30,"nilai2":60}}
█
```

3. Hasil dan Pembahasan

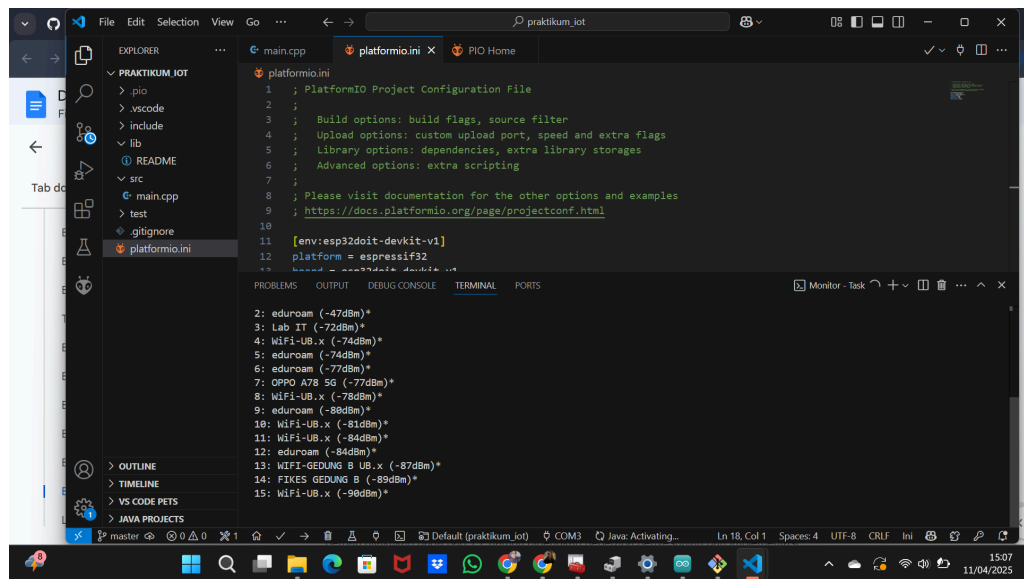
3.1 Hasil Eksperimen

- A. Setelah program diunggah ke ESP32, Serial Monitor menunjukkan proses koneksi ke WiFi dengan tampilan titik-titik (“.....”). Setelah beberapa saat, muncul pesan “WiFi terhubung!” disertai alamat IP seperti 192.168.1.xxx. Hal ini menandakan bahwa ESP32 berhasil terhubung ke jaringan WiFi dan mendapatkan alamat IP dari router.

3.2 Kesimpulan

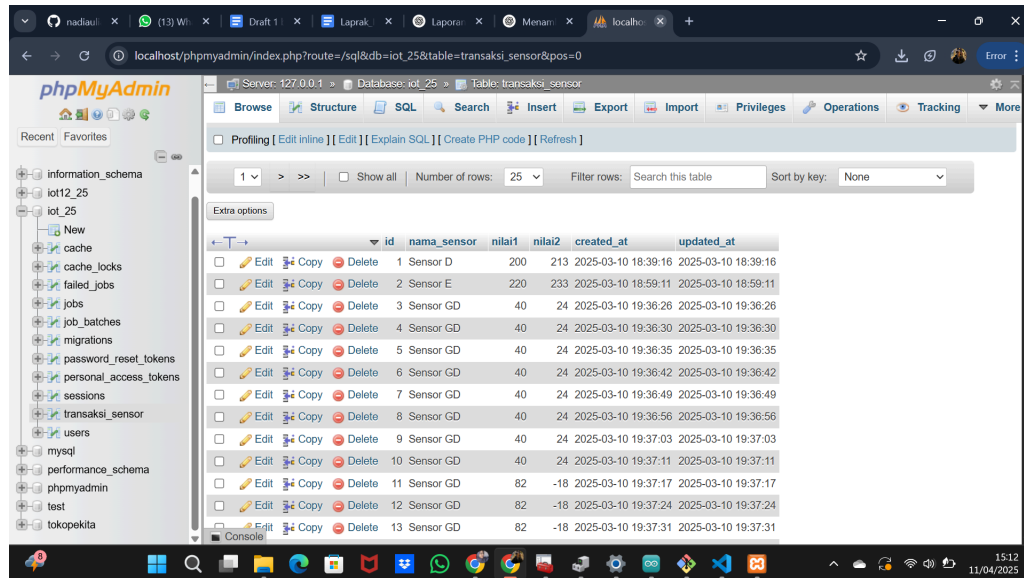
ESP32 dapat terhubung ke jaringan WiFi dengan baik apabila SSID dan password dimasukkan dengan benar. Hal ini ditandai dengan munculnya status “WiFi terhubung” dan alamat IP pada *Serial Monitor*. Keberhasilan koneksi ini menunjukkan bahwa ESP32 siap digunakan dalam proyek IoT yang membutuhkan konektivitas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

4. Lampiran Jika diperlukan



The screenshot shows the VS Code interface with the `platformio.ini` file open. The file contains the following configuration:

```
1 ; PlatformIO Project Configuration File
2
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14
15 ; Board specific configuration
16 ;
17 ; 1: esp32doit-devkit-v1
18 ; 2: eduroam (-47dBm)*
19 ; 3: Lab IT (-72dBm)*
20 ; 4: WiFi-UB.x (-74dBm)*
21 ; 5: eduroam (-74dBm)*
22 ; 6: eduroam (-77dBm)*
23 ; 7: OPPO A78 5G (-77dBm)*
24 ; 8: WiFi-UB.x (-78dBm)*
25 ; 9: eduroam (-80dBm)*
26 ; 10: WiFi-UB.x (-81dBm)*
27 ; 11: WiFi-UB.x (-84dBm)*
28 ; 12: eduroam (-84dBm)*
29 ; 13: WiFi-GEDUNG B UB.x (-87dBm)*
30 ; 14: FIXES GEDUNG B (-89dBm)*
31 ; 15: WiFi-UB.x (-90dBm)*
```



The screenshot shows the phpMyAdmin interface with the `transaksi_sensor` table selected. The table contains 13 rows of data.

id	nama_sensor	nilai1	nilai2	created_at	updated_at
1	Sensor D	200	213	2025-03-10 18:39:16	2025-03-10 18:39:16
2	Sensor E	220	233	2025-03-10 18:59:11	2025-03-10 18:59:11
3	Sensor GD	40	24	2025-03-10 19:36:26	2025-03-10 19:36:26
4	Sensor GD	40	24	2025-03-10 19:36:30	2025-03-10 19:36:30
5	Sensor GD	40	24	2025-03-10 19:36:35	2025-03-10 19:36:35
6	Sensor GD	40	24	2025-03-10 19:36:42	2025-03-10 19:36:42
7	Sensor GD	40	24	2025-03-10 19:36:49	2025-03-10 19:36:49
8	Sensor GD	40	24	2025-03-10 19:36:56	2025-03-10 19:36:56
9	Sensor GD	40	24	2025-03-10 19:37:03	2025-03-10 19:37:03
10	Sensor GD	40	24	2025-03-10 19:37:11	2025-03-10 19:37:11
11	Sensor GD	82	-18	2025-03-10 19:37:17	2025-03-10 19:37:17
12	Sensor GD	82	-18	2025-03-10 19:37:24	2025-03-10 19:37:24
13	Sensor GD	82	-18	2025-03-10 19:37:31	2025-03-10 19:37:31

