

Laporan Praktikum Internet of Things (IoT) Simulasi Pemantauan Suhu dan Kelembaban Berbasis IoT Menggunakan ESP32 dan MQTT

ADITYA FERDIAN RAMDANI
2331407007111018

Teknologi Informasi, Fakultas Vokasi, Universitas Brawijaya
BRINGMETHEHORIZON389@GMAIL.COM

Abstrak

Praktikum ini bertujuan untuk mengembangkan sistem monitoring cuaca berbasis Internet of Things (IoT) menggunakan platform ESP32. Sistem yang dibangun dapat mengambil data cuaca terkini dari API OpenWeatherMap melalui koneksi Wi-Fi. Data yang diperoleh mencakup informasi suhu, deskripsi cuaca, kelembaban, dan kecepatan angin. Seluruh informasi tersebut kemudian ditampilkan pada layar LCD 16x2 dengan menggunakan interface I2C. Sistem ini dilengkapi dengan tombol navigasi yang memungkinkan pengguna untuk berpindah antara halaman-halaman informasi cuaca. Halaman pertama menampilkan lokasi yang dipilih, untuk lokasi di praktikum kali ini difokuskan ke daerah Malang, Jawa Timur; halaman kedua menunjukkan suhu, halaman ketiga menampilkan deskripsi cuaca yang dapat di-scroll, dan halaman keempat menampilkan informasi mengenai kecepatan angin dan kelembaban. Dengan adanya fitur scroll pada halaman ketiga, pengguna dapat melihat deskripsi cuaca meskipun teks yang ditampilkan lebih panjang dari kapasitas layar. Data cuaca diperbarui secara otomatis setiap 60 detik untuk memastikan informasi yang ditampilkan selalu terbaru. Setiap tombol navigasi memiliki fungsi untuk berpindah antar halaman, dengan dukungan pengaturan debounce untuk mencegah input ganda yang tidak diinginkan. Sistem ini memberikan kemudahan bagi pengguna untuk memonitor kondisi cuaca secara real-time, dan memberikan pengalaman interaktif dengan penggunaan tombol untuk mengganti tampilan halaman informasi cuaca. Penggunaan ESP32 sebagai perangkat utama memungkinkan koneksi yang cepat dan stabil untuk mengakses API serta menampilkan data cuaca yang diperbarui secara berkala. Sistem ini juga mengedepankan efisiensi dalam pemrograman dan penggunaan sumber daya perangkat keras yang ada. Kata Kunci—Internet of Things, ESP32, OpenWeatherMap, Wokwi, PlatformIO

Abstract

This practicum aims to develop an Internet of Things (IoT)-based weather monitoring system using the ESP32 platform. The system can retrieve current weather data from the OpenWeatherMap API via a Wi-Fi connection. The data obtained includes temperature information, weather description, humidity, and wind speed. All information is then displayed on a 16x2 LCD screen using an I2C interface. The system is equipped with navigation buttons that allow the user to move between pages of weather information. The first page displays the selected location, for the location in this practicum focused on Malang, East Java, the second page shows the temperature, the third page displays a scrollable weather description, and the fourth page displays information

about wind speed and humidity. With the scroll feature on the third page, users can see the weather description even though the text displayed is longer than the screen capacity. Weather data is automatically updated every 60 seconds to ensure the information displayed is always up-to-date. Each navigation button has the function to move between pages, with support for debounce settings to prevent unwanted double input. The system makes it easy for users to monitor weather conditions in real-time, and provides an interactive experience with the use of buttons to switch the display of weather information pages. The use of ESP32 as the main device enables a fast and stable connection to access the API and display regularly updated weather data. The system also emphasizes efficiency in programming and the use of existing hardware resources.

