

**LAPORAN PRAKTIKUM
MATA KULIAH INTERNET OF THINGS**

**Menampilkan Data Cuaca Suhu, Deskripsi, Kelembaban, dan
Kecepatan Angin pada LCD 16x2 dengan navigasi tombol**

Dosen Pengampu :

Ir. Subairi, ST., MT., IPM



Disusun Oleh :

**Desi Eka Mardiani
233140707111084**

Fakultas Vokasi, Universitas Brawijaya

Email : desiekaa71@student.ub.ac.id

Laporan Praktikum Mata Kuliah Internet of Things
Menampilkan Data Cuaca Suhu, Deskripsi, Kelembaban, dan
Kecepatan Angin pada LCD 16x2 dengan Navigasi Tombol

Desi Eka Mardiani

Program Studi Teknologi Informasi, Fakultas Vokasi, Universitas Brawijaya

Abstrak : Praktikum ini bertujuan merancang sistem Internet of Things (IoT) sederhana, yang dapat menampilkan data cuaca secara *real-time* pada LCD 16x2 dengan sistem navigasi menggunakan tombol. Sistem ini memanfaatkan mikrokontroler ESP32 yang terhubung ke internet untuk mengambil data cuaca dari API OpenWeatherMap, termasuk suhu, deskripsi cuaca, kelembaban, dan kecepatan angin. Data tersebut kemudian ditampilkan pada layar LCD dengan navigasi melalui dua tombol yang memungkinkan pengguna dapat melihat informasi cuaca yang berbeda. Praktikum ini memanfaatkan simulasi Wokwi untuk perancangan dan pengujian sistemnya.

Kata Kunci : IoT, ESP32, OpenWeatherMap API, LCD 16x2, tombol navigasi, suhu, kelembaban, kecepatan angin.

Abstrak : This practicum aims to design a simple Internet of Things (IoT) system, which can display real-time weather data on a 16x2 LCD with a navigation system using buttons. This system utilizes an ESP32 microcontroller connected to the internet to retrieve weather data from the OpenWeatherMap API, including temperature, weather description, humidity, and wind speed. The data is then displayed on the LCD screen with navigation via two buttons that allow users to view different weather information. This practicum utilizes Wokwi simulation for the design and testing of its system..

Keywords : IoT, ESP32, OpenWeatherMap API, 16x2 LCD, navigation buttons, temperature, humidity, wind speed.

1. PENDAHULUAN

1.1. Latar Belakang

Internet of Things (IoT) adalah konsep yang menghubungkan perangkat fisik ke internet untuk bertukar data secara *real-time*. Perkembangan IoT telah membawa perubahan besar dalam berbagai aspek kehidupan, mulai dari sistem industri, pertanian, kesehatan, hingga rumah tangga. IoT memungkinkan perangkat-perangkat seperti sensor, aktuator, dan mikrokontroler untuk saling berkomunikasi, mengumpulkan, mengolah, dan membagikan data tanpa intervensi manusia secara langsung.

Salah satu aplikasi IoT yang relevan di era digital saat ini adalah monitoring cuaca, dimana parameter seperti suhu, kelembaban, deskripsi cuaca, dan kecepatan angin

dapat diambil dari layanan web dan divisualisasikan secara lokal meskipun jaraknya jauh. Data cuaca merupakan informasi penting yang mempengaruhi berbagai aspek kehidupan seperti pertanian untuk menentukan waktu tanam dan panen, transportasi untuk keselamatan perjalanan, dan kegiatan sehari-hari masyarakat seperti perancangan aktivitas luar ruangan.

Pada praktikum ini, digunakan ESP32 sebagai mikrokontroler utama yang memiliki kemampuan konektivitas WiFi, cocok untuk aplikasi IoT yang membutuhkan akses ke internet. ESP32 akan mengambil data cuaca kota tertentu dari API OpenWeatherMap, kemudian memproses dan menampilkannya pada LCD 16x2. Untuk membuat sistem lebih interaktif, ditambahkan dua tombol sebagai navigasi. Dengan tombol ini, pengguna dapat memilih informasi apa yang ingin ditampilkan pada LCD, seperti suhu, kelembaban, deskripsi, cuaca, dan kecepatan angin.

