Journal of Applied Informatics and Computing (JAIC)
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861

Optimalisasi Monitoring Infrastruktur Jaringan Berbasis Otomatisasi PRTG Terintegrasi WhatsApp: Studi Kasus PT. Solnet Indonesia

M.Kelvin Prayoga [1]*, Antoni Haikal [2]**
* Teknik Informatika, Politeknik Negeri Batam
** Rekayasa Keamanan Siber, Politeknik Negeri Batam
kelvinprayoga46@gmail.com [1], antoni@polibatam.ac.id [2]

Article Info

Article history:

Received …
Revised …
Accepted …

Keyword:

Pemantauan jaringan, PRTG, WhatsApp Gateway, Fonnte, otomatisasi notifikasi, ISP.

ABSTRACT

Infrastruktur jaringan merupakan komponen vital bagi operasional perusahaan penyedia jasa internet (Internet Service Provider/ISP), sehingga keterlambatan deteksi gangguan berdampak langsung pada kualitas layanan dan tingkat kepuasan pelanggan. PT. Solnet Indonesia saat ini telah memanfaatkan Paessler Router Traffic Grapher (PRTG) sebagai perangkat utama pemantauan jaringan, namun mekanisme notifikasi yang masih bergantung pada media surel menyebabkan keterlambatan respons dengan median sekitar 7 menit pada jam kerja dan mencapai puluhan menit hingga beberapa jam pada jam non-kerja. Penelitian ini bertujuan mengoptimalkan sistem pemantauan infrastruktur jaringan melalui otomatisasi PRTG yang diintegrasikan dengan aplikasi WhatsApp menggunakan layanan Fonnte WhatsApp Gateway sebagai kanal notifikasi real-time. Metode pengembangan yang digunakan adalah Prototyping, dengan arsitektur berlapis yang menghubungkan PRTG Core Server sebagai monitoring layer, skrip PowerShell (prtg_notifier.ps1) sebagai integration layer, serta Fonnte API sebagai delivery layer menuju perangkat WhatsApp administrator. Sistem diuji pada 100 host pelanggan dengan total 200 sensor aktif (Ping ICMP dan SNMP Traffic) melalui tiga kategori pengujian, yaitu pengujian fungsional, performa, dan keandalan, serta dilengkapi pengujian skenario kegagalan layanan gateway. Pengukuran menggabungkan observasi natural terhadap notifikasi produksi (n = 307-472 sampel), pengukuran terkontrol pada komponen dekomposisi (n = 30), serta wawancara terstruktur dengan tim NOC sebagai pembanding sistem lama. Indikator keberhasilan yang ditetapkan meliputi success rate notifikasi ≥ 99%, response time end-to-end ≤ 30 detik, service availability ≥ 99%, serta message loss rate ≤ 0,5%. Hasil penelitian menunjukkan sistem usulan mencapai response time end-to-end 6,0 detik, success rate 100%, service availability 99,93%, dan message loss 0%. Berdasarkan wawancara terstruktur dengan tim NOC, sistem baru memangkas waktu respons sekitar 70 kali pada jam kerja dan lebih dari 423 kali pada jam non-kerja dibandingkan mekanisme notifikasi konvensional berbasis surel, sekaligus memberikan kontribusi praktis bagi perusahaan ISP dalam membangun sistem pemantauan jaringan yang lebih responsif dan efisien

I. PENDAHULUAN

Di era transformasi digital yang berlangsung pesat, infrastruktur jaringan telah menjadi tulang punggung operasional berbagai sektor strategis, mulai dari perbankan, pemerintahan, pendidikan, hingga industri telekomunikasi. Gangguan jaringan dalam skala kecil sekalipun berpotensi mengganggu produktivitas dan menurunkan kepercayaan pelanggan. Berdasarkan laporan *Cisco Annual Internet Report*, sebanyak 94% organisasi menyatakan keandalan jaringan sebagai faktor krusial bagi kesuksesan bisnis, dengan 42% di antaranya mengalami kerugian finansial lebih dari USD 300.000 setiap tahun akibat *downtime* jaringan [1]. Hal tersebut menjadikan kebutuhan akan sistem pemantauan jaringan yang cerdas, responsif, dan otomatis menjadi semakin mendesak [2]. Pada konteks nasional, jumlah pengguna internet di Indonesia tercatat mencapai 215 juta jiwa pada awal 2024 atau meningkat 7,7% dibanding tahun

e-ISSN: 2548-6861

sebelumnya, sehingga memberikan tekanan tinggi terhadap kapasitas dan kestabilan infrastruktur jaringan nasional [3].

PT. Solnet Indonesia sebagai salah satu penyedia jasa internet (*Internet Service Provider*/ISP) menghadapi tantangan signifikan dalam menjaga kualitas dan kestabilan layanan di tengah meningkatnya permintaan pelanggan. Sistem pemantauan konvensional yang diterapkan saat ini bergantung pada notifikasi internal dan surel sebagai kanal utama eskalasi gangguan, sehingga terbukti lambat dalam merespons insiden. Belum adanya integrasi dengan platform komunikasi instan seperti WhatsApp menyebabkan keterlambatan yang, berdasarkan wawancara dengan tim NOC, memiliki median sekitar 7 menit pada jam kerja dan meningkat tajam hingga puluhan menit bahkan beberapa jam pada jam non-kerja sebelum administrator menyadari adanya gangguan. Proses eskalasi yang belum sepenuhnya otomatis juga mengakibatkan tim teknis kerap terlambat memperoleh informasi krusial [4]. Dalam konteks ini, *Paessler Router Traffic Grapher* (PRTG) hadir sebagai solusi pemantauan otomatis, namun penerapannya yang belum terintegrasi dengan saluran komunikasi populer menjadi hambatan nyata dalam akselerasi respons teknis.

Sejumlah penelitian terdahulu telah membuktikan efektivitas PRTG dalam pemantauan jaringan. Penelitian dalam [2] menunjukkan bahwa kombinasi PRTG dengan *firewall* Sophos mampu meningkatkan keamanan sekaligus mempercepat respons insiden pada jaringan universitas, sementara Daud *et al.* [3] mengkaji penerapan PRTG pada infrastruktur perusahaan ViTrox dan menemukan bahwa sistem tersebut mempercepat deteksi *bottleneck* hingga 40%. Dalam konteks Indonesia, Arvianto *et al.* [5] mengembangkan aplikasi pemantauan jaringan berbasis PRTG yang memungkinkan pengawasan lalu lintas secara real-time, Putri [6] berhasil mengidentifikasi *bottleneck* jaringan FKIP UNS melalui visualisasi PRTG, sementara studi terkini oleh Rachmawati *et al.* [14] menerapkan PRTG di lingkungan akademik UPN "Veteran" Jawa Timur dengan analisis sensor *ping*, latensi, dan *packet loss* yang menunjukkan variasi performa jaringan signifikan pada jam sibuk.

Pada ranah integrasi platform komunikasi instan dengan sistem pemantauan jaringan, berbagai pendekatan telah diteliti. Hidayah *et al.* [7] mendemonstrasikan efektivitas integrasi *WhatsApp Gateway* pada sistem pemantauan *Weigh-In-Motion* berbasis optoelektronik dalam mempercepat notifikasi ke teknisi lapangan, sementara Defani dan Wijayanto [10] mengintegrasikan *Uptime Kuma* dengan sistem router MikroTik menggunakan Prometheus dan Grafana untuk visualisasi pemantauan terpusat. Pada platform monitoring berskala enterprise, Fachrurrozi *et al.* [13] merancang sistem pemantauan berbasis LibreNMS yang diintegrasikan secara simultan dengan Line, Telegram, dan surel di IT Telkom Jakarta untuk mempercepat respons administrator, sedangkan Husna dan Rosyani [15] mengombinasikan Zabbix, Grafana, dan Telegram guna memantau jaringan dan server secara visual sekaligus mengirim peringatan instan. Lebih lanjut, Mohd Fuzi *et al.* [16] memperluas pendekatan ini ke ranah keamanan jaringan dengan mengintegrasikan Zabbix dan Telegram untuk mendeteksi serangan *ping flooding* dan SYN *flooding* secara *real-time*.

Berdasarkan tinjauan tersebut, masih sangat terbatas penelitian kuantitatif yang secara sistematis mengukur efektivitas integrasi PRTG dengan WhatsApp, khususnya dalam konteks operasional ISP di Indonesia. Mayoritas studi terdahulu memang telah membuktikan efektivitas integrasi *messaging app* dengan perangkat pemantauan jaringan, namun fokus utamanya masih pada Zabbix [15], [16] atau LibreNMS [13] dengan Telegram sebagai kanal notifikasi. Sementara itu, integrasi PRTG dengan WhatsApp yang memiliki *open rate* jauh lebih tinggi (98% dalam tiga menit pertama) dan penetrasi pengguna lebih dominan di Indonesia masih sangat terbatas eksplorasinya. Beberapa studi yang tersedia bahkan hanya menempatkan surel sebagai sarana komunikasi pasca-insiden, yaitu pendekatan yang secara faktual kurang efektif dalam komunikasi cepat dan *mobile* [9], serta jarang memanfaatkan indikator kuantitatif seperti waktu rata-rata respons teknis sebelum dan sesudah otomatisasi. Studi lain seperti pada [8] juga masih menggunakan pendekatan kualitatif atau studi kasus terbatas. Kesenjangan inilah yang menjadi celah penelitian (*research gap*) yang ingin diisi melalui studi ini, yaitu pembuktian berbasis data terhadap efektivitas integrasi sistem pemantauan PRTG dengan WhatsApp dalam mempercepat deteksi dan penyelesaian gangguan jaringan.