Keywords—Internet of Things, ESP32, Traffic Light, Wokwi, PlatformIO1. **Introduction (Pendahuluan)**

Internet of Things (IoT) adalah salah satu konsep teknologi yang berkembang pesat dalam beberapa tahun terakhir, yang memungkinkan perangkat untuk saling terhubung dan berkomunikasi melalui jaringan internet. Salah satu aplikasi IoT yang populer adalah sistem monitoring cuaca, yang memungkinkan pengguna untuk memantau kondisi cuaca secara real-time menggunakan perangkat yang terhubung ke internet. Sistem ini memiliki berbagai manfaat, terutama dalam konteks prediksi cuaca, perencanaan kegiatan luar ruangan, dan manajemen sumber daya alam. Dalam praktikum ini, dibangun sebuah sistem monitoring cuaca berbasis IoT menggunakan platform ESP32. ESP32 adalah salah satu mikrokontroler yang populer karena memiliki kemampuan Wi-Fi dan Bluetooth, sehingga sangat cocok untuk aplikasi berbasis IoT. Sistem ini mengakses data cuaca dari API OpenWeatherMap, yang menyediakan informasi cuaca secara global melalui jaringan internet. OpenWeatherMap adalah layanan cuaca online yang menyediakan data cuaca terkini, perkiraan cuaca, dan informasi atmosfer lainnya. API yang disediakan oleh OpenWeatherMap memungkinkan pengguna untuk mengakses data cuaca secara bebas dengan batasan tertentu, dan memberikan informasi seperti suhu, kelembaban, deskripsi cuaca, kecepatan angin, dan banyak lagi, yang dapat digunakan untuk aplikasi berbasis IoT.

Dengan menggunakan ESP32, data yang diperoleh dari API OpenWeatherMap akan ditampilkan secara langsung pada layar LCD 16x2 untuk memudahkan pengguna dalam memantau kondisi cuaca. Sistem ini tidak hanya menampilkan informasi cuaca secara statis, tetapi juga menyediakan fitur navigasi antar halaman dan scrolling teks, sehingga pengguna dapat memperoleh informasi cuaca secara lebih interaktif. Selain itu, sistem ini mampu memperbarui data cuaca setiap 60 detik sekali untuk memastikan informasi yang ditampilkan selalu terbaru. Dengan mengintegrasikan sensor dan API cuaca dalam satu sistem berbasis IoT, praktikum ini menunjukkan potensi besar teknologi IoT dalam mempermudah akses informasi secara real-time.

1.1 Latar Belakang

Perkembangan teknologi Internet of Things (IoT) telah membuka banyak kemungkinan baru dalam berbagai sektor, termasuk dalam pengelolaan data dan pemantauan lingkungan. IoT memungkinkan perangkat-perangkat yang terhubung ke internet untuk saling berkomunikasi, mengirim, dan menerima data secara otomatis. Salah satu aplikasi yang dapat memanfaatkan teknologi IoT adalah sistem monitoring cuaca. Sistem ini memungkinkan pengguna untuk memantau kondisi cuaca secara real-time yang dapat memberikan informasi penting untuk perencanaan kegiatan luar ruangan, pengelolaan sumber daya alam, dan mitigasi bencana alam seperti banjir atau kekeringan.

Di Indonesia, perubahan cuaca yang tidak menentu seringkali berdampak besar terhadap berbagai sektor, seperti pertanian, transportasi, dan pariwisata. Oleh karena itu, penting bagi masyarakat dan pemerintah untuk memiliki akses yang cepat dan akurat terhadap informasi cuaca terkini. Meskipun data cuaca sering tersedia melalui berbagai media, sering kali sulit bagi masyarakat di daerah-daerah terpencil untuk mengakses informasi tersebut secara real-time. Untuk itu, pengembangan sistem monitoring cuaca berbasis IoT yang sederhana, murah, dan mudah diakses menjadi solusi yang potensial. Sistem ini dapat membantu memantau kondisi cuaca dengan

menggunakan perangkat berbasis mikrokontroler, seperti ESP32, yang terhubung dengan API cuaca seperti OpenWeatherMap. OpenWeatherMap menyediakan data cuaca global yang dapat diakses melalui internet, dan memungkinkan pengguna untuk mendapatkan informasi tentang suhu, kelembaban, kecepatan angin, dan kondisi atmosfer lainnya.

