

INDIKATOR SUHU, KELEMBAPAN, & INTENSITAS CAHAYA BERBASIS MIKROKONTROLLER ESP32

Feby Kurnia Putri

*Program Studi Teknologi Informasi, Fakultas Vokasi, Universitas Brawijaya Jl. Veteran No.10-11,
Ketawanggede, Kec. Lowokwaru, Kota Malang, Jawa Timur 65145*

e-mail : febystudentub@gmail.com

*Abstrack -- The development of technology in the field of IoT (Internet of Things) allows real-time monitoring of the environment with sensors integrated in microcontrollers. This research aims to test the temperature, humidity, and light intensity monitoring system using Arduino and display the results on OLED. The tests were conducted in a Wokwi-based simulation and developed using Visual Studio Code as the main programming environment. DHT22 sensors are used to measure temperature and humidity, to measure light intensity. The data obtained from the sensors is processed by the Arduino Uno microcontroller and displayed on the OLED screen. Test results show that the system can read and display data accurately with a fast response. Simulation using Wokwi allows code testing without physical hardware, thus speeding up the development and debugging process. This implementation can be used in various environmental monitoring applications, such as smart farming systems and space condition control.***Kata**

Keyword : Arduino, Wokwi, OLED, temperature, humidity, light intensity.

Abstrak -- Perkembangan teknologi dalam bidang IoT (Internet of Things) memungkinkan pemantauan lingkungan secara real-time dengan sensor yang terintegrasi pada mikrokontroler. Penelitian ini bertujuan untuk menguji sistem pemantauan suhu, kelembapan, dan intensitas cahaya menggunakan Arduino dan menampilkan hasilnya pada OLED. Pengujian dilakukan dalam simulasi berbasis Wokwi dan dikembangkan menggunakan Visual Studio Code sebagai lingkungan pemrograman utama. Sensor DHT22 digunakan untuk mengukur suhu dan kelembapan, untuk mengukur intensitas cahaya. Data yang diperoleh dari sensor diproses oleh mikrokontroler Arduino Uno dan ditampilkan pada layar OLED. Hasil pengujian menunjukkan bahwa sistem dapat membaca dan menampilkan data secara akurat dengan respons yang cepat. Simulasi menggunakan Wokwi memungkinkan pengujian kode tanpa perangkat keras fisik, sehingga mempercepat proses pengembangan dan debugging. Implementasi ini dapat digunakan dalam berbagai aplikasi pemantauan lingkungan, seperti sistem pertanian cerdas dan pengendalian kondisi ruang.**Kata**

Kata Kunci : : Arduino, Wokwi, OLED, suhu, kelembapan, intensitas cahaya.

I. PENDAHULUAN

Pemantauan kondisi lingkungan seperti suhu, kelembapan, dan intensitas cahaya menjadi faktor penting dalam berbagai bidang, termasuk pertanian, industri, dan sistem *smart home*. Dengan berkembangnya teknologi Internet of Things (IoT), pemantauan kondisi lingkungan dapat dilakukan secara real-time menggunakan sensor yang terintegrasi dengan mikrokontroler. Salah satu platform yang sering digunakan untuk pengembangan sistem IoT adalah Arduino, karena bersifat open-source, mudah diakses, dan memiliki banyak dukungan komunitas. Dalam penelitian ini, dilakukan uji coba pemantauan suhu, kelembapan, dan intensitas cahaya menggunakan sensor DHT22. Dengan adanya simulasi ini, pengembangan sistem pemantauan lingkungan menjadi lebih efisien dan dapat diterapkan dalam berbagai kebutuhan. Seperti pertanian cerdas, sistem pengendalian suhu ruangan.

II. METODOLOGI

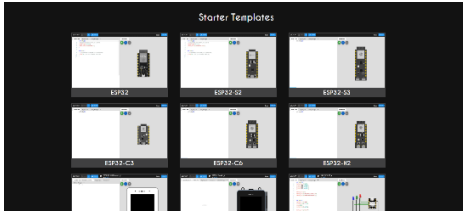
A. ALAT & BAHAN

Mikrokontroler (ESP32), Sensor DHT, OLED Display & LDR Software (Visual Studio Code, PlatformIO IDE, Arduino IDE).

