

PRATIKUM MONITORING SUHU DAN KELEMBABAN MENGUNAKAN PERANGKAT ESP8266 DAN SENSOR DHT11

Mata Kuliah Sistem Berbasis Internet Of Things

Dosen Pengampu : Solichudin, S.Pd., M.T.



Disusun Oleh :

Adam Achsanul Munzali	(2208096055)
Muhammad Ilham Dwi Prasetyo	(2208096065)
Muhammad Azhar Athaya	(2208096076)

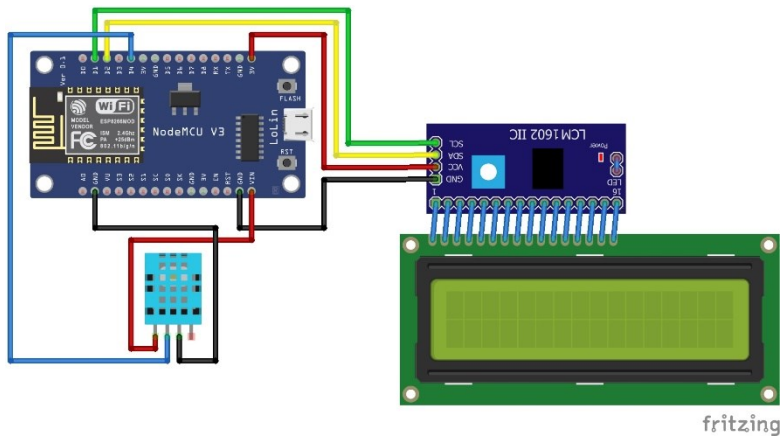
**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG**

2024/2025

A. Tujuan Percobaan

1. Mahasiswa mampu menggunakan NodeMCU ESP8266
2. Mahasiswa mampu menggunakan DHT11
3. Mahasiswa mampu membuat perogram untuk DHT11 ke NodeMCU ESP8266

B. Desain Circuit



C. Tabel Kabel Jumper

Komponen	Pin NodeMCU ESP8266	Pin Komponen	Keterangan
DHT11 (Sensor Suhu dan Kelembapan)	D4	Data	Menghubungkan pin data DHT11 ke pin D4 NodeMCU.
	3.3V	VCC	Memberikan daya 3.3V ke sensor DHT11.
	GND	GND	Ground untuk sensor DHT11.
LCD I2C (16x2)	SDA (D2)	SDA	Pin data serial LCD I2C terhubung ke SDA NodeMCU.
	SCL (D1)	SCL	Pin clock serial LCD I2C terhubung ke SCL NodeMCU.
	5V	VCC	Memberikan daya 5V ke modul LCD I2C.
	GND	GND	Ground untuk modul LCD I2C.

D. Prinsip Kerja

Sistem ini menggunakan **NodeMCU ESP8266** untuk membaca data dari sensor DHT11, menampilkan data tersebut pada LCD I2C, dan mengirimkan data ke platform

ThingSpeak melalui koneksi WiFi. Berikut adalah penjelasan prinsip kerja berdasarkan tabel koneksi:

Langkah Kerja Sistem

1. Inisialisasi Komponen

- **NodeMCU** diinisialisasi untuk mengatur komunikasi dengan sensor DHT11, LCD I2C, dan platform ThingSpeak.
- LCD I2C diinisialisasi untuk menerima data suhu dan kelembapan serta menampilkan informasi.
- Koneksi WiFi diatur menggunakan nama SSID dan password.

2. Pembacaan Data Sensor (DHT11)

- **Sensor DHT11** membaca suhu dan kelembapan lingkungan melalui pin data (terhubung ke pin **D4 NodeMCU**).
- Data yang diperoleh diverifikasi untuk memastikan validitas pembacaan.

3. Tampilan Data pada LCD

- Data suhu dan kelembapan ditampilkan pada **LCD I2C**.
- Komunikasi antara NodeMCU dan LCD dilakukan melalui protokol **I2C**, dengan jalur **SDA** (pin **D2**) dan **SCL** (pin **D1**).
- Modul LCD mendapatkan daya dari **pin 5V NodeMCU**.

4. Pengiriman Data ke ThingSpeak

- NodeMCU mengirim data suhu dan kelembapan ke platform **ThingSpeak** menggunakan library ThingSpeak dan koneksi WiFi.
- Data dikirim ke field yang sesuai pada channel ThingSpeak, yaitu **Field1 untuk suhu** dan **Field2 untuk kelembapan**.

5. Pembaruan Data

- Sistem mengirimkan pembaruan data ke ThingSpeak setiap **20 detik**.
- Jika terjadi kegagalan pengiriman data, sistem akan mencatatnya di Serial Monitor.