Penerapan teknologi ini pada sistem monitoring cuaca berbasis IoT dapat memberikan kemudahan akses informasi cuaca secara real-time dengan perangkat yang sederhana dan dapat dipasang di berbagai lokasi. Praktikum ini dilakukan secara simulasi menggunakan platform Wokwi, sebuah web simulator yang memungkinkan pengguna untuk merancang dan menguji proyek berbasis mikrokontroler tanpa memerlukan perangkat keras asli. Dengan menggunakan Wokwi, praktikum ini dapat dilakukan secara virtual, memungkinkan pengujian dan verifikasi sistem tanpa keterbatasan perangkat keras fisik. Sistem ini diharapkan dapat meningkatkan kesadaran masyarakat tentang pentingnya pemantauan cuaca dan memberikan solusi mudah dalam pengambilan keputusan yang bergantung pada informasi cuaca yang akurat.

1.2 Tujuan Eksperimen

Tujuan dari eksperimen ini adalah untuk membangun dan menguji sistem monitoring cuaca berbasis Internet of Things (IoT) yang dapat menampilkan data cuaca secara real-time menggunakan platform ESP32

1. Mengembangkan Sistem Monitoring Cuaca IoT : Membuat sistem yang dapat mengakses dengan mudah dan menampilkan data cuaca terkini melalui koneksi internet menggunakan API OpenWeatherMap. Sistem ini menggunakan ESP32 untuk mengambil data cuaca, yang meliputi suhu, kelembaban, deskripsi cuaca, dan kecepatan angin.
2. Menggunakan LCD untuk Menampilkan Informasi Cuaca : Menampilkan data cuaca secara real-time pada layar LCD 16x2 dengan interface I2C, yang dapat diakses oleh pengguna.
3. Implementasi navigasi Antar halaman : Membuat sistem navigasi dengan menggunakan tombol yang memungkinkan pengguna untuk berpindah antara berbagai halaman informasi cuaca, termasuk suhu, deskripsi cuaca, kecepatan angin, dan kelembaban.
4. Sistem dapat memperbarui Data Secara Berkala : Mengimplementasikan mekanisme pembaruan data cuaca setiap 60 detik untuk memastikan informasi yang ditampilkan selalu terbaru.
5. Simulasi Proyek di Platform Wokwi dan PlatformIO : Melakukan simulasi dan pengujian sistem menggunakan platform Wokwi dan PlatformIO, yang memungkinkan eksperimen dilakukan secara virtual tanpa memerlukan perangkat keras fisik, sehingga memudahkan perancangan dan pengujian sistem.
6. Penugasan mata kuliah IoT : Praktikum ini juga bertujuan sebagai pemenuhan tugas mata kuliah IoT di semester e

Dengan tujuan-tujuan tersebut, eksperimen ini bertujuan untuk memberikan pemahaman tentang penerapan teknologi IoT dalam pemantauan cuaca serta mengoptimalkan penggunaan platform simulasi untuk merancang sistem berbasis mikrokontroler.

2. Methodology (Metodologi)

2.1 Tools & Materials (Alat dan Bahan)

Untuk mendukung praktikum ini, digunakan alat dan bahan sebagai berikut:

1. Platform simulasi Wokwi, digunakan untuk menyimulasikan sistem secara virtual tanpa perangkat keras asli.
2. Visual Studio Code dengan ekstensi PlatformIO, sebagai tempat pengembangan untuk menulis dan menguji kode program.
3. Mikrokontroler ESP32 (virtual pada Wokwi/PlatformIO), berfungsi sebagai unit pemrosesan utama dalam sistem.
4. LCD 16x2 dengan modul I2C (virtual), menampilkan data cuaca real-time seperti suhu, kelembaban, kondisi cuaca, dan kecepatan angin.

