

# LAPORAN PROYEK PIO IMPLEMENTASI CARA KERJA TEMPERATURE SENSOR ATAU SENSOR SUHU PADA LCD I2C

**Rumiris Butarbutar**

Program Studi Teknologi Informasi, Fakultas Vokasi, Universitas Brawijaya

Email: ririssbtr09@student.ub.ac.id

**Abstrak** – Lampu Sensor suhu merupakan perangkat yang digunakan untuk mengukur dan mendeteksi perubahan suhu di suatu lingkungan. Salah satu cara menampilkan hasil pengukuran sensor suhu adalah dengan menggunakan LCD I2C, yang memungkinkan tampilan data secara real-time dengan koneksi yang lebih sederhana dibandingkan LCD konvensional. Dalam sistem ini, sensor suhu membaca suhu lingkungan dan mengirimkan data ke mikrokontroler untuk diproses. Mikrokontroler kemudian meneruskan informasi tersebut ke LCD I2C melalui komunikasi I2C, yang hanya memerlukan dua jalur data, sehingga lebih efisien dalam penggunaan pin dan konsumsi daya. Teknologi ini banyak diterapkan dalam berbagai bidang, seperti pemantauan suhu ruangan, sistem pendingin otomatis, dan perangkat rumah tangga pintar. Penggunaan LCD I2C mempermudah integrasi dengan berbagai perangkat elektronik dan menjadikannya solusi praktis untuk sistem berbasis mikrokontroler.

**Kata Kunci:** Sensor suhu, LCD I2C, mikrokontroler, komunikasi I2C, teknologi sensor.

*Abstract -- Temperature Sensor Lamp is a device used to measure and detect temperature changes in an environment. One way to display the results of temperature sensor measurements is to use an I2C LCD, which allows data to be displayed in real time with a simpler connection than conventional LCDs. In this system, the temperature sensor reads the ambient temperature and sends the data to the microcontroller for processing. The microcontroller then forwards the information to the I2C LCD via I2C communication, which only requires two data lines, making it more efficient in pin usage and power consumption. This technology is widely applied in various fields, such as room temperature monitoring, automatic cooling systems, and smart home devices. The use of I2C LCDs makes it easy to integrate with various electronic devices and makes it a practical solution for microcontroller-based systems.*

**Keywords:** Temperature sensor, I2C LCD, microcontroller, I2C communication, sensor technology.

lebih sederhana dan konsumsi daya yang lebih rendah, menjadikannya solusi praktis dalam sistem otomatisasi.

## 1. Pendahuluan

### 1.1 Latar Belakang

Sensor suhu banyak digunakan untuk memantau dan mengontrol suhu di berbagai bidang, seperti rumah tangga, industri, dan laboratorium. Untuk menampilkan hasil pengukuran, LCD I2C menjadi pilihan yang lebih efisien dibandingkan LCD konvensional karena hanya memerlukan dua jalur komunikasi, sehingga menghemat penggunaan pin pada mikrokontroler. Dengan teknologi ini, pemantauan suhu dapat dilakukan secara real-time dengan koneksi yang

### 1.2 Tujuan

- Memahami cara kerja sensor suhu dalam membaca perubahan suhu.
- Mempelajari penggunaan LCD I2C untuk menampilkan data secara real-time.
- Mengoptimalkan penggunaan pin mikrokontroler dengan komunikasi I2C.
- Menerapkan sistem pemantauan

suhu yang sederhana dan efisien.

## 2. Metodologi

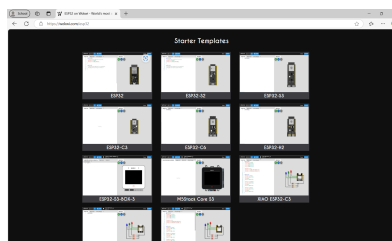
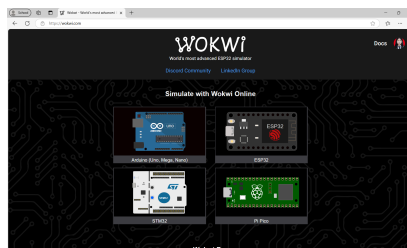
### 2.1 Alat dan Bahan