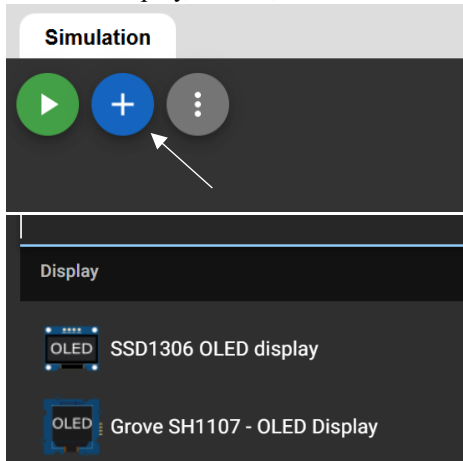
B. LANGKAH IMPLEMENTASI

1. Buka <https://wokwi.com/> pilih ESP32, kemudian pilih *Starter Templates*.

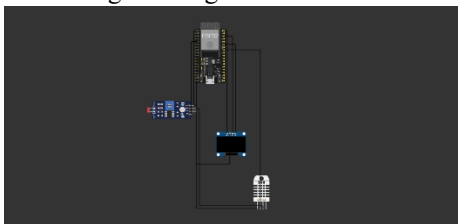




2. Add simulasi untuk memulai simulasi . Add 1 OLED Display, 1 LDR, dan 1 DHT.



3. Atur rangkain diagram menampilkan suhu dan cahaya sesuai angka yang disambungkan dengan *code* nanti.



4. Lakukan pengkodean seperti gambar berikut:

```
sketch.ino • diagram.json • Library Manager
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 DHTPIN 19
8 #define DHTTYPE DHT22
9 #define TIMEDHT 1000
10
11 #define SCREEN_WIDTH 128
12 #define SCREEN_HEIGHT 64
13
14 #define LDR_PIN 34
15
16
17 float humidity, celsius;
18 uint32_t timerDHT = TIMEDHT;
19 DHT dht(DHTPIN, DHTTYPE);
20
21
22 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
23
24
25 void getTemperature() {
26   if ((millis() - timerDHT) > TIMEDHT) {
27     timerDHT = millis();
28     humidity = dht.readHumidity();
29     celsius = dht.readTemperature();
30   }
31 }
```

```
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>
#define DHTPIN 19
```

```
#define DHTTYPE DHT22
#define TIMEDHT 1000
```

```
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
```

```
#define LDR_PIN 34
```

```
float humidity, celsius;
uint32_t timerDHT = TIMEDHT;
DHT dht(DHTPIN, DHTTYPE);
```

```
Adafruit_SSD1306
display(SCREEN_WIDTH,
SCREEN_HEIGHT, &Wire, -1);
```

```
void getTemperature() {
  if ((millis() - timerDHT) > TIMEDHT) {
    timerDHT = millis();
    humidity = dht.readHumidity();
    celsius = dht.readTemperature();
```

```
    if (isnan(humidity) || isnan(celsius)) {
      Serial.println("Gagal membaca sensor
DHT!");
      return;
    }
  }
}
```

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

```
    if
(!display.begin(SSD1306_SWITCHCAPV
CC, 0x3C)) {
  Serial.println("OLED      gagal
diinisialisasi");
  for (;;)
  }
```

```
display.clearDisplay();
}
```

```
void loop() {
  getTemperature();
  int lightValue = analogRead(LDR_PIN);
```

```
Serial.print("Temp: ");
Serial.print(celsius);
Serial.print(" C, Humidity: ");
Serial.print(humidity);
Serial.print("%, Light: ");
Serial.println(lightValue);
display.clearDisplay();
```

```
display.setTextSize(1);
display.setTextColor(WHITE);
```

