



Sistem Notifikasi Suspect Data AWS Center dengan Penyaringan Tanggung Jawab Stasiun Berbasis Push Notification

Rio Bintang Samudra Silalahi*, Kanton Lumban Toruan, Iqbal

Prodi Sarjana Terapan Instrumentasi-MKG, Sekolah Tinggi Meteorologi Klimatologi dan Geofisika, Tangerang, Indonesia

Email: ^{1,*}riosamudera12@gmail.com, ²kanton60@gmail.com, ³iqbal@bmkg.go.id

Email Penulis Korespondensi: riosamudera12@gmail.com

Abstrak—Pemantauan kualitas data peralatan stasiun cuaca otomatis (*Automatic Weather Station/AWS*) pada sistem AWS Center masih mengandalkan pemeriksaan *dashboard* web secara manual, sehingga waktu teknisi menerima informasi *suspect data* tidak terukur dan berpotensi terlambat. Penelitian ini merancang dan membangun aplikasi *mobile* AWSCenter Notif yang menyaring informasi *suspect data* sesuai stasiun tanggung jawab pengguna dan menyampaikannya melalui *push notification*, kemudian mengukur kesesuaian dan ketepatan waktu penyampaiannya. Sistem dikembangkan dengan metode *Rapid Application Development* menggunakan React Native, Cloud Functions, Cloud Firestore, dan Expo Push Service, dengan sinkronisasi data setiap 5 menit dan pemeriksaan notifikasi setiap 10 menit. Pengujian dilakukan pada tiga tahap, yaitu pengujian fungsional *black box* terhadap 11 skenario, pengujian pengalaman pengguna dengan *User Experience Questionnaire* terhadap 12 responden teknisi, serta pengujian kesesuaian dan latensi terhadap 62 unit analisis yang dicatat dari dua sisi secara independen di Stasiun Meteorologi Tanjung Harapan. Seluruh skenario fungsional berhasil dan keenam skala *User Experience Questionnaire* mencapai kategori *Excellent*. Seluruh informasi *flag* tersampaikan secara lengkap dan akurat sesuai kondisi pada sumber data, dengan latensi sistem yang secara konsisten berada di bawah batas atas teoretis yang dirancang. Sebagian besar waktu tunggu berasal dari sisi server sumber di luar kendali sistem, sehingga mempersingkat interval penjadwalan tidak akan mempercepat penyampaian informasi. Kontribusi penelitian ini adalah penyaringan notifikasi berdasarkan pemetaan alat terhadap stasiun tanggung jawab teknisi, penerapannya sebagai lapisan notifikasi aktif di atas sistem *quality control* yang sudah beroperasi, serta kerangka pengukuran empat titik waktu yang menjadikan waktu penerimaan informasi terukur.

Kata Kunci: AWS Center; Firebase; Latensi Diseminasi; *Push Notification*; React Native; *Suspect Data*

Abstract—Monitoring the data quality of automatic weather station (AWS) instruments in the AWS Center system still relies on manual inspection of a web dashboard, leaving the time at which technicians receive suspect data information unmeasured and potentially delayed. This study designs and builds the AWSCenter Notif mobile application, which filters suspect data according to each user's assigned station and delivers it through push notifications, and then measures the conformity and timeliness of that delivery. The system was developed using the Rapid Application Development method with React Native, Cloud Functions, Cloud Firestore, and Expo Push Service, with data synchronisation every five minutes and notification checks every ten minutes. Testing was conducted in three stages, namely black box functional testing of 11 scenarios, user experience testing using the User Experience Questionnaire with 12 technician respondents, and a conformity and latency test on 62 units of analysis recorded independently from two sides at Tanjung Harapan Meteorological Station. All functional scenarios passed and all six User Experience Questionnaire scales reached the Excellent category. All flag information was delivered completely and accurately relative to the source, with system latency consistently remaining below the designed theoretical upper bound. Most of the delay originated from the source server side beyond the system's control, so shortening the scheduling interval would not accelerate information delivery. The contributions of this study are a notification filtering mechanism based on the mapping of instruments to the station each technician is responsible for, its implementation as an active notification layer placed on top of an operational quality control system without altering that system, and a four-timestamp measurement framework that makes the time of information receipt measurable.

Keywords: AWS Center; Dissemination Latency; Firebase; Push Notification; React Native; Suspect Data

1. PENDAHULUAN

Badan Meteorologi, Klimatologi, dan Geofisika menyelenggarakan pengamatan cuaca dan iklim melalui jaringan peralatan stasiun cuaca otomatis yang tersebar di hampir seluruh wilayah Indonesia. Peralatan tersebut beroperasi secara mandiri untuk melakukan pengukuran, perekaman, dan pengiriman data observasi secara *real-time* tanpa memerlukan kehadiran pengamat di lokasi pengamatan. Data yang dihasilkan menjadi dasar penyusunan informasi cuaca, iklim, dan kebencanaan, sehingga kualitas serta kontinuitas data menjadi faktor penentu keandalan layanan informasi yang diberikan. Kualitas data observasi meteorologi umumnya dinilai dari akurasi, kelengkapan, konsistensi, dan kontinuitas pencatatan (Hulu, A. A., & Nasution, M. I. P., 2025) dan pengelolaan data yang tertata serta pemeliharaan peralatan yang tepat menjadi fondasi bagi keandalan layanan informasi yang dihasilkan (Badan Meteorologi Klimatologi dan Geofisika, 2025a)

Untuk mendukung hal tersebut, Direktorat Data dan Komputasi mengembangkan AWS Center, yaitu platform berbasis web yang memantau kualitas data dari berbagai jenis peralatan stasiun cuaca otomatis secara terpusat. Salah satu fungsi utamanya adalah mendeteksi *suspect data*, yaitu data observasi yang dicurigai memiliki masalah kualitas atau reliabilitas. Sistem menerapkan *quality control* berbasis *flagging* yang memberikan nilai *flag* pada setiap parameter sensor melalui tiga jenis pemeriksaan, yaitu *range check* untuk nilai di luar rentang wajar, *step check* untuk perubahan nilai yang terlalu ekstrem dalam waktu singkat, dan *consistency check* untuk ketidaksesuaian hubungan antarparameter meteorologi (Badan Meteorologi, Klimatologi, dan Geofisika, 2025b). Karena gangguan instrumen di lapangan kerap lebih dahulu terlihat sebagai anomali nilai pengamatan, deteksi dini *suspect data* menjadi indikator awal yang penting untuk mengidentifikasi potensi kerusakan sebelum terjadi kehilangan rekaman data observasi. Penerapan sistem pengendalian



mutu digital semacam ini terbukti meningkatkan keandalan produksi data karena pengumpulan dan evaluasi berlangsung otomatis dan berkelanjutan (Hussamadin dkk., 2023).

Meskipun infrastruktur pemantauannya telah tersedia, alur penyampaian informasi kepada teknisi di lapangan masih bersifat pasif. Teknisi harus mengakses laman web AWS Center secara manual dan berkala untuk mengetahui adanya *suspect data* pada peralatan yang menjadi tanggung jawabnya. Ketergantungan pada pemeriksaan manual ini menimbulkan dua persoalan sekaligus. Persoalan pertama, informasi berpotensi diterima terlambat sehingga gangguan instrumen tidak segera ditangani dan rekaman data observasi yang berkualitas berpotensi hilang. Persoalan kedua yang lebih mendasar, tidak ada kepastian mengenai berapa lama informasi tersebut sampai kepada teknisi. Selang waktu antara munculnya indikasi gangguan pada sistem pusat dan diterimanya informasi oleh pihak yang berwenang menanganinya tidak pernah terukur, sehingga alur penyampaian informasi yang berjalan saat ini sulit dijadikan dasar evaluasi kinerja pemeliharaan. Kondisi ini menegaskan perlunya mekanisme penyampaian informasi yang bersifat aktif dan otomatis, alih-alih pasif menunggu pemeriksaan manual, sekaligus terukur waktu penyampaiannya. Padahal ketepatan waktu merupakan salah satu penentu utama kualitas diseminasi informasi digital, di samping akurasi dan relevansi isi yang disampaikan (Siahaan dkk., 2021).

