

Integrasi REST API Rule-Based Dan WebGIS Untuk Otomasi Diseminasi Gempabumi

Iqbal Jabar Marjan^{1*}, Dean J. Samuel Kembuan², Fatih Jalalhaq Ratumboba³, Ottopianus Mellolo⁴, Venny Vita Ponggawa⁵

^{1,2,3,4,5}Program Studi Teknik Informatika, Politeknik Negeri Manado

* iqbalmrjn@gmail.com

Abstrak

Kecepatan penyampaian informasi kebencanaan sering kali terhambat oleh birokrasi data teknis yang panjang. Di lingkungan operasional BMKG Stasiun Geofisika Manado, proses diseminasi informasi pasca-akuisisi dari mesin SeisComP3 rupanya masih sangat bergantung pada intervensi manual *Duty Officer*. Situasi ini kerap memicu antrean pekerjaan yang memakan waktu, terutama ketika petugas harus memilah ratusan anomali getaran gempa mikro sebelum akhirnya bisa menyalin laporan ke grup komunikasi digital. Untuk memecahkan masalah tersebut, penelitian ini menggunakan metode *Research and Development* (R&D) untuk merancang sebuah ekosistem *Zero-Touch Automation*. Penelitian ini mengintegrasikan *backend* REST API yang dibekali algoritma *Rule-Based System* untuk menyeleksi parameter seismik secara mandiri, kemudian dihubungkan secara terpadu dengan *frontend* WebGIS dan Bot Telegram. Hasil pengujian kuantitatif di lapangan menunjukkan bahwa arsitektur ini sukses melakukan rute data cerdas; gempa yang memenuhi ambang batas kewaspadaan akan langsung diteruskan menjadi peringatan dini secara otonom dengan rata-rata waktu tunda (*latency*) kurang dari 3 detik. Sebagai kontribusi ilmiah dan praktis, implementasi sistem yang dilengkapi panel analitik dinamis ini terbukti secara nyata mampu memangkas waktu krusial pelaporan dari hitungan menit menjadi hitungan detik, serta secara efektif menekan risiko *human error* di ruang observasi.

Kata kunci : Otomasi diseminasi, REST API, Rule-Based System, SeisComP3, WebGIS..

Abstract

The speed of disaster information delivery is often hindered by lengthy technical data bureaucracy. At the BMKG Manado Geophysical Station, the dissemination process post-data acquisition from the SeisComP3 machine still heavily relies on the manual intervention of the Duty Officer. This condition triggers work queues, especially when officers must sort through hundreds of micro-earthquake anomalies before copying reports to digital communication groups. To solve this problem, this research uses the Research and Development (R&D) method to design a Zero-Touch Automation ecosystem. We integrated a REST API backend equipped with a Rule-Based System algorithm to independently select seismic parameters, then seamlessly connected it with a WebGIS frontend and Telegram Bot. Quantitative test results in the field indicate that this architecture successfully performs smart data routing; earthquakes meeting the alert threshold are immediately forwarded as early warnings autonomously with an average latency of less than 3 seconds. As a scientific and practical contribution, the implementation of this system, which is equipped with a dynamic analytic panel, is proven to cut crucial reporting times from minutes to seconds, and effectively minimize human error risks in the observation room.

Keywords : Automation dissemination, REST API, Rule-Based System, SeisComP3, WebGIS.

1. Pendahuluan

Posisi geografis Indonesia yang diapit oleh sabuk Cincin Api Pasifik (*Ring of Fire*) membawa konsekuensi logis: wilayah ini menjadi salah satu pusat aktivitas kegempaan paling aktif di tingkat

global. Dalam berbagai kajian mitigasi bencana, banyak pakar menegaskan bahwa seberapa cepat informasi didistribusikan pada rentang *Golden Time* memegang peranan yang setara krusialnya dengan tingkat presisi instrumen

pendeteksi itu sendiri [1]. Sesuai protokol kedaruratan, kelambanan merespons *Golden Time* sesaat setelah gempa terjadi bisa berdampak langsung pada tingginya risiko korban jiwa. Sebagai garda terdepan di Indonesia, Badan Meteorologi Klimatologi dan Geofisika (BMKG) selama ini bergantung pada SeisComP3 guna menyedot data getaran seismik langsung dari stasiun pengamat [2].