Penerapan konsep ini dapat dijadikan sebagai media pembelajaran efektif untuk memahami konsep dasar IoT, pemrograman mikrokontroler, serta integrasi perangkat keras, dan perangkat lunak. Praktikum ini juga melatih keterampilan dalam menggunakan platform simulasi Wokwi, sehingga mahasiswa dapat merancang, menguji, memvalidasi sistem IoT tanpa harus memiliki perangkat fisik secara langsung. Dengan adanya sistem monitoring cuaca berbasis IoT ini, diharapkan dapat meningkatkan efisiensi, akurasi, dan kemudahan dalam memperoleh data cuaca.

1.2. Tujuan Praktikum

Secara spesifik, berikut adalah beberapa tujuan dari praktikum ini :

1. Memahami konsep IoT dengan implementasi pada sistem cuaca.
2. Mempelajari cara mengakses dan mengolah data dari API web menggunakan ESP32.
3. Mengintegrasikan komponen antarmuka pengguna (LCD dan tombol) dengan sistem IoT
4. Mengembangkan keterampilan pemrograman ESP32 untuk aplikasi praktis.
5. Mempelajari teknik parsing data JSON dari respons API.
6. Merancang sistem navigasi sederhana untuk mengakses berbagai informasi pada display terbatas.

2. METODOLOGI

2.1. Alat dan Bahan

Praktikum ini dilaksanakan menggunakan platform Wokwi. Adapun perangkat keras dan sensor yang digunakan adalah Simulasi Mikrokontroler ESP32, LDR atau *Light Dependent Resistor* dengan modul I2C, 2 tombol push bottom (merah dan biru), kabel jumper, dan koneksi internet untuk akses API OpenWeatherMap.

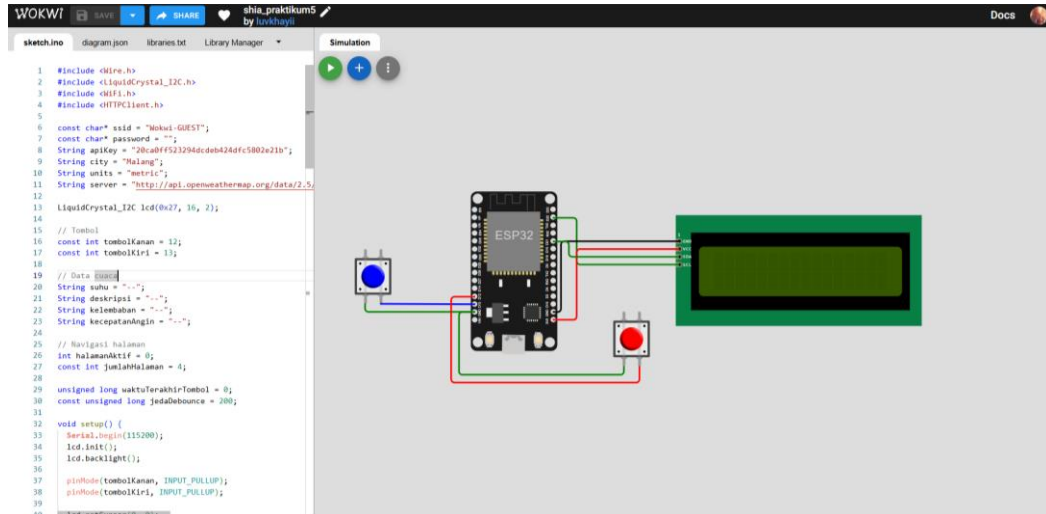
2.2. Langkah Implementasi

1. Persiapan Komponen di Wokwi

- Membuat project baru di simulasi Wokwi
- Menambahkan komponen ESP32, LCD I2C 16x2, dan tombol push button.

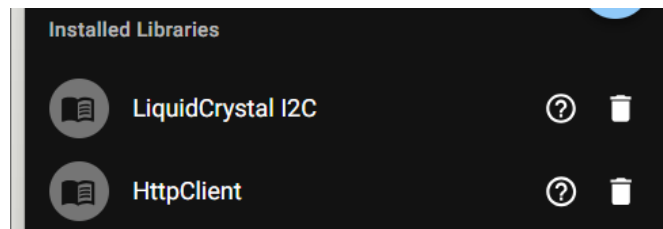
2. Persiapan Komponen di Wokwi

- Menghubungkan LCD I2C ke ESP32 dengan SDA ke pin 21 dan SCL ke pin 22.
- Memasang tombol merah ke pin 12 dan tombol biru ke pin 13.
- Mengatur alamat I2C LCD pada 0x27



