# **LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)**

**Praktik Real Hardware ESP32**

*Ester Yesarela Tamelab*

*Fakultas Vokasi, Universitas Brawijaya*

*Email:* [*estertamelab7@gmail.com*](mailto:estertamelab7@gmail.com)

**Abstrak**

Praktik ini bertujuan untuk menerapkan pengembangan sistem Internet of Things (IoT) secara nyata menggunakan mikrokontroler ESP32. Proses dimulai dengan memastikan perangkat ESP32 dikenali dengan baik oleh komputer melalui pengecekan driver dan port komunikasi. Selanjutnya, proses pemrograman dilakukan menggunakan **Visual Studio Code (VSCode)** dengan ekstensi **PlatformIO** sebagai lingkungan pengembangan. Program pertama yang diunggah bertujuan untuk menguji koneksi perangkat ke jaringan WiFi lokal guna memastikan kemampuan ESP32 dalam menjalin koneksi ke internet.

Pada tahap berikutnya, dilakukan integrasi sensor suhu dan kelembaban ke dalam sistem, dan data dari sensor dikirimkan secara real-time ke API dan database berbasis **Laravel**. Komunikasi antara ESP32 dan server dilakukan melalui *endpoint* publik menggunakan **NGROK** sebagai jembatan antara server lokal dan internet. Praktik ini membuktikan kemampuan ESP32 dalam mengirim data sensor ke server cloud secara langsung menggunakan jaringan WiFi, sekaligus meningkatkan pemahaman terhadap pengembangan sistem IoT berbasis hardware nyata.

***Kata Kunci***

*ESP32, Internet of Things (IoT), PlatformIO, Visual Studio Code, WiFi Connectivity, NGROK, Laravel API, Temperature and Humidity Sensor, Real-time Monitoring, Microcontroller.*

**1**

**PENDAHULUAN**

**1.1 Latar Belakang**

Perkembangan teknologi Internet of Things (IoT) semakin pesat dan telah menjadi bagian penting dalam pengembangan sistem otomatisasi dan pemantauan real-time. Salah satu perangkat utama yang digunakan dalam pengembangan sistem IoT adalah **ESP32**, sebuah mikrokontroler berbasis WiFi dan Bluetooth dengan kemampuan komputasi yang mumpuni serta harga yang terjangkau.

Dalam praktik ini, dilakukan eksplorasi langsung pada **hardware ESP32 nyata** (bukan simulasi) untuk memahami proses pengembangan dan integrasi sistem IoT secara menyeluruh. Langkah awal dimulai dengan memastikan bahwa ESP32 dikenali oleh komputer, sehingga dapat dikonfigurasi dan diprogram melalui **Visual Studio Code (VSCode)** menggunakan **PlatformIO** sebagai lingkungan pengembangannya.

Selanjutnya, eksperimen berfokus pada proses **upload program** ke board ESP32 dan melakukan **pemeriksaan konektivitas WiFi**, memastikan perangkat dapat terhubung ke Access Point di sekitarnya. Praktik ini menjadi dasar sebelum melangkah lebih lanjut ke tahap **implementasi sistem IoT**, yakni menghubungkan sensor suhu dan kelembaban ke **API** dan **database Laravel** yang telah dibangun sebelumnya.

Sistem backend dijalankan kembali menggunakan perintah php artisan serve pada Laravel, dengan konfigurasi host dan port tertentu. Untuk menghubungkan sistem lokal ke internet, digunakan **NGROK** sebagai tunneling tool agar ESP32 dapat mengakses API yang berjalan secara lokal. Implementasi ini mengajarkan bagaimana menyusun dan menjalankan sistem IoT end-to-end secara nyata, mulai dari perangkat keras, komunikasi data, hingga integrasi backend.

**1.2 Tujuan Eksperimen**

Eksperimen ini bertujuan untuk memahami dan menerapkan secara langsung proses pengembangan sistem berbasis **Internet of Things (IoT)** menggunakan **ESP32** sebagai perangkat utama. Praktik dilakukan dengan menggunakan **Visual Studio Code (VSCode)** dan **PlatformIO** sebagai lingkungan pemrograman, serta mengintegrasikan sistem dengan **API dan database Laravel** melalui koneksi internet.

Adapun tujuan spesifik dari eksperimen ini adalah sebagai berikut:

1. Memastikan perangkat **ESP32** dikenali oleh komputer agar dapat diprogram secara langsung.
2. Melakukan **pengkodean dan upload program** ke hardware ESP32 menggunakan PlatformIO.
3. Melakukan **uji konektivitas WiFi** pada ESP32 untuk memastikan perangkat dapat terhubung ke jaringan Access Point di sekitarnya.
4. Mengimplementasikan koneksi antara ESP32 dengan **sensor suhu dan kelembaban**, serta mengirimkan data ke server melalui API Laravel.
5. Menjalankan kembali API Laravel menggunakan php artisan serve dan menghubungkannya dengan **NGROK** agar bisa diakses oleh ESP32 melalui internet.
6. Menguji integrasi sistem secara menyeluruh, mulai dari pengambilan data sensor, pengiriman ke API, hingga verifikasi ke database.
7. Meningkatkan pemahaman praktis dalam pengembangan sistem IoT nyata dan kemampuan mengatasi kendala teknis di luar lingkungan simulasi.