Sejumlah penelitian terdahulu telah menunjukkan efektivitas *push notification* sebagai media penyampaian informasi operasional. Penelitian Wahyudin dkk (2024) menerapkan Firebase Cloud Messaging pada aplikasi manajemen persuratan multiplatform dan menunjukkan bahwa notifikasi *real-time* mempersingkat waktu tanggap pengguna terhadap dokumen yang masuk. mengembangkan aplikasi *mobile* bank sampah berbasis React Native yang terintegrasi dengan Rest API, dan menyimpulkan bahwa kombinasi keduanya mampu menyajikan data secara dinamis dan responsif pada perangkat bergerak. Fernando dkk. (2020) memanfaatkan *push notification* untuk menyampaikan kondisi pintu air kepada petugas lapangan sehingga petugas tidak perlu memeriksa kondisi peralatan secara manual. Voutsas dkk. (2024) menyoroti persoalan kelelahan notifikasi pada sistem pemantauan berbasis *cloud* dan menyaring *alert* menggunakan pendekatan pembelajaran mesin agar hanya peringatan yang relevan yang sampai kepada operator. Secara umum, penerapan *push notification* terbukti meningkatkan keterlibatan sekaligus mempercepat respons pengguna terhadap informasi yang dikirimkan sistem (Wu, 2022).

Meskipun demikian, keempat penelitian tersebut masih menyisakan dua keterbatasan apabila ditinjau dari kebutuhan penelitian ini. Keterbatasan pertama berkaitan dengan relevansi informasi bagi penerima. Wahyudin dkk. (2024) dan Fernando dkk. (2020) mengirimkan notifikasi yang sama kepada seluruh pengguna tanpa membedakan objek yang menjadi tanggung jawab masing-masing, sedangkan Zani dkk. (2024) belum menerapkan notifikasi sama sekali sehingga penyajian data tetap bergantung pada pengguna membuka aplikasi. Voutsas dkk. (2024) memang menyaring notifikasi, namun penyaringannya didasarkan pada karakteristik *alert* melalui model pembelajaran mesin, bukan pada pemetaan objek terhadap penanggung jawab operasionalnya. Keterbatasan kedua berkaitan dengan kedudukan mekanisme notifikasi terhadap sistem sumber. Pada keempat penelitian tersebut, notifikasi dirancang menyatu dengan sistem sumbernya, baik karena aplikasi beserta basis datanya dibangun sendiri maupun karena penyaringan dilakukan di dalam platform pemantauan yang sama, sehingga mekanisme notifikasi menjadi bagian dari sistem yang dirancang sejak awal. Belum ditemukan penelitian yang menempatkan lapisan notifikasi aktif di atas sistem *quality control* yang telah beroperasi tanpa mengubah arsitektur sistem sumber tersebut. Kedua keterbatasan itulah yang menjadi kesenjangan yang diisi oleh penelitian ini, yaitu penyaringan notifikasi berdasarkan pemetaan alat terhadap stasiun tanggung jawab teknisi, yang diterapkan sebagai lapisan *serverless* di atas AWS Center yang sedang berjalan. Sebagai konsekuensi dari perubahan mekanisme tersebut, penelitian ini juga merumuskan kerangka pengukuran empat titik waktu agar penyampaian informasi dapat dinyatakan dalam rentang waktu yang terukur, bukan sekadar dinilai lebih cepat.

Berdasarkan uraian tersebut, penelitian ini bertujuan merancang dan membangun aplikasi *mobile* AWSCenter Notif serta mengevaluasi kesesuaian dan ketepatan waktu penyampaian informasi *suspect data* melalui aplikasi tersebut. Urgensi penelitian terletak pada kebutuhan menjadikan penyampaian informasi *suspect data* sebagai proses yang terukur, bukan sekadar tersedia.

Kontribusi penelitian ini ada tiga. Kontribusi pertama adalah tersedianya mekanisme penyaringan notifikasi berbasis pemetaan alat terhadap stasiun tanggung jawab teknisi, sehingga setiap pengguna hanya menerima informasi *suspect data* pada peralatan di wilayah kerjanya. Kontribusi kedua adalah rancangan lapisan notifikasi *serverless* yang mengubah diseminasi *suspect data* AWS Center dari pasif menjadi aktif tanpa mengubah arsitektur sistem sumbernya, sehingga pola serupa dapat diterapkan pada sistem *quality control* lain yang telah beroperasi. Kontribusi ketiga adalah tersusunnya kerangka pengukuran empat titik waktu yang memisahkan selang waktu pada sisi sumber dari selang waktu pada sisi sistem, sehingga waktu penerimaan informasi dapat dinyatakan sebagai rentang yang terukur dan dijadikan dasar penyusunan standar waktu tanggap pemeliharaan.

2. METODOLOGI PENELITIAN

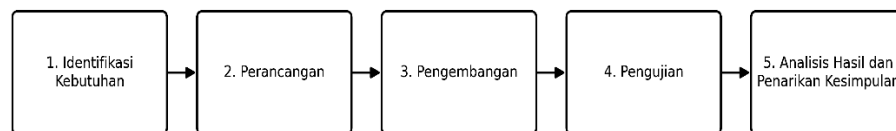
2.1 Objek Penelitian

Objek penelitian ini adalah sistem AWS Center yang dikelola Direktorat Data dan Komputasi, khususnya keluaran data *suspect data* yang dihasilkan dari proses *quality control* berbasis *flagging* terhadap parameter sensor peralatan stasiun cuaca otomatis. Informasi *suspect data* tersebut menjadi masukan utama yang diproses dan disampaikan oleh aplikasi *mobile* AWSCenter Notif yang dibangun dalam penelitian ini. Aplikasi dikembangkan menggunakan React Native

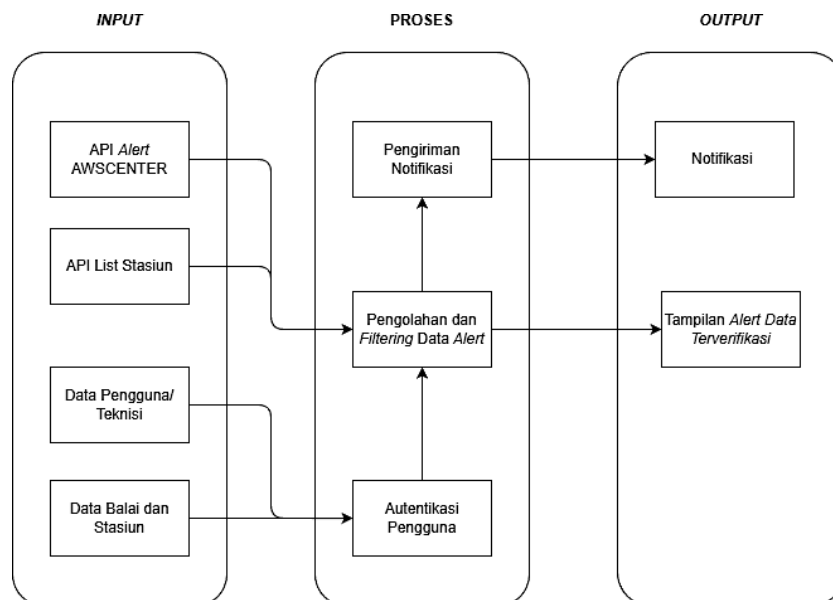
sebagai kerangka kerja sisi klien, serta Firebase yang mencakup Cloud Functions, Cloud Firestore, dan Cloud Scheduler sebagai infrastruktur sisi *backend* yang menangani pengambilan data, pencocokan tanggung jawab stasiun, dan penjadwalan proses. Penyampaian informasi kepada pengguna dilakukan melalui *push notification* dengan Expo Push Service sebagai penyedia layanan pengiriman pesan ke perangkat. Variabel utama yang dievaluasi dalam penelitian ini adalah latensi diseminasi, yaitu selang waktu antara tersedianya informasi *suspect data* pada sumber hingga diterimanya notifikasi oleh pengguna, beserta kesesuaian informasi yang disampaikan terhadap kondisi pada sumber data. Pengujian kesesuaian dan latensi dilakukan pada peralatan yang menjadi tanggung jawab Stasiun Meteorologi Tanjung Harapan, dengan pencatatan dilakukan secara independen dari dua sisi.