5. *Push Button Virtual*, digunakan untuk navigasi antar halaman informasi cuaca.
6. *API OpenWeatherMap*, Penyedia data cuaca yang diakses secara online menggunakan ESP32.
7. Kode program berbasis C/C++, Untuk mengatur pengambilan data cuaca, parsing data JSON, dan menampilkan informasi pada LCD.

2.2 Implementation Steps (Langkah Implementasi)

1. **Persiapan Simulasi:** Membuka platform Wokwi atau Visual Studio Code yang sudah ada template yang diberikan oleh dosen mata kuliah.
2. **Pembuatan Skema Rangkaian:** Berdasarkan template yang sudah ada, mahasiswa tinggal menyesuaikan kode lalu menambahkan button sebanyak dua buah, untuk tombol next dan tombol preview
3. **API key dari OpenWeatherMap :** Mendaftarkan diri di laman OpenWeatherMap dan mengambil Api key yang diberikan dari web.
4. **Pemrograman:** Menyesuaikan api key dari OpenWeatherMap, menyesuaikan kode untuk menambah fungsi tombol next dan tombol preview, menyesuaikan kode agar data-data api dapat tampil di LCD berdasarkan urutan halaman, dan menyesuaikan kode menggunakan bahasa C/C++.
5. **Pengujian Simulasi:** Menjalankan simulasi pada Wokwi dan/atau PlatformIO serta memastikan sistem bekerja sesuai dengan yang diharapkan.
6. **Analisis Hasil:** Mengevaluasi sistem dengan melihat apakah sistem dapat berjalan dengan benar dan sesuai dengan skenario lalu lintas yang telah dirancang.

3. Result dan Discussion (Hasil dan Pembahasan)

3.1 Experimental Result (Hasil Eksperimen)

Praktikum ini merupakan perancangan dan simulasi sistem monitoring cuaca berbasis ESP32 yang terkoneksi ke jaringan Wi-Fi dan mengambil data dari OpenWeatherMap API. Sistem dirancang untuk menampilkan informasi cuaca pada LCD 16x2 dengan menggunakan dua tombol push button. Berikut adalah hasil yang diperoleh dari pengujian sistem:

1. **Koneksi Wi-Fi :** ESP32 berhasil menghubungkan diri ke jaringan Wokwi-GUEST tanpa password. Status koneksi berhasil ditampilkan pada LCD dengan pesan "Connected!".
2. **Pengambilan Data Cuaca :** Sistem mampu melakukan pengambilan data real-time dari server OpenWeatherMap. Data yang diambil adalah

- Suhu udara (dalam satuan Celcius)
- Deskripsi kondisi cuaca (misalnya: "clear sky", "overcast clouds")
- Kelembaban udara (%)
- Kecepatan angin (m/s)
- Data diperbarui secara otomatis setiap 60 detik
- 3. **Tampilan Data pada LCD** menampilkan informasi ke dalam 4 halaman berbeda:
 - Halaman 0: Nama kota ("Malang")
 - Halaman 1: Suhu udara terkini
 - Halaman 2: Deskripsi kondisi cuaca (dengan fitur scrolling text bila panjang deskripsi lebih dari 16 karakter)
 - Halaman 3: Kecepatan angin dan kelembaban udara

Halaman

Baris 1

Baris 2

0

Location:

Malang

1

Temperature:

30.2 C (contoh)

2

Weather:

few clouds (scroll otomatis jika >16 char)

3

Wind: 2.3 m/s (contoh)

Humidity: 60%" (contoh)

4. Navigasi Tombol

Debouncing sebesar 200 ms diterapkan sehingga perpindahan halaman terasa responsif dan stabil. Penggunaan tombol berhasil diuji dengan hasil:

-

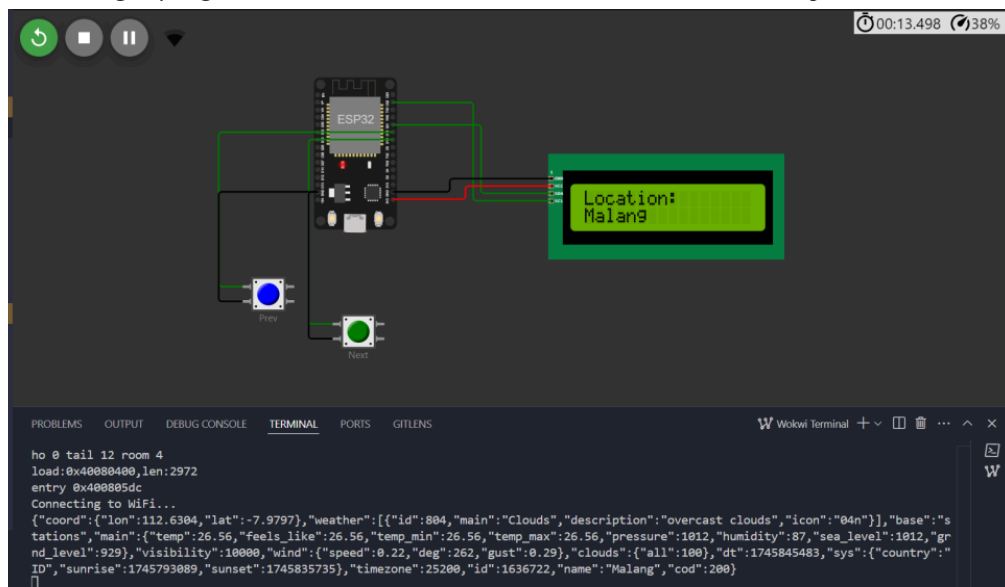
Tombol Next: Berpindah ke halaman berikutnya.

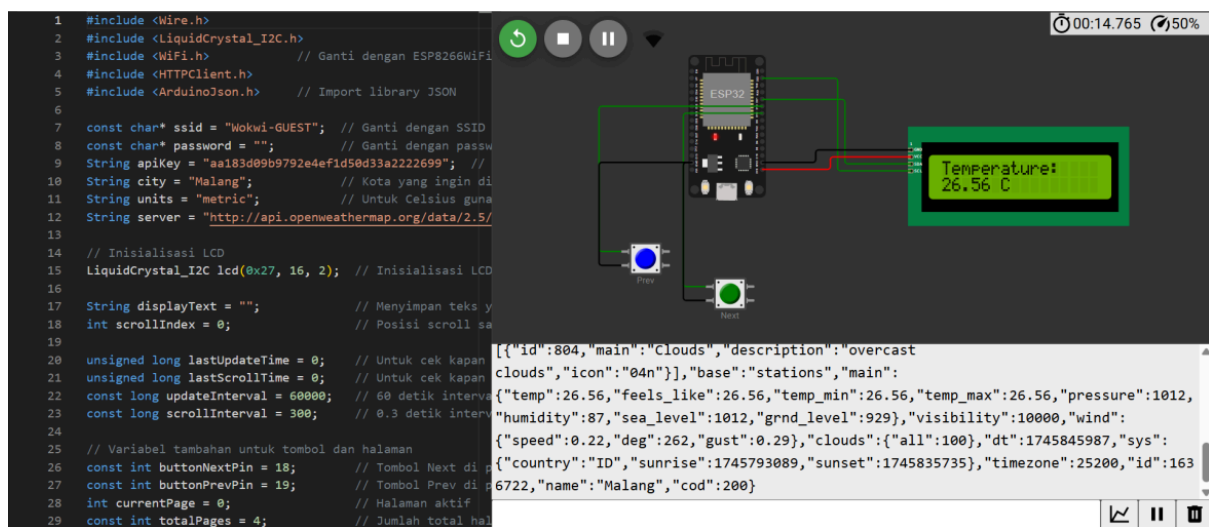
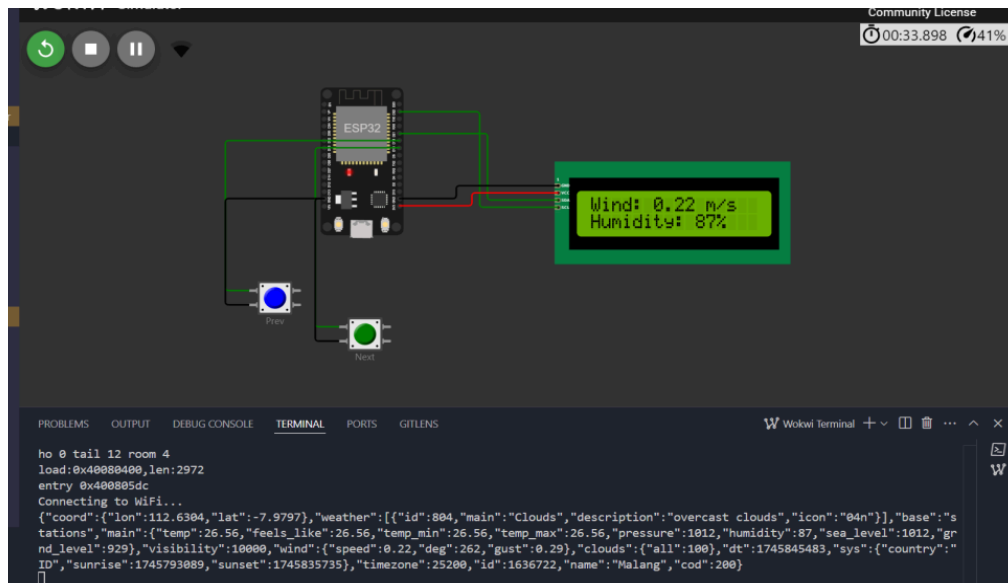
-