**2**

**METODOLOGI**

**2.1 Alat dan Bahan**

Dalam eksperimen ini, sejumlah alat dan bahan diperlukan untuk mendukung proses pengembangan dan pengujian sistem berbasis mikrokontroler ESP32. Berikut adalah daftar alat dan bahan yang digunakan:

1. ESP 32
2. Kabel USB Type-C
3. Kabel Jumper (Female to Male)
4. Sensor DHT11
5. Lampu LED Kecil
6. Breadboard
7. Visual Studio Code (VSCode)
8. PlatformIO
9. Driver CP210x (Silicon Labs)
10. Laravel (API & Database)
11. Ngrok

**2.2 Langkah Implementasi**

Langkah-langkah dalam praktik ini melibatkan tiga tahapan utama, yaitu proses instalasi dan deteksi perangkat, penyusunan sistem dan wiring, serta pengkodean dan pengujian. Berikut adalah tahapan detailnya:

**A. Memastikan Perangkat ESP32 Terdeteksi oleh Komputer**

1. Hubungkan board ESP32 ke komputer menggunakan kabel USB Type-C.
2. Buka **Device Manager** pada sistem operasi Windows.
3. Perhatikan bagian **Ports (COM & LPT)**, pastikan muncul perangkat **Silicon Labs**.
4. Jika belum terinstal otomatis, lakukan langkah berikut:
   * Klik kanan pada perangkat > **Update Driver**.
   * Pilih **Browse my computer for drivers**.
   * Lalu pilih **Let me pick from a list of available drivers on my computer**.
   * Klik **Browse**, arahkan ke folder tempat file driver disimpan (pastikan folder sudah di-*extract*).
   * Klik **Next** untuk memulai instalasi driver.
5. Setelah instalasi selesai, pastikan tidak ada tanda error atau peringatan pada device manager. Ini menandakan bahwa perangkat ESP32 telah dikenali dengan baik oleh komputer.

**B. Menyiapkan Lingkungan Pengembangan dan Wiring Sistem**

1. **Buat folder proyek baru** dan buka melalui **Visual Studio Code (VSCode)**.
2. Pastikan extension **PlatformIO** telah terinstal dan aktif.
3. Pastikan perangkat ESP32 muncul pada daftar board terdeteksi di PlatformIO, misalnya dengan alamat port **COM5**.
4. Buat **proyek baru** di PlatformIO:
   * Pilih board: esp32doit-devkit-v1
   * Framework: Arduino
5. Ubah dan sesuaikan isi file platformio.ini sebagai berikut:

[env:esp32doit-devkit-v1]

platform = espressif32

board = esp32doit-devkit-v1

framework = arduino

upload\_port = COM5

monitor\_port = COM5

1. Lakukan wiring sesuai kebutuhan eksperimen, misalnya:

* Sambungkan **kabel jumper (female to male)** dari pin digital ESP32 ke **LED kecil** (dengan resistor bila perlu).
* Pastikan koneksi ground dan power juga tersambung sesuai standar.

**C. Menulis dan Mengunggah Program ke ESP32**

1. Buka file src/main.cpp di dalam proyek PlatformIO.
2. Tulis atau tempelkan program untuk mengatur logika nyala/mati LED
3. Simpan file, lalu klik menu **Upload** pada toolbar PlatformIO.
4. PlatformIO akan menjalankan proses **compiling** dan **upload** ke perangkat ESP32 melalui port yang telah ditentukan.
5. Jika proses berhasil dan wiring benar, maka **lampu LED akan menyala berkedip sesuai logika program**, sebagai bukti keberhasilan upload dan eksekusi kode.

Langkah-langkah berikut dilakukan untuk menyusun sistem, melakukan pengkodean, dan melakukan pengujian koneksi WiFi pada perangkat mikrokontroler ESP32 menggunakan Visual Studio Code (VSCode) dengan PlatformIO. Tujuan utamanya adalah memastikan ESP32 dapat terhubung ke jaringan WiFi sebagai tahap awal implementasi sistem IoT.

**1. Persiapan Awal**

* Pastikan driver ESP32 (CP210x atau CH340) sudah terinstal dan perangkat dikenali oleh komputer.
* Hubungkan ESP32 ke komputer menggunakan kabel USB Type-C.
* Buka *Device Manager* dan periksa bagian **Ports (COM & LPT)**. Pastikan muncul perangkat seperti "Silicon Labs CP210x".
* Jika belum dikenali, lakukan:
  + Klik kanan → *Update Driver*.
  + Pilih *Browse my computer for drivers* → *Let me pick from a list of available drivers*.
  + Arahkan ke folder driver yang sudah diunduh dan diekstrak, lalu lanjutkan instalasi.

**2. Pengaturan Lingkungan Pengembangan**

* Buka **Visual Studio Code**, lalu install dan aktifkan ekstensi **PlatformIO IDE**.
* Buat **project baru** di PlatformIO dengan memilih board: esp32doit-devkit-v1.
* Modifikasi file platformio.ini

**3. Penulisan Kode untuk Cek Koneksi WiFi**

* Buka file src/main.cpp, lalu tuliskan program untuk menyambungkan ESP32 ke jaringan WiFi

**4. Upload dan Uji Program**

* Klik **Upload** di PlatformIO untuk mengompilasi dan mengunggah kode ke ESP32.
* Buka **Serial Monitor** untuk melihat hasil koneksi:
  + Jika sukses, akan tampil pesan “Terhubung ke WiFi!” dan alamat IP dari ESP32.
  + Jika gagal, pastikan SSID dan password sudah benar, serta sinyal WiFi terjangkau.

**5. Validasi dan Pengujian**

* Ulangi proses dengan jaringan WiFi lain untuk memastikan fleksibilitas koneksi.
* Catat hasil IP address yang muncul sebagai bukti koneksi berhasil.

Setelah hardware ESP32 berhasil melakukan scanning dan koneksi ke jaringan WiFi di sekitarnya, tahap berikutnya adalah mengimplementasikan sistem **Internet of Things (IoT)** secara nyata. Dalam eksperimen ini, ESP32 akan dihubungkan dengan sensor suhu dan kelembaban, kemudian mengirimkan data sensor tersebut ke **API berbasis Laravel** dan menyimpannya ke dalam database melalui endpoint yang telah disiapkan.

Berikut langkah-langkah yang dilakukan:

**1. Menjalankan API Laravel**

* Pastikan project API Laravel yang sebelumnya telah dibuat dalam kondisi siap dijalankan.
* Jalankan kembali server API dengan perintah berikut di terminal: php artisan serve --host=0.0.0.0 --port=8080
* Perintah ini akan menjalankan Laravel di jaringan lokal sehingga dapat diakses oleh ESP32 melalui koneksi internet.