Namun, praktik di lapangan khususnya pada ruang observasi Pusat Gempabumi Regional (PGR) 10 Stasiun Geofisika Manado justru mengungkap adanya rantai birokrasi teknis yang memperlambat fase pasca-akuisisi. Akar permasalahannya berasal dari kepekaan sensor lokal itu sendiri, yang setiap harinya tanpa henti menangkap ratusan getaran gempa mikro atau *noise* [3]. Kondisi ini memaksa *Duty Officer* yang bertugas untuk melakukan penyaringan anomali secara manual, mencatatnya ke log *Quality Control*, dan baru membagikannya ke platform digital. Pola kerja tradisional ini tidak hanya mengikis *Golden Time* secara percuma, tetapi juga sangat rentan disusupi *human error* akibat kondisi kelelahan (*fatigue*) saat petugas harus menghadapi gempuran data secara terus-menerus. Upaya untuk merespons bencana secara digital bukannya belum pernah dilakukan. Sejumlah akademisi sudah mengeksplorasi ranah ini, sebut saja Supriyadi (2021) yang sukses mendemonstrasikan kelincahan Bot Telegram

sebagai sarana penyebar peringatan massal [4], maupun Subagyo (2022) yang mengandalkan kapabilitas WebGIS guna memetakan zona bahaya secara visual [5]. Sayangnya, jika ditelaah lebih dalam, masih terdapat celah atau *gap* penelitian yang belum sepenuhnya terpecahkan.

2. Tinjauan Pustaka

2.1. Penelitian Terkait

Penelitian terdahulu telah mengeksplorasi berbagai metode digitalisasi untuk mitigasi bencana, namun mayoritas masih berfokus pada pendekatan modul tunggal yang terpisah.

1. Pada aspek diseminasi informasi, Supriyadi (2021) mengimplementasikan metode *webhook* pada Bot Telegram untuk mengirimkan notifikasi peringatan secara massal [4].

2. Di sisi lain, pada aspek visualisasi spasial, Subagyo (2022) serta Sholikhah dkk. (2019) menggunakan metode WebGIS untuk melakukan pemetaan wilayah berbasis koordinat [5], [6]. Keandalan WebGIS dalam menyajikan pemetaan interaktif yang responsif juga dibuktikan oleh Sudianto & Sadali (2018) serta Fathurrahman dkk. (2022) [7], [8]. Sementara pada segi pemrosesan *backend*, metode REST API dinilai sangat andal dalam menjaga stabilitas pertukaran *payload* data [9]. Bila dikomparasikan, metode-metode pada penelitian terdahulu memiliki keterbatasan integrasi, di mana sistem peringatan berbasis teks dan peta visual masih berdiri terpisah, serta

operasionalnya masih bergantung pada *input* data manual. Oleh karena itu, penelitian ini mengusulkan kebaruan (*novelty*) dengan cara mengintegrasikan metode REST API, algoritma logika *Rule-Based System* sebagai mesin *filtering* otomatis, serta visualisasi WebGIS ke dalam satu arsitektur terpadu. Berbeda dengan penelitian sebelumnya, sistem terintegrasi ini mampu memproses dan merutekan data kebencanaan secara otonom (*Zero-Touch Automation*)

2.2. Landasan Teori

1. SeisComP3

Dalam operasional pengamatan kegempaan, SeisComP3 difungsikan sebagai perangkat lunak utama untuk mengakuisisi sinyal seismik secara seketika (*real-time*). Sistem ini secara otomatis mengekstraksi parameter krusial gempabumi, seperti magnitudo, kedalaman, dan titik koordinat, langsung dari jaringan sensor yang tersebar di lapangan [2], [3].

2. REST API

Guna menjaga stabilitas pertukaran data antar-sistem tanpa membebani kinerja *server*, penelitian ini menerapkan arsitektur REST API. Pendekatan tersebut secara efektif memisahkan lingkungan pemrosesan *frontend* dan *backend* dengan memanfaatkan format JSON, sehingga *payload* informasi diseminasi dapat ditransmisikan dengan lebih efisien [9], [10].

3. Rule-Based System

Algoritma turunan sistem pakar ini beroperasi dengan mengadopsi nalar pengambilan keputusan berbasis logika *If-Then*. Pada arsitektur otomasi, algoritma tersebut diimplementasikan sebagai filter otonom yang bertugas memvalidasi anomali gempa berdasarkan ambang batas bahaya, sehingga keterlibatan operator secara manual dapat diminimalisasi secara signifikan [11], [12], [13].