Alur Proses

1. NodeMCU menyuplai daya ke sensor DHT11 (3.3V) dan modul LCD (5V).

2. DHT11 membaca suhu dan kelembapan, lalu mengirimkan data ke NodeMCU.
3. NodeMCU memproses data dan menampilkan hasil pada LCD melalui jalur I2C.
4. NodeMCU mengirimkan data suhu dan kelembapan ke platform ThingSpeak menggunakan koneksi WiFi.
5. Proses ini berulang setiap 20 detik untuk pembaruan data.

E. Coding

```

/*****/
//  PRAKTIKUM IOT MONITORING SUHU DAN KELEMBAPAN  //
//  TEKNOLOGI INFORMASI - FAKULTAS SAINS DAN TEKNOLOGI  //
//  UIN WALLISONO SEMARANG  //
/*****/

/*****/
// KELOMPOK: 5          // Nama kelompok
// ANGGOTA:            // Daftar anggota kelompok
// 1. Adam Achsanul Munzali //
// 2. M. Ilham Dwi P      //
// 3. M. Azhar Athaya     //
/*****/

#include <DHT.h> // Library untuk sensor DHT11
#include <ESP8266WiFi.h> // Library untuk konektivitas WiFi
#include <Wire.h> // Library untuk komunikasi I2C
#include <LiquidCrystal_PCF8574.h> // Library untuk LCD dengan antarmuka I2C
#include <ThingSpeak.h> // Library untuk mengirim data ke ThingSpeak

// Konfigurasi Thingspeak
unsigned long channelID = 2920499; // ID channel ThingSpeak (diubah sesuai kebutuhan)
const char* apiKey = "88U9AS22EGE0D29D"; // API Key ThingSpeak (diubah sesuai kebutuhan)

// Konfigurasi WiFi
const char* ssid = "Realmi5i"; // Nama SSID WiFi

```

```

const char* password = "00009999"; // Password WiFi

// Konfigurasi DHT
#define DHTPIN D4 // Pin GPIO NodeMCU tempat sensor DHT11 terhubung
#define DHTTYPE DHT11 // Jenis sensor DHT yang digunakan
DHT dht(DHTPIN, DHTTYPE); // Inisialisasi objek DHT

// Konfigurasi LCD I2C
LiquidCrystal_PCF8574 lcd(0x27); // Alamat I2C LCD (default 0x27)
WiFiClient client; // Inisialisasi klien WiFi

// Fungsi untuk menampilkan informasi kelompok pada LCD
void displayInfo() {
    lcd.clear(); // Membersihkan layar LCD
    lcd.setCursor(0, 0); // Menempatkan kursor di baris 1, kolom 1
    lcd.print("UIN Walisongo"); // Menampilkan teks
    lcd.setCursor(0, 1); // Menempatkan kursor di baris 2, kolom 1
    lcd.print("Semarang"); // Menampilkan teks
    delay(2000); // Menunda selama 2 detik

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("PRODI TI.");
    lcd.setCursor(0, 1);
    lcd.print("FST");
    delay(2000);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("IOT");
    lcd.setCursor(0, 1);
    lcd.print("SK");
    delay(2000);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Kelompok: 5");
    lcd.setCursor(0, 1);
    lcd.print("Azhar,Adam,Ilham");
    delay(3000);
}

```

```

}

void setup() {
  Serial.begin(9600); // Memulai komunikasi serial dengan baud rate 9600
  delay(10); // Menunda selama 10 milidetik

  dht.begin(); // Inisialisasi sensor DHT

  lcd.begin(16, 2); // Inisialisasi LCD dengan ukuran 16x2
  lcd.setBacklight(255); // Menghidupkan backlight LCD
  lcd.clear(); // Membersihkan layar LCD

  displayInfo(); // Menampilkan informasi kelompok pada LCD

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Connecting WiFi"); // Menampilkan status koneksi WiFi
  WiFi.begin(ssid, password); // Menghubungkan ke WiFi dengan SSID dan
password

  while (WiFi.status() != WL_CONNECTED) { // Menunggu hingga WiFi terhubung
    delay(500); // Menunda 500 milidetik
    Serial.print("."); // Menampilkan tanda titik sebagai indikator proses
  }

  Serial.println();
  Serial.println("WiFi Connected!"); // Menampilkan status koneksi berhasil
  Serial.print("IP Address: "); // Menampilkan alamat IP perangkat
  Serial.println(WiFi.localIP());

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("WiFi Connected"); // Menampilkan pesan koneksi berhasil di LCD
  delay(2000);
  lcd.clear();

  ThingSpeak.begin(client); // Inisialisasi ThingSpeak
}

void loop() {