**2. Menyambungkan Laravel ke Internet via NGROK**

* Untuk memungkinkan ESP32 mengakses server Laravel secara publik (dari jaringan luar), gunakan NGROK.
* Jalankan NGROK dengan perintah: ngrok http --scheme=http 8080
* Simpan **URL NGROK** yang muncul, karena alamat ini akan digunakan dalam kode ESP32 sebagai endpoint API tujuan.

**3. Wiring Perangkat Keras**

* Lakukan proses **wiring** antara ESP32 dan sensor suhu & kelembaban (seperti DHT11 atau DHT22) sesuai dengan diagram yang telah dibuat pada bab sebelumnya.
* Pastikan koneksi kabel jumper (VCC, GND, dan Data) sesuai dan stabil agar pembacaan data sensor berjalan dengan baik.

**4. Penyesuaian dan Penulisan Kode pada ESP32**

* Gunakan **PlatformIO di VSCode** untuk membuat proyek baru atau melanjutkan proyek sebelumnya.
* Lakukan modifikasi pada kode program agar:
  + ESP32 membaca data suhu dan kelembaban dari sensor.
  + Data dikirim melalui metode HTTP POST ke alamat API yang dihasilkan oleh NGROK.

**5. Upload dan Pengujian**

* Klik **Upload** pada PlatformIO untuk mengunggah kode ke ESP32.
* Buka **Serial Monitor** untuk memastikan data sensor berhasil dibaca dan respons API diterima dengan baik.
* Lakukan pengecekan ke database untuk melihat apakah data berhasil tersimpan.

**3**

**HASIL DAN PEMBAHASAN**

**3.1 Hasil Eksperimen**

Eksperimen pemrograman **ESP32** menggunakan **Visual Studio Code (VSCode)** dan **PlatformIO** berhasil dilakukan sesuai dengan langkah-langkah yang telah direncanakan. Adapun hasil yang diperoleh dari pelaksanaan praktik ini antara lain:

1. **Pengenalan Perangkat oleh Komputer**

Setelah ESP32 disambungkan ke komputer melalui kabel USB Type-C, perangkat berhasil dikenali oleh sistem dan ditampilkan pada **Device Manager** di bagian **Ports (COM & LPT)** sebagai **Silicon Labs CP210x USB to UART Bridge**. Proses instalasi driver berjalan tanpa kendala.

1. **Pengaturan Lingkungan PlatformIO**

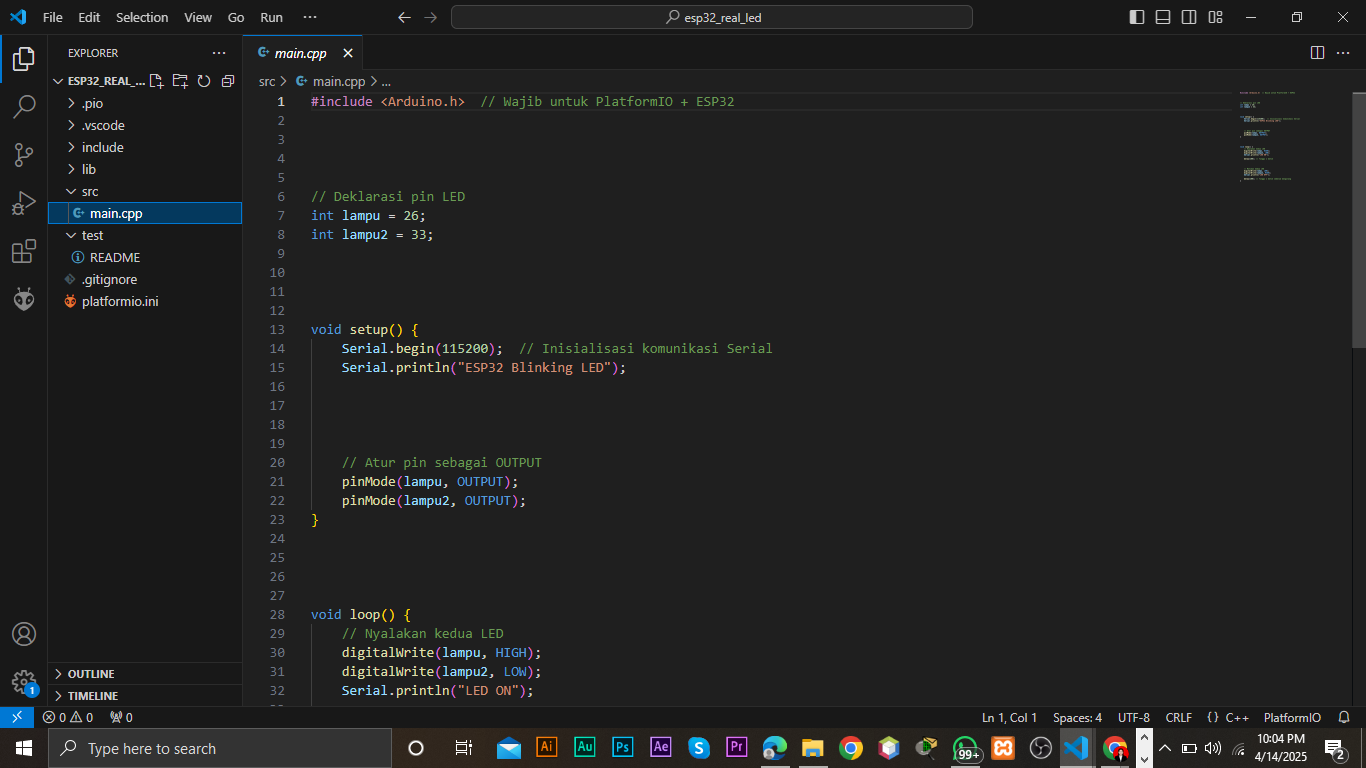
Proyek baru dengan konfigurasi board esp32doit-devkit-v1 berhasil dibuat di PlatformIO. Pengaturan file platformio.ini telah disesuaikan, dan perangkat ESP32 terdeteksi melalui port **COM5**, menandakan bahwa koneksi dengan perangkat telah berlangsung dengan baik.