Sejalan dengan permasalahan tersebut, penelitian ini bertujuan untuk: (1) menganalisis pengaruh sistem pemantauan otomatis berbasis PRTG yang terintegrasi dengan WhatsApp terhadap efisiensi pemantauan jaringan di PT. Solnet Indonesia; (2) mengukur kontribusi notifikasi WhatsApp dalam mempercepat respons teknisi terhadap gangguan jaringan; dan (3) mengevaluasi efektivitas sistem pemantauan otomatis dalam mendukung operasional infrastruktur jaringan secara *real-time*. Adapun lingkup penelitian dibatasi pada infrastruktur jaringan internal PT. Solnet Indonesia, dengan PRTG Network Monitor sebagai perangkat pemantauan dan layanan Fonnte API [11] sebagai jembatan *WhatsApp Gateway*; pengukuran difokuskan pada *response time*, jumlah notifikasi terkirim, dan *recovery time*, tanpa mencakup aspek keamanan data WhatsApp maupun konfigurasi mendalam pada tingkat sistem operasi atau perangkat keras jaringan.

Hasil penelitian ini diharapkan memberikan kontribusi akademik berupa pengayaan khazanah keilmuan di bidang rekayasa keamanan siber dan manajemen jaringan, khususnya pada pemanfaatan otomatisasi pemantauan dan integrasi komunikasi instan sebagai solusi peningkatan efisiensi operasional. Secara praktis, temuan penelitian dapat menjadi dasar rekomendasi bagi perusahaan ISP, terutama PT. Solnet Indonesia, dalam mengembangkan sistem pemantauan

jaringan yang lebih responsif dan adaptif terhadap kebutuhan komunikasi modern berbasis perangkat *mobile*.

II. METODE

Penelitian ini menggunakan metode *Research and Development* (R&D) dengan model pengembangan *Prototyping* dan pendekatan evaluasi kuantitatif. Bagian ini menguraikan studi kasus beserta identifikasi permasalahan, perangkat dan teknologi yang digunakan, rancangan sistem dan alur kerja, serta tahapan pengembangan dan skenario pengujian.

A. Studi Kasus dan Identifikasi Masalah

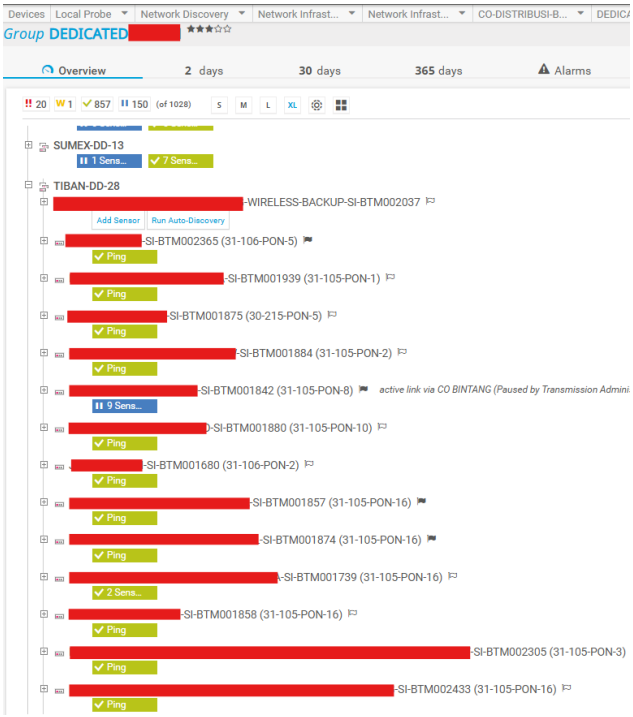
PT. Solnet Indonesia merupakan perusahaan penyedia jasa internet (ISP) yang mengelola infrastruktur jaringan kompleks meliputi server, router, switch, firewall, dan perangkat jaringan lainnya yang tersebar di beberapa lokasi pelanggan skala bisnis maupun rumah tangga. Perusahaan telah mengimplementasikan PRTG Network Monitor sebagai perangkat utama pemantauan dengan konfigurasi *threshold* dan sistem peringatan yang mengirimkan notifikasi melalui surel. Namun, observasi terhadap operasional sistem berjalan mengungkap tujuh permasalahan signifikan sebagaimana dirangkum pada Tabel I.

TABEL I
IDENTIFIKASI PERMASALAHAN SISTEM BERJALAN

NO	PERMASALAHAN	DAMPAK OPERASIONAL
1	Keterlambatan Waktu Respons	Surel mengalami penundaan dengan median sekitar 7 menit pada jam kerja, meningkat hingga puluhan menit bahkan beberapa jam pada jam non-kerja; berpotensi melanggar Service Level Agreement (SLA)
2	Beban Surel Berlebih	50-100 surel notifikasi per hari; peringatan kritis tenggelam pada kotak masuk yang padat
3	Mobilitas Terbatas	Surel kurang ramah perangkat bergerak; administrator terikat pada laptop untuk akses dashboard
4	Ketiadaan Notifikasi Aktif	Tidak ada peringatan instan ke perangkat administrator; pendekatan reaktif alih-alih proaktif
5	Komunikasi Tidak Efisien	Koordinasi antar-administrator masih manual melalui telepon dan WhatsApp pribadi, tanpa sumber informasi tunggal
6	Minimnya Konteks Peringatan	Format surel terbatas tanpa visual; administrator selalu perlu membuka dashboard untuk detail gangguan
7	Kendala Jejak Audit	Sulit melacak waktu respons dari rekaman surel; data tidak akurat untuk perhitungan SLA

Tujuh permasalahan tersebut secara akumulatif berdampak pada peningkatan *Mean Time To Detect* (MTTD) gangguan, potensi pelanggaran SLA, serta menurunnya efektivitas pendekatan pemantauan dari proaktif menjadi reaktif. Penelitian ini memanfaatkan infrastruktur operasional PT. Solnet Indonesia sebagai lingkungan pengujian, dengan total 100 *host* pelanggan yang dipantau secara kontinu sebagai populasi pengujian sistem.

Objek pemantauan dalam penelitian ini adalah 100 host pelanggan yang seluruhnya merupakan perangkat Customer Premises Equipment (CPE) berupa router pada sisi pelanggan layanan internet dedicated korporat, yang terhubung ke jaringan PT. Solnet Indonesia melalui segmen akses fiber optik berbasis Passive Optical Network (PON). Host tersebut dikelola dalam satu grup pemantauan pelanggan dedicated pada PRTG (grup DEDICATED-BATAM), dengan konvensi penamaan perangkat yang memuat identitas pelanggan, kode layanan (SI-BTM), serta identitas port PON asal pada perangkat OLT (contoh: 31-105-PON-16 dan 30-215-PON-3) sehingga posisi logis setiap host pada topologi akses dapat tertelusur. Pada setiap host, sensor Ping ICMP dipasang terhadap alamat IP manajemen CPE untuk memantau ketersediaan, sedangkan sensor SNMP Traffic 64-bit dipasang pada antarmuka uplink fiber optik perangkat (antarmuka ether1-FO) untuk memantau lalu lintas bandwidth, sehingga total 200 sensor aktif beroperasi pada lingkup penelitian. Gambaran struktur device tree grup pemantauan disajikan pada Gambar 1.



Gambar 1. Struktur Device Tree Group Pemantauan, identitas pelanggan disamarkan

B. Perangkat dan Teknologi

PRTG Network Monitor digunakan sebagai perangkat utama pemantauan. PRTG mendukung pemantauan parameter seperti uptime, lalu lintas data, bandwidth, hingga performa server, dengan dukungan protokol SNMP, WMI,

e-ISSN: 2548-6861

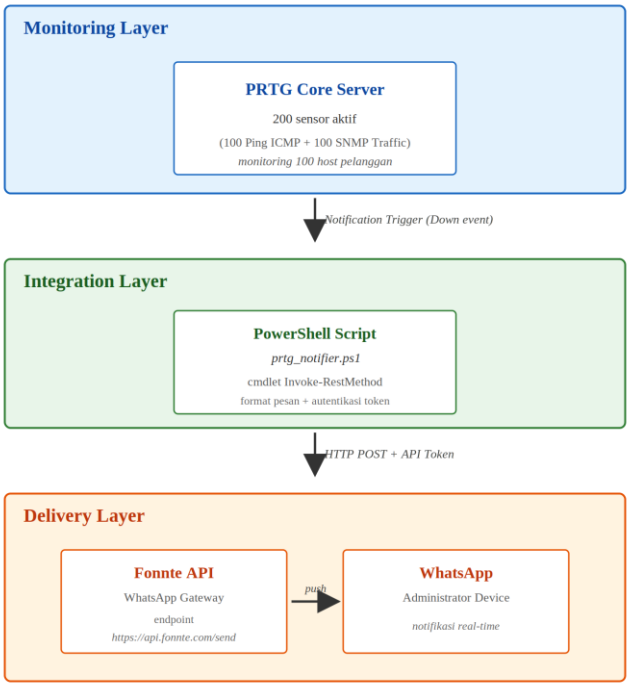
dan HTTP serta mekanisme notifikasi berbasis trigger [2], [3]. Kanal notifikasi yang tersedia mencakup surel, API, dan webhook, sehingga memungkinkan integrasi dengan layanan *WhatsApp Gateway* pihak ketiga.

