E-ISSN : 2988-1986
https://ejournal.cahayailmubangsa.institut/index.php/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 10 No 6 Tahun 2025

PERANCANGAN DAN IMPLEMENTASI APLIKASI MOBILE CUACA "SKYNOW" MENGGUNAKAN API OPENWEATHERMAP

Andre Imanuel Porayou[1]*, Beverly Daniel Velix Rauan[2], Edward Langi[3], Roynaldo Christofel Johanes Mailangkay[4], Ade Yusupa[5], Victor Taringan[6]
[1,2,3,4,5,6]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Sam Ratulangi Kota Manado, Indonesia
E-mail: andreporayou026@student.unsrat.ac.id[1]*, beverlyrauan026@student.unsrat.ac.id[2], edwardlangi026@student.unsrat.ac.id[3], roynaldomailangkay026@student.unsrat.ac.id[4], ade@unsrat.ac.id[5], victortarigan@unsrat.ac.id[6]

ABSTRACT

The development of mobile technology has demanded the availability of fast and accurate information, one of which is weather information. Uncertain climate changes require society to have precise weather forecast data to support daily activities. This study aims to design and build an Android-based weather application named "Skynow" utilizing the OpenWeatherMap API as the main data source. Application development is carried out using the Kotlin programming language with the modern Jetpack Compose UI framework, as well as the implementation of the Model-View-ViewModel (MVVM) architecture to ensure code efficiency. Featured capabilities developed include dynamic interface visual adjustments based on local time zones (Local Timezone Awareness) and interactive temperature graphs. The software development method used is Prototyping with functionality testing using Black Box Testing. Test results show that the application is capable of displaying real-time weather data, hourly forecasts, and daily forecasts with high accuracy, as well as an interface that is responsive to user location time changes.

Keywords: Android, Weather Application, Jetpack Compose, OpenWeatherMap API, Retrofit.

ABSTRAK

Perkembangan teknologi seluler telah menuntut ketersediaan informasi yang cepat dan akurat, salah satunya adalah informasi cuaca. Perubahan iklim yang tidak pasti mengharuskan masyarakat memiliki data prakiraan cuaca yang akurat untuk mendukung aktivitas sehari-hari. Studi ini bertujuan untuk merancang dan membangun aplikasi cuaca berbasis Android bernama "Skynow" dengan memanfaatkan API OpenWeatherMap sebagai sumber data utama. Pengembangan aplikasi dilakukan menggunakan bahasa pemrograman Kotlin dengan kerangka kerja antarmuka pengguna modern Jetpack Compose, serta penerapan arsitektur Model-View-ViewModel (MVVM) untuk memastikan efisiensi kode. Fitur-fitur yang dikembangkan meliputi penyesuaian visual antarmuka secara

Article History
Received: Desember 2025
Reviewed:Desember 2025
Published: Desember 2025

Plagiarism Checker No 235
Prefix DOI : 10.8734/Kohesi.v1i2.365
Copyright : Author
Publish by : Kohesi

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License

dinamis berdasarkan zona waktu lokal (Local Timezone Awareness) dan grafik suhu interaktif. Metode pengembangan perangkat lunak yang digunakan adalah Prototyping dengan pengujian fungsional menggunakan Black Box Testing. Hasil pengujian menunjukkan bahwa aplikasi mampu menampilkan data cuaca real-time, ramalan jam, dan ramalan harian dengan akurasi tinggi, serta antarmuka yang responsif terhadap perubahan waktu lokasi pengguna.

Kata kunci: Android, Aplikasi Cuaca, Jetpack Compose, API OpenWeatherMap, Retrofit.

1. PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini, ketergantungan masyarakat terhadap informasi berbasis mobile semakin meningkat. Smartphone bukan lagi sekadar alat komunikasi, melainkan asisten pribadi yang membantu pengambilan keputusan sehari-hari, termasuk dalam memantau kondisi cuaca. Cuaca merupakan faktor lingkungan yang sangat dinamis dan memiliki dampak signifikan terhadap berbagai sektor kehidupan, mulai dari transportasi, pertanian, hingga perencanaan kegiatan luar ruangan. Ketidakpastian cuaca yang sering terjadi di wilayah tropis mengharuskan adanya akses informasi yang real-time dan akurat.

Sejumlah penelitian terdahulu telah mengembangkan aplikasi cuaca berbasis Android maupun web. Penelitian [1] merancang aplikasi prediksi cuaca berbasis Android menggunakan XML klasik, namun antarmuka yang dihasilkan masih bersifat statis dan kurang adaptif terhadap pengalaman pengguna. Sementara itu, referensi [2] mengembangkan aplikasi tracking cuaca berbasis website, namun aksesibilitas melalui browser dinilai kurang praktis dibandingkan aplikasi native mobile yang dapat memanfaatkan sensor perangkat secara langsung. Penelitian [3] juga menekankan pentingnya penggunaan format data JSON dalam pertukaran data cuaca global, namun implementasi UI yang kaku pada studi terdahulu seringkali masih membingungkan pengguna dalam membaca visualisasi data.

Berdasarkan analisis terhadap penelitian-penelitian tersebut, terdapat celah (gap analysis) berupa kurangnya integrasi antara akurasi data meteorologi dengan pengalaman pengguna (User Experience) yang imersif. Kebanyakan aplikasi yang ada belum sepenuhnya memvalidasi sumber data yang digunakan. Oleh karena itu, penelitian ini menggunakan API OpenWeatherMap yang telah teruji akurasinya dalam studi komparasi meteorologi global [4]. Selain itu, transisi teknologi pengembangan Android dari XML ke Jetpack Compose menawarkan

peluang untuk menciptakan antarmuka yang lebih responsif dan performa yang lebih ringan, namun belum banyak dieksplorasi dalam konteks aplikasi utilitas cuaca di literatur terkini.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang aplikasi cuaca yang mampu menampilkan data *real-time* dan prakiraan cuaca dengan akurasi tinggi menggunakan *OpenWeatherMap API*?
2. Bagaimana mengimplementasikan logika penanganan zona waktu (*Timezone Handling*) agar visualisasi aplikasi menyesuaikan dengan waktu lokal kota tujuan, bukan waktu perangkat pengguna?
3. Bagaimana penerapan *Jetpack Compose* dalam membangun antarmuka grafik suhu yang dinamis dan interaktif?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Membangun aplikasi mobile "Skynow" yang dapat memberikan informasi cuaca *real-time*, prakiraan per jam, dan harian.
2. Menerapkan algoritma penyesuaian latar belakang dinamis (*Dynamic Background*) berdasarkan waktu lokal kota yang dicari untuk meningkatkan pengalaman pengguna.
3. Menguji fungsionalitas aplikasi dalam mengambil dan memparsing data JSON kompleks menjadi visualisasi grafis yang mudah dipahami.

2. METODE PENELITIAN

2.1 Desain Penelitian

Penelitian ini menggunakan metode pengembangan sistem *Rapid Application Development* (RAD) dengan pendekatan *Prototyping*. Metode ini dipilih karena memungkinkan pengembang untuk membuat purwarupa fitur secara cepat, melakukan evaluasi, dan memperbaiki kode secara iteratif, yang sangat sesuai dengan karakteristik kerangka kerja *Jetpack Compose* [5]. Tahapan penelitian meliputi analisis kebutuhan, perancangan sistem, implementasi kode, dan pengujian.

2.2 Metode Pengumpulan Data

Data cuaca yang digunakan dalam aplikasi ini adalah data sekunder yang diperoleh melalui *Application Programming Interface* (API). Sumber data utama adalah OpenWeatherMap. Pemilihan penyedia data ini didasarkan pada studi terdahulu yang menunjukkan konsistensi data OpenWeatherMap dibandingkan stasiun cuaca konvensional [4].