1. **Penulisan dan Pengunggahan Program**

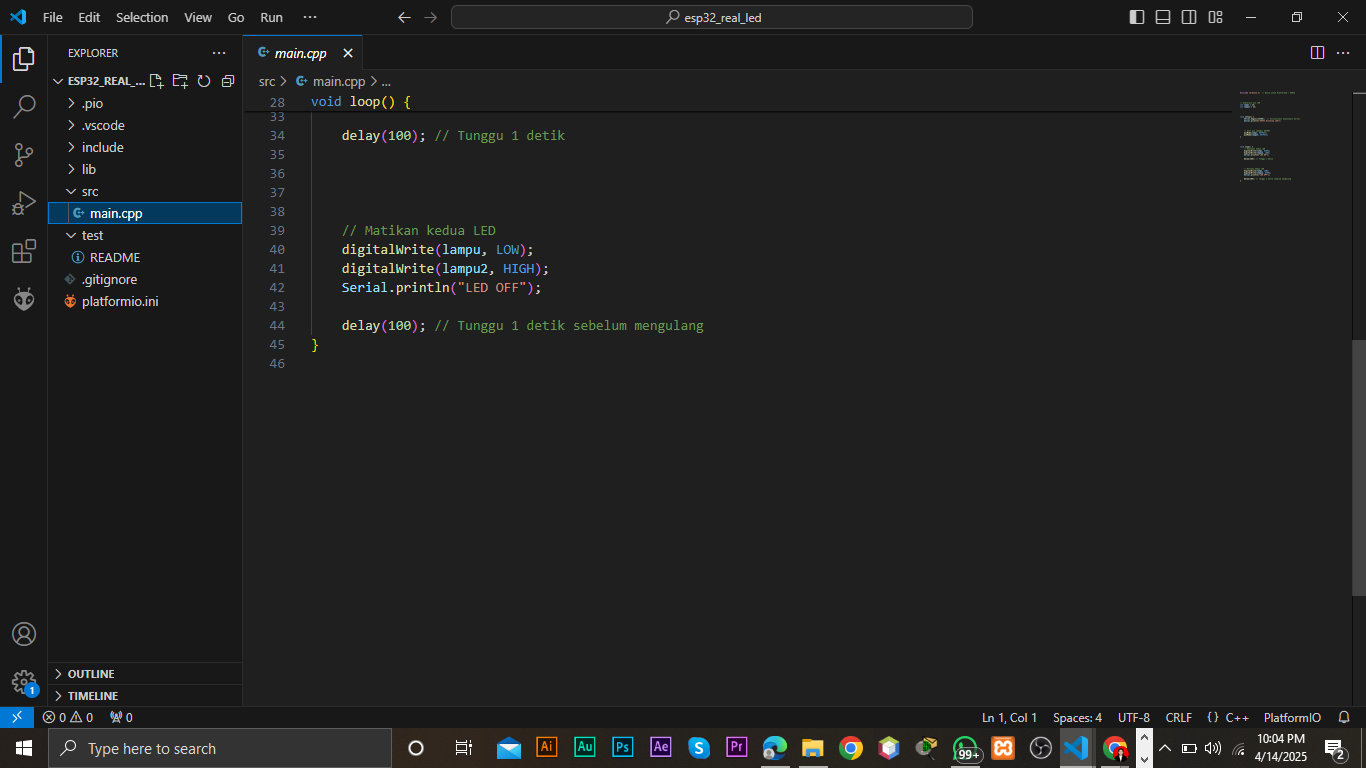
Program sederhana untuk menyalakan dan mematikan **lampu LED eksternal** telah ditulis dalam file main.cpp. Proses **kompilasi** dan **unggah** program ke perangkat ESP32 melalui VSCode berhasil tanpa error.

1. **Hasil Implementasi Fisik**

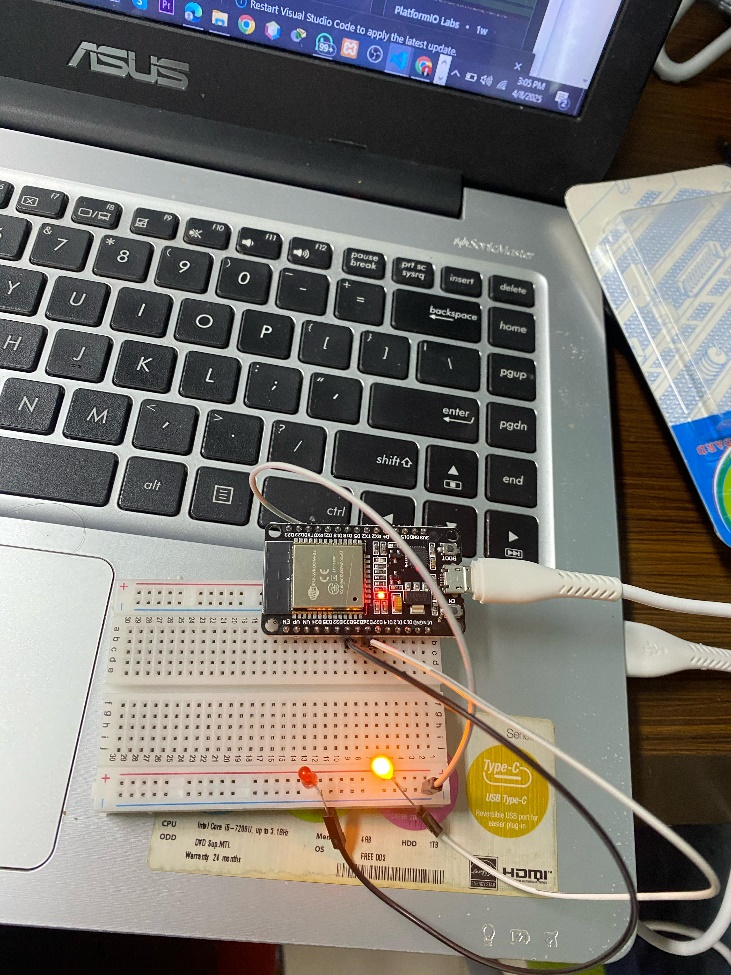
Setelah sistem kabel dan perangkat keras dirangkai sesuai skema, LED berhasil menyala dan berkedip setiap 1 detik, yang menandakan bahwa ESP32 menjalankan program dengan benar.