Sebagai jembatan menuju aplikasi WhatsApp administrator, penelitian ini menggunakan layanan Fonnte API [11], yaitu penyedia *WhatsApp Gateway* berbasis API yang banyak digunakan di Indonesia karena tidak memerlukan proses verifikasi resmi Meta Business yang panjang sebagaimana WhatsApp Business API resmi. Pemilihan Fonnte didasarkan pada tiga pertimbangan utama: (1) proses registrasi dan aktivasi yang cepat tanpa verifikasi bisnis resmi; (2) struktur API yang sederhana sehingga dapat dipanggil langsung dari skrip PowerShell melalui *cmdlet Invoke-RestMethod* tanpa pustaka tambahan; serta (3) biaya berlangganan yang relatif terjangkau dibandingkan WhatsApp Business API resmi sehingga sesuai untuk lingkungan operasional ISP skala menengah. Berdasarkan konfirmasi resmi dari pihak penyedia layanan, Fonnte tidak menerapkan pembatasan laju permintaan (hard rate limit) pada API pengiriman pesan; pemrosesan sepenuhnya berbasis kuota berlangganan, dan apabila kuota habis, API mengembalikan respons status false dengan keterangan insufficient quota yang dapat dideteksi secara programatik oleh skrip integrasi. Adapun risiko pemblokiran nomor oleh pihak WhatsApp merupakan risiko intrinsik penggunaan gateway tidak resmi yang tidak dapat dieliminasi sepenuhnya, melainkan hanya dapat dimitigasi melalui praktik penggunaan nomor yang telah lama aktif, penyimpanan kontak penerima, interaksi dua arah dengan penerima, serta pemberian jeda antar pengiriman [18]. Karakteristik operasional sistem usulan, penerima tunggal yang tetap dan tersimpan (tim NOC), interaksi dua arah yang rutin, serta volume pengiriman moderat berbasis event, sejalan dengan profil penggunaan berisiko rendah tersebut. Rincian seluruh perangkat dan instrumen yang digunakan dalam penelitian disajikan pada Tabel II.

TABEL II
PERANGKAT DAN INSTRUMEN PENELITIAN

KATEGORI	KOMPONEN DAN SPESIFIKASI
Hardware	PRTG Core Server (Windows Server 2019, 16 GB RAM, 4 vCPU); 2-3 smartphone administrator (Android & iOS) untuk uji cross-platform
Monitoring Software	PRTG Network Monitor v25.x (lisensi 500 sensor); PRTG API Browser; PRTG System Administrator
Development	PowerShell 5.1 / 7.x dengan cmdlet Invoke-RestMethod; Postman untuk uji manual endpoint Fonnte
Messaging Gateway	Akun Fonnte WhatsApp Gateway (paket Reguler) + API token; device WhatsApp ter-pairing dengan akun Fonnte
Pencatatan Data	PRTG System Logs; PowerShell Start-Transcript; berkas CSV/Excel untuk rekapitulasi
Pengukuran Waktu	NTP-synchronized timestamp pada PRTG Server dan smartphone administrator untuk perhitungan response time akurat

Arsitektur sistem secara keseluruhan dirancang dalam tiga lapis fungsional sebagaimana ditunjukkan pada Gambar 2. Lapis pertama (*monitoring layer*) berperan mendeteksi gangguan melalui sensor *Ping* ICMP dan SNMP *Traffic* pada PRTG *Core Server*. Lapis kedua (*integration layer*) yang dijalankan oleh skrip PowerShell *prtg_notifier.ps1* menerima parameter dari PRTG *Notification Trigger* dan menyusun struktur pesan beserta autentikasi token. Lapis ketiga (*delivery layer*) meneruskan pesan tersebut melalui Fonnte API menuju aplikasi WhatsApp administrator secara *real-time*.

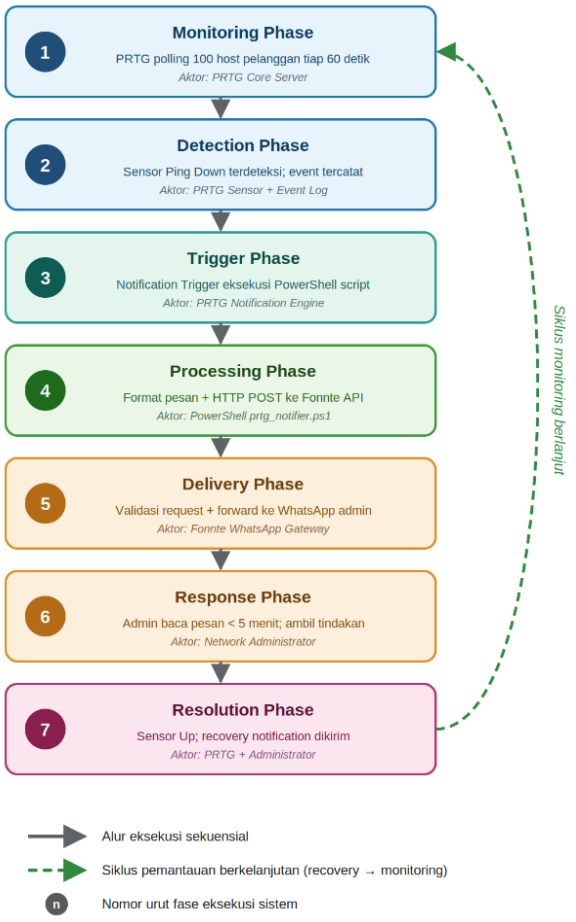


Gambar 2. Arsitektur sistem integrasi PRTG dan WhatsApp Gateway

C. Rancangan Sistem dan Alur Kerja

Topologi pemantauan dirancang dengan menempatkan dua jenis sensor pada setiap *host* pelanggan, yaitu sensor *Ping* ICMP untuk memantau ketersediaan *host* dan sensor SNMP *Traffic* untuk memantau lalu lintas *bandwidth* pada antarmuka perangkat pelanggan. Dengan konfigurasi ini, total terdapat 200 sensor aktif yang dipantau secara kontinu oleh PRTG *Core Server*. Fokus *trigger* notifikasi ditetapkan pada sensor *Ping* karena status *Down* pada sensor tersebut merepresentasikan kondisi gangguan paling kritikal yang memerlukan respons segera, sementara data SNMP *Traffic* diperlakukan sebagai data pendukung analisis performa tanpa men-*trigger* notifikasi WhatsApp secara langsung pada lingkup pengujian.

Sistem usulan beroperasi berdasarkan pendekatan *event-driven architecture* yang dirinci ke dalam tujuh fase eksekusi sebagaimana divisualisasikan pada Gambar 3. PRTG *Core Server* melakukan *polling* terhadap 100 *host* pelanggan setiap 60 detik (*Monitoring Phase*). Apabila sensor *Ping* tidak merespons ICMP, status sensor berubah menjadi *Down* dan event tercatat pada log database (*Detection Phase*). PRTG *Notification Engine* kemudian mengeksekusi skrip PowerShell dengan parameter berupa nama *device*, jenis sensor, status, pesan, dan *timestamp* (*Trigger Phase*). Skrip memformat pesan sesuai template dan mengirim HTTP POST ke *endpoint* Fonnte dengan API *token* (*Processing Phase*). Fonnte memvalidasi *request* dan meneruskannya sebagai *push notification* ke perangkat administrator (*Delivery Phase*). Administrator membaca notifikasi dalam waktu kurang dari 5 menit dan mengambil tindakan (*Response Phase*), hingga gangguan teratasi dan sistem mengirim notifikasi pemulihan melalui alur yang sama (*Resolution Phase*).



