

LAPORAN PROYEK UAS

PERSONAL WEATHER DASHBOARD

UAS Matakuliah: Proyek Integrasi Sistem



Disusun Oleh:

Seluruh Mahasiswa Jurusan Teknologi informasi Lokal: TI23B

UNIVERSITAS ISLAM NEGERI ANTASARI BANJARMASIN
PRODI TEKNOLOGI INFORMASI
FAKULTAS DAKWAH DAN ILMU KOMUNIKASI
TAHUN 2025

DAFTAR ISI

LAPORAN PROYEK UAS PERSONAL WEATHER DASHBOARD.....	1
DAFTAR ISI.....	2
BAB I PENDAHULUAN	4
1.1. Ringkasan Eksekutif	4
1.2 Latar Belakang	4
1.3 Rumusan Masalah.....	4
1.4 Tujuan	4
1.5 Manfaat	5
1.6. Ruang Lingkup.....	5
1.7 Anggota dan pembagian timnya	5
BAB II ANALISIS KEBUTUHAN.....	6
2.1Kebutuhan Fungsional	6
2.2 Kebutuhan Non-Fungsional.....	6
2.3 Pemodelan Sistem.....	6
BAB III LANDASAN TEORI.....	7
3.1 OpenWeatherMap API.....	7
3.2 Teknologi yang Digunakan.....	7
BAB IV PERANCANGAN SISTEM.....	9
4.1 Arsitektur Sistem	9
4.2 Desain Antarmuka	10
4.3 Rencana Fitur	11
BAB V IMPLEMENTASI.....	13
5.1 Struktur Folder	13
5.2 Ringkasan code	13
BAB VI PENGUJIAN SISTEM.....	14
6.1 Metodologi.....	14
6.2 Hasil Pengujian	14
BAB VII KESIMPULAN DAN SARAN.....	15

7.1 Kesimpulan	15
7.2 Rekomendasi.....	15
LAMPIRAN.....	16
DAFTAR PUSTAKA	19

BAB I

PENDAHULUAN

1.1 Ringkasan Eksekutif

Penelitian ini menghadirkan Personal Weather Dashboard, sebuah aplikasi web berbasiskan client-side yang memanfaatkan OpenWeatherMap API untuk menyajikan data cuaca secara real-time. Sistem dirancang untuk memenuhi kebutuhan pengguna dalam memperoleh informasi kondisi atmosfer (temperatur, kelembapan, kecepatan angin, dan status cuaca) dari berbagai lokasi, baik melalui pencarian nama kota maupun deteksi lokasi otomatis. Fitur tambahan seperti penyimpanan riwayat pencarian ke LocalStorage, pilihan tema (siang/gelap), dan visualisasi grafik suhu harian turut diimplementasikan guna meningkatkan kenyamanan dan daya tarik antarmuka. Hasil pengujian menunjukkan sistem berfungsi sesuai spesifikasi dan dapat dioperasikan pada berbagai perangkat dengan responsivitas optimal.

1.2 Latar Belakang

Informasi cuaca memegang peran krusial dalam perencanaan aktivitas sehari-hari, mulai dari pertanian, transportasi, hingga rutinitas personal. Kemajuan teknologi web memungkinkan penyajian data cuaca secara dinamis dan interaktif tanpa memerlukan backend kompleks. Berdasarkan hal tersebut, dikembangkan sebuah *dashboard* cuaca yang responsif dan mudah diakses melalui peramban.

1.3 Rumusan Masalah

1. Bagaimana merancang dan mengimplementasikan antarmuka web yang intuitif untuk menampilkan data cuaca real-time?
2. Bagaimana mengintegrasikan OpenWeatherMap API agar data yang disajikan akurat dan terkini?
3. Bagaimana menyimpan dan memulihkan riwayat pencarian pengguna secara lokal tanpa server?

1.4 Tujuan

1. Merancang antarmuka pengguna (UI) yang sederhana namun informatif.
2. Mengintegrasikan OpenWeatherMap API untuk memperoleh data cuaca global secara real-time.
3. Menerapkan mekanisme penyimpanan lokal dengan LocalStorage untuk riwayat pencarian.
4. Menyediakan opsi personalisasi tampilan melalui tema siang/gelap dan grafik interaktif.