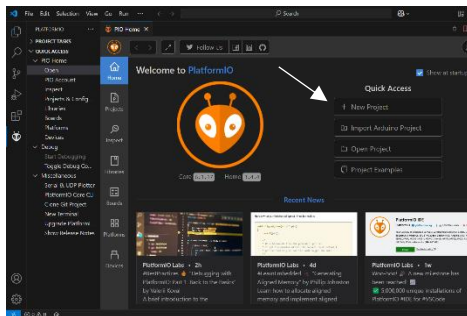
```
display.setCursor(0, 0);
display.print("Temp: ");
display.print(celsius);
display.println(" C");
```

```
display.setCursor(0, 16);
display.print("Humidity: ");
display.print(humidity);
display.println(" %");
```

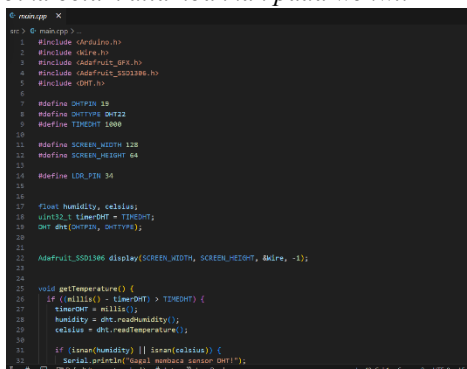
```
display.setCursor(0, 32);
display.print("Light: ");
display.print(lightValue);
```

```
display.display();
delay(1000);
}
```

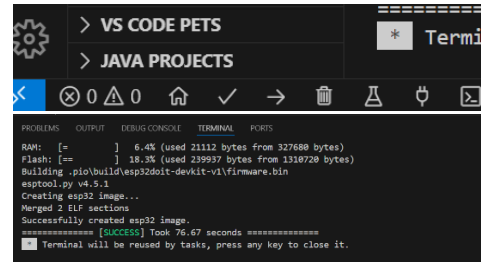
4. Buka VScode, kemudian install PlatformIO IDE. Buat proyek baru seperti gambar berikut:



5. Setelah itu kembali ke *Wokwi* untuk meng *copy code sketch.io*, lalu *paste code* pada *Visual Studio Code* pada bagian *src-main.cpp*. Tambahkan *Include <Arduino.h>* Jika belum ditambahkan pada *wokwi*.



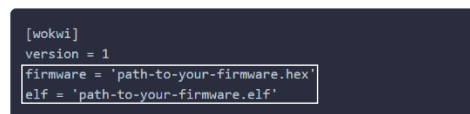
6. Jika sudah, lakukan build *Wokwi* dengan cara klik centang pada bagian bawah dari *Visual Studio Code*, seperti gambar berikut.



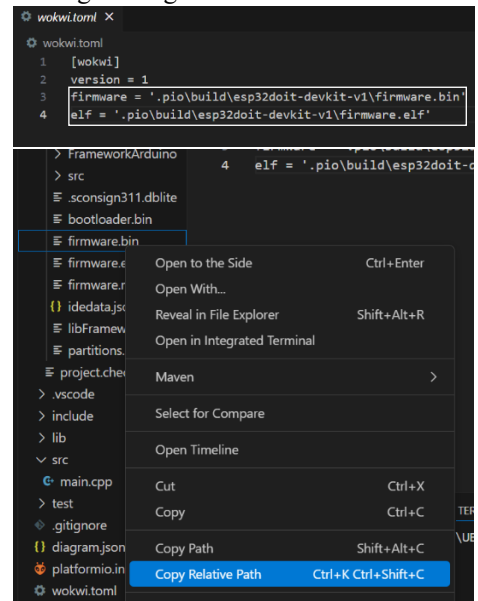
7. Buat file baru pada *Visual Studio Code*, kemudian buka *wokwi.toml* pada *google*, kemudian *copy* ke file baru yang diberi nama *wokwi.toml* di *VScode*.

wokwi.toml

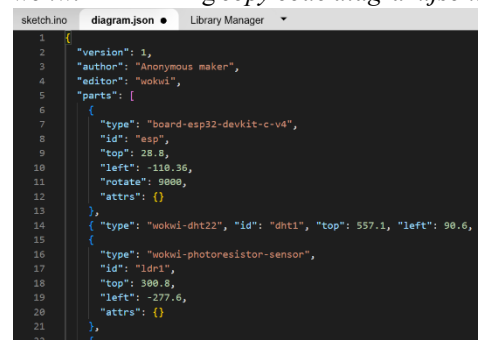
A basic *wokwi.toml* file looks like this:



8. Setelah itu ganti menjadi *firmware bin*, dan *esp32 firmware elf* pada *Visual Studio Code* masing-masing.



9. Buat file baru bernama *diagram.json* pada *Visual Studio Code*. Kemudian pergi ke *wokwi* untuk meng *copy code diagram.json*.



```

{
  "version": 1,
  "author": "Anonymous maker",
  "editor": "wokwi",
  "parts": [
    {
      "type": "board-esp32-devkit-c-v4",
      "id": "esp",
      "top": 28.8,
      "left": -110.36,
      "rotate": 9000,
      "attrs": {}
    },
    {
      "type": "wokwi-dht22", "id": "dht1",
      "top": 557.1, "left": 90.6, "attrs": {} },
    {
      "type": "wokwi-photoresistor-sensor",
      "id": "ldr1",
      "top": 300.8,
      "left": -277.6,
      "attrs": {}
    },
    {
      "type": "board-ssd1306",
      "id": "oled1",
      "top": 406.34,
      "left": -38.17,
      "attrs": { "i2cAddress": "0x3c" }
    }
  ]
}

```

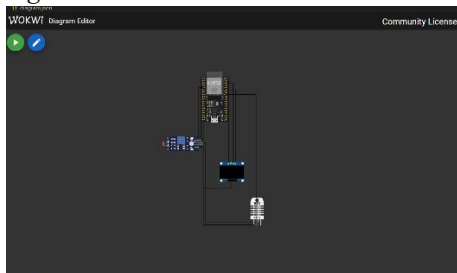
10. Paste ke *Visual Studio Code*, pada file *diagram.json*. Jika *diagram* tidak berfungsi, maka lakukan edit pada *diagram.json*, kemudian *paste code* yang sudah di *copy* pada *wokwi*.

```

1 // diagram.json
2 {
3   "version": 1,
4   "author": "Anonymous maker",
5   "editor": "wokwi",
6   "parts": [
7     {
8       "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 28.8, "left": -110.36, "rotate": 9000, "attrs": {} },
9     {
10      "type": "wokwi-led",
11      "id": "led1",
12      "top": 32.8,
13      "left": 141,
14      "power": 50,
15      "color": "red", "flip": "" }
16    ],
17    {
18      "type": "wokwi-led",
19      "id": "led2",
20      "top": 75.8,
21      "left": 141,
22      "power": 50,
23      "color": "green", "flip": "" }
24  ]
25 }

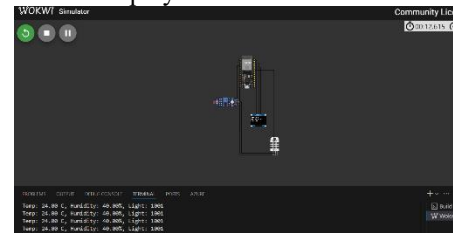
```

11. Kemudian *wokwi* akan menampilkan hasil dari *diagram.json* pada simulasi *Traffic Light* di *Visual Studio Code*.



III. HASIL DAN PEMBAHASAN

Hasil indikator Suhu, tingkat kelembapan dan intensitas cahaya, ditampilkan pada OLED Display.



IV. KESIMPULAN

Setelah melakukan perancangan dan pengujian terhadap alat secara keseluruhan. Maka dapat diambil kesimpulan :

1. Perangkat yang digunakan oleh penulis dapat bekerja dengan baik sesuai dengan yang diharapkan.
2. Mikrokontroler ESP32 yang digunakan sebagai pengendali utama dapat bekerja dalam menjalankan perintah yang diberikan

V. DAFTAR PUSTAKA

Rohatgi, A. (2023). BH1750 Light Sensor Tutorial

<https://media.neliti.com/media/publications/439068-none-037851e2.pdf>

Espressif Systems. (2024). I2C Communication Protocol.

<https://docs.espressif.com/>

Adafruit. (2024). *DHT22 Sensor Guide*.

<https://learn.adafruit.com/dht>