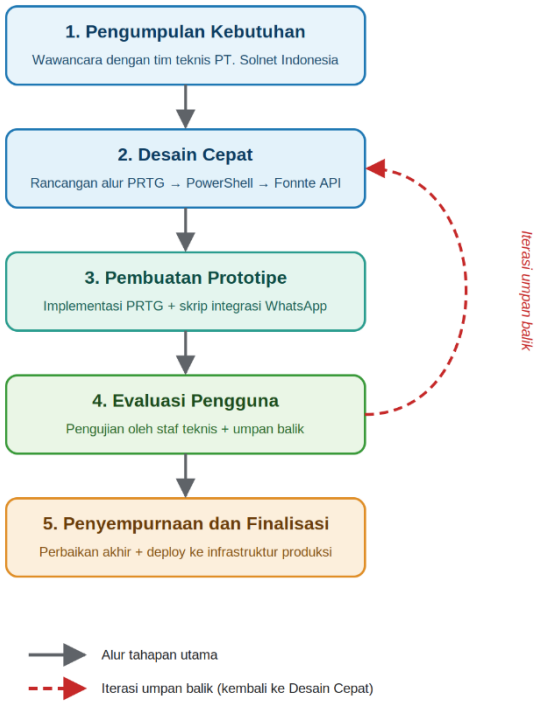
Gambar 3. Flowchart alur kerja sistem usulan dalam tujuh fase eksekusi

Pada lapisan *trigger* PRTG, *notification trigger* dikonfigurasi dengan parameter latensi penundaan sebesar 300 detik (5 menit) sebelum memanggil skrip *prtg_notifier.ps1*. Konfigurasi tersebut berfungsi sebagai mekanisme *debounce* yang menyaring gangguan transien (*transient outage*), yaitu kondisi sensor menunjukkan status Down dalam durasi singkat akibat *packet loss* sesaat, *micro-outage* pada perangkat jaringan, atau fluktuasi propagasi pada jalur *fiber optic*. Tanpa penyaringan ini, setiap fluktuasi singkat yang umum terjadi pada infrastruktur ISP akan memicu notifikasi WhatsApp secara langsung, sehingga menimbulkan fenomena *alert fatigue* pada tim NOC dan menurunkan responsivitas teknisi terhadap insiden yang benar-benar memerlukan eskalasi. Ambang batas 300 detik ditetapkan berdasarkan karakteristik operasional di PT. Solnet Indonesia, dimana gangguan dengan durasi di bawah lima menit umumnya pulih secara otomatis melalui mekanisme *failover* maupun *re-routing* tanpa memerlukan intervensi manual. Mekanisme *debounce* dengan parameter identik (latensi 300 detik pada lapisan *notification trigger* PRTG) juga diterapkan pada sistem notifikasi email yang dikomparasi, sehingga kedua sistem dievaluasi pada kondisi penyaringan *false positive* yang konsisten. Konsekuensinya, seluruh pengukuran *response time* dan *throughput* pada bab Hasil dan Pembahasan mengisolasi performa intrinsik sistem notifikasi, yaitu dihitung sejak skrip eksekusi berhasil dipanggil (pasca-*debounce*) hingga pesan diterima pada perangkat penerima, sehingga tidak mencakup periode penundaan 300 detik tersebut

D. Tahapan Pengembangan dan Skenario Pengujian

Metode *Research and Development* (R&D) dengan model pengembangan *Prototyping* dan pendekatan evaluasi kuantitatif dipilih karena memungkinkan pembuatan model awal sistem yang dapat diuji dan disempurnakan berdasarkan umpan balik pengguna [12], sehingga sangat sesuai untuk sistem yang membutuhkan integrasi antarmuka dan komunikasi cepat. Tahapan yang ditempuh divisualisasikan pada Gambar 4 dan dijabarkan sebagai berikut

e-ISSN: 2548-6861



Gambar 4. Alur metode Prototyping dengan lima tahap pengembangan

1) Pengumpulan Kebutuhan: Tahap ini mengidentifikasi kebutuhan sistem dari sisi teknis maupun pengguna melalui wawancara dengan tim teknis PT. Solnet Indonesia, dengan fokus pada pemetaan kebutuhan pemantauan dan jalur eskalasi notifikasi.

2) Desain Cepat: Rancangan awal sistem integrasi PRTG dengan *WhatsApp Gateway* disusun dengan menetapkan alur data dari sensor jaringan menuju layanan notifikasi melalui skrip PowerShell yang dipicu oleh *Notification Trigger* PRTG.

3) Pembuatan Prototipe: Implementasi sistem pemantauan menggunakan PRTG dilakukan beriringan dengan pembuatan koneksi ke WhatsApp melalui Fonnte API. Prototipe diuji menggunakan data nyata dari jaringan PT. Solnet Indonesia.

4) Evaluasi Pengguna: Pengujian dilakukan oleh staf teknis untuk mengevaluasi kecepatan notifikasi dan efektivitas pengiriman pesan. Umpan balik dicatat secara sistematis sebagai dasar perbaikan pada iterasi berikutnya.

5) Penyempurnaan dan Finalisasi: Berdasarkan hasil evaluasi, sistem diperbaiki dan difinalisasi untuk digunakan secara penuh dalam infrastruktur PT. Solnet Indonesia.

Skenario pengujian gangguan disimulasikan melalui tiga pendekatan untuk menjamin variasi kondisi: (1) pemutusan konektivitas pada beberapa *host* pelanggan secara terjadwal di lingkungan *testbed*; (2) penggunaan fitur *Pause/Resume* sensor PRTG untuk memicu transisi status terkontrol; dan (3) penonaktifan *interface* perangkat tujuan untuk menghasilkan kondisi *true Down*. Pendekatan pengukuran dibedakan menurut karakteristik parameter. Parameter fungsional dan performa (Gambar 7 dan 8) diukur melalui observasi natural terhadap seluruh notifikasi produksi yang terjadi selama periode pengamatan, menghasilkan populasi 307-472 sampel sehingga bersifat sensus, bukan sekadar sampel terbatas. Parameter dekomposisi t_1 dan t_4 (Gambar 9) yang memerlukan kondisi terkendali diukur dengan minimal 30 sampel sesuai kaidah Central Limit Theorem. Parameter keandalan (Gambar 11) diperoleh dari pengamatan 28 hari melalui pemindaian log sistem. Khusus parameter MTTR, insiden diklasifikasikan terlebih dahulu menurut mekanisme kegagalannya berdasarkan pola log Windows Event Viewer: insiden terisolasi, yaitu kegagalan layanan tunggal yang pulih tanpa kegagalan susulan; dan insiden beruntun (cascade incident), yaitu rangkaian dua atau lebih crash layanan berurutan (Event ID 7034) sebelum pemulihan penuh. Karena kedua kelas merepresentasikan mekanisme kegagalan yang berbeda — kegagalan terisolasi tertangani pemulihan otomatis bawaan, sedangkan kegagalan beruntun memerlukan penanganan akar masalah, MTTR dilaporkan terpisah untuk masing-masing kelas tanpa menghilangkan data apa pun dari pelaporan. Adapun data pembanding sistem lama berbasis surel dikumpulkan melalui wawancara terstruktur dengan tim NOC menggunakan Critical Incident Technique [17]. Sebagai pelengkap, dilakukan pula pengujian terhadap empat skenario kegagalan pada sisi layanan *gateway* — autentikasi gagal (token tidak valid), kuota layanan habis, permintaan dari alamat IP di luar daftar putih, dan pemblokiran nomor — menggunakan akun uji terpisah agar tidak mengganggu layanan produksi, dengan hasil dirangkum pada Tabel IV.

TABEL III
PARAMETER PENGUJIAN DAN INDIKATOR KEBERHASILAN

KATEGORI	PARAMETER	FORMULA / METODE PENGUKURAN	TARGET
Fungsional	Success Rate	(Notifikasi terkirim / Total trigger) × 100%	≥ 99%
	Accuracy Alert	(Alert akurat / Total alert) × 100%	100%
	Format Compliance	Validasi regex terhadap template pesan	100%
	Trigger Coverage	(Sensor Down ber-notif / Total Sensor Down) × 100%	100%
Performa	Response Time	t1 (detection) + t2 (script) + t3 (API) + t4 (WA delivery)	≤ 30 detik avg; ≤ 60 detik worst
	Throughput	Jumlah notifikasi terkirim per menit	≥ 30 notif/menit
	Fonnte API Latency	Round-trip HTTP POST (Measure-Command)	≤ 1.000 ms avg; ≤ 3.000 ms p95
	PRTG Resource	Beban tambahan CPU & Memory saat eksekusi script	CPU ≤ 30%; Memory ≤ 30%

KATEGORI	PARAMETER	FORMULA / METODE PENGUKURAN	TARGET
Keandalan	Service Availability	(Uptime efektif / Total waktu, di luar maintenance) × 100%	≥ 99,3%
	MTBF	Mean Time Between Failures (rata-rata jam antar insiden terkonsolidasi)	≥ 168 jam (≈7 hari)
	MTTR	Mean Time To Recovery (rata-rata waktu pemulihan)	≤ 25 menit (1.500 detik)
	Message Loss	(Pesan gagal kirim / Total pesan) × 100%	≤ 0,5%
	Script Reliability	(Eksekusi PowerShell tanpa error / Total eksekusi) × 100%	≥ 99%

III. HASIL DAN PEMBAHASAN

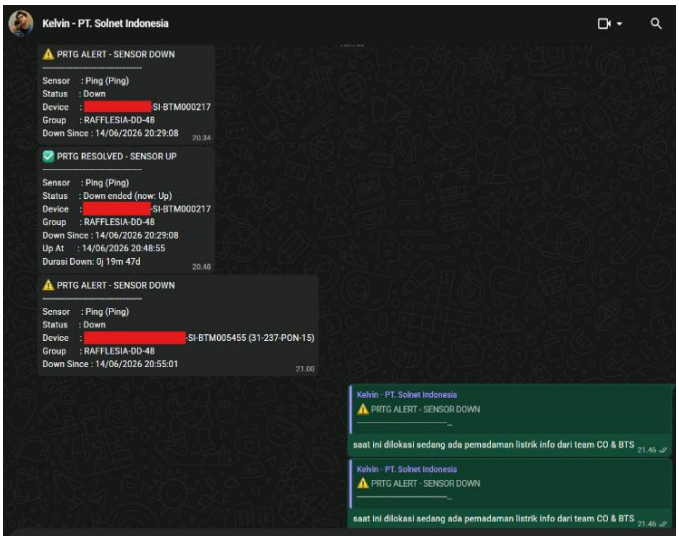
Bagian ini menguraikan hasil implementasi sistem dan pengujian yang dilaksanakan pada infrastruktur operasional PT. Solnet Indonesia. Pengujian fungsional dan performa memanfaatkan observasi natural terhadap insiden produksi nyata, mencakup 307 notifikasi yang terkirim selama insiden PoP Outage pada 14 Juni 2026 serta 472 eksekusi skrip yang terekam pada periode 14-16 Juni 2026. Pengujian keandalan dilaksanakan melalui pemindaian log Windows Event Viewer selama 28 hari observasi (20 Mei - 16 Juni 2026). Diskusi pada bagian akhir membandingkan capaian sistem usulan dengan target indikator keberhasilan, dengan data pembanding sistem lama yang diperoleh melalui wawancara terstruktur tim NOC, serta dengan studi terdahulu yang relevan.