4. WebGIS (*Web Geographic Information System*)

Integrasi antara Sistem Informasi Geografis dan teknologi internet menghasilkan platform WebGIS yang andal untuk keperluan pemetaan. Pemanfaatan teknologi ini memungkinkan *plotting* episenter gempabumi divisualisasikan secara interaktif dan dinamis, serta dapat diakses secara langsung melalui peramban web standar guna keperluan kebencanaan [5], [6], [7], [8], [14].

5. Telegram Bot API

Antarmuka pemrograman aplikasi berbasis *cloud* dari Telegram menyediakan fasilitas untuk mengintegrasikan skrip pihak ketiga ke dalam ekosistem perpesanan. Fitur integrasi ini dimanfaatkan untuk mengonfigurasi sistem agar mampu mengeksekusi pengiriman peringatan dini (*push notification*) secara seketika ke ruang komunikasi observasi [4].

6. Docker

Sebagai langkah mitigasi terhadap potensi benturan dependensi perangkat lunak, Docker

diimplementasikan sebagai platform virtualisasi tingkat sistem operasi. Teknologi *containerization* ini mengisolasi lingkungan aplikasi guna memastikan stabilitas dan konsistensi kinerja sistem saat tahap pemindahan (*deployment*) ke *server* produksi ^[15].

7. Next.js

Pengembangan antarmuka visual pada riset ini dibangun menggunakan *framework* Next.js yang.

3. Metode Penelitian

3.1. Desain dan Pendekatan Penelitian

Penelitian ini menggunakan pendekatan *Research and Development* (R&D) guna merancang arsitektur otomasi pelaporan gempa. Kerangka kerja rekayasa perangkat lunak yang diimplementasikan adalah *Incremental Model* ^[17]. Pemilihan model ini didasarkan pada kompleksitas arsitektur sistem yang menuntut integrasi beberapa modul independen secara paralel (seperti REST API, Telegram Bot, dan WebGIS). Melalui pendekatan inkremental, proses pengembangan dapat dibagi ke dalam beberapa modul fungsional (*increment*) secara bertahap. Hal ini memungkinkan setiap fungsionalitas modul diuji secara praktis dan dievaluasi pada lingkungan *server* lokal Stasiun Geofisika Manado tanpa mengganggu stabilitas perangkat lunak SeisComP3 utama yang sedang beroperasi.

3.2. Lokasi Penelitian dan Teknik Pengumpulan Data

Studi kasus pada penelitian ini dilaksanakan secara langsung di Pusat Gempabumi Regional (PGR) 10 Stasiun Geofisika Manado. Dalam merumuskan spesifikasi kebutuhan sistem, tahap awal pengumpulan data dilakukan melalui metode observasi lapangan. Pengamatan difokuskan pada alur kerja operasional *Duty Officer* BMKG saat merespons *event* gempabumi melalui antarmuka SeisComP3. Observasi ini esensial guna mengidentifikasi titik penumpukan beban kerja (*bottleneck*) yang memakan durasi paling signifikan. Selanjutnya, guna memastikan purwarupa arsitektur dapat diuji menggunakan parameter yang valid, penelitian ini melakukan ekstraksi data mentah dari *server* *QuakeLink* pada jaringan intranet BMKG. Perlu ditegaskan bahwa proses penarikan data ini tidak menggunakan teknik *web scraping* non-prosedural, melainkan murni memanfaatkan jalur *Application Programming Interface* (API) resmi yang telah mendapatkan otorisasi legal dari pihak Stasiun Geofisika Manado untuk keperluan riset pengembangan. Proses ekstraksi data ini memanfaatkan protokol standar FDSNWS dengan format keluaran JSON. Adapun dataset yang ditarik untuk keperluan pengujian fungsionalitas sistem berjumlah 4.106 arsip data historis gempabumi yang terekam selama periode pelaksanaan observasi.