Tombol Prev: Berpindah ke halaman sebelumnya.

5. Simulasi pada Wokwi

Seluruh eksperimen dilakukan pada platform Wokwi dan VSC menggunakan simulasi ESP32, LCD I2C, dan dua buah push button. Hasil simulasi menunjukkan sistem bekerja sesuai dengan yang direncanakan. Berikut adalah screenshot hasil simulasi pada Wokwi:





4. Appendix (Lampiran)

4.1 Kode Program (Main.cpp)

```

#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h> // Ganti dengan ESP8266WiFi.h jika menggunakan ESP8266
#include <HTTPClient.h>
#include <ArduinoJson.h> // Library untuk parsing JSON
// Konfigurasi Wi-Fi dan API OpenWeatherMap
const char* ssid = "Wokwi-GUEST"; // Ganti dengan SSID Wi-Fi kamu
const char* password = ""; // Ganti dengan password Wi-Fi kamu
String apiKey = "aa183d09b9792e4ef1d50d33a2222699"; // API Key dari OpenWeatherMap
String city = "Malang"; // Kota yang ingin ditampilkan
String units = "metric"; // Untuk Celsius gunakan "metric", untuk Fahrenheit "imperial"
String server =
"http://api.openweathermap.org/data/2.5/weather?q=Malang&units=metric&appid=aa183d09b9792e4ef1d50d33a2222699";
// Inisialisasi LCD I2C