A. Hasil Implementasi Sistem

Sistem berhasil diimplementasikan pada lingkungan operasional PT. Solnet Indonesia dengan total 200 sensor aktif yang memantau 100 *host* pelanggan. Tampilan PRTG *System Status Log* yang merekam kronologi event sensor selama pemantauan disajikan pada Gambar 5. Log tersebut menampilkan enam kolom informasi yang terdiri atas *DateTime*, *Parent* (nama *device* pelanggan atau *group* jaringan), *Type* (jenis sensor), *Object* (subtipe sensor atau *interface*), *Status*, dan *Message* yang memuat detail teknis setiap kejadian. Lingkungan pengujian mencakup beragam pelanggan korporat dan segmen jaringan internal PT. Solnet Indonesia dengan kombinasi sensor *Ping* ICMP dan SNMP *Traffic* 64-bit yang masing-masing merekam nilai latensi dalam satuan *msec* dan *throughput bandwidth* dalam satuan *kbit/s*.

Gambar 5. Tampilan PRTG System Status Log yang mencatat kronologi event sensor dan aktivasi State Trigger pada pemantauan jaringan pelanggan PT. Solnet Indonesia

Adapun tampilan notifikasi yang diterima administrator melalui aplikasi WhatsApp disajikan pada Gambar 6. Format pesan memuat informasi terstruktur mencakup nama *device*, jenis sensor, status, *group device,* dan durasi down, sehingga administrator dapat melakukan analisis awal tanpa harus membuka dasbor PRTG terlebih dahulu. Notifikasi pemulihan (*Recovery*) otomatis dikirim ketika sensor kembali ke status *Up*, lengkap dengan informasi total durasi gangguan.



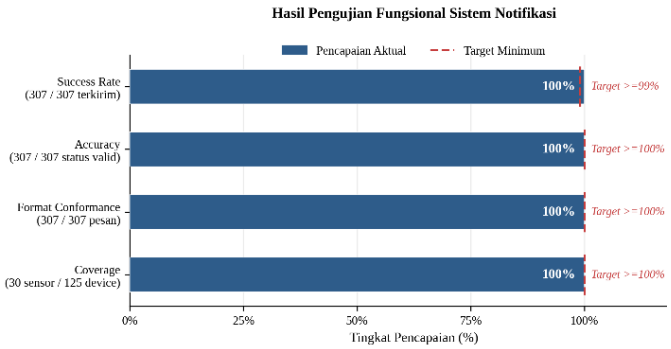
Gambar 6. Tampilan notifikasi WhatsApp berisi peringatan Down dan Recovery yang diterima administrator

B. Hasil Pengujian Fungsional

Pengujian fungsional dilaksanakan melalui observasi natural terhadap 307 notifikasi yang terkirim selama insiden

e-ISSN: 2548-6861

produksi nyata. Hasil pengukuran keempat parameter fungsional disajikan pada Gambar 7. Seluruh parameter fungsional terpenuhi: *Success Rate* mencapai 100%, demikian pula *Accuracy Alert*, *Format Compliance*, dan *Trigger Coverage* masing-masing mencapai 100%. Capaian ini secara langsung menjawab tujuh permasalahan yang teridentifikasi pada Tabel I, terutama keterlambatan respons dan ketiadaan notifikasi aktif.

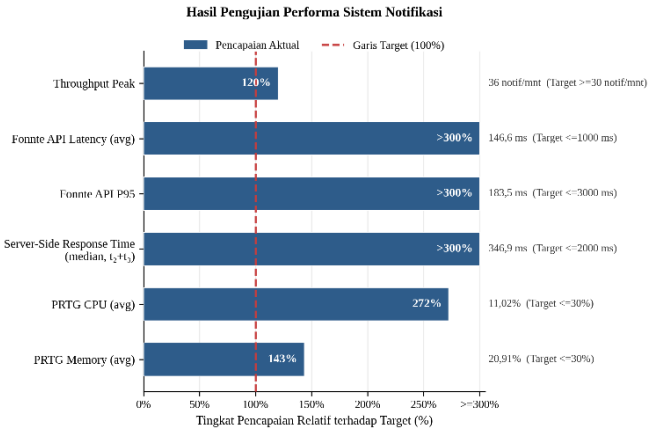


Gambar 7. Hasil Pengujian Fungsional

Berdasarkan Gambar 7, tidak ditemukan kegagalan pengiriman sepanjang periode observasi: seluruh 307 notifikasi terkirim dengan respons *status* berhasil dari Fonnte API, tanpa penolakan maupun kesalahan skrip. Validasi cakupan menunjukkan notifikasi berhasil dipicu dari 30 jenis sensor berbeda yang tersebar pada 125 perangkat unik, memperlihatkan bahwa sistem mampu menangani keragaman tipe sensor PRTG dalam kondisi produksi nyata. Capaian *Success Rate* 100% ini diperoleh dari insiden produksi alami, sehingga memiliki validitas ekologis yang lebih tinggi dibandingkan pengujian sintetik terkendali.

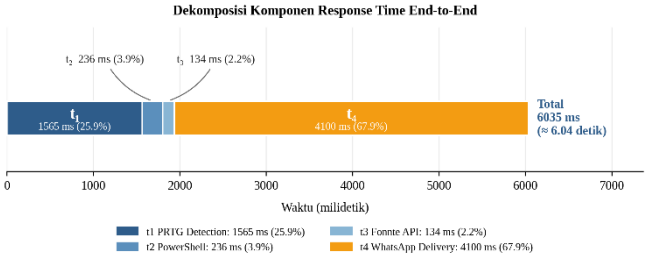
C. Hasil Pengujian Performa

Hasil pengukuran lima parameter performa selama pengujian disajikan pada Gambar 8. *Response Time* sisi server (t_2+t_3) dengan median sebesar 0,347 detik telah jauh melampaui target ≤ 2 detik. Nilai median dipilih sebagai representasi tipikal karena distribusi cenderung condong ke kanan; persentil ke-95 sebesar 2,44 detik tetap berada dalam ambang ≤ 3 detik. Komponen response time end-to-end secara lengkap, termasuk deteksi sensor PRTG dan propagasi WhatsApp, diuraikan terpisah pada Gambar 9. *Throughput* puncak 36 notifikasi per menit (jendela bergulir 60 detik) melampaui target 30 notif/menit, sementara latensi Fonnte API rata-rata 146,63 ms berada jauh dalam ambang ≤ 1.000 ms. Beban *resource* pada proses PRTG tercatat 11,02% CPU dan 20,91% Memori (setara 13.696 MB), keduanya berada di bawah ambang ≤ 30% yang ditetapkan mengikuti rekomendasi vendor Paessler untuk instalasi PRTG skala enterprise.



Gambar 8. Hasil Pengujian Performa

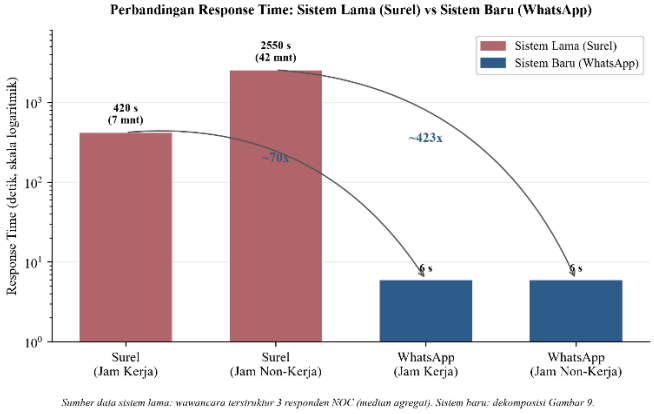
Untuk mengidentifikasi komponen yang paling berpengaruh terhadap *Response Time*, dilakukan dekomposisi waktu antar lapis sistem sebagaimana disajikan pada Gambar 9. Hasil dekomposisi mengungkap bahwa tahap WhatsApp Delivery (t_4) menyumbang kontribusi terbesar dengan rata-rata 4.100 ms (67,9%), yang merupakan karakteristik intrinsik platform pesan *over-the-top* dan tidak dapat dioptimalkan dari sisi server. Tahap PRTG Sensor Detection (t_1) berkontribusi 1.565 ms (25,9%), sedangkan PowerShell Script Execution (t_2) dan Fonnte API Response (t_3) masing-masing hanya 236 ms (3,9%) dan 134 ms (2,2%). Dengan demikian, komponen yang sepenuhnya dapat dikendalikan oleh implementasi (t_1+t_2+t_3) hanya berjumlah 1.935 ms, menunjukkan bahwa beban pemrosesan sistem usulan telah sangat efisien dan total *response time* end-to-end 6.035 ms (sekitar 6 detik) sebagian besar ditentukan oleh latensi propagasi jaringan WhatsApp yang berada di luar kendali sistem.