1.5 Manfaat

1. Bagi pengguna
2. Bagi mahasiswa
3. Untuk pembelajaran sistem integrasi

1.6. Ruang Lingkup

Proyek ini membatasi ruang lingkup pada pengembangan sisi klien (frontend) menggunakan HTML5, CSS3, dan JavaScript (ES6+), tanpa melibatkan server-side processing atau basis data eksternal.

1.7 Anggota dan pembagian timnya

Role:

1. UI/UX Developer

- Muhammad Fahrul Bahri (koor)
- Husni Majedi
- Muhammad Lutfan
- Sabilia Rizqina Majid
- Siti Alayda Azzahro
- Najwa Dinda Aprilia

3. Feature Developer

- Muhammad Raihan Azmi (koor)
- Muhammad Naufal Jihadi
- Muhammad Riduwan
- Aisyah Safitri
- Ghefira Nur Fatima
- Syti Masidah

2. Backend Integration Specialist

- M. Delfian Tirta Nugraha (koor)
- Akhmad Hafidz Ardianto
- Muhammad Reyhan
- Muna Rahimatal Olya
- Siti Maghfiratun Warahmah
- Husna Norgina

4. Documentation & QA

- Husin Nafarin Ramadhani (koor)
- Muhammad Azhiem Fadhilah
- Febi Novia Putri
- Siti Mauliana Rahmah
- Nurmiyaty
- Mahdalina

BAB II

ANALISIS KEBUTUHAN

2.1 Kebutuhan Fungsional

1. Sistem harus menampilkan data cuaca berdasarkan input nama kota.
2. Sistem harus mendeteksi koordinat geografis pengguna melalui Geolocation API dan menampilkan data cuaca setempat.
3. Sistem harus menyimpan dan memulihkan riwayat pencarian kota menggunakan LocalStorage.
4. Sistem harus menyediakan notifikasi kondisi cuaca ekstrem.
5. Sistem harus mengekspor riwayat pencarian dalam format yang dapat dibaca (mis. CSV).

2.2 Kebutuhan Non-Fungsional

1. Antarmuka harus responsif pada berbagai ukuran layar (mobile, tablet, desktop).
2. Waktu muat halaman tidak melebihi 2 detik pada koneksi broadband.
3. Kode sumber terstruktur dan terdokumentasi dengan baik.

2.3 Pemodelan Sistem

1. Aktor: Pengguna (User)
2. Use Case Diagram: (Lampiran Gambar 1)
3. Arsitektur: Aplikasi single-page berbasis JavaScript yang berkomunikasi langsung ke REST API.

BAB III

LANDASAN TEORI

3.1 OpenWeatherMap API

OpenWeatherMap API adalah layanan berbasis web yang menyediakan data cuaca real-time dan prakiraan cuaca dari berbagai lokasi di seluruh dunia. API ini dapat digunakan secara gratis (dengan batasan tertentu) dan juga menyediakan paket berbayar untuk kebutuhan lanjutan.

Cara Kerja:

API ini bekerja dengan prinsip request-response, di mana pengguna mengirim permintaan HTTP dengan parameter tertentu (misalnya nama kota), dan server akan mengembalikan data cuaca dalam format JSON.

Contoh URL Endpoint:

```
https://api.openweathermap.org/data/2.5/weather?q=Banjarmasin&appid=API_KEY&units=metric
```

Penjelasan Parameter:

- q=Banjarmasin → nama kota
- appid=API_KEY → kunci API unik pengguna
- units=metric → satuan suhu (Celsius)

3.2 Teknologi yang Digunakan

1. HTML5

Digunakan untuk menyusun struktur dasar halaman web, termasuk elemen form, tombol, dan kontainer tampilan. HTML5 mendukung elemen semantik seperti <header>, <main>, dan <section> yang memudahkan pengembangan web yang lebih terstruktur dan mudah diakses.

2. CSS3

Digunakan untuk menata tampilan halaman, termasuk warna, layout, dan responsivitas. CSS3 mendukung media query, yang memungkinkan desain antarmuka menyesuaikan ukuran layar perangkat (mobile-first design).