- Wokwi Simulator
- Visual Studio Code (VS Code)
- Mikrokontroler virtual (ESP32)
- Sensor suhu virtual (DHT11 atau DS18B20)
- LCD I2C virtual (20 x 4)
- Library Arduino (Wire.h, LiquidCrystal\_I2C.h, dan DHT.h)

### 2.2 Langkah Implementasi

#### Implementasi dan Konfigurasi

1. Menyiapkan akun dan login pada website Wokwi (<https://wokwi.com>)



2. Untuk menghubungkan ESP32 ke sumber daya, sambungkan pin VCC ke 3.3V dan GND ke GND agar mikrokontroler dapat beroperasi dengan baik.

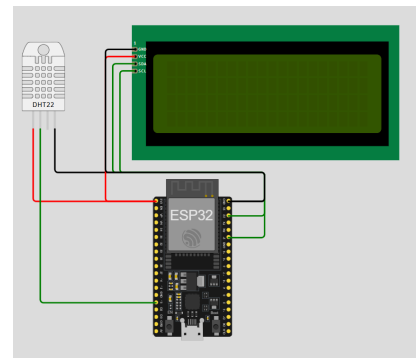
Selanjutnya, sambungkan sensor suhu DHT22 ke ESP32:

- a. Hubungkan pin VCC DHT22 ke 3.3V ESP32.
- b. Hubungkan pin GND DHT22 ke GND ESP32.
- c. Hubungkan pin Data

DHT22 ke salah satu GPIO ESP32.

Lalu sambungkan LCD I2C ke ESP32:

- a. Hubungkan pin VCC LCD ke 3.3V ESP32.
- b. Hubungkan pin GND LCD ke GND ESP32.
- c. Hubungkan pin SDA LCD ke pin SDA ESP32.
- d. Hubungkan pin SCL LCD ke pin SCL ESP32.



3. Tulis kode program untuk membaca suhu dari DHT22 dan menampilkannya di LCD I2C.

```
sketch.ino  diagram.json  libraries.txt  Library Manager

1 //RUMAHIRIS OUTARROBITAR
2 //23314870711138
3 //LCD I2C dengan sensor suhu
4
5 #include <Wire.h>
6 #include <LiquidCrystal_I2C.h>
7 #include "DHTesp.h"
8
9 LiquidCrystal_I2C lcd(0x27,16,2);
10
11 float temperature; //tipe data temperature
12 const int DHT_PIN = 13; //kaki DHT-22 pada ESP 32
13
14 DHTesp dhtSensor;
15
16 void setup() {
17   Wire.begin();
18   Serial.begin(115200);
19   dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
20   lcd.begin();
21   lcd.backlight();
22 }
23
24 void loop() {
25   TempAndHumidity data = dhtSensor.getTempAndHumidity();
26   lcd.clear();
27
28   lcd.setCursor(0,0); //Menampilkan text di posisi kolom ke 0 baris ke 0
29   lcd.println("Tampilan Sensor Suhu");
30
31   lcd.setCursor(3,2); //Menampilkan text di posisi kolom ke 3 baris ke 3
32   lcd.println("Suhu : " + String(data.temperature, 2) + " C");
33
34   lcd.setCursor(1,3);
35   lcd.println("Kelembapan: " + String(data.humidity, 2) + " %");
36
37   delay(2000); //Delay selama 2s
38
39 }
40
```

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

```
#include "DHTesp.h"

LiquidCrystal_I2C
lcd(0x27,16,2);

float temperature; //tipe data
temperature

const int DHT_PIN = 13; //kaki
DHT-22 pada ESP 32

DHTesp dhtSensor;

void setup() {

  Wire.begin();

  Serial.begin(115200);

  dhtSensor.setup(DHT_PIN,
DHTesp::DHT22);

  lcd.init();

  lcd.backlight();

}

void loop() {

  TempAndHumidity data =
dhtSensor.getTempAndHumidity(
);

  lcd.clear();

  lcd.setCursor(0,0);
//Menampilkan text di posisi
kolom ke 0 baris ke 0

  lcd.println ("Tampilan Sensor
Suhu");

  lcd.setCursor(3,2);
//Menampilkan text di posisi
```