3. Pemrograman ESP32

- Mendeklarasikan library yang dibutuhkan Wire, LiquidCrystal_I2C, WiFi, dan HTTPClient.



```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3  #include <WiFi.h>
4  #include <HTTPClient.h>
```

- Mengkonfigurasi parameter WiFi dan API OpenWeatherMap.

```
10 const char* ssid = "Wokwi-GUEST";
11 const char* password = "";
12 String apiKey = "20ca0ff523294dcdeb424dfc5802e21b";
13 String city = "Malang";
14 String units = "metric";
15 String server = "http://api.openweathermap.org/data/2.5/weather?q=" + city + "&units=" + unit
```

- Implementasi fungsi untuk koneksi WiFi dan pengambilan data cuaca.
- Perancangan sistem navigasi halaman menggunakan tombol

```

29 // Navigasi halaman
30 int halamanAktif = 0;
31 const int jumlahHalaman = 4;
32 unsigned long waktuTerakhirTombol = 0;
33 const unsigned long jedaDebounce = 200;

```

Jika sudah memastikan tidak ada masalah dan simulasi berhasil, langkah berikutnya adalah mengimplementasikan proyek ini di perangkat keras ESP32 fisik. Proses ini melibatkan pemindahan kode dari Wokwi ke Visual Studio Code dengan PlatformIO, memastikan library yang sesuai telah diinstal, dan menghubungkan ESP32 fisik ke komputer untuk mengunggah kode program.

3. HASIL DAN PEMBAHASAN

3.1. Hasil

- ESP32 berhasil terhubung ke jaringan WiFi simulasi Wokwi-GUSET

```

Connecting to WiFi...
{"coord":{"lon":112.6304,"lat":-7.9797},"weather":[{"id":804,"main":"clouds","description":"overcast clouds","icon":"04d"}],"base":"stations","main":{"temp":27.7,"feels_like":33.44,"temp_min":27.7,"temp_max":27.7,"pressure":1011,"humidity":94,"sea_le

```

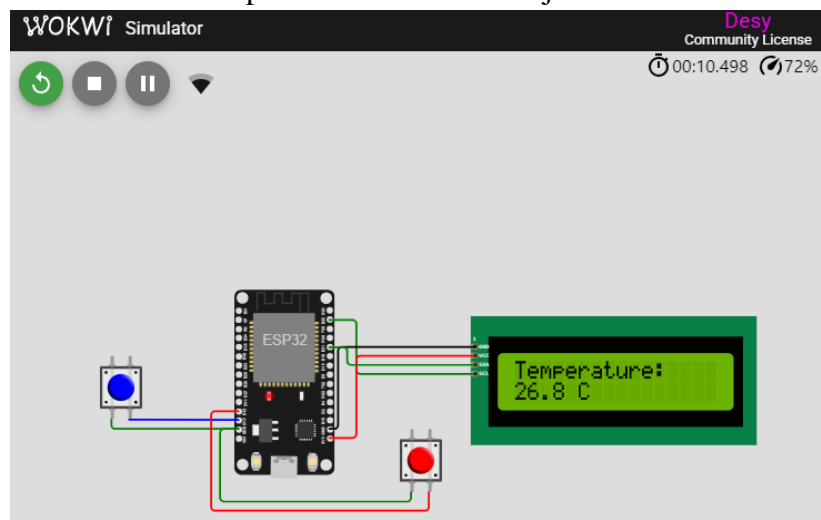
- Berhasil mengambil data cuaca kota Malang dari API OpenWeatherMap

```

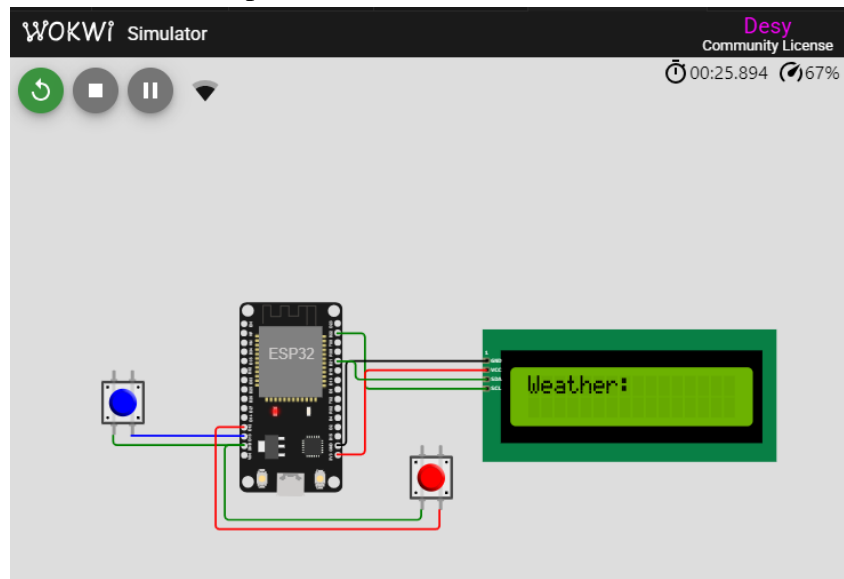
8},"visibility":10000,"wind":
{"speed":0.31,"deg":40,"gust":0.32},"clouds":
{"all":96},"dt":1746576799,"sys":
{"country":"ID","sunrise":1746570733,"sunset":1746613191},"time
zone":"25200","id":1636722,"name":"Malang","cod":200}