Contoh media query:

```
@media (max-width: 768px) {  
    .weather-card {  
        flex-direction: column;  
    }  
}
```

3. JavaScript

Bahasa pemrograman yang digunakan untuk logika aplikasi, seperti mengambil data dari API menggunakan `fetch()`, memanipulasi DOM (tampilan dinamis), dan menyimpan data kota ke `localStorage`.

Contoh penggunaan `fetch`:

```
fetch('https://api.openweathermap.org/data/2.5/weather?q=Banjarmasin&appid=${API  
_KEY}&units=metric')  
.then(response => response.json())  
.then(data => {  
  console.log(data);  
});
```

4. LocalStorage

Fitur penyimpanan lokal pada browser yang memungkinkan data disimpan secara permanen di sisi pengguna tanpa database. Dalam proyek ini digunakan untuk menyimpan daftar kota favorit atau kota yang terakhir diakses.

Contoh:

```
localStorage.setItem("kotaFavorit", "Banjarmasin");  
let kota = localStorage.getItem("kotaFavorit");
```

5. Figma

Figma adalah alat desain UI/UX berbasis web yang digunakan untuk membuat wireframe dan prototype antarmuka aplikasi sebelum diimplementasikan dalam HTML dan CSS. Tim menggunakan Figma untuk merancang layout yang responsif dan estetis.

6. GitHub

GitHub digunakan sebagai version control system (VCS) berbasis Git. Semua anggota proyek menyimpan dan mengelola kode sumber secara kolaboratif, membuat *branch* fitur, dan melakukan *pull request* untuk review dan integrasi.

BAB IV

PERANCANGAN SISTEM

4.1 Arsitektur Sistem

Sistem *Personal Weather Dashboard* dirancang untuk memberikan informasi cuaca secara real-time menggunakan teknologi web berbasis client-side. Artinya, seluruh proses pengambilan dan penampilan data dilakukan di sisi pengguna (browser), tanpa memerlukan server backend tersendiri.

a. Komponen Arsitektur

1. Pengguna (Client / Browser):

Tempat semua proses berlangsung. Pengguna berinteraksi dengan antarmuka aplikasi untuk mencari informasi cuaca berdasarkan nama kota atau lokasi mereka saat ini.

2. OpenWeatherMapAPI:

Layanan pihak ketiga yang menyuplai data cuaca secara real-time berdasarkan permintaan HTTP.

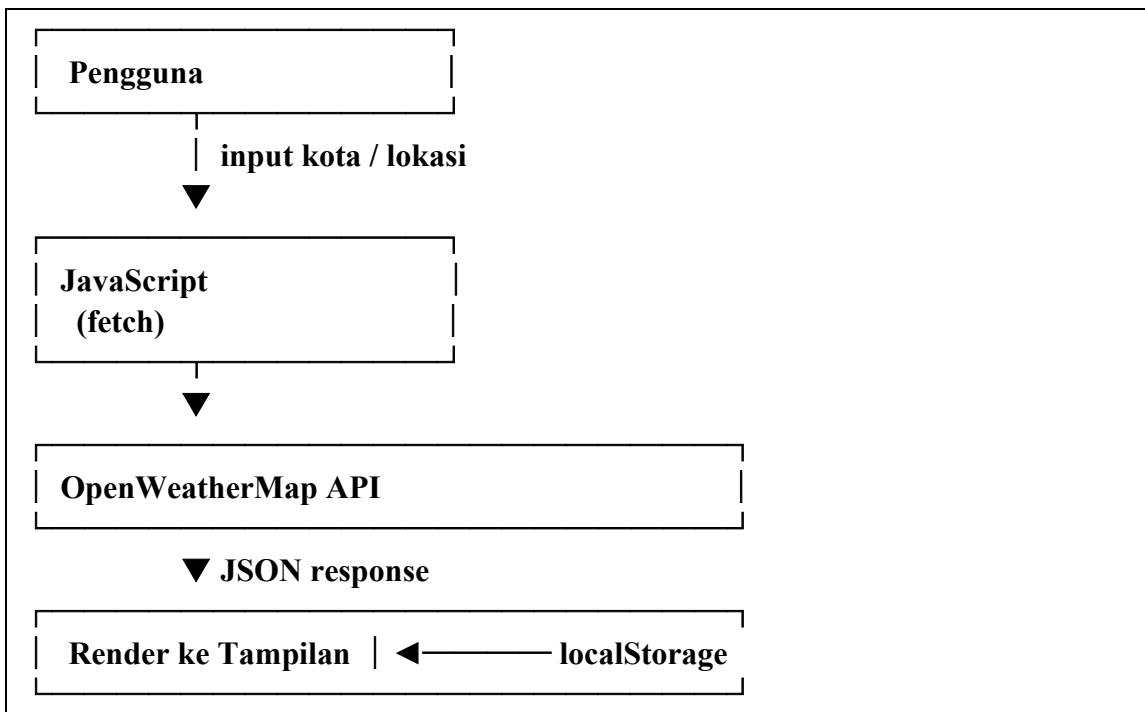
3. LocalStorage (Browser):