kolom ke 3 baris ke 3

```
    lcd.println ("Suhu : " +
String(data.temperature, 2) + "
C");

    lcd.setCursor(1,3);

    lcd.print("Kelembapan: " +
String(data.humidity, 2) + " %");

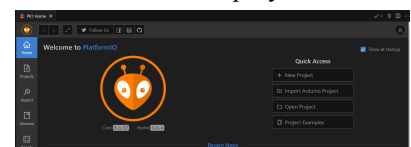
    delay (2000); //Delay selama 2s

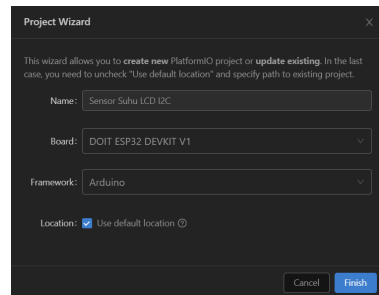
}
```

4. Tambahkan Library Manager di Wokwi agar perangkat dapat berfungsi dengan benar, melalui tombol plus di pojok kanan atas.
  - a. Tambahkan LiquidCrystal I2C untuk mengontrol tampilan LCD I2C.
  - b. Tambahkan DHT Sensor Library for ESPx agar ESP32 dapat membaca data dari sensor DHT22.
  - c. Tambahkan DHT22 sebagai pustaka tambahan untuk memastikan sensor dapat dikonfigurasi dengan baik.

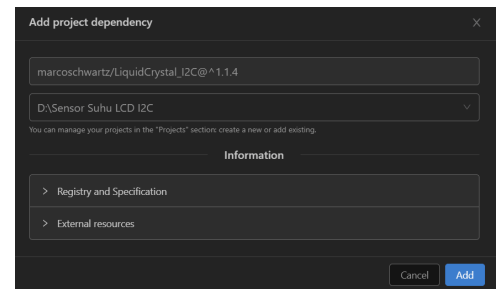
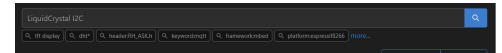
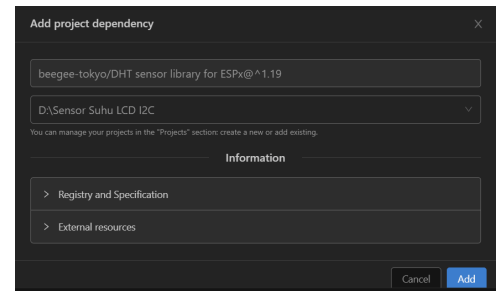
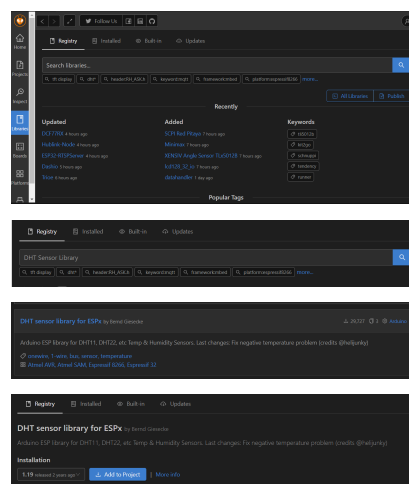