```

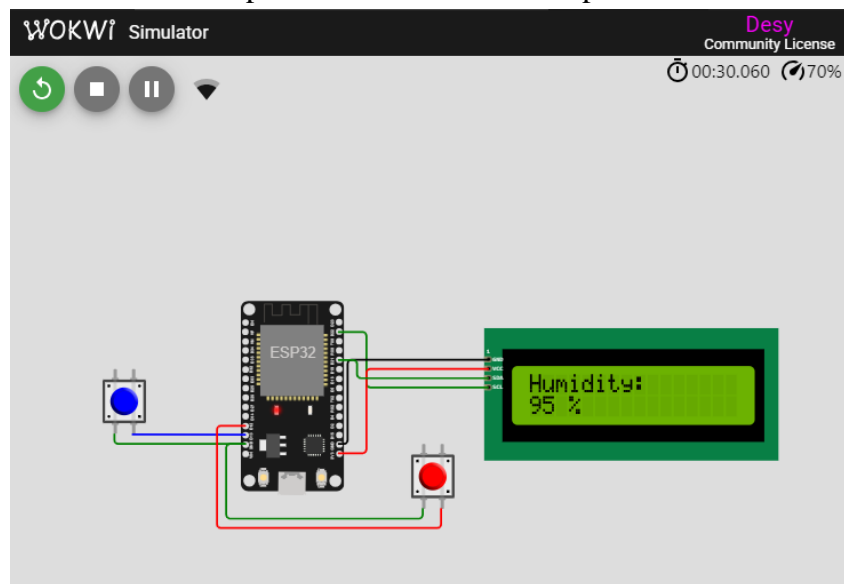
- Tampilan data LCD 16x2
 - Halaman 1 menampilkan suhu dalam derajat Celcius



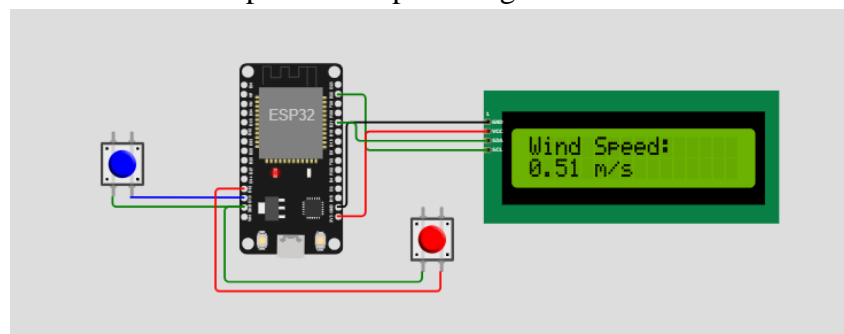
- Halaman 2 menampilkan cuaca



- Halaman 3 menampilkan kelembaban dalam persen



- Halaman 4 menampilkan kecepatan angin dalam m/s



d. Menambahkan kode program di beberapa bagian dalam Visual Studio Code.