Fitur penyimpanan lokal di browser yang digunakan untuk menyimpan daftar kota yang telah dicari atau ditandai sebagai favorit.

b. Alur Proses Sistem

1. Pengguna membuka aplikasi.
2. Aplikasi mendeteksi lokasi pengguna (jika izin diberikan) atau menunggu input kota.
3. Aplikasi mengirim permintaan ke API cuaca.
4. API mengembalikan data cuaca dalam format JSON.
5. Aplikasi menampilkan data ke pengguna (DOM rendering).
6. Kota yang dicari disimpan di localStorage

c. Diagram Arsitektur Sederhana



- Pengguna memasukkan nama kota → fungsi `getWeatherByCity()` dipanggil → data cuaca ditampilkan → data disimpan ke LocalStorage.
- Pengguna memilih deteksi lokasi → fungsi `getCurrentLocationWeather()` dijalankan → data cuaca setempat ditampilkan.

4.2 Desain Antarmuka

Desain antarmuka dirancang agar intuitif, ringan, dan dapat diakses di berbagai perangkat (desktop, tablet, dan mobile).

a. Tools yang Digunakan

- **Figma:** Untuk membuat mockup dan wireframe desain halaman.
- **HTML5 + CSS3:** Untuk membuat struktur dan tampilan antarmuka.
- **Media Queries:** Agar tampilan responsif di berbagai ukuran layar.

b. Struktur Antarmuka

- **Header:** Judul aplikasi dan toggle dark/light mode (opsional).
- **Search Section:** Input text + tombol cari + tombol lokasi.
- **Display Section:** Kartu cuaca berisi data seperti suhu, status cuaca, kelembaban, angin.
- **Footer:** Informasi sumber data dan hak cipta.

c. Elemen Visual

Elemen UI	Fungsi
Search Bar	Untuk memasukkan nama kota dan memicu fetch data cuaca.
Tombol Lokasi	Untuk meminta akses geolocation dari pengguna.
Kartu Cuaca	Menampilkan hasil cuaca dalam bentuk visual per kota.
Ikon Cuaca	Simbol visual dari kondisi cuaca (berawan, hujan, cerah).
Riwayat Kota	Daftar kota yang sebelumnya dicari, disimpan di localStorage.

d. Responsivitas

Tampilan menyesuaikan secara otomatis tergantung ukuran layar. Contoh kode media query:

```
@media (max-width: 768px) {
  .weather-card {
    flex-direction: column;
    width: 100%;
  }
}
```

4.3 Rencana Fitur

Fitur-fitur aplikasi dibagi menjadi dua kategori: wajib (core features) dan tambahan (enhancement/optional).

A. Fitur Wajib

Fitur	Deskripsi
Pencarian Kota	Pengguna dapat memasukkan nama kota dan menampilkan informasi cuaca melalui fetch API.
Geolocation	Jika disetujui, aplikasi akan otomatis menampilkan cuaca lokasi pengguna saat ini.
Tampilan Multi Kota	Dashboard mendukung tampilan beberapa kota sekaligus (misalnya 3-5 kota terakhir dicari).

Fitur	Deskripsi
Responsif	Antarmuka aplikasi bekerja baik di semua ukuran layar.
Error Handling	Jika terjadi kesalahan seperti kota tidak ditemukan atau API error, akan ditampilkan pesan.
LocalStorage	Kota favorit atau riwayat pencarian disimpan di browser agar muncul kembali saat reload.