2.2 Tahapan Penelitian

Penelitian ini dilaksanakan melalui lima tahapan yang berurutan. Tahap pertama adalah identifikasi kebutuhan, yang dilakukan melalui studi pustaka serta diskusi dengan pengelola AWS Center pada Direktorat Data dan Komputasi untuk menentukan fungsi utama aplikasi sekaligus memperoleh akses layanan API. Tahap kedua adalah perancangan, meliputi penyusunan arsitektur sistem, diagram alir pemrosesan data dan pengiriman notifikasi, struktur basis data, serta rancangan antarmuka. Pemodelan perancangan menggunakan Unified Modelling Language yang berfungsi sebagai media komunikasi rancangan sistem berorientasi objek (Narulita dkk., 2024). Tahap ketiga adalah pengembangan, yaitu penerjemahan rancangan ke dalam kode program menggunakan Visual Studio Code sebagai *code editor* (Seo & Rogge, 2023). Tahap keempat adalah pengujian, yang terdiri atas pengujian fungsional, pengujian pengalaman pengguna, serta pengujian kesesuaian dan ketepatan waktu. Tahap kelima adalah analisis hasil pengujian dan penarikan kesimpulan. Tahap kedua hingga keempat dijalankan mengikuti metode *Rapid Application Development*, yang dipilih karena kebutuhan aplikasi bersifat iteratif dan memerlukan umpan balik berkelanjutan dari calon pengguna, sebagaimana penerapannya pada pengembangan aplikasi pemantauan berbasis *mobile* yang menuntut siklus pembangunan singkat dengan penyesuaian fitur yang fleksibel (Tan dkk., 2023). Alur kelima tahapan tersebut ditunjukkan pada Gambar 1. Diagram blok penelitian yang menggambarkan masukan, proses, dan keluaran sistem ditunjukkan pada Gambar 2.



Gambar 1. Diagram tahapan penelitian



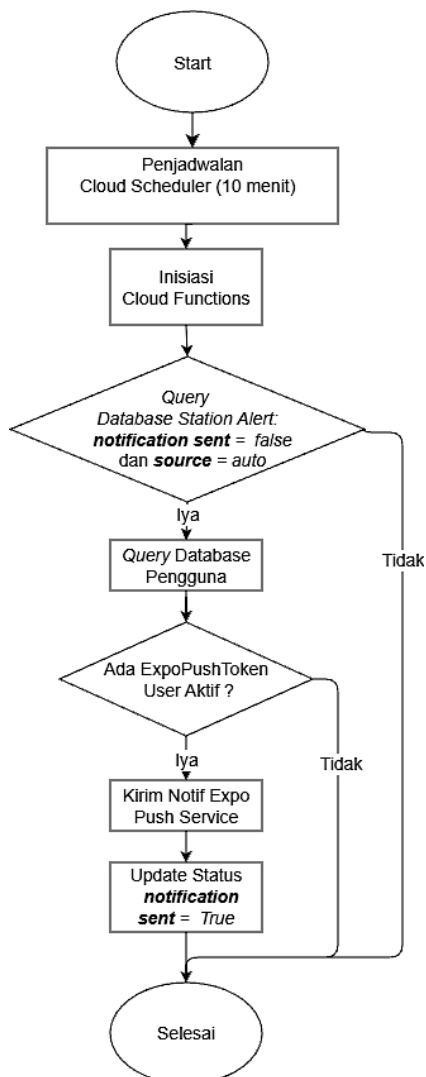
Gambar 2. Diagram blok masukan, proses, dan keluaran sistem

2.3 Arsitektur dan Mekanisme Sistem

Sistem terdiri atas tiga lapisan. Lapisan sumber data adalah layanan API AWS Center, yang menyediakan API *alert* berisi informasi *flag* pada setiap parameter sensor beserta identitas alat, dan API daftar stasiun yang memetakan identitas alat terhadap identitas stasiun. Lapisan *backend* dijalankan pada Firebase dengan memanfaatkan Cloud Functions sebagai pengolah data, Cloud Firestore sebagai basis data, dan Cloud Scheduler sebagai pemicu terjadwal. Pendekatan *serverless* dipilih karena logika *backend* dapat dieksekusi sebagai respons terhadap peristiwa tertentu tanpa perlu mengelola infrastruktur server secara langsung (Rajan, 2020), sekaligus memberikan efisiensi biaya dan skalabilitas yang lebih baik pada aplikasi *mobile* (Birajdar & Banubakode, 2025). Lapisan klien adalah aplikasi Android yang dibangun menggunakan bahasa TypeScript dan *framework* React Native, yang memungkinkan pembangunan aplikasi lintas platform dengan

tampilan dan perilaku mendekati aplikasi *native* (Meta Platforms, 2025). Autentikasi pengguna ditangani Firebase Authentication dengan metode Email/Password, yang berperan sebagai gerbang pembatas akses sistem (Toyib dkk., 2025).

Aplikasi *mobile* tidak mengakses API AWS Center secara langsung. Seluruh pengambilan dan pengolahan data dilakukan pada sisi *backend*, sehingga proses pemantauan tetap berjalan meskipun aplikasi pada perangkat pengguna sedang tidak aktif. Pemrosesan data dipicu Cloud Scheduler setiap 5 menit, dan dapat pula dipicu secara manual melalui tombol *reload* pada aplikasi. Pemilihan arsitektur berbasis penjadwalan (bukan protokol *event-driven* seperti WebSocket atau MQTT) mempertimbangkan karakteristik pembaruan data pada sumber. API *alert* AWS Center memperbarui data secara berkala setiap sekitar 10 menit, bukan secara *streaming* kontinu, sehingga penerapan protokol dengan koneksi persisten berlatensi sangat rendah tidak akan memberikan manfaat sepadan dengan kompleksitas infrastruktur tambahan yang diperlukan, seperti pengelolaan *broker* pesan atau server dengan koneksi yang harus dipertahankan secara terus-menerus. Pendekatan *serverless* dipilih karena tidak memerlukan infrastruktur khusus yang harus dikelola dan dipelihara sepanjang waktu, sekaligus tetap mampu menghasilkan latensi sistem yang berada di bawah interval pembaruan sumber data. Pada setiap eksekusi, sistem mengambil data dari kedua API, menyaring parameter dengan nilai *flag* lebih besar dari nol, lalu mencocokkan identitas alat dengan pemetaan alat terhadap stasiun dan dengan identitas stasiun yang menjadi tanggung jawab setiap pengguna terdaftar. Hasil pencocokan disimpan pada *collection station_alerts* beserta waktu pengambilan datanya. Pemeriksaan pengiriman notifikasi dijalankan oleh Cloud Scheduler terpisah setiap 10 menit dan hanya memproses dokumen yang belum pernah dikirimkan sebagai notifikasi serta berasal dari sinkronisasi otomatis, sehingga pembaruan manual oleh pengguna tidak memicu notifikasi ganda. Penerima ditentukan dari kesesuaian stasiun tanggung jawab dan preferensi notifikasi pengguna, dengan Expo Push Token sebagai alamat tujuan perangkat (Expo, 2025). Alur pengiriman notifikasi tersebut ditunjukkan pada Gambar 3.



Gambar 3. Diagram alir pengiriman notifikasi

Kedua interval penjadwalan tersebut ditetapkan menyesuaikan pola pembaruan data pada API *alert* AWS Center yang berlangsung sekitar setiap 10 menit. Berdasarkan rancangan sinkronisasi 5 menit dan pemeriksaan notifikasi 10 menit, nilai harapan latensi sistem secara teoretis adalah 7,5 menit dengan batas atas 15 menit.

Kegagalan sinkronisasi ditangani dengan memanfaatkan sifat setiap siklus yang berjalan independen. Jika pengambilan data dari API AWS Center gagal, misalnya karena jaringan bermasalah atau server sumber mengembalikan status non-2xx, siklus sinkronisasi saat itu berhenti dan galatnya otomatis tercatat pada Cloud Logging. Sistem tidak melakukan pengulangan otomatis untuk siklus yang gagal, tetapi hal ini tidak berisiko karena data baru hanya ditulis ke Cloud Firestore apabila berbeda dari data pada siklus sebelumnya. Dengan begitu, satu siklus yang gagal tidak menyebabkan data hilang atau ganda, melainkan hanya menunda deteksi informasi *flag* terbaru sampai siklus berikutnya berhasil dijalankan lima menit kemudian. Pada pengiriman notifikasi, sistem juga menangani token perangkat yang sudah tidak valid: jika Expo Push Service melaporkan status *DeviceNotRegistered*, token tersebut otomatis dihapus dan notifikasi ke pengguna itu dinonaktifkan agar tidak terus dicoba kirim ke perangkat yang sudah tidak terdaftar. Mekanisme ini belum mencakup pengulangan otomatis maupun pemisahan kegagalan antar stasiun dalam satu siklus notifikasi, sehingga hal ini menjadi salah satu arah pengembangan lanjutan.

