

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

(Simulasi Kelembapan Tanah Berbasis IoT dengan ESP32 Menggunakan Platform
Wokwi yang Diintegrasikan dengan VS Code)



Muhammad Kadavi
Fakultas Vokasi, Universitas Brawijaya
Email: kadavi2945@student.ub.ac.id

Abstrak

Praktikum ini dilakukan untuk mensimulasikan sistem kelembapan tanah berbasis IoT yang bekerja dengan menggunakan mikrokontroler ESP32 dan mengintegrasikannya ke platform Wokwi serta menerapkan aplikasi VS Code dengan memanfaatkan ekstensi PlatformIO. Dalam proses simulasi dibuat simulator pemantau kelembapan tanah menggunakan sensor kelembapan virtual DHT22 dan lalu menampilkan data yang berubah kepada terminal vscode. Sistem kerja simulasi ini diatur dalam pengaruh dari logika yang diprogramkan kondisional dimana ESP32 mengolah sinyal dari sensor dengan merespon perubahan data kelembapan tanah yang telah diatur dari sensor DHT22 yang dimilikinya. Berdasarkan hasil dari percobaan yang dilakukan, sistem ini dapat bekerja dengan stabil dalam membaca dan membuat respon terhadap perubahan nilai kelembapan yang sesuai masing-masing algoritmanya.

Pendahuluan

Perkembangan teknologi Internet of Things membuat sistem monitoring lingkungan seperti kelembapan tanah dapat bekerja dengan lebih efisien dan terintegrasi. Pemantauan kelembapan tanah di bidang pertanian atau budidaya tanaman sangat penting karena dapat mengoptimalkan tanaman dan menggunakan sumber air dengan efektif. Namun, pengembangan sistem IoT secara langsung memerlukan perangkat fisik dan proses debugging yang ribet. Praktikum ini bertujuan untuk mensimulasikan sistem monitoring kelembapan tanah berbasis IoT menggunakan mikrokontroler ESP32 dan Platform Wokwi. Simulasi ini menggunakan sensor kelembapan virtual dan interface sederhana untuk menampilkan data dan respons sistem. Tujuannya adalah untuk memahami bagaimana sistem IoT bekerja dalam membaca kondisi tanah, mengolah data, dan menanggapi secara otomatis. Dengan integrasi PlatformIO di VS Code, proses debugging dan pemrograman menjadi lebih mudah, sehingga pengguna dapat fokus pada pengembangan logika sistem tanpa hambatan teknis yang signifikan.

Latar Belakang

Salah satu komponen yang sangat penting bagi pertumbuhan tanaman, terutama dalam sektor pertanian dan budidaya, adalah tingkat kelembapan tanah. Tingkat kelembapan yang tidak sesuai dapat menunda perkembangan, menurunkan produktivitas, dan mematikan tanaman. Hingga saat ini, telah dilakukan pemantauan kelembapan tanah sebagian besar manual per lusin, yang jelas tidak mencukupi untuk skala yang lebih besar. Di pihak lain, akan semakin sulit untuk menemukan sumber air bersih di masa depan, karena Sistem Irigasi Terkontrol yang mampu. Solusinya terletak pada perangkat tersebut: teknologi IoT pada tahun terkini memungkinkan penggunaan sistem monitoring dan kontrol otomatis yang menggunakan data real-time. Mikrokontroler, seperti ESP32, sangat diinginkan untuk sistem Internet of Things, karena mereka mampu menghubungkan perangkat mereka ke internet, memproses data, dan mengatur perangkat luar seperti sensor atau aktuator. Namun, sebagian besar sulit dan mahal untuk pengembangan sistem IoT secara fisik, karena setup-nya kecil dan sulit di-debug. Karena sistem IoT, sedikit berbeda halnya. Tidak mendebug ingin memerlukan setup berdasarkan perangkat keras, dan simulasi adalah pilihan terbaik untuk menerapkan sistem IoT menggunakan platform seperti Wokwi dan membantu terakhir dalam studi sistem IoT. Dapat dikatakan bahwa pengguna berhak merancang, memprogram, dan mendebug logika dengan IoT virtual, dan perangkat lain. Simulasi ini juga membuat iterasi dan algoritma analisis lebih cepat daripada kebanyakan kerusakan sistem keras yang aktif.

Tujuan

Eksperimen ini dilakukan untuk memahami cara kerja dari IoT berbasis sistem monitoring kelembapan tanah, mulai dari logika pembacaan data sensor, pengolahan sinyal, hingga respons otomatis yang diterapkan. Selanjutnya, praktikum ini juga diadakan untuk memahami simulasi sistem IoT yang diintegrasikan dengan Platform Wokwi ke dalam text editor VS Code, dengan demikian, user dapat secara virtual mengetahui proses algoritma dan antarmuka yang disediakan tanpa harus memakai perangkat secara fisik.

