

PRATIKUM MONITORING SUHU DAN KELEMBABAN MENGUNAKAN SIMULATOR NODEMCU ESP32 DAN SENSOR DHT22

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)

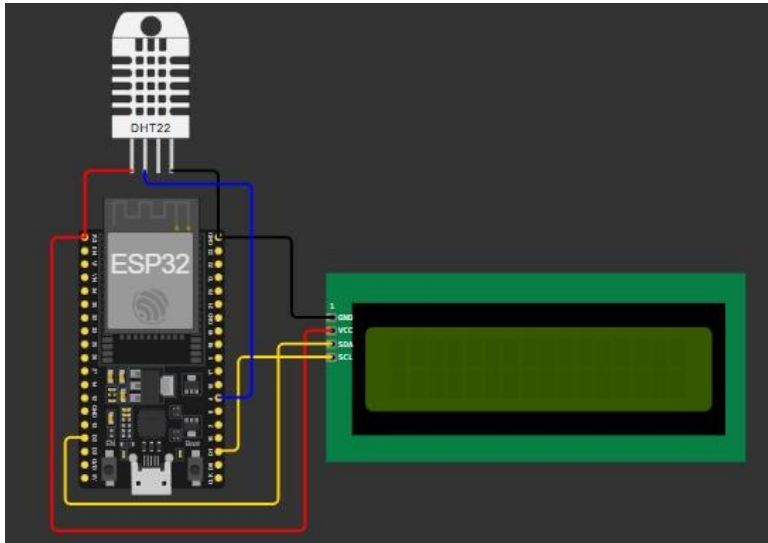
**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG
2024/2025**

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. Link Simulator ESP32 <https://wokwi.com/projects/428373386313072641>

C. Desain Circuit



D. Tabel Kabel Jumper

Komponen	Pin	Terhubung ke Pin ESP32	Keterangan
DHT22	VCC	3.3V	Tegangan untuk sensor
	GND	GND	Ground
	DATA	GPIO 4	Sesuai definisi <code>#define DHTPIN 4</code>
LCD I2C	VCC	5V atau 3.3V	Tegangan (sesuai modul LCD, biasanya 5V)
	GND	GND	Ground
	SDA	GPIO 21	Pin I2C SDA default ESP32
	SCL	GPIO 22	Pin I2C SCL default ESP32

E. Coding

```
//////////////////////////////////////////
//          MODUL PRAKTIKUM IOT - 01          //
//          MONITORING SUHU DAN                //
//          KELEMBABAN                        //
//  TEKNOLOGI INFORMASI - FAKULTAS SAINS DAN TEKNOLOGI  //
//          UIN WALISONGO SEMARANG            //
//////////////////////////////////////////

/// kelompok :                               ///
/// Anggota  : 1. Adam Achsanul M            ///
///           2. M Azhar Athaya              ///
///           3. M Ilham Dwi P               ///

// Library untuk komunikasi I2C
#include <Wire.h>

// Library untuk LCD I2C 16x2
#include <LiquidCrystal_I2C.h>

// Library untuk sensor DHT (suhu & kelembaban)
#include <DHT.h>

// Mendefinisikan pin yang digunakan untuk data DHT22 (GPIO 4)
#define DHTPIN 4

// Menentukan jenis sensor yang digunakan (DHT22)
#define DHTTYPE DHT22

// Membuat objek sensor DHT dengan pin dan tipe yang telah didefinisikan
DHT dht(DHTPIN, DHTTYPE);

// Membuat objek LCD dengan alamat I2C 0x27 dan ukuran 16x2 karakter
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
    // Memulai komunikasi serial pada baud rate 115200
    Serial.begin(115200);

    // Inisialisasi LCD
    lcd.init();

    // Mengaktifkan lampu latar LCD
}
```

```

lcd.backlight();

// Menetapkan kursor LCD ke baris 0 kolom 0
lcd.setCursor(0, 0);

// Menampilkan teks awal di LCD
lcd.print("Wokwi DHT22");

// Memulai sensor DHT
dht.begin();

// Delay untuk memberikan waktu inisialisasi
delay(2000);
}

void loop() {
    // Membaca suhu dari sensor DHT
    float temp = dht.readTemperature();

    // Membaca kelembaban dari sensor DHT
    float hum = dht.readHumidity();

    // Mengecek apakah hasil pembacaan valid
    if (isnan(temp) || isnan(hum)) {
        // Menampilkan pesan error di Serial Monitor
        Serial.println("Gagal membaca sensor DHT!");

        // Menghapus tampilan LCD
        lcd.clear();

        // Menetapkan posisi kursor LCD
        lcd.setCursor(0, 0);

        // Menampilkan pesan error di LCD
        lcd.print("Sensor Error");

        // Delay sebelum mencoba lagi
        delay(2000);
        return;
    }

    // Menampilkan data suhu ke Serial Monitor
    Serial.print("Suhu: ");
    Serial.print(temp);
    Serial.print(" C | Kelembaban: ");

```

```

Serial.print(hum);
Serial.println(" %");

// Menghapus tampilan LCD
lcd.clear();

// Menetapkan posisi kursor dan menampilkan suhu di baris pertama LCD
lcd.setCursor(0, 0);
lcd.print("Suhu: ");
lcd.print(temp, 1); // Menampilkan 1 angka di belakang koma

// Menampilkan kelembaban di baris kedua LCD
lcd.setCursor(0, 1);
lcd.print("Lembab: ");
lcd.print(hum, 1); // Menampilkan 1 angka di belakang koma
lcd.print(" %");

// Delay selama 2 detik sebelum pembacaan berikutnya
delay(2000);
}

```

F. Hasil Uji Coba

Suhu: 24.00 C		Kelembaban: 40.00 %
Suhu: 24.00 C		Kelembaban: 40.00 %
Suhu: 24.00 C		Kelembaban: 40.00 %
Suhu: 24.00 C		Kelembaban: 40.00 %
Suhu: 24.00 C		Kelembaban: 40.00 %

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.