Gambar 9. Dekomposisi Komponen Response Time End-to-End

Temuan ini menunjukkan bahwa sebagian besar response time end-to-end ditentukan oleh tahap propagasi WhatsApp (t_4) yang berada di luar kendali sistem, sehingga ruang optimasi dari sisi implementasi relatif terbatas. Di antara komponen yang masih dapat dikendalikan, interval polling PRTG (t_1) merupakan kontributor terbesar, sementara optimasi pada skrip integrasi maupun pemilihan layanan gateway hanya akan memberikan dampak marginal karena keduanya telah sangat efisien. Perbandingan kinerja *Response Time* antara sistem lama berbasis surel dan sistem usulan berbasis WhatsApp divisualisasikan pada Gambar 10.

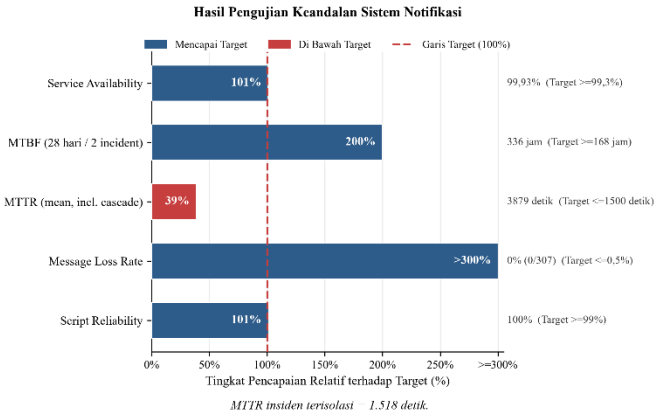
Berdasarkan wawancara terstruktur dengan tim NOC, response time sistem lama memiliki median sekitar 7 menit pada jam kerja dan 42,5 menit pada jam non-kerja, sehingga sistem usulan yang menghasilkan notifikasi end-to-end sekitar 6 detik tercatat sekitar 70 kali lebih cepat pada jam kerja dan lebih dari 423 kali lebih cepat pada jam non-kerja. Perbedaan yang jauh lebih besar pada jam non-kerja terjadi karena sistem usulan mengirimkan *push notification* seketika tanpa bergantung pada aktivitas administrator membuka surel secara manual.



Gambar 10. Perbandingan response time sistem lama (surel) dan sistem usulan (WhatsApp) pada skala logaritmik

D. Hasil Pengujian Keandalan

Pengujian keandalan dilaksanakan melalui pemindaian log Windows Event Viewer selama 28 hari observasi (20 Mei - 16 Juni 2026). Hasil pengukuran kelima parameter keandalan disajikan pada Gambar 11. *Service Availability* tercatat 99,93%, jauh melampaui target ≥ 99,3%. Nilai ini mengikuti standar SLA industri yang mengecualikan *scheduled maintenance window* berupa proses backup harian PRTGProbeService; apabila seluruh *downtime* terjadwal turut diperhitungkan, *Total Uptime* tetap mencapai 99,79%. MTBF sebesar 336 jam menandakan rata-rata interval antar insiden terkonsolidasi mencapai sekitar 14 hari, dua kali lipat dari target ≥ 168 jam. Tingkat kehilangan pesan (*Message Loss Rate*) tercatat 0% (0 dari 307 pesan) dan *Script Reliability* sempurna pada 100% (472 dari 472 eksekusi). Capaian script reliability tersebut turut didukung oleh mekanisme inspeksi status respons serta penanganan kesalahan (try-catch) pada skrip integrasi yang mencatat status respons Fonnte termasuk kondisi insufficient quota, ke dalam berkas log kesalahan terpisah, sehingga kegagalan pengiriman yang bersumber dari sisi gateway dapat terdeteksi dan terdokumentasi secara otomatis tanpa intervensi manual. Adapun MTTR rata-rata sebesar 3.879 detik (sekitar 64,7 menit) tidak memenuhi target ≤ 1.500 detik; analisis penyebabnya diuraikan sebagai studi kasus pada bagian berikut.



Gambar 11. Hasil Pengujian Keandalan

Kegagalan pencapaian target MTTR memerlukan analisis lebih lanjut sebagai studi kasus. Nilai MTTR rata-rata yang tinggi disebabkan oleh satu *cascade incident* pada 9 Juni 2026, ketika PRTGAppServer mengalami *crash* beruntun yang terdeteksi melalui Windows Event ID 7034, dengan waktu pemulihan penuh mencapai 6.240 detik (sekitar 1 jam 44 menit). Insiden ini menarik nilai rata-rata keseluruhan ke atas. Sebaliknya, insiden tunggal pada 10 Juni 2026 berhasil dipulihkan secara otomatis oleh mekanisme *Windows Service Recovery* dalam 1.518 detik (sekitar 25 menit), atau hanya 18 detik di atas target. Sesuai klasifikasi mekanisme kegagalan yang ditetapkan pada Bab II.D, kedua insiden pada periode observasi terbagi ke dalam dua kelas: insiden 9 Juni 2026 memenuhi kriteria *cascade incident* (rangkaian *crash* beruntun Event ID 7034; pemulihan penuh 6.240 detik), sedangkan insiden 10 Juni 2026 merupakan insiden terisolasi yang pulih otomatis dalam 1.518 detik. Dengan demikian MTTR keseluruhan tercatat 3.879 detik, sementara MTTR kelas insiden terisolasi sebesar 1.518 detik secara praktis telah mencapai target ≤ 1.500 detik. Pelaporan terpisah ini menunjukkan bahwa target MTTR realistis bagi kelas kegagalan terisolasi yang tertangani mekanisme pemulihan otomatis, sedangkan kelas kegagalan beruntun memerlukan mitigasi tambahan. Temuan ini menunjukkan bahwa mekanisme pemulihan otomatis bawaan sudah memadai untuk kegagalan terisolasi, namun kegagalan beruntun memerlukan penanganan pada akar masalah. Rekomendasi mitigasi untuk pengembangan lanjutan mencakup penerapan strategi *restart* berbasis *exponential backoff* serta pemantauan kesehatan layanan berbasis ambang penggunaan memori untuk mendeteksi degradasi sebelum *crash* terjadi.

E. Pengujian Skenario Kegagalan Fonnte Gateway

Menindaklanjuti ketergantungan sistem terhadap layanan pihak ketiga, dilakukan pengujian terhadap empat skenario kegagalan pada sisi Fonnte Gateway menggunakan akun uji terpisah (paket gratis) agar tidak mengganggu layanan produksi, dengan satu pengujian dasar (baseline) bertoken valid sebagai pembanding. Temuan utama pengujian adalah bahwa seluruh kondisi kegagalan dikembalikan API dengan

e-ISSN: 2548-6861

kode HTTP 200, sehingga kegagalan tidak terindikasi pada lapisan protokol HTTP melainkan pada isi respons berupa field status bernilai false disertai field reason; dengan demikian, deteksi kegagalan pada skrip integrasi dilakukan melalui inspeksi field status respons, sedangkan blok try-catch menangani kelas kegagalan berbeda berupa gangguan konektivitas HTTP (misalnya timeout atau layanan tidak terjangkau). Pada skenario token tidak valid, API mengembalikan status false dengan keterangan invalid token; pada skenario kuota habis, keterangan insufficient quota dikembalikan sesuai konfirmasi resmi penyedia layanan; dan pada skenario permintaan dari alamat IP di luar daftar putih (IP whitelist), keterangan unauthorized token usage dikembalikan, yang sekaligus menunjukkan bahwa fitur daftar putih IP berfungsi sebagai kontrol mitigasi penyalahgunaan token dari luar server yang sah. Ketiga kondisi tersebut terdeteksi oleh skrip melalui inspeksi respons, tercatat pada berkas log kesalahan, dan menghasilkan exit code 1 sehingga turut terekam pada log PRTG. Skenario pemblokiran nomor tidak disimulasikan secara nyata karena menyengajakan kondisi tersebut berpotensi melanggar kebijakan platform; berdasarkan konfirmasi resmi penyedia layanan, pemulihan pemblokiran umumnya berlangsung sekitar enam jam dan kanal notifikasi surel eksisting yang berjalan paralel berfungsi sebagai kanal cadangan. Rangkuman hasil disajikan pada Tabel IV.