B. Fitur Tambahan

Fitur	Deskripsi
Auto-refresh	Data cuaca diperbarui otomatis setiap beberapa menit.
Forecast 5 Hari	Menampilkan prakiraan cuaca lima hari ke depan untuk tiap kota.
Dark/Light Mode	Pengguna dapat mengganti tema tampilan sesuai preferensi.
Export Data	Fitur untuk mengekspor data cuaca ke file .csv atau .json.

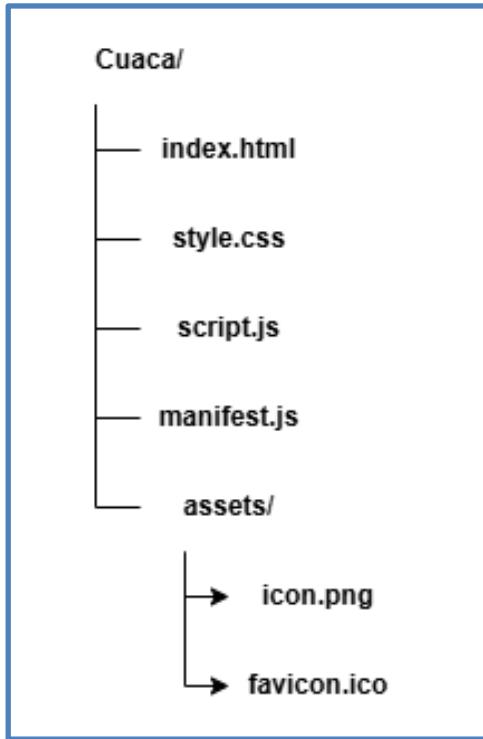
C. Detail Alur Interaksi Modul

Modul	Fungsi
Input Handler	Menerima input pengguna dan memicu permintaan data ke API.
Fetch API Module	Mengambil data cuaca dari OpenWeatherMap dan menangani response.
LocalStorage Handler	Menyimpan dan mengambil data kota dari penyimpanan lokal browser.
DOM Renderer	Mengubah data JSON menjadi elemen HTML yang bisa ditampilkan ke pengguna.
Error Manager	Menangani error teknis atau input tidak valid, dan memberi feedback yang jelas.

BAB V

IMPLEMENTASI

5.1 Struktur Folder



5.2 Ringkasan code

- index.html: Mengatur kerangka HTML, memuat CSS, Font Awesome, Chart.js, dan manifest.
- style.css: Menerapkan desain responsif, tema, dan animasi.
- script.js:
 - Inisialisasi API_KEY, BASE_URL.
 - Fungsi addCity(), getWeatherByCity(), getCurrentLocationWeather().
 - Penanganan error dan validasi input.
 - Penyimpanan serta pemulihan data dari LocalStorage.
 - Pengaturan dan pemeliharaan tema.

BAB VI

PENGUJIAN SISTEM

6.1 Metodologi

Pengujian dilakukan menggunakan pendekatan black-box untuk memvalidasi fungsi sesuai spesifikasi.

6.2 Hasil Pengujian

No.	Kasus Uji	Langkah	Hasil yang Diharapkan	Status
1	Pencarian kota valid	Masukkan “Banjarmasin” → tekan Enter	Muncul data cuaca Banjarmasin	Lulus <input checked="" type="checkbox"/>
2	Pencarian kota kosong	Biarkan input kosong → tekan Enter	Notifikasi “Nama kota kosong”	Lulus <input checked="" type="checkbox"/>
3	Deteksi lokasi	Klik tombol lokasi → ijinakan akses posisi	Muncul data cuaca lokasi Anda	Lulus <input checked="" type="checkbox"/>
4	Tema siang/gelap	Klik tombol tema	Tampilan berganti sesuai tema	Lulus <input checked="" type="checkbox"/>
5	Penyimpanan & pemulihan riwayat	Tambah beberapa kota → refresh halaman	Daftar kota tetap muncul	Lulus <input checked="" type="checkbox"/>

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

- Aplikasi berhasil memenuhi seluruh kebutuhan fungsional dan non-fungsional.
- Desain antarmuka user-friendly dan responsif.
- Integrasi API terjamin keandalannya dengan penanganan error yang baik.