- **Endpoint API:** Menggunakan layanan Forecast 5 days / 3 hour.
- **Format Data:** Data diterima dalam format JSON (*JavaScript Object Notation*) yang berisi parameter suhu, kelembaban, kecepatan angin, dan kode ikon cuaca.

2.3 Analisis dan Perancangan Sistem

Arsitektur aplikasi dirancang menggunakan pola **MVVM (Model-View-ViewModel)** untuk memisahkan logika bisnis dari antarmuka pengguna.

1. **Model:** Merepresentasikan struktur data JSON dari API (didefinisikan dalam WeatherModel.kt dan ForecastResponse).
2. **View:** Tampilan antarmuka yang dibangun dengan *Jetpack Compose* (WeatherScreen.kt).
3. **ViewModel:** Penghubung yang melakukan pemanggilan jaringan secara *asynchronous* menggunakan *Coroutines*.

Alat pengembangan (*Tools*) yang digunakan meliputi:

- **IDE:** Android Studio Narwal 3 Feature Drop.
- **Bahasa:** Kotlin 1.9.
- **Library Networking:** Retrofit 2.9 untuk manajemen HTTP Request.
- **Image Loading:** Coil untuk memuat ikon cuaca secara asinkron.

2.4 Perancangan Antarmuka (UI/UX)

Perancangan antarmuka difokuskan pada konsep Edge-to-Edge dan minimalisme. Pendekatan desain ini mengacu pada strategi optimalisasi UI/UX modern yang bertujuan mempermudah interaksi pengguna dengan elemen visual aplikasi [6]. Aplikasi dirancang memiliki satu halaman utama yang memuat:

- **Header:** Informasi kota, suhu saat ini, dan kondisi awan.
- **Hourly Section:** Grafik suhu menggunakan Canvas custom.
- **Daily Section:** List prakiraan cuaca 5 hari ke depan.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

Pengembangan aplikasi "Skynow" berhasil diimplementasikan sesuai dengan rancangan. Aplikasi ini tidak menggunakan XML untuk tata letak, melainkan sepenuhnya menggunakan fungsi *Composable* dalam Kotlin.

A. Tampilan Halaman Utama

Halaman utama menampilkan informasi cuaca terkini. Pengguna dapat melihat suhu aktual yang diambil dari parameter main.temp pada respons API. Latar belakang aplikasi diimplementasikan menggunakan Brush.verticalGradient yang warnanya berubah secara programatik.



Gambar 1. Tampilan Halaman Utama

B. Fitur Pencarian dan Networking

Fitur pencarian kota diimplementasikan menggunakan OutlinedTextField. Saat pengguna menekan tombol *search*, aplikasi memicu fungsi RetrofitClient untuk melakukan *request* ke server OpenWeatherMap. Berikut adalah implementasi antarmuka API pada kode program:

```
// Potongan kode WeatherApi.kt
@GET("data/2.5/forecast")
suspend fun getForecast(
    @Query("q") city: String,
    @Query("appid") apiKey: String,
    @Query("units") units: String = "metric"
): ForecastResponse
```

Penggunaan suspend fun memastikan bahwa proses pengambilan data berjalan di *background thread*, sehingga tidak menyebabkan aplikasi macet (*freeze*) saat menunggu respon server.

3.2 Pembahasan dan Analisis Teknis

A. Penanganan Zona Waktu (Timezone Handling)

Salah satu tantangan teknis terbesar dan kebaruan (*novelty*) dalam proyek ini adalah menampilkan waktu yang relevan dengan lokasi kota, bukan lokasi pengguna. API OpenWeatherMap memberikan parameter `timezone` berupa offset detik dari UTC.