```

```

LiquidCrystal_I2C lcd(0x27, 16, 2); // Alamat I2C untuk LCD adalah 0x27, ukuran 16x2 karakter
String displayText = ""; // Teks yang akan ditampilkan dan di-scroll
int scrollIndex = 0; // Posisi indeks scroll saat ini
unsigned long lastUpdateTime = 0; // Waktu terakhir update data cuaca
unsigned long lastScrollTime = 0; // Waktu terakhir scroll teks
const long updateInterval = 60000; // Interval update data cuaca setiap 60 detik
const long scrollInterval = 300; // Interval scroll teks LCD setiap 300ms
// Variabel untuk tombol dan navigasi antar halaman
const int buttonNextPin = 18; // Pin tombol Next di D18
const int buttonPrevPin = 19; // Pin tombol Prev di D19
int currentPage = 0; // Halaman yang sedang aktif
const int totalPages = 4; // Jumlah halaman yang tersedia
unsigned long lastDebounceTime = 0; // Waktu debounce tombol
const long debounceDelay = 200; // Delay debounce untuk tombol
// Variabel untuk data cuaca
String temp = ""; // Menyimpan suhu
String desc = ""; // Menyimpan deskripsi cuaca
String humidity = ""; // Menyimpan kelembaban
String wind = ""; // Menyimpan kecepatan angin
// Fungsi deklarasi
void updateWeather();
void showPage(int page);
void scrollDisplay();
void setup() {
  Serial.begin(115200); // Mulai komunikasi serial untuk debug
  // Inisialisasi LCD
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Weather Info:");
  // Inisialisasi tombol
  pinMode(buttonNextPin, INPUT_PULLUP); // Tombol Next menggunakan mode INPUT_PULLUP
  pinMode(buttonPrevPin, INPUT_PULLUP); // Tombol Prev menggunakan mode INPUT_PULLUP
  // Inisialisasi koneksi Wi-Fi
  WiFi.begin(ssid, password);
  lcd.setCursor(0, 1);
  lcd.print("Connecting...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to Wi-Fi...");
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Connected!");
  delay(2000); // Tunggu 2 detik sebelum lanjut
  lcd.clear();
  // Ambil data cuaca pertama kali setelah Wi-Fi terkoneksi
  updateWeather();
  showPage(currentPage); // Tampilkan halaman pertama
}
void loop() {

```

```

unsigned long currentMillis = millis(); // Waktu saat ini untuk timer
// Update data cuaca setiap 60 detik
if (currentMillis - lastUpdateTime >= updateInterval) {
  updateWeather();
  lastUpdateTime = currentMillis; showPage(currentPage); // Refresh halaman setelah update data
}
// Scroll teks LCD setiap 300ms di halaman 2 (deskripsi cuaca)
if (currentPage == 2) {
  if (currentMillis - lastScrollTime >= scrollInterval) {
    scrollDisplay();
    lastScrollTime = currentMillis;
  }
}
// Baca tombol Next (pindah ke halaman berikutnya)
if (digitalRead(buttonNextPin) == LOW) {
  if (millis() - lastDebounceTime > debounceDelay) {
    currentPage++;
    if (currentPage >= totalPages) currentPage = 0; // Kembali ke halaman pertama jika sudah di
    halaman terakhir
    showPage(currentPage);
    lastDebounceTime = millis();
  }
}
// Baca tombol Prev (pindah ke halaman sebelumnya)
if (digitalRead(buttonPrevPin) == LOW) {
  if (millis() - lastDebounceTime > debounceDelay) {
    if (currentPage == 0) currentPage = totalPages - 1; // Kembali ke halaman terakhir jika di
    halaman pertama
    else currentPage--;
    showPage(currentPage);
    lastDebounceTime = millis();
  }
}
// Fungsi untuk mengambil data cuaca dari API
void updateWeather() {
  if (WiFi.status() == WL_CONNECTED) { // Periksa apakah Wi-Fi terhubung
    HTTPClient http;
    http.begin(server); // Tentukan URL untuk request
    int httpCode = http.GET(); // Lakukan request HTTP GET
    if (httpCode > 0) { // Jika request berhasil
      String payload = http.getString(); // Ambil data dari API
      Serial.println(payload); // Tampilkan data mentah (JSON) ke Serial Monitor
      // Parsing data JSON
      StaticJsonDocument<1024> doc;
      deserializeJson(doc, payload);
      // Ambil data dari JSON dan simpan ke variabel
      temp = String(doc["main"]["temp"].as<float>());
      desc = doc["weather"][0]["description"].as<String>();
      humidity = String(doc["main"]["humidity"].as<int>());
      wind = String(doc["wind"]["speed"].as<float>());
      // Gabungkan data untuk ditampilkan di layar LCD displayText = "| Temp: " + temp + " C | " + desc;
    }
  }
}