3.3. Prosedur Pengembangan Sistem

Pengerjaan teknis dalam riset ini berpegang pada konsep dasar *Zero-Touch Automation*. Maksudnya, penelitian ini berupaya semaksimal mungkin untuk mengeliminasi campur tangan manusia dari alur perpindahan data. Agar lebih terarah, prosedur ini dibagi menjadi tiga fase utama:

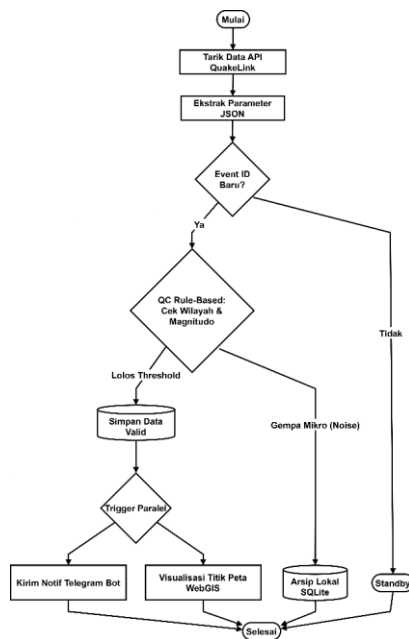
1. Pembangunan Monorepo : Darpada repot mengurus folder yang terpisah, proses pengembangan *server* Flask dan antarmuka Next.js sengaja digabungkan dalam satu repositori (*monorepo*). Tujuannya agar koordinasi antarkode lebih sinkron. Setelah itu, seluruh kode dibungkus menggunakan *container* Docker. Ini adalah trik agar aplikasi kebal terhadap masalah perbedaan versi *library* saat nanti dipasang di *server* mana pun.

2. Injeksi Rule-Based System : Di bagian *backend*, sistem ditanamkan logika "Jika-Maka" sebagai saringan utamanya. Cara kerjanya begini: setiap kali ada data JSON baru yang masuk, algoritma akan otomatis mengecek batasan wilayah atau *Geofencing* (khusus PGR 10) beserta skala magnitudonya. Kalau sistem mendeteksi getaran tersebut sebagai anomali sinyal palsu (*noise*) atau berada jauh di luar batas wilayah pantauan (luar *Geofencing* PGR 10), datanya secara diam-diam hanya diarsipkan ke SQLite lokal. Namun, jika data tervalidasi sebagai gempa riil di wilayah PGR 10 (termasuk gempa

mikro sesuai SOP observasi), sistem akan langsung mengklasifikasikan tingkat kewaspadaannya dan memicu modul pelaporan Telegram secara seketika.

3. Integrasi Diseminasi dan Dashboard: Pada tahap penyelesaian, data yang berhasil lolos seleksi tidak dibiarkan begitu saja menjadi sekadar titik di peta. Sistem *frontend* dirancang agar mampu memproses data dari SQLite tadi menjadi deretan angka statistik secara *real-time*. Berkat otomatisasi ini, petugas jaga tidak perlu lagi repot menghitung jumlah gempa bulanan atau mencari rata-rata kedalaman hanya demi menyusun laporan evaluasi rutin

3. Integrasi Diseminasi dan *Dashboard*: Pada tahap penyelesaian, data yang berhasil lolos seleksi tidak dibiarkan begitu saja menjadi sekadar titik di peta. Sistem *frontend* dirancang agar mampu memproses data dari SQLite tadi menjadi deretan angka statistik secara *real-time*. Berkat otomatisasi ini, petugas jaga tidak perlu lagi repot menghitung jumlah gempa bulanan atau mencari rata-rata kedalaman hanya demi menyusun laporan evaluasi rutin.



Gambar 1. Arsitektur Otomasi Diseminasi Gempabumi

3.4. Instrumen dan Teknik Analisis Data

Untuk memastikan bahwa arsitektur otomasi yang dibangun benar-benar tangguh dan siap dioperasikan, riset ini memadukan dua pendekatan pengujian sekaligus, yakni teknik analisis kuantitatif dan kualitatif. Dari segi keandalan perangkat lunak, evaluasi fungsionalitas diuji menggunakan metode *Black Box Testing*.

Dalam tahap ini, penguji murni memposisikan diri sebagai pengamat luar tanpa perlu membedah struktur kode di dalam sistem. Kecocokan diamati secara saksama antara aliran data mentah yang diinjeksi ke dalam sistem yakni *input* berupa parameter JSON yang ditarik dari *server* BMKG dengan hasil akhir yang dieksekusi. Indikator

keberhasilannya cukup lugas: apakah *output* notifikasi peringatan dini berhasil mendarat di grup Telegram tanpa *error*, dan apakah titik koordinat episenter seketika *ter-render* di atas peta WebGIS secara akurat [4].