Gambar 3.1



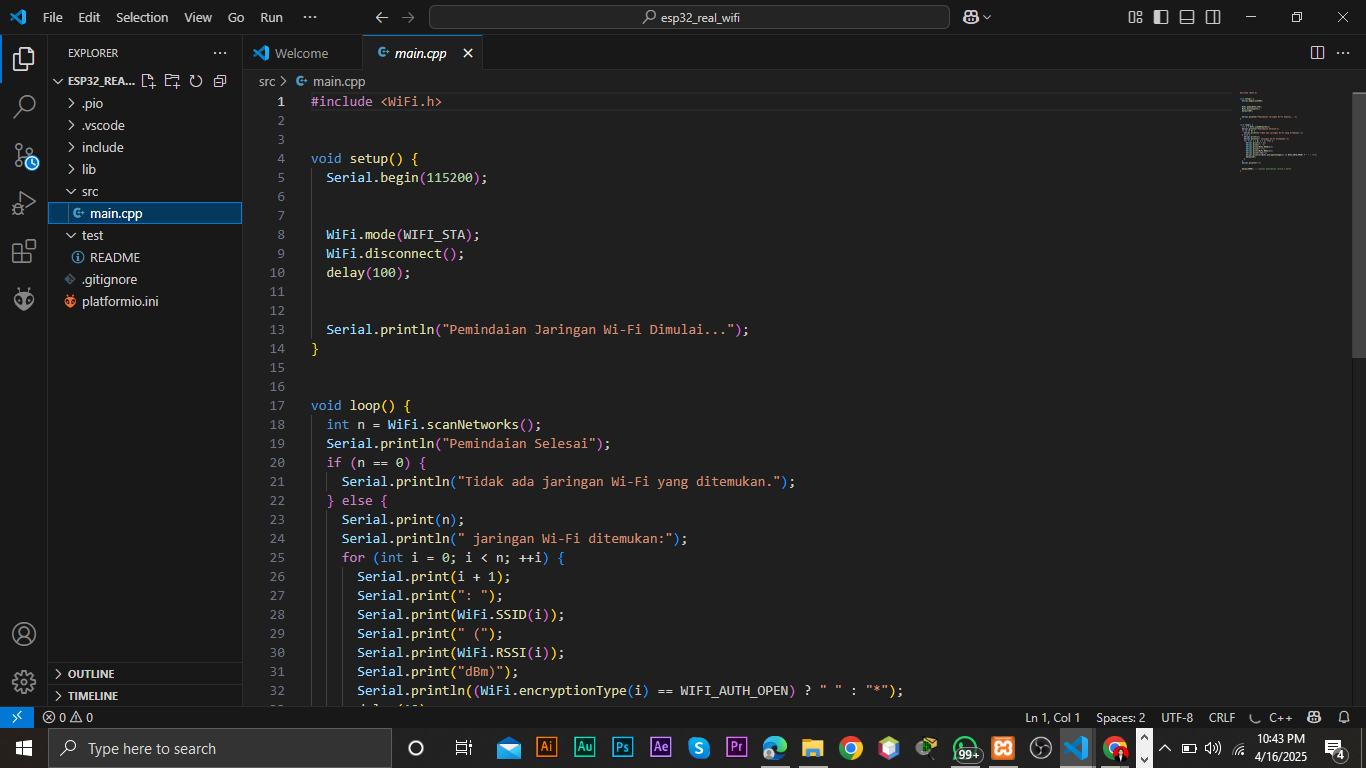
Gambar 3.2



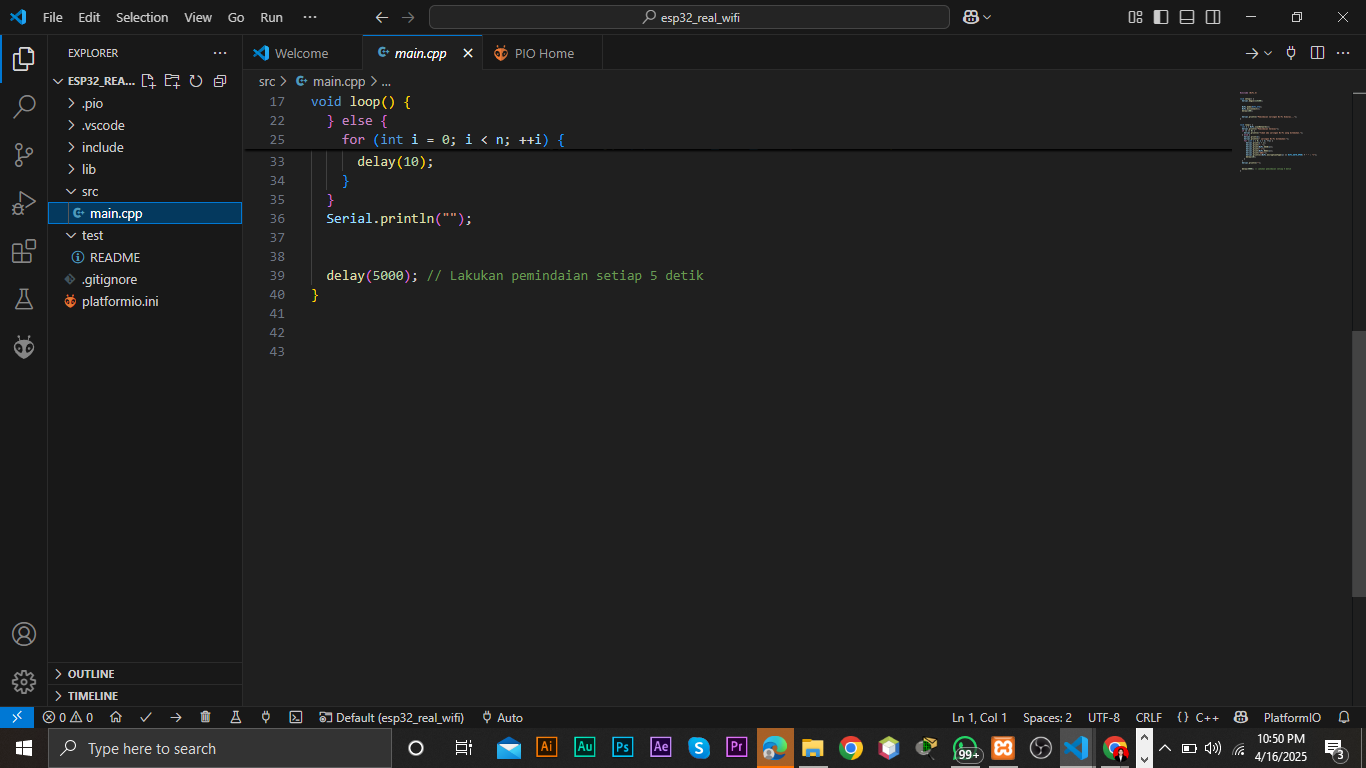
Gambar 3.3

Pada eksperimen ini, dilakukan pengujian terhadap kemampuan perangkat keras ESP32 dalam mendeteksi dan terhubung ke jaringan WiFi yang tersedia di sekitarnya. Eksperimen dimulai dengan melakukan modifikasi kode pada file main.cpp yang digunakan dalam proyek PlatformIO.