TABEL IV
HASIL PENGUJIAN SKENARIO KEGAGALAN GATEWAY

SKENARIO	METODE UJI	RESPONS API (HTTP 200)	DETEKSI	MITIGASI
Baseline (valid)	Simulasi (akun uji)	status: true, "message in queue"	-	-
Token tidak valid	Simulasi (akun uji)	status: false, "invalid token"	Inspeksi status → log kesalahan; exit code 1	Perbaikan / rotasi token
Kuota habis	Simulasi (akun uji)	status: false, "insufficient quota"	Inspeksi status → log kesalahan; exit code 1	Monitoring kuota berkaa; fallback surel
IP di luar whitelist	Simulasi (akun uji)	status: false, "unauthorized token usage"	Inspeksi status → log kesalahan; exit code 1	Whitelist IP server PRTG (kontrol keamanan)
Nomor terblokir	Analitik (konfirmasi penyedia)	Kanal WA terhenti (±6 jam)	Kegagalan kirim beruntun pada log	Fallback surel; migrasi API resmi (saran)

F. Pembahasan Komprehensif

Sintesis hasil dari ketiga kategori pengujian menunjukkan bahwa sistem usulan berhasil memenuhi 12 dari 13 parameter indikator keberhasilan yang ditetapkan pada Tabel III. Satu-satunya parameter yang tidak tercapai, yaitu MTTR rata-rata, telah dibahas sebagai studi kasus pada Sub-bab sebelumnya dan terbukti dipengaruhi oleh satu *cascade incident* yang bersifat *outlier*. Perbandingan ringkas antara sistem lama berbasis surel dan sistem baru berbasis WhatsApp pada enam aspek operasional yang paling berdampak disajikan pada Tabel V. Komparasi pada Tabel V dilaksanakan dalam kondisi penyaringan yang setara, dimana kedua sistem (notifikasi berbasis email pada konfigurasi lama maupun notifikasi WhatsApp pada konfigurasi baru) sama-sama menerapkan *debounce* 300 detik pada lapisan *notification trigger* PRTG. Dengan demikian, perbedaan *response time* yang terukur sepenuhnya merepresentasikan perbedaan performa intrinsik kanal pengiriman pesan, bukan artefak konfigurasi *trigger*.

TABEL V
PERBANDINGAN KINERJA SISTEM LAMA DAN SISTEM USULAN

ASPEK OPERASIONAL	SISTEM LAMA (SUREL)	SISTEM BARU (WHATSAPP)	PENINGKATAN
Response Time (jam kerja)	~7 menit (rentang 4-10)	~6 detik	~70× cepat
Response Time (jam non-kerja)	~42,5 menit (rentang 12,5-90)	~6 detik	~423× cepat
Push Notification	Tidak tersedia	Tersedia	✓
Open Rate (30 menit pertama)	~50% (rentang 30-70)	~100%	~2×
Mobilitas Akses	Penuh (smartphone)	Penuh (smartphone)	✗
Audit Trail	Manual	Otomatis (log terpusat)	✓

Capaian *Response Time* end-to-end sebesar 6 detik lebih unggul dibandingkan studi Fachrurrozi *et al.* [13] yang melaporkan latensi notifikasi 22-35 detik pada sistem LibreNMS dengan kombinasi Line, Telegram, dan surel. Keunggulan ini diduga berasal dari arsitektur tiga lapis sederhana yang mengeliminasi kebutuhan *message queue* terpisah dan layanan *gateway* perantara. Demikian pula, *open rate* notifikasi berbasis platform pesan instan yang menurut literatur dapat mencapai sekitar 98% pembukaan secara signifikan lebih tinggi dibandingkan rerata pembukaan surel teknis; pada tim NOC PT. Solnet, hasil wawancara menunjukkan rerata pembukaan surel dalam 30 menit pertama hanya sekitar 50%, sebagaimana pola serupa juga dilaporkan oleh Hidayah *et al.* [7] dalam konteks notifikasi sistem *Weigh-In-Motion*.

Data pembanding sistem lama pada Tabel V diperoleh melalui wawancara terstruktur dengan tiga anggota tim NOC

PT. Solnet Indonesia yang mewakili tiga tingkat peran, yaitu Supervisor, Engineer Senior, dan Engineer Junior. Wawancara menggunakan *Critical Incident Technique* [17], yang meminta setiap responden mengingat insiden konkret beserta estimasi waktunya, sehingga data yang diperoleh bersumber dari pengalaman aktual dan bukan sekadar opini umum. Nilai *response time* sistem lama dilaporkan dalam bentuk median agregat ketiga responden beserta rentang antar responden untuk menjaga transparansi variasi. Variasi yang cukup lebar pada jam non-kerja (rentang 12,5-90 menit) mencerminkan ketergantungan tinggi mekanisme surel terhadap pola individual masing-masing administrator dalam memeriksa kotak masuk, sebuah karakteristik yang justru dieliminasi oleh mekanisme *push notification* pada sistem usulan.

Ditinjau dari perspektif rekayasa keamanan siber, implementasi sistem memiliki tiga permukaan risiko yang perlu dikemukakan secara konkret. Pertama, keamanan kredensial: pada implementasi awal, token API tersimpan dalam bentuk teks polos pada badan skrip di server PRTG, sehingga kompromi terhadap server atau repositori skrip memungkinkan penyalahgunaan token untuk pengiriman pesan atas nama akun perusahaan; sebagai tindak lanjut, token telah dipindahkan ke environment variable dengan pembatasan hak akses pada tingkat sistem, disertai rotasi token, dan penyimpanan pada fasilitas kredensial terlindungi seperti Windows Credential Manager direkomendasikan sebagai penguatan lanjutan. Kedua, risiko kebocoran isi notifikasi: pesan memuat informasi operasional sensitif (nama perangkat pelanggan, grup jaringan, durasi gangguan) yang dapat bocor melalui perangkat penerima yang tidak terkunci, keanggotaan grup yang tidak terkontrol, maupun sisi infrastruktur *gateway* pihak ketiga; mitigasinya meliputi minimalisasi data pada templat pesan serta pengendalian keanggotaan grup penerima secara berkala. Ketiga, proteksi kanal komunikasi: meskipun WhatsApp menerapkan enkripsi ujung-ke-ujung antar pengguna dan komunikasi skrip menuju API telah menegakkan HTTPS dengan TLS 1.2, penggunaan *gateway* tidak resmi menempatkan penyedia layanan sebagai perantara yang memproses isi pesan pada sisi servernya, sehingga kerahasiaan turut bergantung pada praktik keamanan penyedia. Evaluasi keamanan menyeluruh, termasuk pengujian penetrasi alur integrasi dan migrasi ke WhatsApp Business API resmi, direkomendasikan sebagai penelitian lanjutan.

Meskipun mayoritas parameter terpenuhi, penelitian ini memiliki sejumlah keterbatasan yang perlu diperhatikan. Pertama, sistem masih bergantung pada layanan pihak ketiga (Fonnte WhatsApp Gateway) yang ketersediaannya tidak dijamin oleh SLA formal, dan penggunaannya sebagai *gateway* tidak resmi mengandung risiko pemblokiran nomor oleh pihak WhatsApp yang berdasarkan konfirmasi penyedia layanan hanya dapat dimitigasi, tidak dieliminasi. Kedua, lingkup pengujian dibatasi pada 100 *host* pelanggan dan sensor *Ping* ICMP, sehingga generalisasi ke skala lebih besar (1.000+ *host*) atau jenis sensor kompleks (CPU, Memory, atau *Service Monitoring*) perlu pengujian lebih lanjut. Ketiga, faktor risiko keamanan informasi terkait penggunaan WhatsApp sebagai kanal notifikasi operasional belum dievaluasi secara mendalam. Keempat, data pembanding sistem lama bersumber dari estimasi *self-report* tiga responden NOC, bukan dari pencatatan log historis terstruktur yang tidak tersedia pada infrastruktur sebelumnya, sehingga angka komparatif sebaiknya dimaknai sebagai indikasi besaran ordo perbedaan dan bukan nilai presisi mutlak. Sebagai penegasan atas keterbatasan-keterbatasan tersebut, seluruh temuan kuantitatif diperoleh dari satu lingkungan operasional dengan karakteristik spesifik, satu perusahaan ISP skala menengah, satu topologi jaringan, 100 *host*, dan dominasi sensor Ping ICMP, sehingga capaian yang dilaporkan hendaknya dimaknai sebagai bukti kelayakan (*proof of feasibility*) pada konteks serupa, bukan klaim yang berlaku umum bagi seluruh lingkungan jaringan; generalisasi ke skala, topologi, atau jenis sensor berbeda memerlukan validasi tersendiri. Penelitian lanjutan disarankan mencakup ekspansi skala pengujian, integrasi dengan sumber data WhatsApp Business API resmi serta evaluasi aspek keamanan keseluruhan komunikasi end-to-end, validasi kuantitatif sistem lama melalui pencatatan otomatis apabila memungkinkan.

IV. KESIMPULAN

Penelitian ini berhasil merancang, mengimplementasikan, dan mengevaluasi sistem notifikasi pemantauan jaringan berbasis otomatisasi PRTG yang terintegrasi dengan WhatsApp melalui layanan Fonnte WhatsApp Gateway pada infrastruktur operasional PT. Solnet Indonesia. Berdasarkan pengujian terhadap aspek fungsional, performa, dan keandalan, sistem usulan memenuhi 12 dari 13 indikator keberhasilan yang ditetapkan. Pada aspek fungsional, sistem mencapai success rate 100% dari 307 notifikasi produksi nyata yang diobservasi, dengan akurasi status, kesesuaian format, dan cakupan pemicuan yang seluruhnya sempurna. Pada aspek performa, sistem mencatat median response time sisi server sebesar 0,35 detik, latensi Fonnte API rata-rata 146,63 ms, throughput puncak 36 notifikasi per menit, serta penggunaan sumber daya proses PRTG sebesar 11,02% CPU dan 20,91% memori, yang seluruhnya berada dalam ambang target.