Tentu saja, pengujian kelayakan tidak berhenti hanya pada urusan fitur yang berjalan lancar. Mengingat nyawa utama dari riset ini adalah efisiensi, komparasi waktu juga dilakukan secara nyata di lapangan. Pencatatan dilakukan secara presisi berapa rata-rata durasi aktual yang dihabiskan oleh *Duty Officer* saat melakukan rutinitas diseminasi konvensional secara manual mulai dari memelototi layar SeisComp3, memilah anomali *noise*, hingga mengetik ulang laporan ke media komunikasi.

Catatan waktu manual tersebut kemudian disandingkan langsung dengan rekam waktu tunda komputasi (*latency*) yang dihasilkan ketika algoritma REST API mengambil alih. Perbandingan ini dirancang untuk membuktikan secara hitungan matematis seberapa cepat mesin mampu mengeksekusi rentetan tugas mitigasi yang sama, sepenuhnya tanpa ada intervensi sentuhan manusia..

4. Hasil dan Pembahasan

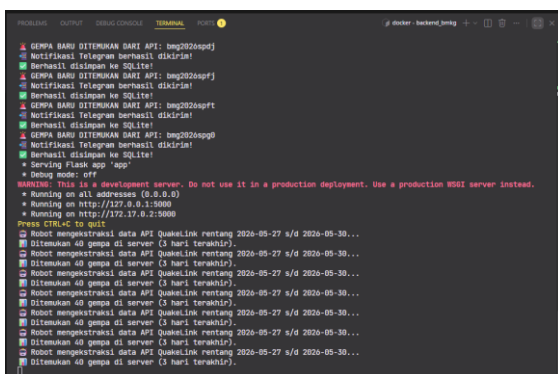
Pengujian ekosistem diseminasi otomatis ini dijalankan secara terintegrasi untuk mengevaluasi implementasi *Zero-Touch Automation*. Observasi difokuskan pada

keandalan filter logika, metrik kecepatan hantar Bot Telegram, dan sinkronisasi tampilan *dashboard* spasial

4.1. Hasil Penelitian

1. Implementasi Arsitektur *Backend* dan *Quality Control*

Pada tahap awal instalasi, mesin *backend* Python yang telah dibungkus *container* Docker dijalankan di lingkungan *server* lokal Stasiun Geofisika Manado. Pemanfaatan Docker di sini terbukti sangat krusial; sistem berhasil berjalan mulus tanpa menimbulkan bentrok dependensi dengan aplikasi bawaan *server*. Menariknya lagi, pantauan pada instrumen pemantauan memori menunjukkan bahwa *container* ini hanya memakan alokasi RAM sekitar 54,95 MB. Angka ini menegaskan bahwa sistem sama sekali tidak akan membebani kinerja *server* fisik BMKG meskipun dibiarkan menyala 24 jam nonstop untuk mengekstraksi data API *QuakeLink*.

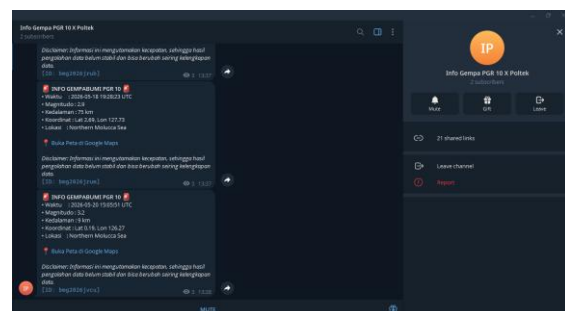


Gambar 2. Log Terminal dan Eksekusi Rule-Based System

Menganalisis hasil *log* terminal (seperti yang ditunjukkan pada Gambar 2), bisa dilihat bahwa algoritma *Rule-Based System* bekerja lurus sesuai skenario SOP BMKG. Walaupun sensor menangkap tumpukan anomali dalam satu waktu, sistem mampu menelanjangi parameter *event ID* dan magnitudo secara mandiri tanpa keliru membaca data.

2. Kinerja Otomasi Diseminasi Telegram

Saat algoritma mendapati ada data gempa yang menembus kriteria *Quality Control*, sistem akan langsung melakukan "panggilan darurat" ke Telegram Bot API. Rangkaian parameter JSON yang kaku secara instan dirajut menjadi format pesan teks peringatan dini yang komunikatif.