Analisis kode pada modul Helper.kt menunjukkan logika konversi sebagai berikut:

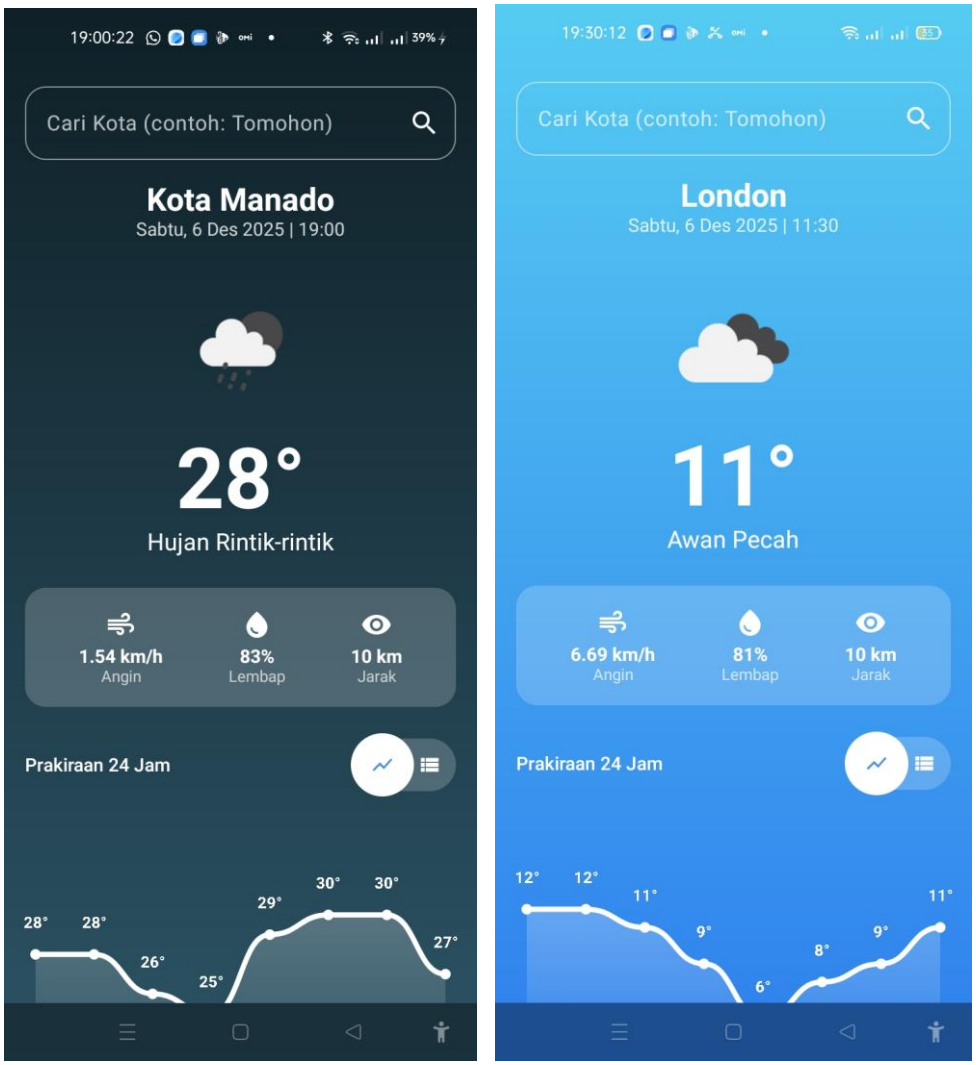
1. Waktu UTC dari sistem diambil.
2. Ditambahkan dengan offset detik dari kota tujuan.
3. Hasilnya diformat menjadi jam lokal.

```
// Logika pada Helper.kt untuk konversi waktu
fun formatTime(dateTime: String, offsetSeconds: Long): String {
    val inputFormat = SimpleDateFormat("yyyy-MM-dd HH:mm:ss")
    inputFormat.timeZone = TimeZone.getTimeZone("UTC")
    val date = inputFormat.parse(dateTime)

    val outputFormat = SimpleDateFormat("HH:mm")
    outputFormat.timeZone = SimpleTimeZone((offsetSeconds *
1000).toInt(), "LocalTime")
```

```
    return outputFormat.format(date)
}
```