2.4 Rancangan Pengujian

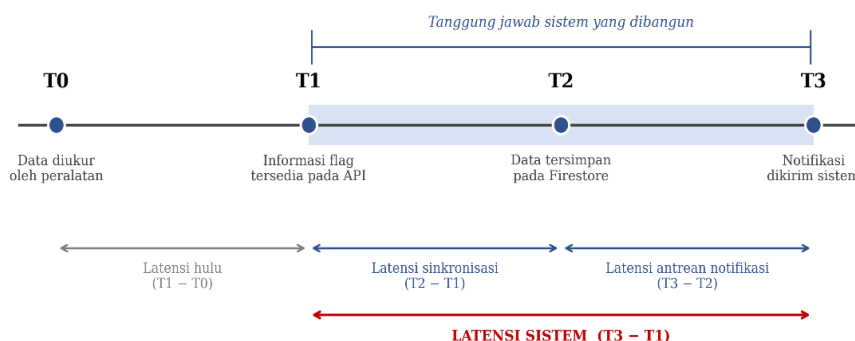
Evaluasi dilakukan melalui tiga tahap yang saling melengkapi. Pengujian fungsional menggunakan metode *black box* terhadap 11 skenario yang dikelompokkan ke dalam manajemen autentikasi, sinkronisasi dan penyaringan data, notifikasi, serta manajemen profil. Pengujian pengalaman pengguna menggunakan *User Experience Questionnaire* yang terdiri atas enam skala, yaitu Daya Tarik, Kejelasan, Efisiensi, Keandalan, Stimulasi, dan Kebaruan, dengan setiap item dinilai pada skala semantik diferensial bernilai 1 sampai 7 (Kushendriawan dkk., 2021). Aplikasi didistribusikan dalam bentuk berkas APK beserta panduan penggunaan kepada 12 responden yang merupakan teknisi dengan keterkaitan terhadap pemantauan peralatan stasiun cuaca otomatis, dan hasil kuesioner diolah menggunakan perangkat analisis resmi *User Experience Questionnaire* untuk memperoleh nilai rata-rata, varians, serta kategori tolok ukur setiap skala.

Pengujian ketiga disusun karena dua pengujian sebelumnya belum mengukur sejauh mana informasi *flag* pada sumber benar-benar tersampaikan kepada pengguna dan berapa lama waktu yang dibutuhkan. Pengujian dilakukan dengan mencatat kejadian yang sama dari dua sisi secara independen lalu membandingkannya. Sisi pertama adalah informasi *flag* pada API *alert* AWS Center, dicatat oleh program perekam yang dijalankan terpisah dari sistem dengan interval pengambilan data setiap 1 menit. Sisi kedua adalah informasi *flag* yang telah diproses sistem dan tersimpan pada Cloud Firestore, beserta waktu pengiriman notifikasinya. Cakupan pengujian dibatasi pada tujuh alat yang berada di bawah stasiun koordinator akun pengguna uji, yaitu Stasiun Meteorologi Tanjung Harapan dengan identitas stasiun 96525. Empat titik waktu yang diamati disajikan pada Tabel 1.

Tabel 1. Titik waktu pengamatan penyampaian informasi *flag*

Kode	Titik Waktu	Sumber Data
T0	Data diukur oleh peralatan	Atribut tanggal pada data API <i>alert</i>
T1	Informasi <i>flag</i> tersedia pada API <i>alert</i> AWS Center	Program perekam, interval 1 menit
T2	Data tersimpan pada Cloud Firestore	Atribut <i>fetchedAt</i> ISO, tercatat otomatis
T3	Sistem mengirimkan notifikasi	Atribut <i>notificationSentAt</i> , tercatat otomatis

Dari keempat titik tersebut dihitung tiga parameter latensi, yaitu latensi hulu yang merupakan selisih T1 terhadap T0 dan menggambarkan jeda pada sisi server AWS Center, latensi sinkronisasi yang merupakan selisih T2 terhadap T1, serta latensi antrean notifikasi yang merupakan selisih T3 terhadap T2. Penjumlahan dua parameter terakhir disebut latensi sistem, yaitu selang waktu yang menjadi tanggung jawab sistem yang dibangun dan menjadi parameter utama yang dilaporkan. Pemisahan ini penting agar kinerja sistem tidak dinilai berdasarkan jeda yang berada di luar kendalinya. Pengukuran diakhiri pada T3 karena tahap distribusi melalui Expo Push Service dijalankan oleh layanan pihak ketiga. Hubungan keempat titik waktu tersebut digambarkan pada Gambar 3.



Gambar 4. Garis waktu penyampaian informasi *flag*

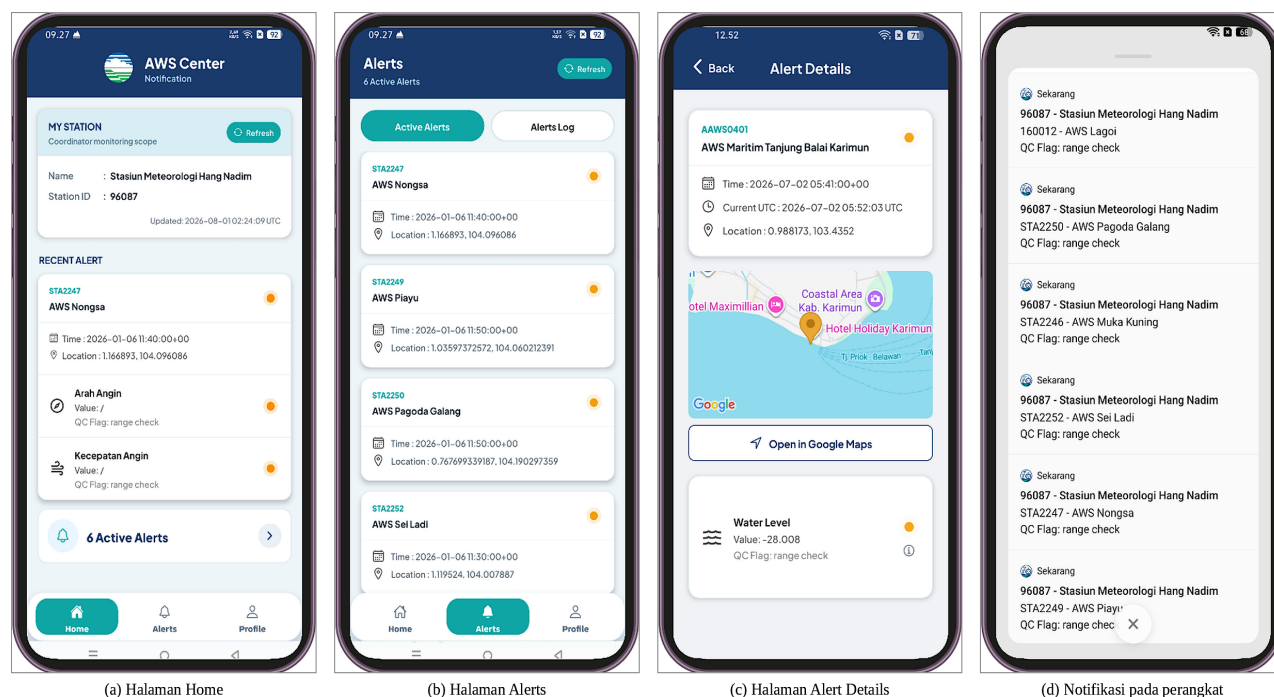
Unit analisis pada pengujian ini adalah kombinasi identitas alat, parameter sensor, dan nilai *flag*. Setiap unit diklasifikasikan sebagai sesuai apabila terdapat pada kedua sisi dengan jenis *flag* yang sama, tidak tersampaikan apabila hanya terdapat pada API, tidak berpadanan apabila hanya terdapat pada aplikasi, atau jenis *flag* berbeda apabila

tersampaikan namun jenis flagnya tidak sama. Dari klasifikasi tersebut dihitung tingkat keterkirimian informasi, tingkat ketepatan informasi, dan akurasi klasifikasi *flag*.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