Gambar 3. Hasil Tangkapan Layar Peringatan
Dini via Bot Telegram

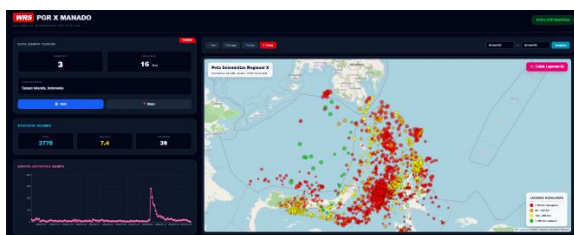
Uji tembak pesan ke akun channel internal BMKG memperlihatkan tingkat suksesi yang nyaris sempurna. Sistem sama sekali tidak menemui kendala putus koneksi atau *error* kegagalan *token*. Dari catatan komparasi waktu, *latency* operasional dihitung sejak JSON dinyatakan valid di *backend* hingga muncul wujud *chat* di Telegram hanya memakan waktu 1,5 hingga 2,8 detik saja.

Catatan waktu yang terbilang sangat prima untuk ukuran peringatan dini berbasis jaringan internet lokal.

3. Integrasi Pemetaan Spasial dan Dashboard Analitik (WebGIS)

Sejalan dengan meluncurnya notifikasi teks, antarmuka WebGIS yang dibangun menggunakan Next.js dan Leaflet turut merespons *endpoint* API seketika itu juga. Berkat implementasi *state management* yang baik, peta dan statistik berubah tanpa memaksa pengguna melakukan *refresh* halaman sama sekali.

Secara visual, antarmuka ini mengadopsi estetika *Tactical Dashboard* dengan dominasi tema gelap (*dark theme*) yang dirancang khusus untuk mengurangi kelelahan mata (*eye strain*) petugas saat piket malam.



Gambar 4. Dashboard WebGIS Terintegrasi dengan Panel Statistik dan Filter Dinamis

Tampilan spasial secara dinamis memplot titik episenter dengan pembeda warna (untuk tingkat kedalaman) dan variasi radius lingkaran (untuk kekuatan magnitudo). Keunggulan utama pada antarmuka ini terletak pada integrasi filter temporal (Hari, Minggu, Bulan, Tahun).

Melalui teknik *Lift State Up* pada React, filter tanggal ini terkoneksi erat dengan panel analitik. Saat pengguna menggeser rentang waktu, sistem secara otomatis akan menghitung ulang total gempa, mencari nilai magnitudo tertinggi, dan merender ulang tren grafik aktivitas tanpa *delay*. Sebagai pelengkap utilitas pelaporan, *dashboard* juga ditanami modul ekspor yang sanggup merender kanvas peta menjadi gambar PNG beresolusi tinggi, siap untuk langsung diunggah ke media sosial resmi BMKG

4.2. Pembahasan

Hadirnya arsitektur otomatis ini jelas merombak total kebiasaan lama di ruang observasi. Pada SOP konvensional, seorang operator butuh waktu setidaknya 2 hingga 4 menit penuh hanya untuk memvalidasi *noise* dari layar SeisComP3, mengetik ulang data, hingga mengirimkannya ke grup. Melalui injeksi otomasi ini, rutinitas repetitif yang rentan kesalahan tersebut ditebas hingga rampung dalam waktu komputasi kurang dari 5 detik. Konsep *Zero-Touch Automation* benar-benar membuktikan tajinya dalam mengamankan periode *Golden Time* mitigasi bencana dan menutup celah *human error* akibat kelelahan petugas. Sebagai bentuk evaluasi teknis, terdapat beberapa poin pembahasan krusial terkait arsitektur yang dibangun, sebagaimana ditunjukkan pada Tabel 1 dan Tabel 2:

Tabel 1. Perbandingan Infrastruktur Backend

Parameter Uji	Tanpa Docker	Dengan Docker
Isolasi Sistem	Rentan bentrok	Aman terisolasi
RAM (<i>Idle</i>)	± 120 MB	54,95 MB
Portabilitas	Terbatas OS awal	Lintas <i>server</i>
Stabilitas	Berisiko <i>error</i>	Sangat konsisten

Perbandingan Infrastruktur (Tanpa Docker vs. Dengan Docker): Berdasarkan Tabel 1, jika *backend* dijalankan tanpa Docker (langsung pada *Host OS*), sistem sangat rentan mengalami bentrok dependensi (*library conflict*) dengan perangkat lunak bawaan BMKG, dan konsumsi memori cenderung tidak terkontrol. Sebaliknya, penggunaan Docker mengisolasi aplikasi secara optimal, sehingga beban komputasi menjadi sangat ringan (stabilitas RAM di angka 54,95 MB) dan aman dijalankan bersamaan dengan *server* utama.