Analisis response time end-to-end sebesar 6,04 detik menunjukkan bahwa 67,9% waktu dipengaruhi oleh tahap propagasi WhatsApp yang merupakan karakteristik intrinsik platform, sedangkan komponen yang dapat dikendalikan oleh implementasi sistem hanya sebesar 1,9 detik, sehingga membuktikan efisiensi pemrosesan sistem. Pada aspek keandalan, sistem mencatat service availability 99,93%, MTBF 336 jam, message loss rate 0%, dan script reliability 100% selama 28 hari observasi. Satu-satunya parameter yang belum memenuhi target adalah MTTR rata-rata sebesar 3.879

detik yang dipengaruhi oleh satu insiden kelas *cascade*; adapun MTTR kelas insiden terisolasi sebesar 1.518 detik secara praktis telah mencapai target, yang menunjukkan bahwa mekanisme pemulihan otomatis bawaan telah memadai untuk menangani kegagalan terisolasi.

Berdasarkan perbandingan dengan sistem lama berbasis surel yang datanya diperoleh melalui wawancara terstruktur dengan tim NOC, sistem usulan terbukti mampu memangkas waktu respons secara signifikan, yaitu sekitar 70 kali lebih cepat pada jam kerja dan lebih dari 423 kali lebih cepat pada jam non-kerja. Hasil tersebut menunjukkan bahwa integrasi PRTG dengan WhatsApp secara kuantitatif mampu meningkatkan responsivitas pemantauan infrastruktur jaringan serta mempercepat deteksi dan penyelesaian gangguan. Meskipun demikian, penelitian ini masih memiliki keterbatasan, antara lain ketergantungan pada layanan pihak ketiga, pengujian yang terbatas pada sensor Ping ICMP, belum dievaluasinya aspek keamanan komunikasi secara mendalam, serta penggunaan data pembanding sistem lama yang berasal dari estimasi self-report. Oleh karena itu, penelitian selanjutnya disarankan untuk memperluas skala pengujian pada jumlah host dan tipe sensor yang lebih beragam, mengintegrasikan WhatsApp Business API resmi, mengevaluasi keamanan komunikasi end-to-end, menerapkan strategi exponential backoff untuk menekan MTTR pada kegagalan beruntun, serta melakukan validasi kuantitatif sistem lama melalui pencatatan log otomatis.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada PT Solnet Indonesia atas dukungan fasilitas dan lingkungan pengujian yang diberikan dalam pelaksanaan penelitian ini. Ucapan terima kasih juga penulis sampaikan kepada orang tua, dosen pembimbing, Senior Technical Manager, SPV Monitoring, rekan-rekan tim monitoring, seluruh pihak yang telah memberikan bantuan, arahan, dan dukungan, baik secara langsung maupun tidak langsung, sehingga penelitian ini dapat terselesaikan dengan baik. Terakhir, kepada seseorang yang pernah menjadi bagian dari perjalanan hidup penulis, meskipun namanya kini tak lagi dapat disebutkan. Walau pada akhirnya ia tidak berhasil menunggu proses kelulusan penulis hingga akhir, terima kasih atas dukungan, kebersamaan, dan pelajaran berharga yang pernah diberikan. Kepergian Anda membuat penulis memahami bahwa setiap orang memiliki masanya, dan setiap masa memiliki orangnya. Luka yang tertinggal bukanlah akhir dari segalanya, melainkan awal dari proses bertumbuh menjadi pribadi yang lebih baik.

DAFTAR PUSTAKA

[1] Cisco Systems, "Cisco annual internet report (highlights dashboard)," Cisco Systems, San Jose, CA, USA, 2025. [Daring]. Tersedia: https://www.cisco.com/site/us/en/solutions/annual-internet-report/index.html

[2] A. Fathima dan G. S. Devi, "Enhancing university network management and security: A real-time monitoring, visualization & cyber attack detection approach using Paessler PRTG and Sophos firewall," *Int. J. Syst. Assur. Eng. Manag.*, hlm. 1–17, Agu. 2024, doi: 10.1007/s13198-024-02448-y.

[3] A. Y. Daud, J. X. Tan, dan W. J. Ooi, "Paessler router traffic grapher (PRTG) network monitoring: An implementation process in Vitrox," *J. Digit. Syst. Dev.*, vol. 2, no. 2, hlm. 1–11, 2024, doi: 10.32890/jdsd2024.2.2.1.

[4] A. Barbadekar, A. Amrutkar, dan A. Khan, "Real-time anomaly detection system for network administrators," dalam *Computing and Machine Learning* (Lecture Notes in Networks and Systems, vol. 1108), J. C. Bansal, S. Borah, S. Hussain, dan S. Salhi, Ed. Singapore: Springer, 2024, hlm. 315–326, doi: 10.1007/978-981-97-6588-1_24.

[5] D. Arvianto, M. Kamisutara, dan W. A. Kristiana, "Development of Paessler router traffic grapher network monitoring application," *IJEEIT: Int. J. Electr. Eng. Inf. Technol.*, vol. 7, no. 2, hlm. 80–87, Des. 2024, doi: 10.29138/ijeeit.v7i2.2912.

[6] P. M. Putri, "Penerapan PRTG traffic grapher untuk monitoring & evaluasi jaringan di lingkungan FKIP UNS," Tugas Akhir, Jurusan Teknik Informatika, Universitas Islam Indonesia, Yogyakarta, Indonesia, 2017. [Daring]. Tersedia: https://dspace.uii.ac.id/handle/123456789/32377

[7] F. N. Hidayah, A. Jamaldi, dan S. A. Kristiawan, "Real-time weigh in motion (WIM) monitoring system based on optoelectronics and WhatsApp gateway," *Indones. J. Appl. Phys.*, vol. 15, no. 1, hlm. 156–167, 2025, doi: 10.13057/ijap.v15i1.92986.

[8] C. Matowe, "Using deep learning to classify community network traffic," Tesis M.Sc., Dept. Computer Science, Faculty of Science, University of Cape Town, Cape Town, South Africa, 2022. [Daring]. Tersedia: https://open.uct.ac.za/handle/11427/37511

[9] M. Al Homsi, "Network Guard: A Python-powered network monitoring solution," *Higher Education Diploma Thesis*, Dept. of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden, 2024. [Daring]. Tersedia: https://www.divaportal.org/smash/get/diva2:1888393/FULLTEXT01.pdf

[10] R. Defani dan D. Wijayanto, "Integration of Uptime Kuma application monitoring and MikroTik router system using Prometheus and Grafana," *SISTEMASI: J. Sist. Inf.*, vol. 14, no. 5, hlm. 2395–2409, Sep. 2025, doi: 10.32520/stmsi.v14i5.5326.

[11] Fonnte, "Fonnte WhatsApp API documentation," 2024. [Daring]. Tersedia: https://docs.fonnte.com

[12] A. Gaeta, V. Loia, A. Lorusso, F. Orciuoli, dan A. Pascuzzo, "Towards a LLM-based intelligent system for detecting propaganda within textual content," *Comput. Electr. Eng.*, vol. 128, Art. no. 110765, 2025, doi: 10.1016/j.compeleceng.2025.110765.

[13] N. R. Fachrurrozi, A. A. Wirabudi, dan S. A. Rozano, "Design of network monitoring system based on LibreNMS using Line Notify, Telegram, and Email notification," *SINERGI*, vol. 27, no. 1, hlm. 111–122, Feb. 2023, doi: 10.22441/sinergi.2023.1.013.

[14] L. N. Rachmawati, M. E. A. Octaviana, dan Agussalim, "Monitoring jaringan menggunakan PRTG (Studi kasus: Fakultas Ekonomi Bisnis UPN 'Veteran' Jatim)," *J. Ilm. Teknol. Inf. dan Robot.*, vol. 6, no. 1, hlm. 44–53, 2024, doi: 10.33005/jifti.v6i1.149.

[15] M. A. Husna dan P. Rosyani, "Implementasi sistem monitoring jaringan dan server menggunakan Zabbix yang terintegrasi dengan Grafana dan Telegram," *JURIKOM: J. Ris. Komput.*, vol. 8, no. 6, hlm. 247–255, Des. 2021, doi: 10.30865/jurikom.v8i6.3631.

[16] M. F. Mohd Fuzi, N. F. Mohammad Ashraf, dan M. N. F. Jamaluddin, "Integrated network monitoring using Zabbix with push notification via Telegram," *J. Comput. Res. Innov. (JCRINN)*, vol. 7, no. 1, hlm. 158–166, Mar. 2022, doi: 10.24191/jcrinn.v7i1.282.

[17] J. C. Flanagan, "The critical incident technique," *Psychol. Bull.*, vol. 51, no. 4, hlm. 327–358, Jul. 1954, doi: 10.1037/h0061470.

[18] Fonnte, "Informasi terkait banned," 2026. [Daring]. Tersedia: https://fonnte.com/tutorial/informasi-terkait-banned/