Metodologi

Mikrokontroler ESP32 digunakan untuk menguji sistem monitoring kelembapan tanah. Selanjutnya, sensor kelembapan tanah virtual dipergunakan untuk mengetahui tingkat kelembapan tanah. Sedangkan, logika pemrograman dipergunakan untuk memperlakukan respons sistem, misalnya menampilkan nilai pada antarmuka. Simulasi eksperimen pada Platform Wokwi dipergunakan untuk menguji pembacaan nilai yang tepat dan memperlakukan respons sistem dari hasil algoritma yang telah tersusun. Lalu, penulisan sintaks memberitahukan ke tata bahasa pemrograman. Dengan kedua integrasi sistem tersebut, uji coba dan debug dapat dilakukan dengan mudah dan cepat tanpa memakai perangkat fisik yang diperlukan.

Software & Hardware

Software	Hardware
Wokwi	Esp32 (virtual)
Extension PlatformIO	Sensor Kelembapan DHT22(virtual)
VSCode	Laptop

2.2 Implementation Steps (Langkah Implementasi)

1. Persiapan

1. Buka Wokwi (<https://wokwi.com>)
2. Buat proyek baru dengan ESP32, lalu tambahkan komponen:
 - Sensor Kelembapan DHT22
3. Hubungkan komponen virtual ke ESP32 menggunakan diagram JSON.

2. Instalasi VS Code dan PlatformIO

1. Download dan Install VS Code dari <https://code.visualstudio.com/>
2. Buka VS Code, lalu buka Extensions Marketplace (Ctrl+Shift+X).
3. Cari "PlatformIO", lalu install extension PlatformIO.
4. Restart VS Code setelah instalasi selesai.

3. Membuat Proyek di PlatformIO

1. Buka VS Code dan klik ikon PlatformIO (di sidebar kiri).
2. Buat proyek baru dengan:
 - Board: DOIT ESP32 DEVKIT V1
 - Framework: Arduino
 - Nama Proyek: TrafficLightESP32
3. Tambahkan library yang diperlukan dengan perintah

```
pio lib install "Adafruit GFXLibrary"
pio lib install "Adafruit ILI9341"
```
4. Tambahkan file wokwi.toml
5. Lalu build yang berada di pojok kanan atas,
6. setelah itu salin path berikut ini,

```
'.pio/build/esp32doit-devkit-v1/firmware.bin'
'.pio/build/esp32doit-devkit-v1/firmware.elf'
```

4. Menulis Kode ESP32 di VS Code

1. Buka file `src/main.cpp` dan masukkan kode berikut:

```
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "DHT.h"
```

```
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define SCREEN_ADDRESS 0x3C // Alamat I2C dari SSD1306
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

```

#define DHTPIN 27
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx + OLED Test"));

  dht.begin();

  if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 initialization failed!"));
    for (;;);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(10, 10);
  display.println("DHT22 Sensor Ready!");
  display.display();
  delay(2000);
}

void loop() {
  delay(2000); // Sensor DHT22 memiliki delay minimal 2 detik

  float h = dht.readHumidity();
  float t = dht.readTemperature(); // Celsius
  float f = dht.readTemperature(true); // Fahrenheit

  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    display.clearDisplay();
    display.setCursor(10, 10);
    display.println("Sensor Error!");
    display.display();
    return;
  }

  Serial.print(F("Humidity: ")); Serial.print(h);
  Serial.print(F("% Temperature: ")); Serial.print(t);
  Serial.print(F("°C ")); Serial.print(f);
  Serial.println(F("°F"));

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(10, 10);
  display.print("Temp: ");
  display.print(t);
  display.print(" C");

```

```

display.setCursor(10, 30);
display.print("Humidity: ");
display.print(h);
display.print(" %");