Aplikasi AWSCenter Notif berhasil diimplementasikan pada platform Android sesuai rancangan. Aplikasi menyajikan halaman beranda yang memuat identitas stasiun tanggung jawab pengguna beserta ringkasan *alert* terkini, halaman daftar *alert* yang terbagi menjadi *tab alert* aktif dan *tab riwayat alert* dengan batas penyimpanan tiga hari terakhir, halaman rincian *alert* yang menampilkan parameter sensor bermasalah beserta nilai pembacaan dan penjelasan jenis *flag*, serta halaman profil untuk pengelolaan akun dan preferensi notifikasi. Seluruh informasi yang ditampilkan telah disaring sesuai stasiun yang menjadi tanggung jawab pengguna yang sedang masuk, sehingga teknisi tidak perlu menelusuri data peralatan di luar wilayah kerjanya. Tampilan keempat bagian utama aplikasi ditunjukkan pada Gambar 4.



Gambar 5. Tampilan halaman *Home*, halaman *Alerts*, halaman *Alert Details*, dan notifikasi pada perangkat

Gambar 5(a) memperlihatkan halaman *Home* yang menampilkan identitas stasiun tanggung jawab pengguna beserta *alert card* terkini lengkap dengan parameter sensor yang bermasalah, sehingga kondisi peralatan dapat diketahui begitu aplikasi dibuka. Gambar 5(b) memperlihatkan halaman *Alerts* yang memuat seluruh *alert* aktif dalam bentuk daftar gulir. Pada tangkapan layar tersebut terlihat enam *alert* yang seluruhnya berasal dari alat di bawah satu stasiun koordinator yang sama, yaitu bukti visual bahwa penyaringan berdasarkan tanggung jawab stasiun bekerja sebagaimana dirancang. Halaman ini terbagi menjadi *tab Active Alerts* untuk *alert* yang masih aktif dan *tab Alerts Log* untuk riwayat *alert* dalam batas tiga hari terakhir yang dilengkapi penyaringan berdasarkan rentang waktu dan instrumen tertentu.

Gambar 5(c) memperlihatkan halaman *Alert Details* yang memuat identitas alat, waktu kejadian, koordinat lokasi yang terhubung dengan peta, serta nilai pembacaan dan jenis *flag* pada setiap parameter sensor yang terindikasi *suspect data*. Penyertaan koordinat dan peta ditujukan untuk membantu teknisi mengetahui posisi peralatan apabila diperlukan penanganan di lapangan. Gambar 5(d) memperlihatkan notifikasi yang diterima pada bilah notifikasi perangkat, yang memuat identitas stasiun, identitas alat, dan jenis *flag* sehingga teknisi dapat menilai tingkat urgensi tanpa membuka aplikasi terlebih dahulu. Indikator warna pada setiap *alert card* digunakan untuk membedakan tingkat keparahan *flag*. Halaman splash screen, halaman autentikasi, dan halaman profil tidak ditampilkan karena ketiganya merupakan fungsi pendukung yang tidak berkaitan langsung dengan penyajian informasi *suspect data*.

Pada sisi *backend*, fungsi sinkronisasi berjalan setiap 5 menit dan menuliskan hasil penyaringan ke *collection station alerts* dengan *timestamp* pengambilan data, sumber pemicu, serta status pengiriman notifikasi. Fungsi pemeriksaan notifikasi berjalan setiap 10 menit dan menandai setiap dokumen yang telah dikirimkan sehingga dokumen yang sama tidak diproses ulang pada siklus berikutnya. Rancangan ini membuat proses pemantauan berjalan sepenuhnya di sisi layanan *cloud* dan tidak bergantung pada apakah pengguna membuka aplikasi. Implementasi tersebut memenuhi kebutuhan yang ditetapkan pada tahap perancangan dan menjadi dasar bagi ketiga pengujian yang dilaporkan pada bagian berikutnya.



3.2 Hasil Pengujian Fungsional

Pengujian fungsional dilakukan terhadap 11 skenario yang tersebar pada empat kelompok fungsi. Rekapitulasi hasil pengujian disajikan pada Tabel 2.

Tabel 2. Rekapitulasi hasil pengujian fungsional metode *black box*

Kelompok Fungsi	Jumlah Skenario	Berhasil	Gagal
Manajemen autentikasi	3	3	0
Sinkronisasi dan penyaringan data	3	3	0
Notifikasi Expo Push Service	3	3	0
Manajemen profil	2	2	0
Total	11	11	0

Berdasarkan Tabel 2, seluruh skenario memperoleh status berhasil. Pada kelompok autentikasi, sistem terbukti menolak kredensial yang tidak valid disertai pesan peringatan dan meneruskan pengguna ke halaman utama ketika kredensial valid. Pada kelompok sinkronisasi dan penyaringan data, fungsi terjadwal berhasil dieksekusi sesuai interval, tombol *reload* berhasil memicu pengambilan data di luar jadwal, dan aplikasi hanya menampilkan *alert* yang sesuai dengan stasiun tanggung jawab pengguna. Pada kelompok notifikasi, notifikasi berhasil diterima baik ketika aplikasi berada di latar depan maupun latar belakang, dan ketukan pada notifikasi berhasil mengarahkan pengguna langsung ke halaman rincian *alert* yang relevan. Pada kelompok manajemen profil, pengguna berhasil mengubah data profil dan mengakhiri sesi. Hasil ini menunjukkan bahwa seluruh fungsi yang dirancang bekerja sesuai spesifikasi, sehingga sistem layak dilanjutkan ke pengujian yang menilai kualitas informasi dan waktu penyampaiannya.

3.3 Hasil Pengujian Pengalaman Pengguna

Pengujian pengalaman pengguna diikuti 12 responden yang seluruhnya merupakan teknisi dengan keterkaitan terhadap pemantauan peralatan stasiun cuaca otomatis. Nilai rata-rata, varians, dan kategori tolok ukur setiap skala disajikan pada Tabel 3.

Tabel 3. Nilai rata-rata, varians, dan kategori tolok ukur setiap skala

Skala	Rata-rata	Varians	Kategori Tolok Ukur
Daya Tarik	2,389	0,31	<i>Excellent</i>
Kejelasan	2,479	0,11	<i>Excellent</i>
Efisiensi	2,208	0,29	<i>Excellent</i>
Keandalan	2,146	0,48	<i>Excellent</i>
Stimulasi	2,375	0,32	<i>Excellent</i>
Kebaruan	2,167	0,53	<i>Excellent</i>

Berdasarkan Tabel 3, seluruh skala memperoleh nilai rata-rata di atas 0,8 sehingga termasuk dalam rentang evaluasi positif, dan keenamnya berada pada kategori tolok ukur *Excellent* yang berarti hasil evaluasi aplikasi termasuk dalam rentang sepuluh persen hasil terbaik pada data perbandingan. Skala Kejelasan memperoleh nilai tertinggi sebesar 2,479 sekaligus varians terendah sebesar 0,11, yang menunjukkan bahwa penilaian responden pada aspek kemudahan pemahaman relatif seragam. Keceragaman ini masuk akal karena alur penggunaan aplikasi hanya terdiri atas beberapa langkah dan informasi yang ditampilkan telah disaring sehingga responden tidak perlu memilah data yang tidak relevan.

Skala Keandalan memperoleh nilai terendah sebesar 2,146 dengan varians 0,48, sedangkan skala Kebaruan memiliki varians tertinggi sebesar 0,53. Sebaran penilaian yang lebih lebar pada kedua skala tersebut dapat dipahami karena persepsi keandalan dan kebaruan sangat bergantung pada pengalaman masing-masing responden terhadap aplikasi sejenis serta pada durasi pemakaian yang relatif singkat selama pengujian. Meskipun demikian, kedua skala tetap berada pada kategori *Excellent*. Hasil ini menunjukkan bahwa aplikasi dapat diterima oleh calon penggunanya, namun perlu dicatat bahwa dengan ukuran sampel 12 responden hasil tersebut bersifat gambaran awal dan belum dapat digeneralisasi ke seluruh populasi teknisi.