Implementasi ini berhasil memecahkan masalah perbedaan waktu. Misalnya, saat pengguna di Indonesia (WIB) mencari cuaca "London", aplikasi dengan tepat menampilkan jam lokal London dan menyesuaikan latar belakang menjadi terang jika di London sedang siang hari, meskipun di Indonesia sedang malam hari.



Gambar 2. Implementasi Perbedaan Waktu Pada Kota yang Berbeda

B. Visualisasi Grafik Suhu Custom

Berbeda dengan aplikasi yang menggunakan *library* grafik pihak ketiga, Skynow mengimplementasikan grafik suhu menggunakan Canvas bawaan Compose (TemperatureGraph.kt). Kurva suhu digambar menggunakan *Bezier curve* (path.cubicTo) untuk

menghasilkan garis yang halus dan estetik. Posisi titik Y dihitung secara dinamis berdasarkan rumus:

$$Y = Height - \left(\frac{Suhu - MinSuhu}{MaxSuhu - MinSuhu}\right) \times Height \qquad (1)$$

Posisi vertikal setiap titik pada grafik ditentukan secara proporsional terhadap tinggi area gambar (canvas). Perhitungan koordinat Y dilakukan menggunakan Persamaan (1), di mana variabel Height merepresentasikan tinggi total komponen grafik dalam satuan piksel



Gambar 4. Tampilan Prakiraan Suhu untuk 24 Jam Kedepan

3.3 Pengujian Sistem (Black Box Testing)

Pengujian sistem dilakukan menggunakan metode Black Box Testing untuk memvalidasi fungsionalitas antarmuka serta integritas data yang ditampilkan [7]. Selain menguji responsivitas tombol dan navigasi, pengujian ini juga memverifikasi akurasi data dengan

membandingkan output suhu pada aplikasi 'Skynow' terhadap data sumber pada website resmi OpenWeatherMap secara real-time untuk memastikan tidak terjadi kesalahan parsing data.

No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1.	Input nama kota valid (misal: "Manado")	Menampilkan suhu dan lokasi Manado	Sesuai	Berhasil
2.	Input nama kota tidak valid	Menampilkan pesan error	Sesuai	Berhasil
3.	Scroll daftar prakiraan per jam	List dapat digeser secara horizontal	Sesuai	Berhasil
4.	Klik item harian (Expand)	Menampilkan detail grafik harian	Sesuai	Berhasil
5.	Validasi Data Suhu Aktual	Suhu di aplikasi sama dengan data di website OpenWeatherMap.org pada jam yang sama	Sesuai (App: 28°C, Web: 28°C)	Berhasil
6.	Validasi Kesesuaian Ikon Cuaca	Ikon kondisi cuaca (Cerah/Hujan) sesuai dengan deskripsi teks dari API	Sesuai (Ikon: Awan, Teks: Berawan)	Berhasil

Tabel 1. Tabel Pengujian Sistem

4. KESIMPULAN

Penelitian ini telah berhasil merancang dan mengimplementasikan aplikasi cuaca "Skynow" yang mampu menyajikan informasi meteorologi secara *real-time* dan akurat. Integrasi antara teknologi pengembangan antarmuka modern *Jetpack Compose* dengan data dinamis dari *OpenWeatherMap API* terbukti efektif dalam menghasilkan aplikasi yang tidak hanya fungsional tetapi juga responsif. Penerapan arsitektur MVVM memastikan kode program terstruktur dengan baik, memisahkan logika bisnis dari lapisan tampilan, sehingga mempermudah pemeliharaan dan pengembangan lebih lanjut di masa mendatang.