display.display();
}

```

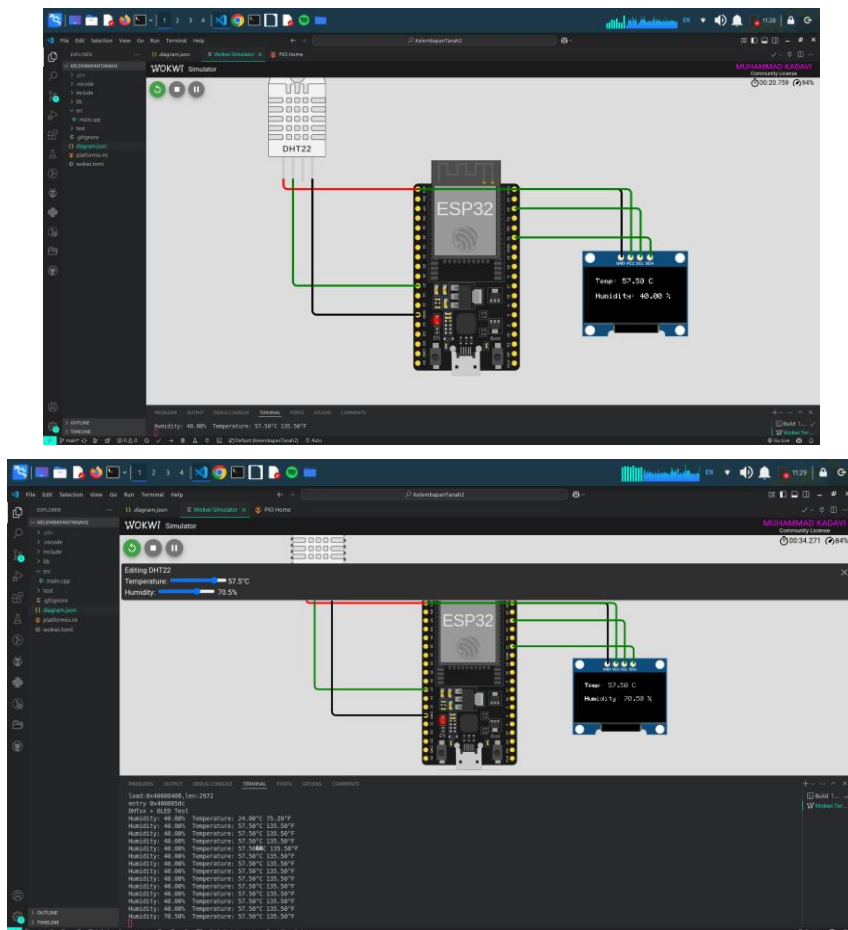
5. Menjalankan Simulasi di Wokwi

1. Buka proyek Wokwi yang sudah dibuat.
2. Tambahkan file diagram JSON agar projeknya bisa di jalankan di vscode
3. Klik tombol "RUN" untuk menjalankan simulasi dan melihat hasilnya.

3. Results and Discussion (Hasil dan Pembahasan)

Simulasi memberikan hasil yang menarik bahwa sistem pemantauan kelembapan tanah dapat menghitung nilai kelembapan virtual secara akurat dengan menggunakan logika pemrograman sebagai acuan. Data kelembapan ditampilkan pada antarmuka simulasi dalam real-time, dan mid-value dalam simulasi tetap berdasarkan kondisi yang disimulasikan. Dari perspektif pembahasan, tidak ada kesulitan berarti selain dari integrasi input dan output antarmuka yang melibatkan sensor virtual, mikrokontroler ESP32 dan antarmuka antarmuka pada platform wokwi. Fakta bahwa proses pembacaan data dan pembaruan tampilan dilakukan pada interval 2 detik menunjukkan logika algoritma telah bekerja sebagaimana mestinya. Disamping itu pengujian sintaks dan debugging Platform Wokwi dalam VS Code terhubung dengan PlatformIO memberikan penyederhanaan dalam pengembangan sistem yang tidak melibatkan perangkat fisik.

3.1 Experimental Results (Hasil Eksperimen)



4. Appendix (Lampiran, jika diperlukan)

```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5 #include "DHT.h"
6
7 #define SCREEN_WIDTH 128
8 #define SCREEN_HEIGHT 64
9 #define OLED_RESET -1
10 #define SCREEN_ADDRESS 0x3C // Alamat I2C dari SSD1306
11
12 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
13
14 #define DHTPIN 27
15 #define DHTTYPE DHT22
16
17 DHT dht(DHTPIN, DHTTYPE);
18
19 void setup() {
20   Serial.begin(9600);
21   Serial.println(F("DHTxx + OLED Test"));
22
23   dht.begin();
24
25   if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
26     Serial.println(F("SSD1306 initialization failed!"));
27     for (;;);
28   }
29
30   display.clearDisplay();
31   display.setTextSize(1);
32   display.setTextColor(WHITE);
33   display.setCursor(10, 10);
34   display.println("DHT22 Sensor Ready!");
35   display.display();
36   delay(2000);
37 }
38
39 void loop() {
40   delay(2000); // Sensor DHT22 memiliki delay minimal 2 detik
41
42   float h = dht.readHumidity();
43   float t = dht.readTemperature(); // Celsius
44   float f = dht.readTemperature(true); // Fahrenheit
45
46   if (isnan(h) || isnan(t) || isnan(f)) {
47     Serial.println(F("Failed to read from DHT sensor!"));
48     display.clearDisplay();
49     display.setCursor(10, 10);
50     display.println("Sensor Error!");
51     display.display();
52     return;
53   }
54
55   Serial.print(F("Humidity: ")); Serial.print(h);
56   Serial.print(F("% Temperature: ")); Serial.print(t);
57   Serial.print(F("°C ")); Serial.print(f);
58   Serial.println(F("°F"));
59
60   display.clearDisplay();
61   display.setTextSize(1);
62   display.setTextColor(WHITE);
63
64   display.setCursor(10, 10);
65   display.print("Temp: ");
66   display.print(t);
67   display.print(" C");
68
69   display.setCursor(10, 30);
70   display.print("Humidity: ");
71   display.print(h);
72   display.print(" %");
73
74   display.display();
75 }
76
```