1. Main.cpp

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <HTTPClient.h>

// Deklarasi fungsi
void updateDisplay();
void updateWeatherData();

const char* ssid = "Wokwi-GUEST";
const char* password = "";
String apiKey = "20ca0ff523294dcdeb424dfc5802e21b";
String city = "Malang";
String units = "metric";
String server = "http://api.openweathermap.org/data/2.5/weather?q=" +
city + "&units=" + units + "&appid=" + apiKey;

LiquidCrystal_I2C lcd(0x27, 16, 2);

// Tombol
const int tombolKanan = 13;
const int tombolKiri = 12;

// Data cuaca
String suhu = "---";
String deskripsi = "---";
String kelembaban = "---";
String kecepatanAngin = "---";

// Navigasi halaman
int halamanAktif = 0;
const int jumlahHalaman = 4;
unsigned long waktuTerakhirTombol = 0;
const unsigned long jedaDebounce = 200;

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

  // Inisialisasi LCD
  lcd.init();
  lcd.backlight();
```

```

// Inisialisasi tombol
pinMode(tombolKanan, INPUT_PULLUP);
pinMode(tombolKiri, INPUT_PULLUP);

lcd.setCursor(0, 0);
lcd.print("Weather Info:");

// Koneksi WiFi
WiFi.begin(ssid, password);
lcd.setCursor(0, 1);
lcd.print("Connecting...");

while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
}

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

// Ambil data cuaca pertama kali
updateWeatherData();
}

void loop() {
    // Cek tombol kanan (next page)
    if (digitalRead(tombolKanan) == LOW && millis() -
waktuTerakhirTombol > jedaDebounce) {
        halamanAktif = (halamanAktif + 1) % jumlahHalaman;
        waktuTerakhirTombol = millis();
        updateDisplay();
    }

    // Cek tombol kiri (previous page)
    if (digitalRead(tombolKiri) == LOW && millis() -
waktuTerakhirTombol > jedaDebounce) {
        halamanAktif = (halamanAktif - 1 + jumlahHalaman) %
jumlahHalaman;
        waktuTerakhirTombol = millis();
        updateDisplay();
    }
}

```



```

// Update data cuaca setiap menit
static unsigned long lastUpdate = 0;
if (millis() - lastUpdate > 60000 || lastUpdate == 0) {
    updateWeatherData();
    lastUpdate = millis();
}
}

void updateWeatherData() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(server);
        int httpCode = http.GET();

        if (httpCode > 0) {
            String payload = http.getString();
            Serial.println(payload);

            // Parse suhu
            int tempIndex = payload.indexOf("\"temp\":");
            if (tempIndex > 0) {
                suhu = payload.substring(tempIndex + 7, payload.indexOf(",",
tempIndex));
            }

            // Parse deskripsi
            int descIndex = payload.indexOf("\"description\":");
            if (descIndex > 0) {
                deskripsi = payload.substring(descIndex + 14, payload.indexOf("\"",
descIndex + 14));
            }

            // Parse kelembaban
            int humIndex = payload.indexOf("\"humidity\":");
            if (humIndex > 0) {
                kelembaban = payload.substring(humIndex + 11, payload.indexOf(",",
humIndex));
            }

            // Parse kecepatan angin
            int windIndex = payload.indexOf("\"speed\":");
            if (windIndex > 0) {

```

```

        kecepatanAngin = payload.substring(windIndex + 8,
payload.indexOf(",", windIndex));
    }

```

```

        updateDisplay();
    } else {
        Serial.println("Error on HTTP request");
    }
    http.end();
}
}

```

```

void updateDisplay() {
    lcd.clear();

```

```

    switch (halamanAktif) {
        case 0: // Suhu
            lcd.setCursor(0, 0);
            lcd.print("Temperature:");
            lcd.setCursor(0, 1);
            lcd.print(suhu);
            lcd.print(" C");
            break;

```

```

        case 1: // Deskripsi
            lcd.setCursor(0, 0);
            lcd.print("Weather:");
            lcd.setCursor(0, 1);
            lcd.print(deskripsi);
            break;

```

```

        case 2: // Kelembaban
            lcd.setCursor(0, 0);
            lcd.print("Humidity:");
            lcd.setCursor(0, 1);
            lcd.print(kelembaban);
            lcd.print(" %");
            break;

```

```

        case 3: // Kecepatan angin
            lcd.setCursor(0, 0);
            lcd.print("Wind Speed:");
            lcd.setCursor(0, 1);
            lcd.print(kecepatanAngin);

```

```

    lcd.print(" m/s");
    break;
}
}

```

2. Membuat File wokwi.toml

firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'

elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'

3. Diagram.json