```



```

scrollIndex = 0; // Reset indeks scroll
} else {
Serial.println("Error on HTTP request");
}
http.end(); // Hapus resource HTTP setelah selesai
}
}
// Fungsi untuk menampilkan halaman sesuai dengan currentPage
void showPage(int page) {
lcd.clear();
switch (page) {
case 0:
lcd.setCursor(0, 0);
lcd.print("Location:");
lcd.setCursor(0, 1);
lcd.print(city);
break;
case 1:
lcd.setCursor(0, 0);
lcd.print("Temperature:");
lcd.setCursor(0, 1);
lcd.print(temp + " C");
break;
case 2:
lcd.setCursor(0, 0);
lcd.print("Weather:");
lcd.setCursor(0, 1);
if (desc.length() > 16) {
lcd.print(desc.substring(0, 16)); // Potong jika deskripsi lebih dari 16 karakter
} else {
lcd.print(desc);
}
break;
case 3:
lcd.setCursor(0, 0);
lcd.print("Wind: " + wind + " m/s");
lcd.setCursor(0, 1);
lcd.print("Humidity: " + humidity + "%");
break;
}
}
// Fungsi untuk scroll teks di halaman deskripsi cuaca
void scrollDisplay() {
if (displayText.length() > 0) {
lcd.setCursor(0, 1);
if (scrollIndex + 16 <= displayText.length()) {
lcd.print(displayText.substring(scrollIndex, scrollIndex + 16)); // Tampilkan potongan teks
} else {
String part1 = displayText.substring(scrollIndex);
String part2 = displayText.substring(0, 16 - part1.length());
lcd.print(part1 + part2); // Efek looping scroll
}
}
}

```

```

scrollIndex++;
if (scrollIndex >= displayText.length()) {
scrollIndex = 0; // Jika sudah habis, mulai dari awal lagi
}
}
}

```

4.2 Kode Program (diagram.json)

```

{
"version": 1,
"author": "Cagita Dian A'yunin",
"editor": "wokwi",
"parts": [
{ "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -50, "left": -100, "attrs": {} },
{ "type": "wokwi-lcd1602", "id": "lcd", "top": 50, "left": 200, "attrs": { "pins": "i2c" } },
{
"type": "wokwi-pushbutton",
"id": "btnNext",
"top": 255.8,
"left": -76.8,
"attrs": { "color": "green", "label": "Next" }
},
{
"type": "wokwi-pushbutton",
"id": "btnPrev",
"top": 207.8,
"left": -192,
"attrs": { "color": "blue", "label": "Prev" }
}
],
"connections": [
[ "esp:TX0", "$serialMonitor:RX", "", [] ],
[ "esp:RX0", "$serialMonitor:TX", "", [] ],
[ "lcd:SCL", "esp:D22", "green", [ "h-100", "v-40" ] ],
[ "lcd:SDA", "esp:D21", "green", [ "h-90", "v-20" ] ],
[ "lcd:VCC", "esp:3V3", "red", [ "h-110", "v60" ] ],
[ "lcd:GND", "esp:GND.1", "black", [ "h-130", "v40" ] ],
[ "btnNext:1.1", "esp:D18", "green", [ "h-30", "v-20" ] ],
[ "btnNext:2.1", "esp:GND.2", "black", [ "h-30", "v20" ] ],
[ "btnPrev:1.1", "esp:D19", "green", [ "h-30", "v-20" ] ],
[ "btnPrev:2.1", "esp:GND.2", "black", [ "h-30", "v20" ] ]
],
"dependencies": {}
}

```