5. Buka PlatformIO IDE pada VScode dan buatlah proyek baru

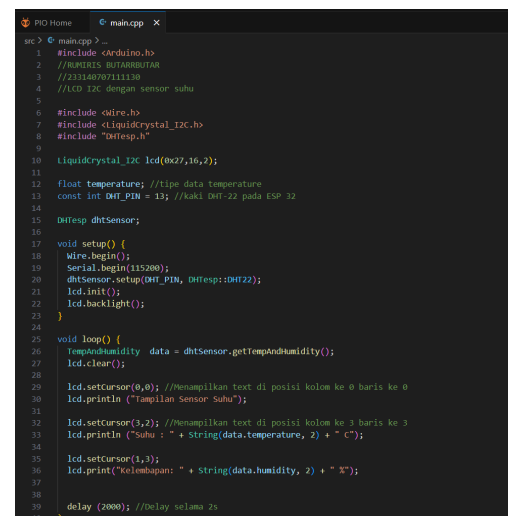




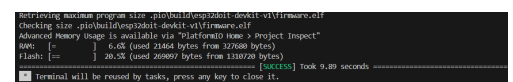
6. Untuk menghindari error saat menjalankan program di VS Code, diperlukan penambahan beberapa library menggunakan Library Manager di PlatformIO atau melalui Arduino IDE jika menggunakan ekstensi Arduino. Library yang harus ditambahkan meliputi LiquidCrystal I2C, yang digunakan untuk mengontrol tampilan LCD I2C, serta DHT Sensor Library for ESPx agar ESP32 dapat membaca data dari sensor DHT22. Selain itu, pastikan library DHT22 juga terpasang agar sensor dapat dikonfigurasi dengan benar. Setelah semua library ditambahkan, pastikan includePath telah diperbarui agar kode dapat dikenali tanpa error.



7. *Copy code* pada file sketch.ino pada website wokwi, setelah itu *paste* pada file main.cpp pada *Visual Studio Code*



8. Selanjutnya lakukan *built* hingga *success*



9. Buat file baru dengan format wokwi.toml pada *Visual Studio Code*. Yang mana isi file tersebut ialah salinan dari file wokwi.toml pada *google*.

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

```
[wokwi]
version = 1
firmware = 'path-to-your-firmware.hex'
elf = 'path-to-your-firmware.elf'
```

10. Rubah *firmware bin* dan *elf* sesuai versi *Visual Studio Code* masing-masing

```
wokwi.toml
1 [wokwi]
2 version = 1
3 firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'
4 elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'
```

11. Selanjutnya buat file diagram pada *Visual Studio Code* yang mana isi file tersebut ialah hasil *copy* dari file `diagram.json` pada *wokwi*

```
diagram.json
1 {
2   "version": 1,
3   "author": "RUMIRIS BUTARBUTAR",
4   "editor": "wokwi",
5   "parts": [
6     {
7       "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 86.4, "left": 4.84, "attrs": {} },
8     {
9       "type": "wokwi-lcd2004",
10      "id": "lcd2",
11      "top": 118.4,
12      "left": -23.2,
13      "attrs": { "pins": "i2c" }
14    }
15  ],
16  "connections": [
17    [ "esp:TX", "serialMonitor:RX", "" ],
18    [ "esp:RX", "serialMonitor:TX", "" ],
19    [ "dht1:GND", "esp:GND.2", "black", [ "v57.6", "h278.4", "v38.4" ] ],
20    [ "dht1:VCC", "esp:3V3", "red", [ "v96", "h163.2" ] ],
21    [ "esp:22", "lcd2:SCL", "green", [ "h48", "v-57.6", "h-192", "v-134.7" ] ],
22    [ "esp:21", "lcd2:SDA", "green", [ "h48", "v-86.4", "h-201.6", "v-144.2" ] ],
23    [ "lcd2:VCC", "esp:3V3", "red", [ "h-38.4", "v192.1" ] ],
24    [ "lcd2:GND", "esp:GND.2", "black", [ "h-38.4", "v163.2", "h211.2", "v38.4" ] ],
25    [ "dht1:SDA", "esp:13", "green", [ "v0" ] ]
26  ],
27  "dependencies": {}
28 }
```

```
{
  "version": 1,

  "author": "RUMIRIS BUTARBUTAR",

  "editor": "wokwi",

  "parts": [

    {
      "type":
"board-esp32-devkit-c-v4", "id":
"esp", "top": 86.4, "left": 4.84,
"attrs": {} },

    {
      "type": "wokwi-dht22", "id":
"dht1", "top": -95.7, "left":
-168.6, "attrs": {} },

    {
```

```
    "type": "wokwi-lcd2004",

    "id": "lcd2",

    "top": -118.4,

    "left": -23.2,

    "attrs": { "pins": "i2c" }

  ]

  "connections": [

    [
      "esp:TX",
      "serialMonitor:RX", "", [] ],

    [
      "esp:RX",
      "serialMonitor:TX", "", [] ],

    [ "dht1:GND", "esp:GND.2",
"black", [ "v57.6", "h278.4",
"v38.4" ] ],

    [ "dht1:VCC", "esp:3V3",
"red", [ "v96", "h163.2" ] ],

    [ "esp:22", "lcd2:SCL",
"green", [ "h48", "v-57.6",
"h-192", "v-134.7" ] ],

    [ "esp:21", "lcd2:SDA",
"green", [ "h48", "v-86.4",
"h-201.6", "v-144.2" ] ],

    [ "lcd2:VCC", "esp:3V3",
"red", [ "h-38.4", "v192.1" ] ],

    [ "lcd2:GND", "esp:GND.2",
"black", [ "h-38.4", "v163.2",
"h211.2", "v38.4" ] ],

    [ "dht1:SDA", "esp:13",
"green", [ "v0" ] ]

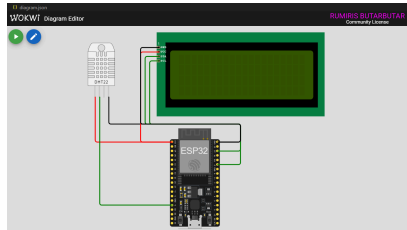
  ],

}
```

```
"dependencies": {}
```