3.4 Hasil Pengujian Kesesuaian Informasi

Pengamatan efektif berlangsung sekitar 3,5 jam yang terbagi ke dalam dua segmen. Dari periode tersebut diperoleh 62 unit analisis yang berasal dari empat alat, karena dari tujuh alat yang berada dalam cakupan pengujian hanya empat alat yang memunculkan informasi *flag* selama periode pengamatan berlangsung. Hasil klasifikasi unit analisis disajikan pada Tabel 4.

Tabel 4. Hasil uji kesesuaian informasi *flag*

Kategori	Jumlah	Persentase (%)
Sesuai	62	100,00
Tidak tersampaikan	0	0,00
Tidak berpadanan	0	0,00

Kategori	Jumlah	Persentase (%)
Jenis <i>flag</i> berbeda	0	0,00
Total unit analisis	62	100,00

Berdasarkan Tabel 4, seluruh informasi *flag* yang terdapat pada API *alert* AWS Center selama periode pengamatan berhasil tersaji pada aplikasi dengan jenis *flag* yang sama. Tingkat keterkiriman informasi, tingkat ketepatan informasi, dan akurasi klasifikasi *flag* ketiganya bernilai 100 persen. Tidak ditemukan unit yang berkategori tidak tersampaikan, yang berarti tidak ada informasi yang hilang dalam proses pengolahan, dan tidak ditemukan unit yang berkategori tidak berpadanan, yang berarti aplikasi tidak memunculkan informasi tanpa dasar pada sumbernya. Ketiadaan unit yang tidak berpadanan sekaligus memverifikasi bahwa penyaringan berdasarkan stasiun tanggung jawab pengguna bekerja sesuai rancangan, karena seluruh *alert* yang muncul pada akun pengguna uji memang berasal dari alat di bawah stasiun koordinatonya.

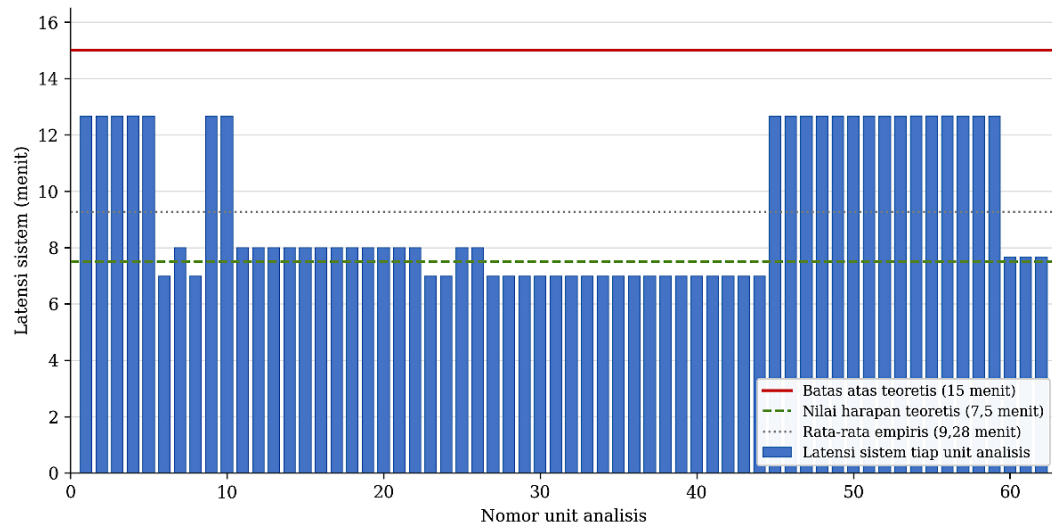
3.5 Hasil Pengukuran Latensi

Statistik ketiga parameter latensi beserta latensi sistem yang dihitung dari 62 unit analisis disajikan pada Tabel 5.

Tabel 5. Statistik parameter latensi penyampaian informasi *flag* (menit)

Parameter Latensi	Rata-rata	Median	Minimum	Maksimum	Simpangan Baku
Latensi hulu	12,77	14,04	8,37	15,05	2,58
Latensi sinkronisasi	2,54	2,69	2,02	3,03	0,40
Latensi antrean notifikasi	6,74	4,97	4,96	9,98	2,39
Latensi sistem	9,28	7,99	6,99	12,67	2,54

Berdasarkan Tabel 5, latensi sistem memiliki nilai rata-rata 9,28 menit dengan nilai maksimum 12,67 menit dan nilai minimum 6,99 menit. Kontribusi terbesar terhadap latensi sistem berasal dari latensi antrean notifikasi dengan rata-rata 6,74 menit, sedangkan latensi sinkronisasi hanya menyumbang rata-rata 2,54 menit dengan simpangan baku 0,40 menit yang menunjukkan keteraturan eksekusi penjadwal. Latensi hulu memiliki rata-rata 12,77 menit, yaitu lebih besar daripada seluruh latensi yang menjadi tanggung jawab sistem yang dibangun. Apabila dihitung sejak data diukur oleh peralatan, total waktu penyampaian mencapai 22,05 menit. Sebaran latensi sistem pada setiap unit analisis ditunjukkan pada Gambar 6.



Gambar 6. Sebaran latensi sistem pada setiap unit analisis

Gambar 5 memperlihatkan bahwa seluruh unit analisis berada di bawah garis batas atas teoretis 15 menit, dan sebagian besar unit tersebar di sekitar nilai harapan teoretis 7,5 menit. Pola bertingkat pada grafik tersebut merupakan konsekuensi langsung dari mekanisme penjadwalan, karena setiap informasi *flag* harus menunggu siklus sinkronisasi berikutnya lalu menunggu siklus pemeriksaan notifikasi berikutnya. Pengamatan langsung terhadap lima kejadian notifikasi menunjukkan bahwa notifikasi diterima pada perangkat dalam hitungan sekitar satu detik setelah waktu pengiriman yang tercatat sistem, sehingga tahap distribusi melalui Expo Push Service tidak memberikan kontribusi berarti terhadap latensi keseluruhan dan pengukuran yang diakhiri pada T3 telah merepresentasikan waktu penyampaian informasi kepada pengguna.

3.6 Pembahasan

Akurasi klasifikasi *flag* sebesar 100 persen merupakan hasil yang memang diharapkan dan tidak boleh dibaca sebagai keunggulan sistem, karena sistem menyalin deskripsi *flag* dari API tanpa melakukan pengolahan ulang. Nilai yang lebih bermakna dari Tabel 4 justru terletak pada tidak adanya unit berkategori tidak tersampaikan maupun tidak berpadanan,



karena kedua kategori itulah yang akan muncul apabila terdapat kesalahan pada logika pengambilan data maupun pada logika penyaringan berbasis stasiun. Dengan demikian, hasil pengujian kesesuaian sebaiknya dimaknai sebagai verifikasi terhadap rantai pengolahan data, bukan sebagai klaim mengenai ketepatan algoritma *quality control* yang berjalan pada sisi AWS Center.

Nilai rata-rata latensi sistem sebesar 9,28 menit mendekati nilai harapan teoretis 7,5 menit yang dihitung dari rancangan penjadwalan. Kedekatan kedua nilai tersebut menunjukkan bahwa mekanisme penjadwalan berjalan sesuai rancangan tanpa penundaan berarti pada sisi *backend*, dan bahwa selisih di antara keduanya berasal dari sebaran kemunculan *flag* yang tidak merata terhadap awal siklus penjadwalan. Temuan yang paling berimplikasi terhadap rancangan adalah bahwa latensi sistem sebesar 9,28 menit lebih kecil daripada periode pembaruan data pada API *alert* AWS Center yang berlangsung sekitar setiap 10 menit. Artinya, setiap informasi *flag* telah disampaikan kepada pengguna sebelum pembaruan berikutnya tersedia pada sumber. Konsekuensinya, mempersingkat interval penjadwalan tidak akan mempercepat penyampaian informasi selama periode pembaruan pada sumber data tidak berubah, dan justru hanya akan menambah beban pemanggilan API serta biaya eksekusi fungsi. Perlu digarisbawahi bahwa temuan ini bersumber dari pengamatan efektif sekitar 3,5 jam pada empat alat di satu stasiun koordinator sehingga kesimpulan ini merepresentasikan kondisi pada periode dan lokasi pengujian tersebut. Generalisasi terhadap variasi beban jaringan dan karakteristik operasional di seluruh wilayah kerja BMKG masih memerlukan pengujian lanjutan dengan cakupan waktu dan stasiun yang lebih luas. Percepatan yang berarti hanya dapat dicapai apabila sumber data menyediakan mekanisme berbasis peristiwa, misalnya melalui broker pesan dengan pola *publish-subscribe*, sehingga sistem tidak perlu menunggu siklus pemeriksaan berkala.