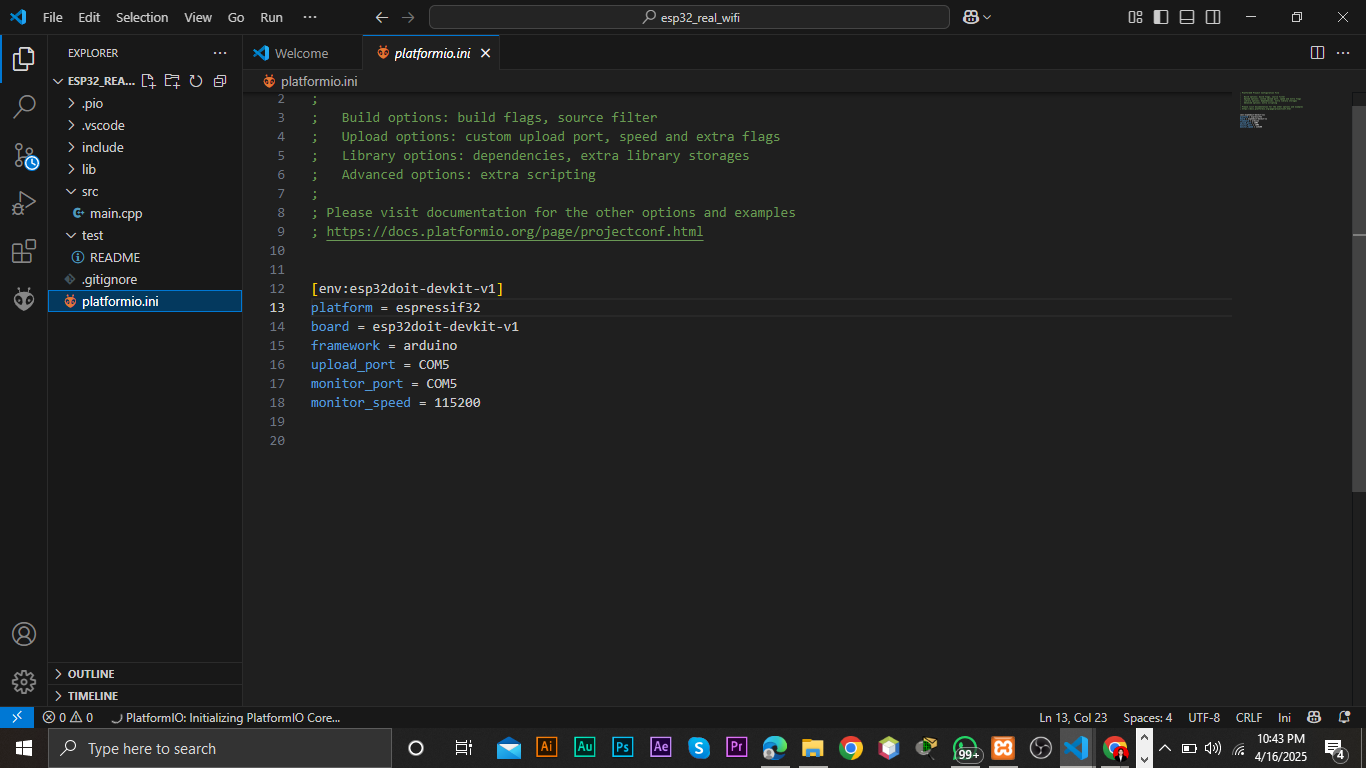
* + 1. Kode program dibuat agar ESP32 melakukan *scanning* jaringan WiFi dan menampilkan daftar SSID (nama jaringan) beserta kekuatan sinyal (RSSI) melalui **Serial Monitor**.
    2. Setelah kode diunggah ke perangkat ESP32 menggunakan fitur **Upload** pada PlatformIO, hasil yang diperoleh ditampilkan melalui Serial Monitor. ESP32 berhasil mendeteksi beberapa jaringan WiFi.