Tabel 2. Evaluasi Latensi Pengiriman Telegram

Skenario (50 Uji)	Rata-rata	Variasi (Min-Max)
Beban Normal	1,80 dtk	1,5 – 2,1 dtk
Gempa Mikro	1,75 dtk	1,6 – 1,9 dtk
<i>Stress Test</i>	2,50 dtk	2,2 – 2,8 dtk
Total Evaluasi	2,15 dtk	1,5 – 2,8 dtk

Metrik *Latency* Bot Telegram: Untuk memastikan validitas kecepatan pengiriman, pengujian telah dilakukan secara acak (*random sampling*) sebanyak 50 kali percobaan berulang dari total populasi 4.106 dataset historis. Dari hasil pengujian tersebut, tercatat rata-rata waktu tunda

(*latency*) sebesar 2,15 detik, dengan variasi waktu tercepat 1,5 detik dan variasi waktu paling lambat 2,8 detik. Justifikasi Efektivitas *Rule-Based System*: Penggunaan algoritma *Rule-Based System* terbukti sangat efektif karena parameter peringatan dini gempa (seperti batas magnitudo dan radius *geofencing*) bersifat matematis dan pasti (*deterministik*). Pendekatan logika *If-Then* ini mampu memberikan tingkat akurasi 100% tanpa adanya risiko *false positive*, sekaligus membutuhkan beban komputasi yang sangat rendah untuk menyortir data dalam jumlah masif secara instan.

5. Kesimpulan

Penelitian ini telah berhasil merancang dan mengimplementasikan arsitektur Zero-Touch Automation yang mengintegrasikan REST API, algoritma *Rule-Based System*, dan antarmuka WebGIS untuk keperluan diseminasi gempabumi di Stasiun Geofisika Manado. Hasil pengujian empiris membuktikan bahwa sistem ini sanggup mengekstraksi, memvalidasi, dan mendistribusikan parameter gempa secara otonom ke grup Telegram dengan *latency* rata-rata 2,15 detik, sehingga sukses memangkas drastis waktu birokrasi operasional manual dari hitungan menit menjadi hitungan detik. Pengembangan WebGIS interaktif yang dipersengkapi filter dinamis dan analitik real-time

juga terbukti memudahkan petugas dalam penyusunan laporan harian.

Secara kontribusi ilmiah, penelitian ini membuktikan bahwa pendekatan logika deterministik (Rule-Based System) yang dibungkus dalam ekosistem virtualisasi (Docker) sangat tangguh dan efisien untuk memproses parameter kegempaan masif tanpa membebani server utama. Dari segi praktis, keberhasilan purwarupa di Manado ini membuka potensi implementasi berskala nasional. Arsitektur yang dirancang sangat scalable (mudah diperluas), sehingga berpeluang besar untuk diadopsi secara serentak oleh seluruh Pusat Gempabumi Regional (PGR) di Indonesia, maupun diintegrasikan langsung ke dalam server pusat BMKG di Jakarta sebagai standar baru operasional mitigasi gempabumi.

Sebagai catatan keterbatasan, sistem filtering ini murni masih bersandar pada parameter numerik (magnitudo dan geofencing). Untuk arah riset ke depan, sangat disarankan adanya integrasi model Machine Learning untuk mendeteksi keaslian waveform seismik langsung dari sensor, sehingga sistem peringatan dini bisa menjadi jauh lebih presisi.