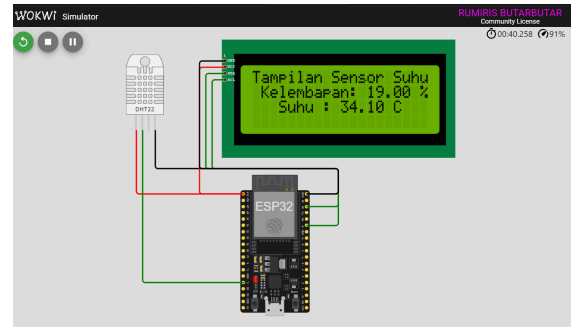
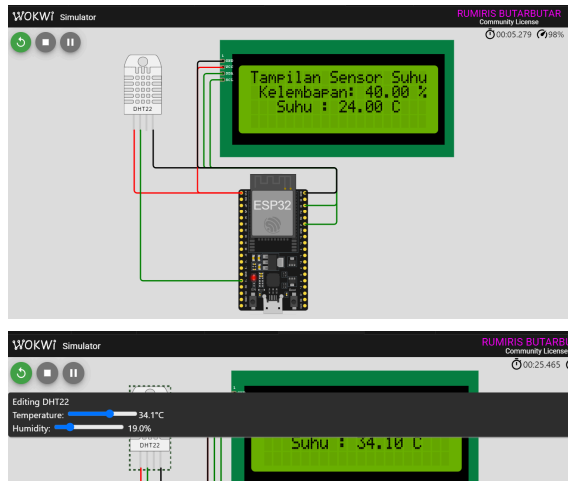
```
}
```

12. Selanjutnya *rename* file diagram pada Visual Studio Code menjadi diagram.json, maka akan muncul simulasi seperti berikut



### 3. Pengujian Hasil dan Pembahasan

Pengujian dilakukan dengan menjalankan simulasi di Wokwi dan mengunggah program ke ESP32 melalui VS Code. Sensor DHT22 berhasil membaca suhu dan kelembaban lingkungan, lalu menampilkan data secara real-time di LCD I2C. Hasil pengujian menunjukkan bahwa sistem bekerja dengan baik dan sesuai dengan yang diharapkan.



### 4. Kesimpulan

Implementasi sensor suhu menggunakan **ESP32**, **DHT22**, dan **LCD I2C** berhasil dilakukan tanpa kendala. Sistem dapat membaca suhu serta kelembaban dan menampilkannya dengan akurat. Dengan metode ini, pemantauan suhu secara real-time menjadi lebih efisien dan mudah diterapkan dalam berbagai keperluan.