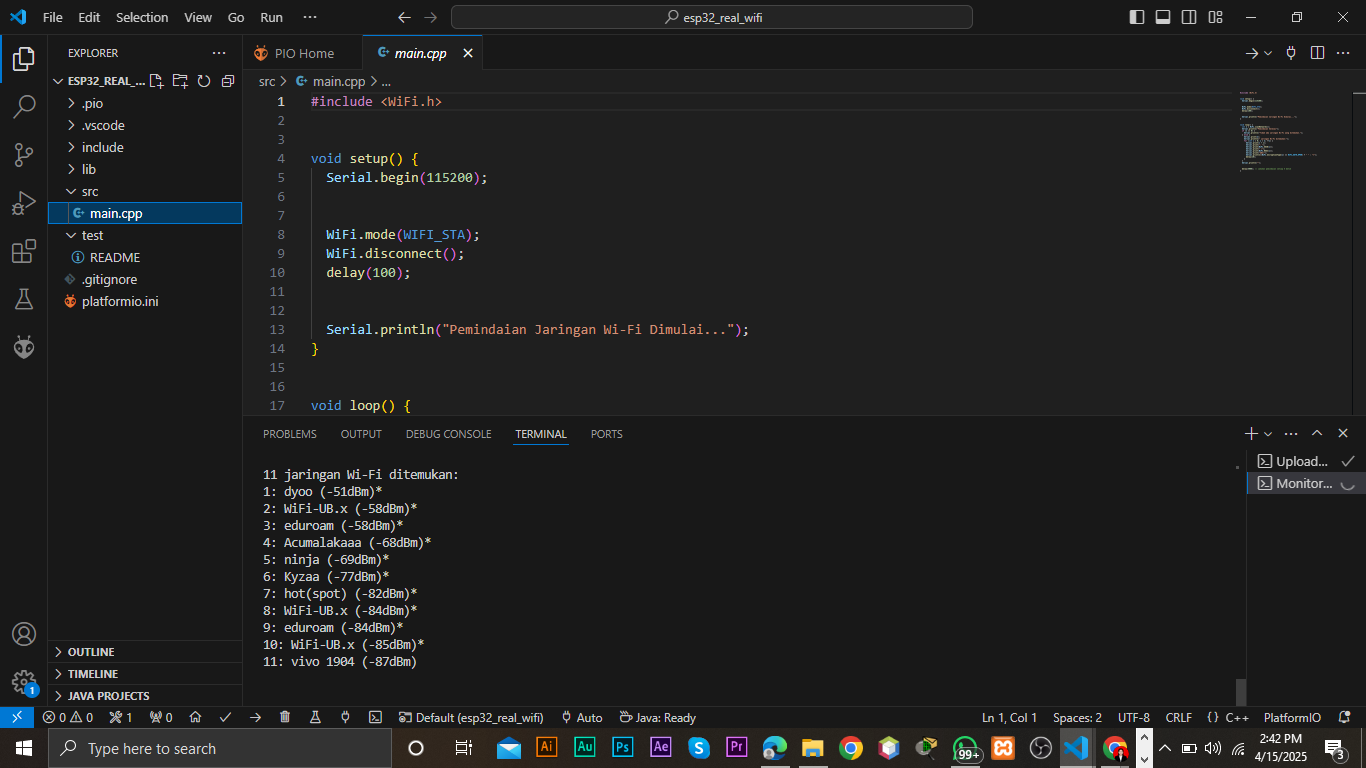
Gambar 3.4



Gambar 3.5



Gambar 3.6



Gambar 3.7

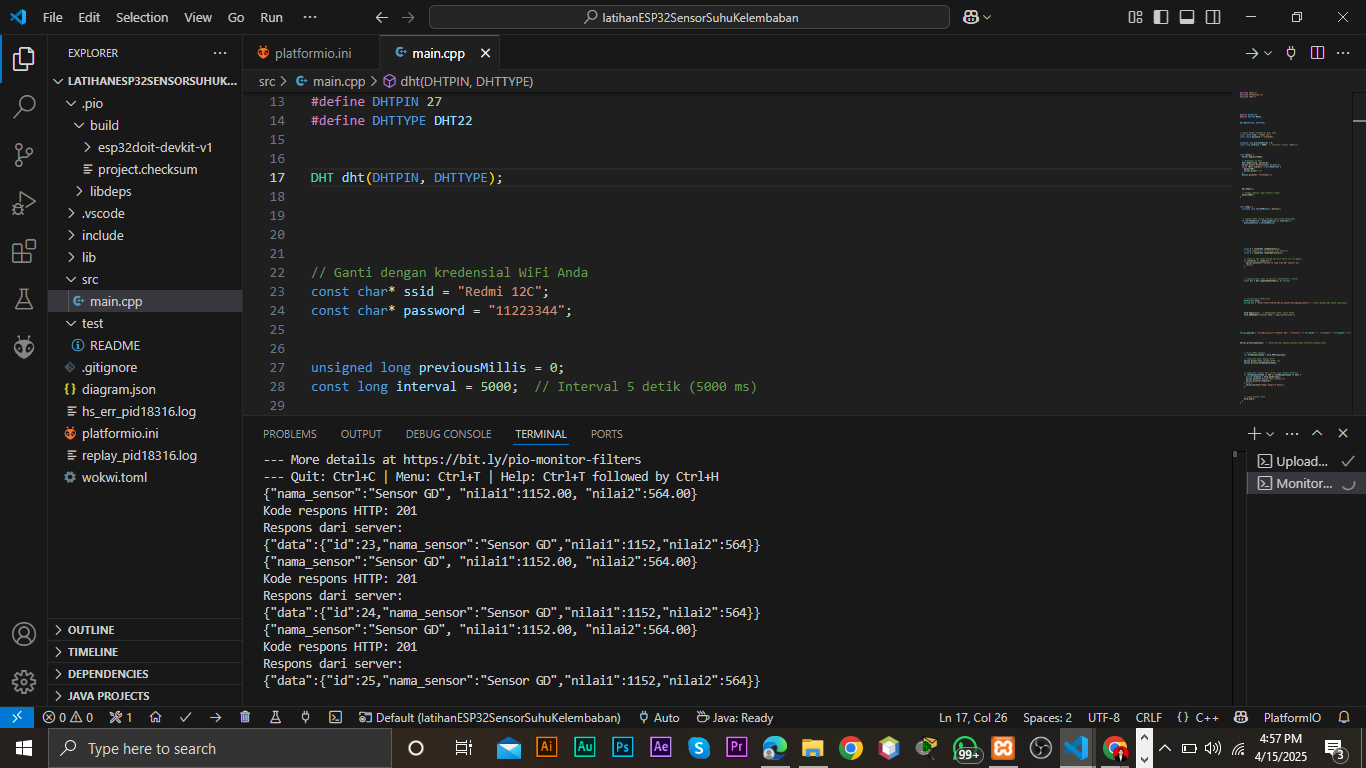
Pada tahap ini, eksperimen dilakukan untuk menguji koneksi dan komunikasi antara perangkat ESP32 nyata dengan jaringan WiFi serta integrasinya ke dalam sistem Internet of Things (IoT). Hardware ESP32 telah berhasil melakukan proses pemindaian (scanning) terhadap jaringan WiFi di sekitarnya. Hasil dari scanning ditampilkan melalui **Serial Monitor** yang menunjukkan daftar SSID dan kekuatan sinyal masing-masing jaringan yang terdeteksi.