Apabila dibandingkan dengan penelitian terdahulu, sistem yang dibangun pada penelitian ini memiliki capaian fungsional yang setara namun memberikan informasi evaluatif yang berbeda. Wahyudin dkk. (2024) dan Fernando dkk. (2020) sama-sama menunjukkan bahwa *push notification* berhasil menyampaikan informasi kepada pengguna, tetapi keduanya tidak melaporkan berapa lama waktu yang dibutuhkan sejak informasi tersedia pada sumber hingga diterima pengguna. Zani dkk. (2024) melaporkan responsivitas aplikasi berbasis React Native secara kualitatif tanpa angka latensi. Penelitian ini melengkapi temuan tersebut dengan menunjukkan bahwa sebagian besar waktu tunggu justru berada pada sisi sumber data, yaitu 12,77 menit dari total 22,05 menit, sehingga upaya optimalisasi pada sisi aplikasi memiliki ruang perbaikan yang terbatas. Temuan ini sejalan dengan pandangan Wu (2022) bahwa nilai *push notification* terletak pada kecepatan respons pengguna, namun menambahkan syarat bahwa kecepatan tersebut hanya dapat direalisasikan sejauh sumber data memungkinkannya.

Dibandingkan dengan mekanisme sebelumnya yang mengharuskan teknisi membuka laman web AWS Center secara manual, kontribusi utama sistem ini bukan semata pada kecepatan melainkan pada kepastian. Pada mekanisme manual, waktu penerimaan informasi bergantung sepenuhnya pada kapan teknisi memeriksa *dashboard* sehingga bersifat terbuka dan tidak terukur. Dengan sistem yang dibangun, waktu penerimaan menjadi terbatas pada rentang yang terukur dengan batas atas yang dapat dihitung langsung dari rancangan penjadwalan. Karakteristik inilah yang memungkinkan penyampaian informasi *suspect data* dijadikan dasar evaluasi kinerja pemeliharaan peralatan observasi, karena keterlambatan penanganan kini dapat dipisahkan antara yang bersumber dari sistem informasi dan yang bersumber dari respons teknisi.

Pada tataran praktis, hasil pengukuran ini dapat langsung dimanfaatkan sebagai dasar penyusunan standar waktu tanggap pemeliharaan. Dengan diketahuinya bahwa informasi *flag* sampai kepada teknisi dalam rentang 6,99 sampai 12,67 menit sejak tersedia pada API, pengelola dapat menetapkan tenggat penanganan yang dihitung dari waktu pengiriman notifikasi, bukan dari waktu teknisi kebetulan membuka *dashboard*. Angka latensi hulu sebesar 12,77 menit juga memberikan informasi bagi pengelola AWS Center mengenai besarnya jeda yang terjadi pada sisi server, sehingga upaya percepatan dapat diarahkan pada komponen yang benar-benar menjadi penyumbang terbesar. Selain itu, penyaringan berbasis tanggung jawab stasiun mengurangi jumlah notifikasi yang diterima setiap teknisi hanya pada peralatan di wilayah kerjanya, sehingga mengurangi risiko kelelahan notifikasi yang lazim terjadi pada sistem peringatan dengan volume pesan tinggi (Voutsas, F., Violos, J., & Leivadeas, A., 2024). Ketiga hal tersebut menunjukkan bahwa manfaat sistem tidak berhenti pada otomatisasi pengiriman pesan, melainkan pada tersedianya dasar kuantitatif untuk mengelola proses pemeliharaan peralatan observasi.

Penelitian ini memiliki sejumlah keterbatasan yang perlu diperhatikan dalam menafsirkan hasilnya. Pengamatan kesesuaian dan latensi berlangsung sekitar 3,5 jam efektif pada empat alat di satu stasiun koordinator, sehingga sebaran latensi yang diperoleh belum tentu mewakili kondisi beban jaringan dan variasi waktu yang berbeda. Pengujian pengalaman pengguna melibatkan 12 responden sehingga hasilnya bersifat gambaran awal. Selain itu, aplikasi baru dikembangkan untuk platform Android dan proses autentikasinya masih terpisah dari akun web AWS Center. Penelitian lanjutan disarankan memperpanjang periode pengamatan hingga beberapa minggu, memperluas cakupan alat dan stasiun, membandingkan sebaran latensi antar skenario interval penjadwalan, serta menguji mekanisme berbasis peristiwa apabila sumber data telah menyediakannya.

4. KESIMPULAN

Penelitian ini menghasilkan aplikasi mobile AWSCenter Notif yang menyajikan informasi *suspect data* sesuai stasiun tanggung jawab pengguna dan menyampaikannya secara otomatis melalui *push notification*. Seluruh pengujian



fungsional dan pengalaman pengguna menunjukkan hasil positif, sedangkan pengujian terhadap 62 unit analisis mencatat tingkat keterkirrimian dan akurasi 100 persen dengan latensi sistem rata-rata 9,28 menit, seluruhnya di bawah batas atas teoretis. Kontribusi penelitian ini mencakup tiga hal, yaitu tersedianya mekanisme penyaringan notifikasi berbasis pemetaan alat terhadap stasiun tanggung jawab teknisi, rancangan lapisan notifikasi *serverless* yang mengubah diseminasi *suspect data* dari pasif menjadi aktif tanpa mengubah arsitektur sistem sumber, serta tersusunnya kerangka pengukuran empat titik waktu yang memisahkan selang waktu pada sisi sumber dari selang waktu pada sisi sistem. Ketiga kontribusi tersebut mengubah ketidakpastian waktu penerimaan informasi yang sebelumnya terbuka menjadi rentang yang terukur, sekaligus menunjukkan bahwa penyumbang terbesar terhadap total waktu penyampaian, yaitu 12,77 menit, berasal dari jeda pembaruan pada sisi server sumber AWS Center. Latensi hulu tersebut berpotensi dimitigasi tanpa mengubah arsitektur *backend* secara menyeluruh, misalnya dengan mengusulkan mekanisme *Webhook* pada sisi server AWS Center yang mengirimkan pembaruan data segera setelah tersedia, sehingga sistem tidak perlu menunggu siklus pembaruan berkala pada sumber. Pendekatan ini berpotensi memangkas latensi hulu secara signifikan tanpa menambah kompleksitas pada sisi klien, meskipun implementasinya bergantung pada kesediaan pengelola AWS Center untuk menyediakan antarmuka tersebut. Penelitian ini memiliki keterbatasan pada periode pengamatan yang hanya sekitar 3,5 jam efektif pada satu stasiun serta jumlah responden pengujian pengalaman pengguna yang terbatas. Penelitian selanjutnya disarankan memperpanjang periode pengamatan, memperluas cakupan alat dan stasiun, menguji integrasi *Webhook* sebagai mekanisme penyampaian berbasis peristiwa dari sisi sumber data, menambahkan mekanisme percobaan ulang (*retry*) otomatis pada proses sinkronisasi, serta memperluas cakupan platform aplikasi di luar Android.

