

PRATIKUM MONITORING SUHU BERBASIS IOT MENGUNAKAN ESP32 DAN SENSOR DHT11 DENGAN UJI COBA PANAS EKSTERNAL

Mata Kuliah Sistem Berbasis Internet Of Things

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



Disusun Oleh :

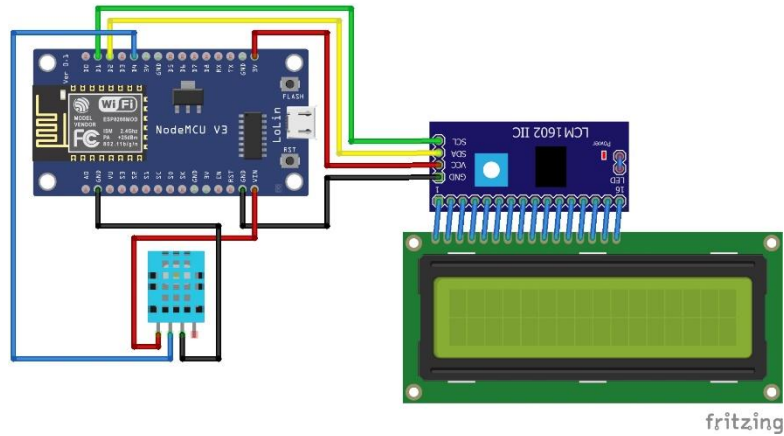
Adam Achsanul Munzali	(2208096055)
Muhammad Ilham Dwi Prasetyo	(2208096065)
Muhammad Azhar Athaya	(2208096076)

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

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



Gambar 1. Desain circuit

C. Tabel Kabel Jumper

Komponen	Pin ESP8266	Pin Komponen	Keterangan
DHT11	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**.

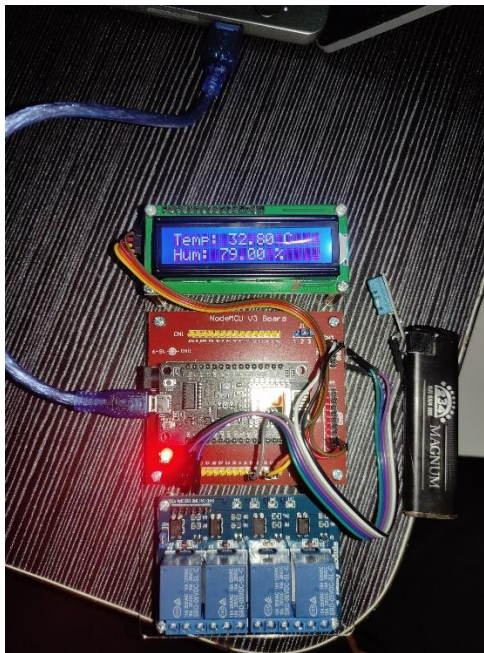
- Jika terjadi kegagalan pengiriman data, sistem akan mencatatnya di Serial Monitor.

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. Hasil Uji Coba

- Rangkaian dan tampilan pada LCD



Gambar 2. Rangkaian Uji Coba ESP32 dan DHT11

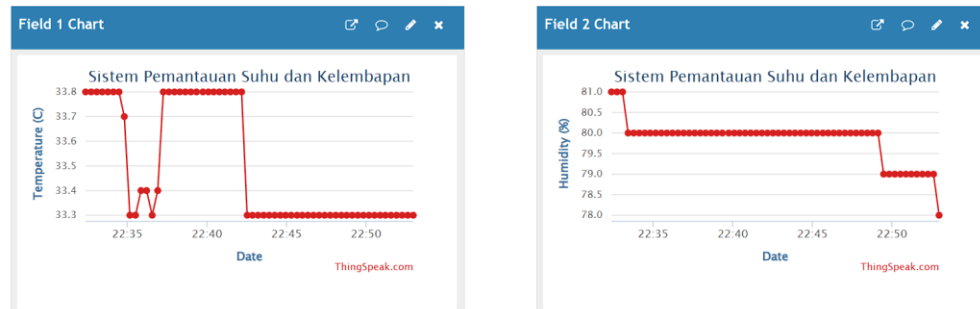
- **Hasil grafik pada web ThinkSpeak**

Channel Stats

Created: [8 days ago](#)

Last entry: [38 minutes ago](#)

Entries: 72



Gambar 3. Grafik inputan sensor DHT11 pada web ThinkSpeak

F. 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.

REFERENSI

1. **DHT Sensor Library** - Adafruit.
Tersedia di: <https://github.com/adafruit/DHT-sensor-library>
2. **ESP8266WiFi Library** - Arduino.
Tersedia di: <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>
3. **ThingSpeak Documentation** - MathWorks.
Tersedia di: <https://www.mathworks.com/help/thingspeak/>
4. **Wokwi Simulator** - Wokwi IoT Simulator for Arduino and ESP32.
Tersedia di: <https://wokwi.com/>

LAMPIRAN PROGRAM

```
//*****//
//          PRAKTIKUM IoT: MONITORING SUHU & KELEMBAPAN          //
//          TEKNOLOGI INFORMASI - FAKULTAS SAINS DAN TEKNOLOGI    //
//          UIN WALISONGO SEMARANG                                //
//*****//

/*****/
// KELOMPOK: 5                                                    //
// ANGGOTA:                                                         //
//   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 I2C
#include <ThingSpeak.h>         // Library untuk koneksi ke
ThingSpeak

// Konfigurasi ThingSpeak
unsigned long channelID = 2920499; // Ganti sesuai
Channel ID ThingSpeak
const char* apiKey = "88U9AS22EGE0D29D"; // Ganti sesuai
API Key ThingSpeak

// Konfigurasi WiFi
const char* ssid = "Realmi5i"; // SSID WiFi
const char* password = "00009999"; // Password
WiFi

// Konfigurasi DHT
#define DHTPIN D4 // Pin sensor
DHT11
#define DHTTYPE DHT11 // Tipe sensor
DHT dht(DHTPIN, DHTTYPE); // Inisialisasi
objek sensor

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

// Fungsi tampilan info kelompok
void displayInfo() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("UIN Walisongo");
    lcd.setCursor(0, 1);
```

```

    lcd.print("Semarang");
    delay(2000);

    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 Praktikum");
    lcd.setCursor(0, 1);
    lcd.print("Monitoring");
    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);
    delay(10);

    dht.begin();
    lcd.begin(16, 2);
    lcd.setBacklight(255);
    lcd.clear();

    displayInfo(); // Tampilkan info kelompok

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Connecting WiFi...");
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println();
    Serial.println("WiFi Connected!");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("WiFi Connected");
    delay(2000);
    lcd.clear();
}

```



```

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

void loop() {
    float h = dht.readHumidity();           // Baca kelembapan
    float t = dht.readTemperature();        // Baca suhu

    if (isnan(h) || isnan(t)) {
        Serial.println("Sensor DHT11 gagal membaca data!");
        lcd.setCursor(0, 0);
        lcd.print("Sensor Error!");
        delay(2000);
        return;
    }

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

    // Tampilkan di LCD
    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(" % ");

    // Kirim ke ThingSpeak
    ThingSpeak.setField(1, t);
    ThingSpeak.setField(2, h);
    int response = ThingSpeak.writeFields(channelID, apiKey);

    if (response == 200) {
        Serial.println("Data berhasil dikirim ke ThingSpeak!");
    } else {
        Serial.print("Gagal kirim data: ");
        Serial.println(response);
    }

    delay(20000); // Delay 20 detik sebelum pengulangan
}

```