Setelah berhasil terkoneksi dengan WiFi, langkah selanjutnya adalah menghubungkan ESP32 dengan sensor suhu dan kelembaban (seperti DHT11 atau DHT22) serta mengirimkan data hasil pembacaan sensor tersebut ke sistem API berbasis Laravel yang telah dikembangkan sebelumnya.

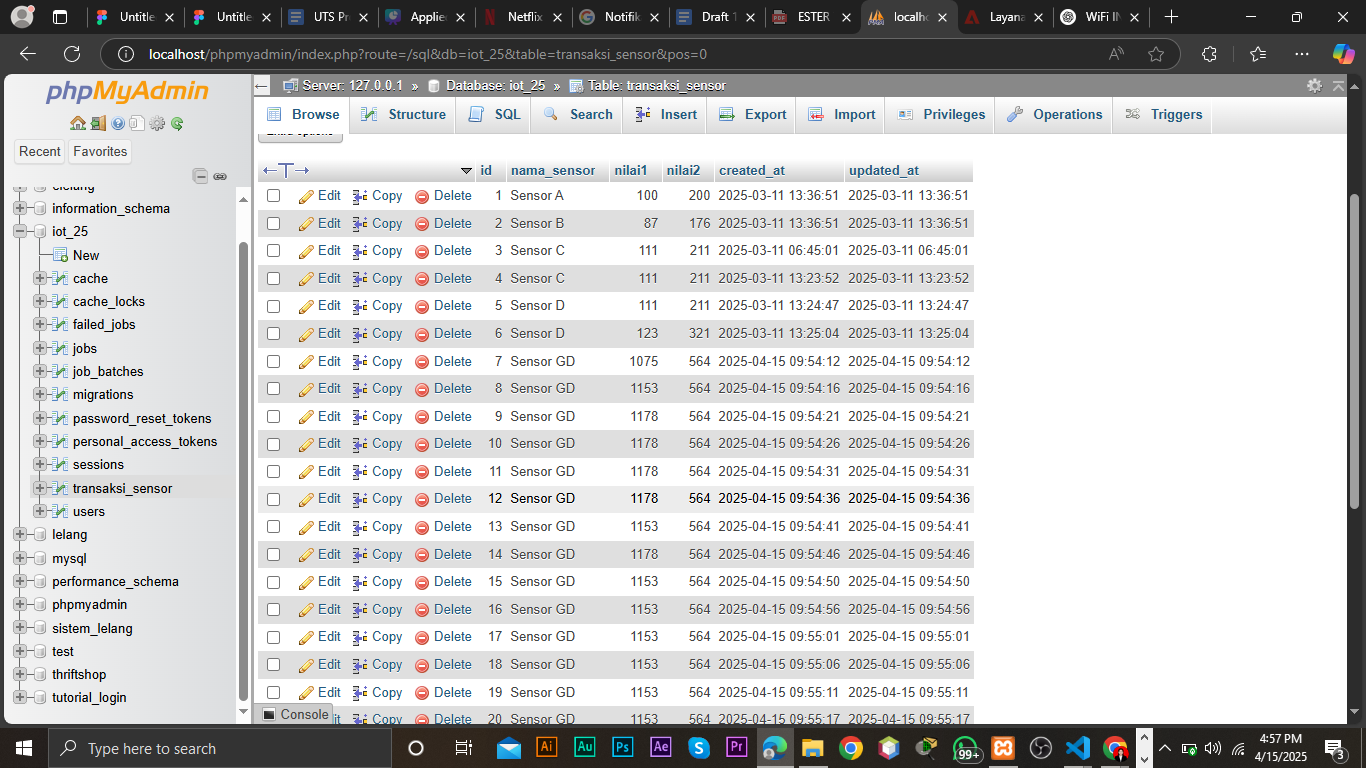
Hasil eksperimen menunjukkan bahwa ESP32 dapat:

* Terhubung ke jaringan WiFi dengan stabil.
* Melakukan pembacaan data suhu dan kelembaban dari sensor.
* Mengirimkan data ke API Laravel secara berkala melalui koneksi internet.

Data yang berhasil dikirimkan dapat diverifikasi melalui tampilan **log API**, **database**, atau melalui **endpoint JSON** yang telah tersedia.



Gambar 3.8



Gambar 3.

**4**

**PENUTUPAN**

**4.1 Kesimpulan**

Berdasarkan hasil praktik yang telah dilakukan, dapat disimpulkan bahwa perangkat **ESP32** dapat digunakan secara nyata (*real hardware*) untuk mendukung pengembangan sistem berbasis **Internet of Things (IoT)**. Proses dimulai dari pengenalan dan instalasi driver hingga perangkat berhasil dikenali oleh komputer, kemudian dilanjutkan dengan pengkodean dan upload program menggunakan **Visual Studio Code** dan **PlatformIO**.

ESP32 berhasil melakukan pemindaian jaringan WiFi di sekitar, serta dapat terhubung ke Access Point yang tersedia. Tahap selanjutnya berupa integrasi dengan sensor suhu dan kelembaban telah dilakukan dengan menyesuaikan kode dari praktik sebelumnya (Bab 13), dan berhasil mengirimkan data ke API Laravel melalui koneksi internet menggunakan layanan **Ngrok** sebagai jembatan akses publik.

Praktik ini membuktikan bahwa ESP32 dapat dijadikan sebagai komponen utama dalam proyek IoT, dengan kemampuan menghubungkan perangkat fisik ke dalam sistem digital berbasis API dan database. Dengan pendekatan ini, sistem monitoring atau kontrol jarak jauh berbasis internet dapat diwujudkan dengan efektif dan efisien.