REFERENCES

- Badan Meteorologi, Klimatologi, dan Geofisika. (2025a). *Rencana strategis Badan Meteorologi, Klimatologi, dan Geofisika tahun 2025–2029*. BMKG. <https://jdih.bmkg.go.id/infografis/3>
- Badan Meteorologi, Klimatologi, dan Geofisika. (2025b). *Standar operasional prosedur pemantauan dan evaluasi kuantitas dan kualitas data pada AWS Center di lingkungan BMKG*. BMKG. <https://share.google/iDu1YsA6dyXxiS1b8>
- Birajdar, S., & Banubakode, A. (2025). Optimizing AI-powered mobile applications with serverless cloud computing: A performance, cost, and sustainability analysis in Flutter. *International Journal of Advanced Research in Science, Communication and Technology*, 5(12), 1–10. <https://www.ijarsct.co.in/May5i12.html>
- Expo. (2025). *Expo notifications documentation*. <https://docs.expo.dev/>
- Fernando, F., Arini, & Fahrianto, F. (2020). Push notification monitoring sistem pintu air berbasis Android menggunakan Firebase Cloud Messaging. *Jurnal Teknik Informatika*, 13(1), 1–10. <https://doi.org/10.15408/jti.v13i1.15884>
- Hulu, A. A., & Nasution, M. I. P. (2025). Analisis komparatif dimensi kualitas data pada sistem informasi publik: Studi sintesis literatur 2015–2024. *Jurnal Ilmiah Nusantara*, 2(4), 120–129. <https://doi.org/10.61722/jinu.v2i4.4974>
- Hussamadin, R., Jansson, G., & Mukkavaara, J. (2023). Digital quality control system A tool for reliable production. *Buildings*, 13(9), 2143. <https://doi.org/10.3390/buildings13020358>
- Kushendriawan, M. A., Santoso, H. B., Putra, P. O. H., & Schrepp, M. (2021). Evaluating user experience of a mobile health application Halodoc using User Experience Questionnaire and usability testing. *Jurnal Sistem Informasi*, 17(1), 58–71. <https://doi.org/10.21609/jsi.v17i1.1063>
- Meta Platforms. (2025). *React Native: Learn once, write anywhere*. <https://reactnative.dev/docs/getting-started>
- Narulita, S., Nugroho, A., & Abdillah, M. Z. (2024). Diagram Unified Modelling Language (UML) untuk perancangan sistem informasi manajemen penelitian dan pengabdian masyarakat. *Bridge: Jurnal Publikasi Sistem Informasi dan Telekomunikasi*, 2(3), 244–256. <https://doi.org/10.62951/bridge.v2i3.174>
- Rajan, R. A. P. (2020). A review on serverless architectures Function as a service (FaaS) in cloud computing. *TELKOMNIKA*, 18(1), 530–537. <https://doi.org/10.12928/TELKOMNIKA.v18i1.12169>
- Seo, J. Y., & Rogge, M. (2023). Coding non-visually in Visual Studio Code: Collaboration towards accessible development environment for blind programmers. Dalam *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility*. Association for Computing Machinery. <https://doi.org/10.1145/3597638.3614550>
- Siahaan, C., Tampubolon, J. A., & Sinambela, N. B. (2021). Diseminasi informasi melalui media online sebagai transformasi media konvensional. *Jurnal Signal*, 10(2), 322–334. <https://doi.org/10.33603/signal.v9i2.6288>
- Tan, J. P. Y., Jauhar, M. N. A., Hussain, A., & Mkuzi, M. S. (2023). mHealth app to facilitate remote care for patients with COVID-19: Rapid development of the DrCovid+ app. *JMIR Formative Research*, 7, e41894. <https://doi.org/10.2196/38555>
- Toyib, R., Sani, A., Kanedi, I., & Setiawan, A. (2025). Indonesian sign system introduction application with TensorFlow Lite and Firebase Authentication. *Jurnal Teknik Informatika dan Sistem Informasi*, 11(1), 31–48. <https://doi.org/10.28932/jutisi.v11i1.9678>
- Wahyudin, A., Anisyah, A., & Ahmaddifa, D. (2024). Pengembangan sistem notifikasi real-time untuk aplikasi manajemen persuratan multiplatform menggunakan Firebase Cloud Messaging dan Application Programming Interface. *Jurnal Teknik Informatika UNIKA Santo Thomas*, 9(2), 1–12. <https://ejournal.ust.ac.id/index.php/JTIUST/article/view/4437>



- Voutsas, F., Violos, J., & Leivadeas, A. (2024). Mitigating alert fatigue in cloud monitoring systems: A machine learning perspective. *Computer Networks*, 250, Article 110543. <https://doi.org/10.1016/j.comnet.2024.110543>
- Wu, H. T. (2022). Establish a digital real-time learning system with push notifications. *Frontiers in Psychology*, 13, 767389. <https://doi.org/10.3389/fpsyg.2022.767389>
- Zani, A. W. R., Kartini, K., & Rizki, A. M. (2024). Rancang bangun aplikasi mobile bank sampah menggunakan framework React Native dan Rest API. *Uranus: Jurnal Ilmiah Teknik Elektro, Sains dan Informatika*, 2(3), 112–124. <https://doi.org/10.61132/uranus.v2i3.254>

Rangkuman Revisi Reviewer

No.	Bagian Naskah	Komentar Reviewer 1	Komentar Reviewer 2
1	Judul	Penulisan judul sudah cukup baik dan sesuai	Judul sudah cukup deskriptif, namun

No.	Bagian Naskah	Komentar Reviewer 1	Komentar Reviewer 2
			penggunaan istilah 'AWS Center' di sini berpotensi menimbulkan ambiguitas dengan Amazon Web Services. Penulis wajib mengklarifikasi akronim AWS di awal untuk menghindari kerancuan dengan penyedia layanan cloud global.
2	Abstrak	Penulisan abstrak sudah cukup baik dan sesuai	Abstrak cukup informatif, namun penyajian data latensi (9,28 menit) yang terlalu detail di abstrak membuat alur kurang padat. Fokus seharusnya pada nilai kebaruan dan kontribusi sistem daripada rincian angka yang sebenarnya lebih cocok diletakkan di bagian hasil.
3	Pendahuluan	Isi pendahuluan sudah cukup baik dan sesuai	Pendahuluan terlalu panjang lebar pada latar belakang teoretis. Hubungan antara sistem existing dan urgensi notifikasi terasa kurang tajam. Penulis perlu menjelaskan lebih eksplisit mengapa sistem notifikasi berbasis Firebase dipilih dibandingkan opsi lain yang lebih mungkin mendukung latensi rendah, seperti WebSocket atau MQTT, mengingat latensi sistem ini masih cukup tinggi (9,28 menit).
4	Metode Penelitian	Sebaiknya pada bagian metodologi penelitian ditambahkan subbab yang secara singkat menjelaskan objek-objek penelitian yang sesuai dengan kata kunci di abstrak, serta berikan penjelasan / gambar diagram alur tahapan penelitian.	Metodologi menggunakan RAD dianggap tepat, namun diagram alir pada Gambar 2 terlalu sederhana. Penulis tidak menjelaskan mekanisme penanganan jika terjadi kegagalan sinkronisasi (error handling) pada lapisan Firebase Functions. Selain itu, pemilihan interval 5 menit untuk sinkronisasi tidak disertai justifikasi teknis mengenai trade-off antara efisiensi sumber daya dan urgensi data.
5	Hasil dan Pembahasan	Isi hasil dan pembahasan sudah cukup baik dan sesuai	Hasil pengujian menunjukkan latensi 9,28 menit. Sangat kritis: penulis menyatakan 'mempersingkat interval tidak akan mempercepat penyampaian', namun ini adalah klaim yang



No.	Bagian Naskah	Komentar Reviewer 1	Komentar Reviewer 2
			lemah jika hanya berdasarkan pada 3,5 jam pengamatan. Periode pengamatan ini secara statistik tidak representatif untuk menyimpulkan perilaku sistem di seluruh wilayah operasional BMKG.
6	Kesimpulan	<i>(Tidak ada catatan khusus pada bagian ini)</i>	Kesimpulan mengulang apa yang sudah ada di hasil. Kontribusi 'terukurnya waktu' memang bagus, namun penulis belum memberikan rekomendasi mitigasi teknis terkait latensi hulu sebesar 12,77 menit selain hanya menyebutnya di luar kendali. Kesimpulan seharusnya memberikan arah pengembangan masa depan yang lebih konkret, misalnya integrasi Webhook dari sisi server sumber.
7	Referensi / Daftar Pustaka	untuk bagian referensi sebaiknya gunakan Mendeley , lalu sebaiknya referensi harus memiliki metadata yang lengkap yg minimal 20 penelitian yg terbit dalam 5 tahun terakhir	Setiap referensi yang bersumber dari jurnal wajib di lengkapi dengan DOI atau URL jurnal.