

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

Membuat Tampilan Interface Web Dashboard IoT

Intan Tania

Fakultas Vokasi, Universitas Brawijaya

Email: intantania2412@gmail.com

Abstract

The Internet of Things (IoT) is a concept that connects physical devices to the internet, allowing them to exchange data automatically without human intervention. One of the key components in developing IoT systems is the microcontroller, such as the ESP32, which comes with built-in Wi-Fi and Bluetooth connectivity. This practical activity aims to provide a basic understanding of using the ESP32 as real hardware to control simple output devices like LEDs. The activity includes hardware setup, programming using PlatformIO, and an introduction to using GPIO pins on the ESP32. Through this practice, students are expected to gain foundational knowledge of embedded systems and be prepared to develop more complex IoT applications in the future.

Keywords -- *Internet of Things, ESP32, Microcontroller, Real Hardware.*

1. Introduction

1.1 Latar belakang

Perkembangan teknologi saat ini sangat pesat, terutama dalam bidang otomasi dan komunikasi. Salah satu teknologi yang sedang berkembang adalah Internet of Things (IoT). IoT adalah sistem yang menghubungkan perangkat-perangkat fisik seperti sensor dan alat elektronik ke internet, sehingga bisa saling bertukar data secara otomatis tanpa bantuan manusia secara langsung.

Untuk membuat sistem IoT, dibutuhkan perangkat keras seperti mikrokontroler. Salah satu mikrokontroler yang banyak digunakan adalah ESP32. ESP32 sudah memiliki fitur Wi-Fi dan Bluetooth, sehingga bisa langsung digunakan untuk menghubungkan perangkat ke internet. Selain itu, ESP32 memiliki banyak pin input/output (GPIO) yang bisa digunakan untuk membaca sensor atau mengontrol perangkat seperti LED, motor, atau relay.

Dalam praktikum ini, ESP32 digunakan secara langsung (*real hardware*) untuk menyalakan LED. Tujuannya adalah agar mahasiswa bisa memahami cara kerja ESP32, mulai dari menyusun rangkaian, membuat program, sampai melihat hasilnya secara langsung. Praktikum ini juga melatih mahasiswa untuk memahami cara mengendalikan perangkat elektronik menggunakan kode program dan bagaimana ESP32 bisa digunakan dalam sistem IoT.

Dengan praktikum ini, mahasiswa mendapatkan pengalaman nyata dalam membuat sistem sederhana berbasis IoT dan bisa mengembangkan pengetahuan tersebut ke proyek yang lebih kompleks di masa depan.

1.2 Tujuan eksperimen

Tujuan dari eksperimen atau praktikum ini adalah sebagai berikut:

- Mengetahui ESP32 sebagai mikrokontroler untuk sistem IoT.
- Melatih penggunaan pin GPIO untuk membaca input dan mengendalikan output.
- Mengetahui proses pemrograman ESP32 menggunakan PlatformIO.
- Memahami cara kerja perangkat fisik secara langsung.

2. Methodology

2.1 Tools & Materials

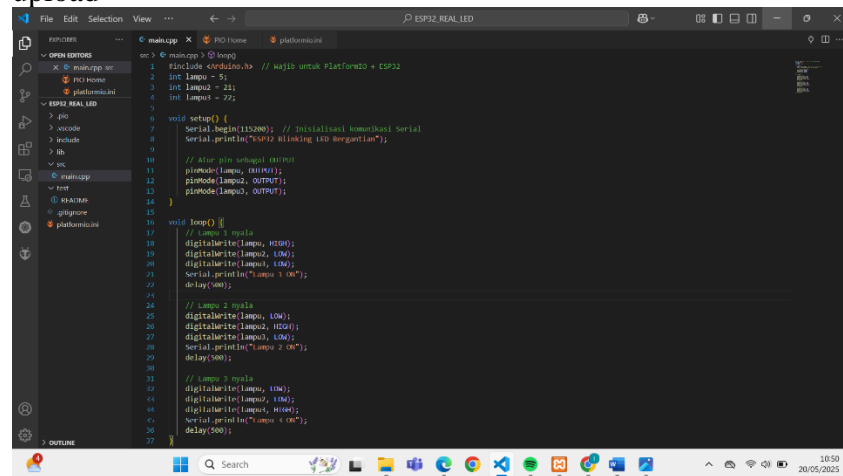
- ESP32
- LED
- Kabel jumper
- Breadboard
- Komputer dengan PlatformIO

2.2 Implementation Steps

1. Sambungkan ESP32 ke komputer dan pastikan sudah terdeteksi di bagian *Device Manager* → *Ports (COM & LPT)*. Biasanya akan tertera sebagai *Silicon Labs*.
2. Lakukan prosedur wiring kabel dan sensor sesuai dengan simulasi yang telah dibuat sebelumnya.
3. Buat sebuah folder dengan nama REAL_HARDWARE.
4. Buat project di platform.io dengan nama ESP32_Real_LED.
5. Buka proyek baru di PlatformIO. Sesuaikan file platformio.ini dengan board yang digunakan seperti berikut:

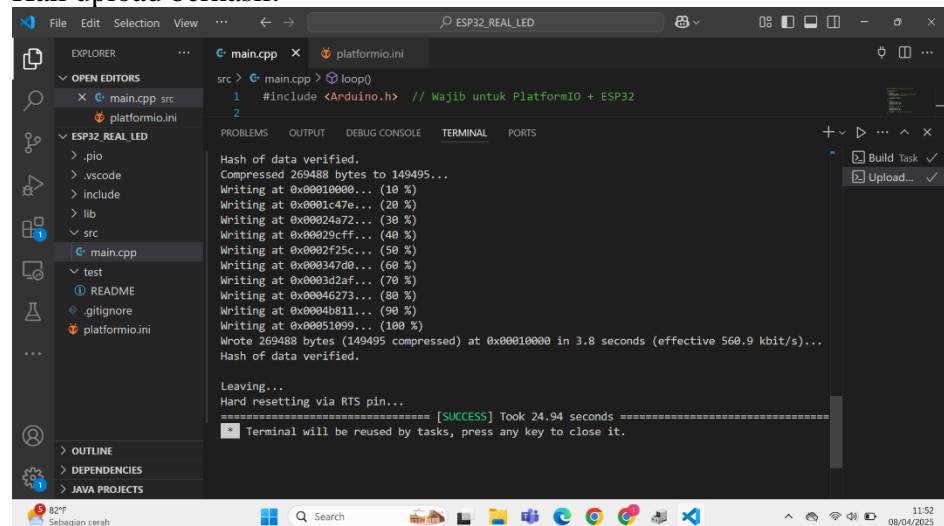
```
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14 framework = arduino
15 upload_port = COM4
16 monitor_port = COM4
```

6. Masukkan kode berikut ke dalam file main.cpp setelah itu lakukan upload



```
1 #include <Arduino.h> // wajib untuk PlatformIO + ESP32
2 int lampu = 5;
3 int lampu2 = 21;
4 int lampu3 = 22;
5
6 void setup() {
7   Serial.begin(115200); // Inisialisasi komunikasi Serial
8   Serial.println("ESP32 blinking led bergantian");
9
10  // Ator pin sebagai output
11  pinMode(lampu, OUTPUT);
12  pinMode(lampu2, OUTPUT);
13  pinMode(lampu3, OUTPUT);
14
15
16  void loop() {
17    // lampu 1 nyala
18    digitalWrite(lampu, HIGH);
19    digitalWrite(lampu, LOW);
20    digitalWrite(lampu, HIGH);
21    Serial.println("lampu 1 ON");
22    delay(1000);
23
24    // lampu 2 nyala
25    digitalWrite(lampu, LOW);
26    digitalWrite(lampu, HIGH);
27    digitalWrite(lampu, LOW);
28    Serial.println("lampu 2 ON");
29    delay(1000);
30
31    // lampu 3 nyala
32    digitalWrite(lampu, HIGH);
33    digitalWrite(lampu, LOW);
34    digitalWrite(lampu, HIGH);
35    Serial.println("lampu 3 ON");
36    delay(1000);
37  }
```

Hail upload berhasil:



```
src > main.cpp > loop()
1 #include <Arduino.h> // wajib untuk PlatformIO + ESP32
2
Hash of data verified.
Compressed 269488 bytes to 149495...
Writing at 0x00010000... (10 %)
Writing at 0x00010470... (20 %)
Writing at 0x0002a272... (30 %)
Writing at 0x00029cfc... (40 %)
Writing at 0x0002f25c... (50 %)
Writing at 0x000347d0... (60 %)
Writing at 0x0003d2af... (70 %)
Writing at 0x00046273... (80 %)
Writing at 0x0004b811... (90 %)
Writing at 0x00051099... (100 %)
Wrote 269488 bytes (149495 compressed) at 0x00010000 in 3.8 seconds (effective 560.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 24.94 seconds =====
Terminal will be reused by tasks, press any key to close it.
```

Lampu LED akan MENYALA

3. Results and Discussion

Pada praktikum ini, percobaan dilakukan dengan menghubungkan tiga buah LED ke mikrokontroler ESP32 pada pin GPIO 25, 26, dan 27. LED diatur agar menyala satu per satu secara bergantian, masing-masing selama satu detik, menggunakan fungsi digitalWrite() dan delay() dalam bahasa pemrograman Arduino.

Setelah program berhasil di-*upload* melalui PlatformIO, ketiga LED menyala secara bergantian sesuai urutan yang sudah diprogram. Hasil ini menunjukkan bahwa ESP32 dapat mengontrol lebih dari satu output secara bergiliran dengan baik.

Dari hasil ini, dapat disimpulkan bahwa:

- ESP32 dapat mengontrol banyak perangkat output secara bersamaan.

- Waktu jeda antar perubahan status LED bekerja sesuai dengan nilai `delay(1000)`.
- Serial Monitor sangat membantu untuk melihat proses yang sedang berjalan dan mendeteksi kesalahan jika terjadi.

Selain itu, tidak ditemukan kendala dalam proses upload program, dan koneksi perangkat keras (wiring) bekerja dengan baik. Praktikum ini memberikan pemahaman penting tentang penggunaan pin GPIO dan bagaimana cara ESP32 mengeksekusi perintah logika untuk mengontrol perangkat output.

Beriku adalah hasil praktikum :