```

```

float h = dht.readHumidity(); // Membaca kelembapan dari sensor DHT11
float t = dht.readTemperature(); // Membaca suhu dari sensor DHT11

if (isnan(h) || isnan(t)) { // Memeriksa jika pembacaan gagal
  Serial.println("Sensor DHT11 gagal membaca data!");
  lcd.setCursor(0, 0);
  lcd.print("Sensor Error!"); // Menampilkan pesan error di LCD
  delay(2000);
  return;
}

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

lcd.setCursor(0, 0);
lcd.print("Temp: ");
lcd.print(t);
lcd.print(" C  ");
lcd.setCursor(0, 1);
lcd.print("Hum: ");
lcd.print(h);
lcd.print(" %  ");

ThingSpeak.setField(1, t); // Mengirim suhu ke Field1 ThingSpeak
ThingSpeak.setField(2, h); // Mengirim kelembapan ke Field2 ThingSpeak

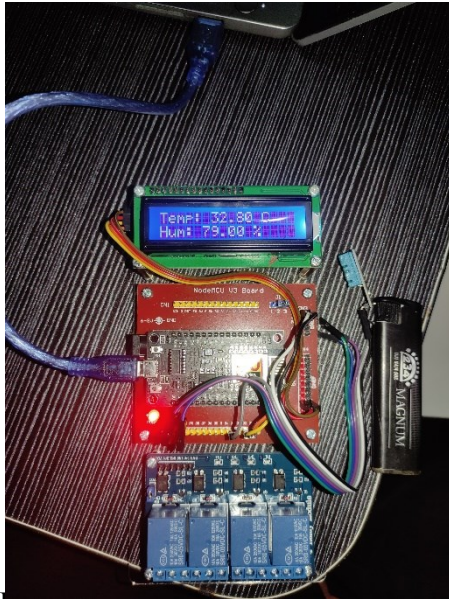
int response = ThingSpeak.writeFields(channelID, apiKey); // Mengirim data ke
ThingSpeak
if (response == 200) {
  Serial.println("Data berhasil dikirim ke Thingspeak!");
} else {
  Serial.print("Error mengirim data ke Thingspeak: ");
  Serial.println(response);
}

delay(20000); // Menunda pengulangan loop selama 20 detik
}

```

F. Hasil Uji Coba

- Rangkaian dan tampilan pada LCD



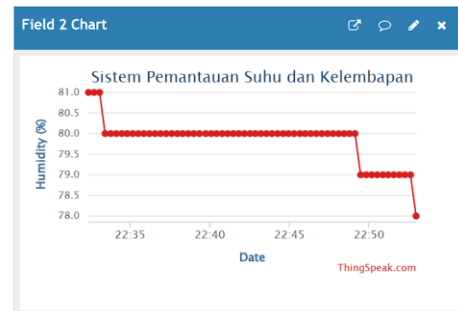
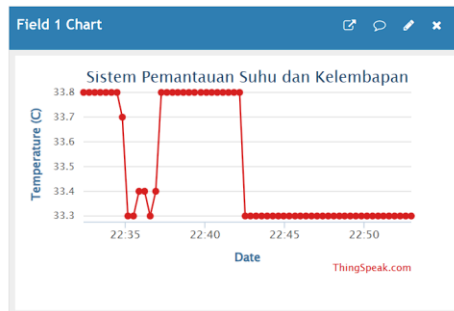
- Hasil grafik pada web Thingspeak

Channel Stats

Created: 8 days ago

Last entry: 38 minutes ago

Entries: 72



G. Kesimpulan

Dari hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa:

1. NodeMCU ESP32 dapat digunakan sebagai mikrokontroler yang handal untuk membaca dan memproses data dari sensor lingkungan seperti DHT22.
2. Sensor DHT22 mampu mengukur suhu dan kelembaban secara real-time, dengan tingkat akurasi yang cukup untuk kebutuhan monitoring dasar.

3. Data suhu dan kelembaban yang diperoleh berhasil ditampilkan secara serial di Serial Monitor dan secara visual melalui LCD 16x2 I2C, menunjukkan keberhasilan komunikasi antara perangkat-perangkat yang terlibat (ESP32, DHT22, dan LCD).
4. Penggunaan simulator Wokwi sangat membantu dalam pengujian dan pengembangan awal sistem monitoring tanpa memerlukan perangkat keras fisik secara langsung.
5. Pemrograman berbasis Arduino IDE memungkinkan integrasi berbagai komponen IoT dengan mudah dan efisien melalui penggunaan pustaka-pustaka pendukung seperti DHT.h dan LiquidCrystal_I2C.h.