7.2 Rekomendasi

- Mengembangkan modul backend ringan untuk menyimpan data pengguna secara terpusat.
- Menambahkan fitur prediksi cuaca multi-hari (forecast).
- Mengimplementasikan Progressive Web App (PWA) untuk akses offline.

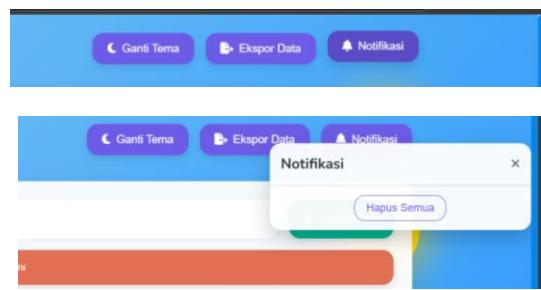
LAMPIRAN



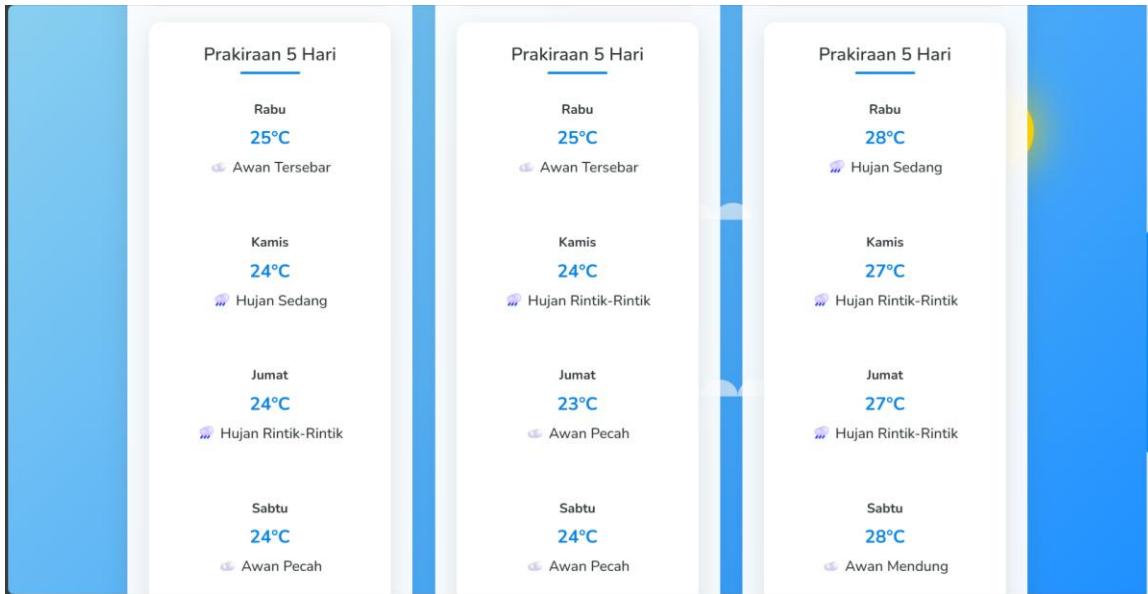
Gambar 1. Use Case Diagram



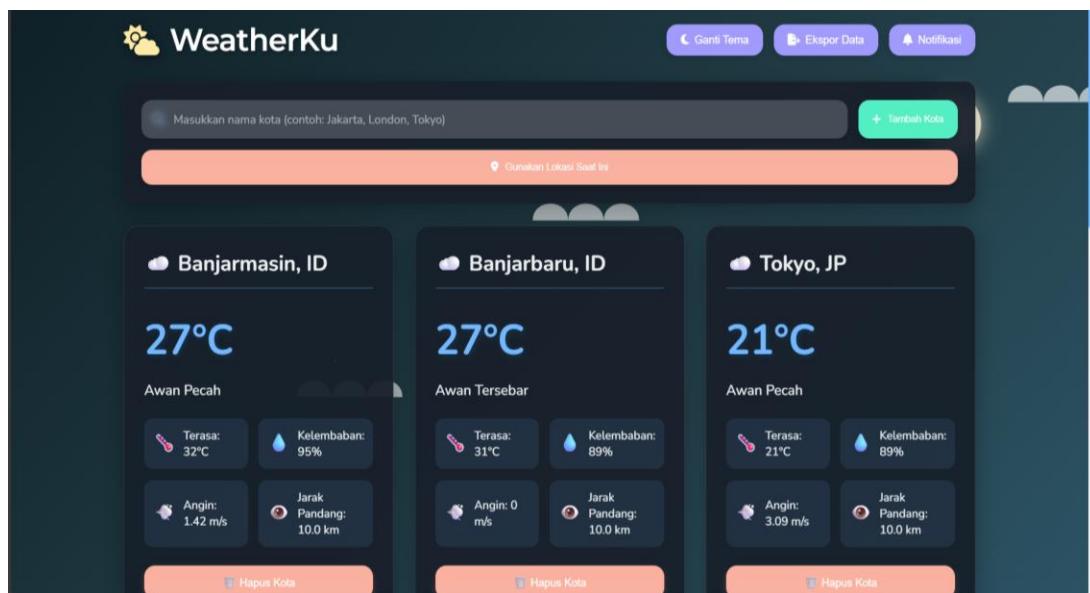