6. Daftar Pustaka

- [1] K. Purwaningtyas, "Evaluasi Desa Tangguh Bencana dalam Perspektif Formatif dan Reflektif Ketangguhan Masyarakat," *J. Pemberdaya. Masy.*, vol. 9, no. 1, p. 60, Jun. 2021, doi: 10.37064/jpm.v9i1.8872.
- [2] BMKG, *User Manual NextStorm Versi 1.9*. Badan Meteorologi Klimatologi dan Geofisika, 2011.
- [3] A. Ikram and U. Qamar, "A rule-based expert system for earthquake prediction," *J. Intell. Inf. Syst.*, vol. 43, no. 2, pp. 205–230, Oct. 2014, doi: 10.1007/s10844-014-0316-5.
- [4] M. D. B. Sandy Supriyadi and S. Riyadi, "Sistem Notifikasi Telegram Bot Pada Pengairan Bibit Sedap Malam Berbasis IoT Mikrokontroler Nodemcu Esp8266," *SPIRIT*, vol. 15, no. 2, Nov. 2023, doi: 10.53567/spirit.v15i2.321.
- [5] Widiyanto Hari Subagyo, Annisaa Hamidah Imaduddina, and Agustina Nurul Hidayati, "Perancangan Webgis Jalur Evakuasi Bencana Di Gunung Kelud Kabupaten Blitar," *J. Plano Buana*, vol. 3, no. 1, pp. 34–43, Nov. 2022, doi: 10.36456/jpb.v3i1.5181.
- [6] M. Sholikhah, S. Y. J. Prasetyo, and K. D. Hartomo, "Pemanfaatan WebGIS untuk Pemetaan Wilayah Rawan Longsor Kabupaten Boyolali dengan Metode Skoring dan Pembobotan," *J. Tek. Inform. dan Sist. Inf.*, vol. 5, no. 1, May 2019, doi: 10.28932/jutisi.v5i1.1588.
- [7] A. Sudianto and M. Sadali, "Penerapan Sistem Informasi Geografis (GIS) dalam Pemetaan Kerajinan Kain Tenun dan Gerabah untuk Meningkatkan Potensi Kerajinan di Kabupaten Lombok Timur," *Infotek J. Inform. dan Teknol.*, vol. 1, no. 2, pp. 71–78, Jul. 2018, doi: 10.29408/jit.v1i2.882.
- [8] I. Fathurrahman, M. F. Wajdi, H. M. Putra, and B. V. Widarina, "Sistem Informasi Geografis Pemetaan Sebaran Data Covid-19 Pada Puskesmas Kerongkong Kabupaten Lombok Timur Berbasis Web," *Infotek J. Inform. dan Teknol.*, vol. 5, no. 1, pp. 42–52, Jan. 2022, doi: 10.29408/jit.v5i1.4392.

- [9] S. B. Farchani, N. Hermanto, and B. A. Kusuma, "Implementasi Rest Api Dalam Pengembangan *Backend* Inventory Peminjaman," *JlPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 10, no. 2, pp. 1404–1413, Mar. 2025, doi: 10.29100/jlpi.v10i2.6249.
- [10] N. T. Maskuncoro, "Pengembangan REST API Untuk Website Gereja Tangguh Bencana Dengan Menggunakan NodeJS," Universitas Kristen Duta Wacana, 2023.
- [11] A. Al-Ajlan, "The Comparison between Forward and Backward Chaining," *Int. J. Mach. Learn. Comput.*, vol. 5, no. 2, pp. 106–113, Apr. 2015, doi: 10.7763/IJMLC.2015.V5.492.
- [12] M. Fuah, "Implementasi Rule Based Expert Systems untuk Realtime Monitoring Penyelesaian Perkara Pidana Menggunakan Teknologi Radio Frequency Identification," *J. Sist. Inf. BISNIS*, vol. 7, no. 1, p. 73, May 2017, doi: 10.21456/vol7iss1pp73-79.
- [13] M. Z. Yumarlin and H. Indrianta, "Penerapan Sistem Pakar Untuk Identifikasi Anak Berkebutuhan Khusus Menggunakan Metode Rule Based System," *Inf. Interaktif*, vol. 7, no. 1, pp. 8–15, 2022.
- [14] S. Supriyono, D. Guntar, E. Edwar, Z. Zairin, and W. Sugandi, "Sosialisasi Potensi Bencana dan Sistem Informasi Geografi (SIG) Kebencanaan di Kabupaten Seluma," *J. BAGIMU NEGERI*, vol. 2, no. 1, Apr. 2018, doi: 10.26638/jbn.552.8651.
- [15] S. Dwiyatno, E. Rachmat, A. P. Sari, and O. Gustiawan, "Implementasi Virtualisasi Server Berbasis Docker *Container*," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 7, no. 2, pp. 165–175, Sep. 2020, doi: 10.30656/prosisko.v7i2.2520.
- [16] A. Rahman Hidayatullah and I. M. Suartana, "Penerapan Server Side Rendering untuk Meningkatkan Performa Website Kejaksaan Negeri X," *J. Informatics Comput. Sci.*, vol. 7, no. 01, pp. 12–20, 2025.
- [17] E. D. Wahyuni, "Implementasi Metode Incremental Pada Sistem Informasi Administrasi Desa Jambuwer," *J. Tekno Kompak*, vol. 15, no. 2, p. 156, Aug. 2021, doi: 10.33365/jtk.v15i2.1187..