Salah satu pencapaian signifikan dalam penelitian ini adalah keberhasilan implementasi logika *Timezone Handling* yang memberikan nilai tambah pada pengalaman pengguna (*User Experience*). Aplikasi mampu beradaptasi secara cerdas terhadap konteks waktu lokasi yang dicari, mengubah nuansa visual antarmuka dari terang ke gelap sesuai kondisi lokal kota tujuan, bukan sekadar mengikuti waktu perangkat pengguna. Selain itu, visualisasi grafik suhu menggunakan *Canvas* kustom menunjukkan bahwa pengembangan komponen visual kompleks dapat dilakukan secara mandiri tanpa bergantung pada pustaka eksternal yang memberatkan ukuran aplikasi, sekaligus memberikan kontrol penuh terhadap estetika desain yang dihasilkan.

Meskipun tujuan utama penelitian telah tercapai, terdapat ruang untuk pengembangan lebih lanjut guna meningkatkan kegunaan aplikasi. Penelitian selanjutnya disarankan untuk mengintegrasikan fitur geolokasi (GPS) agar aplikasi dapat mendeteksi lokasi pengguna secara otomatis tanpa input manual. Penambahan sistem notifikasi dini terkait peringatan cuaca ekstrem juga direkomendasikan untuk menjadikan "Skynow" sebagai alat mitigasi risiko yang lebih proaktif dan bermanfaat bagi aktivitas harian penggunanya.

DAFTAR PUSTAKA

[1] H. A. Tambunan dan D. Saputra, “Rancang Bangun Aplikasi Prediksi Cuaca Berbasis Android,” *Jurnal Bisantara Informatika*, vol. 6, no. 2, pp. 1–7, Des. 2022. [Online]. Available: https://bisantara.amikparbinanusantara.ac.id/index.php/bisantara/article/view/78

[2] D. Sopyandi, A. T. Sati, A. Prabowo, D. M. P. Setya, dan F. A. Nugroho, “Bangun Aplikasi Tracking Cuaca (Weather App) Menggunakan Public API Berbasis Website,” *Jurnal Ilmiah Wahana Pendidikan*, vol. 10, no. 20, pp. 209–215, Okt. 2024. doi: https://doi.org/10.5281/zenodo.14274718

[3] G. N. Sofian, W. K. Raharja, dan Jalinas, “Perancangan Aplikasi Informasi Cuaca Berbasis Android Menggunakan API dan JSON,” *Jurnal Ilmiah KOMPUTASI*, vol. 21, no. 1, pp. 115–122, 2022. doi: https://doi.org/10.32409/jikstik.21.1.2908

[4] A. Musah *et al.*, "An evaluation of the OpenWeatherMap API versus INMET using weather data from two Brazilian cities: Recife and Campina Grande," *Data*, vol. 7, no. 8, Art. no. 106, Aug. 2022. doi: https://doi.org/10.3390/data7080106

[5] R. S. Pressman, Software Engineering: A Practitioner's Approach, 9th ed. New York: McGraw-Hill Education, 2020.

[6] I. Paendong, J. Lelemboto, A. Yusupa, dan V. Tarigan, "Optimalisasi UI/UX dalam Aplikasi Mobile: Strategi Meningkatkan Pengalaman dan Keterlibatan Pengguna," *Variable Research Journal*, vol. 2, no. 2, pp. 615–624, 2025. Available: https://variablejournal.my.id/index.php/VRJ/article/view/213

[7] F. I. H. Jihaadi, Hozairi, dan Hoiriyah, “Analisis Penerapan Metode Waterfall dan Pengujian Sistem Aplikasi Persewaan Peralatan Pesta Berbasis Mobile Menggunakan Black Box Testing,” *JOUTICA (Journal of Informatic Engineering)*, vol. 9, no. 2, pp. 111–124, 2024. doi: https://doi.org/10.30736/informatika.v9i2.1273