Gambar 2. Tampilan Halaman Utama

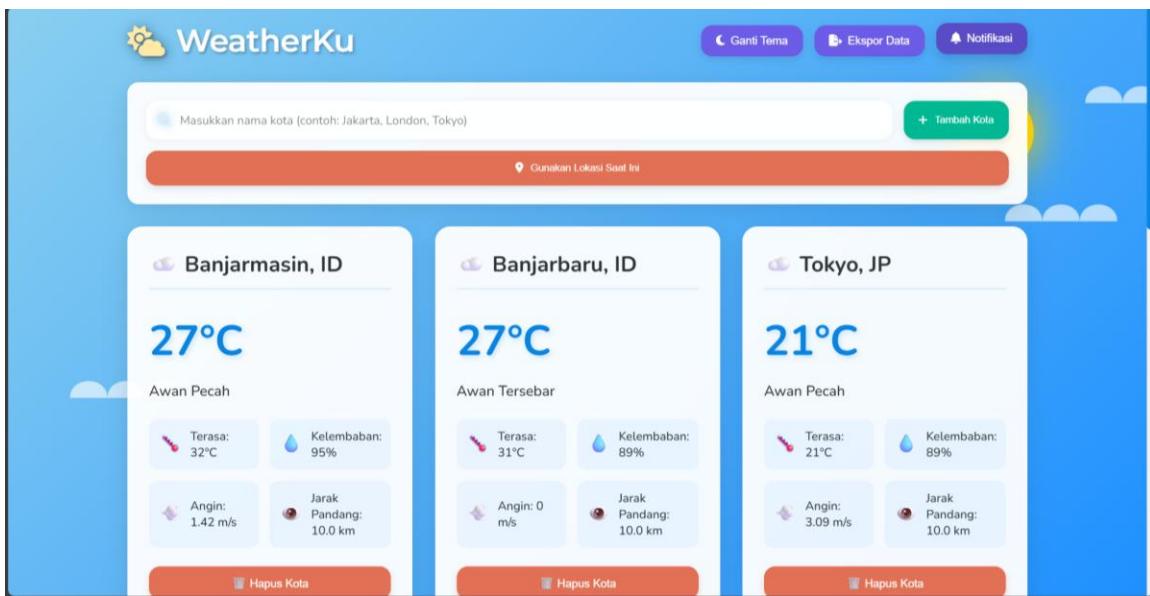


Gambar 3. Panel Notifikasi Cuaca



Gambar 4. Grafik Suhu Harian





Gambar 5. Mode Siang vs Gelap

DAFTAR PUSTAKA

OpenWeatherMap. (2025). OpenWeatherMap API Documentation.

Chart.js. (2025). Chart.js User Guide.

Mozilla Developer Network. (2025). MDN Web Docs: Web APIs.

W3C. (2018). HTML5 Specification.

W3C. (2018). CSS3 Specification.