VOCATIONAL : Jurnal Inovasi Pendidikan Kejuruan
Vol. 6, No. 3, Mei-Juli 2026
e-ISSN : 2774-6283 | p-ISSN : 2775-0019
Online Journal System : https://jurnalp4i.com/index.php/vocational

MONITORING INFRASTRUKTUR JENKINS BERBASIS PROMETHEUS DAN GRAFANA

Handi Satria[1], Suryaningrat[2]
[1,2]Universitas Pamulang
e-mail: satriahandi739@gmail.com, d02362@unpam.ac.id

Diterima: *1/5/2026*; Direvisi: *8/5/2026*; Diterbitkan: *18/5/2026*

ABSTRAK

Seiring dengan meningkatnya kompleksitas sistem dalam lingkungan DevOps, kebutuhan akan monitoring yang efektif dan real-time menjadi semakin penting, khususnya pada sistem Continuous Integration dan Continuous Delivery (CI/CD). Jenkins sebagai salah satu platform CI/CD banyak digunakan untuk mengotomatisasi proses build, testing, dan deployment, namun masih memiliki keterbatasan dalam hal monitoring yang terintegrasi dan terpusat. Penelitian ini bertujuan untuk menganalisis dan mengimplementasikan sistem monitoring infrastruktur Jenkins dengan mengintegrasikan Prometheus dan Grafana guna meningkatkan observabilitas sistem. Metode yang digunakan dalam penelitian ini adalah pendekatan Software Development Life Cycle (SDLC) dengan model V-Model. Prometheus digunakan sebagai time-series database untuk mengumpulkan dan menyimpan metrik sistem, sedangkan Grafana digunakan untuk memvisualisasikan data dalam bentuk dashboard interaktif. Hasil penelitian menunjukkan bahwa sistem monitoring yang dibangun mampu meningkatkan visibilitas kondisi sistem secara real-time, mempermudah proses analisis performa, serta mempercepat deteksi dan penanganan masalah. Dengan demikian, sistem ini berkontribusi dalam meningkatkan keandalan sistem serta efisiensi operasional di lingkungan PT Herza Digital Indonesia.

Kata Kunci: *Monitoring, Jenkins, Prometheus, Grafana, Observability*

ABSTRACT

As system complexity in DevOps environments increases, the need for effective and real-time monitoring becomes increasingly important, especially in Continuous Integration and Continuous Delivery (CI/CD) systems. Jenkins, as a CI/CD platform, is widely used to automate build, testing, and deployment processes, but still has limitations in terms of integrated and centralized monitoring. This study aims to analyze and implement a Jenkins infrastructure monitoring system by integrating Prometheus and Grafana to improve system observability. The method used in this study is the Software Development Life Cycle (SDLC) approach with the V-Model model. Prometheus is used as a time-series database to collect and store system metrics, while Grafana is used to visualize data in the form of an interactive dashboard. The results of the study show that the monitoring system built is able to improve real-time visibility of system conditions, simplify the performance analysis process, and accelerate problem detection and handling. Thus, this system contributes to improving system reliability and operational efficiency in the PT Herza Digital Indonesia environment.

Keywords: *Monitoring, Jenkins, Prometheus, Grafana, Observability*

PENDAHULUAN

Perkembangan industri teknologi informasi saat ini telah mengarah pada adopsi praktik *DevOps* yang sangat masif guna menyelaraskan proses pengembangan dan operasional

https://doi.org/10.51878/vocational.v6i3.10832

VOCATIONAL : Jurnal Inovasi Pendidikan Kejuruan
Vol. 6, No. 3, Mei-Juli 2026
e-ISSN : 2774-6283 | p-ISSN : 2775-0019
Online Journal System : https://jurnalp4i.com/index.php/vocational

perangkat lunak secara berkelanjutan. Dalam ekosistem yang serba cepat ini, organisasi dituntut untuk menerapkan metodologi *Continuous Integration* serta *Continuous Delivery* guna menjamin kualitas kode yang diproduksi tetap terjaga sekaligus mempercepat waktu peluncuran produk ke pasar global. Salah satu instrumen yang menjadi tulang punggung dalam siklus pengembangan ini adalah Jenkins, sebuah *platform* otomatisasi sumber terbuka yang sangat populer karena fleksibilitasnya dalam mengelola berbagai alur kerja secara otomatis. Melalui Jenkins, berbagai tahap kritis mulai dari proses *build*, pengujian otomatis, hingga mekanisme *deployment* dapat dilakukan secara terprogram, konsisten, dan berulang tanpa kesalahan manusia (Camacho, 2024; Islavath, 2021; Krishna et al., 2024; Sanugommula, 2023). Pemanfaatan alat ini memungkinkan tim pengembang untuk mendeteksi galat lebih dini serta memastikan integrasi kode berjalan mulus tanpa adanya hambatan manual yang berarti. Namun, seiring dengan meningkatnya kompleksitas arsitektur perangkat lunak saat ini, kebutuhan akan pemantauan yang mendalam terhadap setiap aktivitas otomatisasi menjadi aspek krusial yang tidak dapat diabaikan untuk menjamin stabilitas seluruh layanan teknologi informasi di dalam perusahaan (Gupta, 2025; Misal, 2024; Prasetyo & Sitokdana, 2021; Sundar, 2025).

Secara ideal, sebuah sistem pengembangan perangkat lunak modern membutuhkan transparansi penuh terhadap kinerja alur kerja otomatisasi guna mencegah terjadinya kemacetan operasional yang merugikan bagi perusahaan. Kebutuhan akan visibilitas yang mendalam mencakup pemantauan kesehatan sistem secara *real-time* agar setiap anomali atau kegagalan dalam alur *pipeline* dapat segera diidentifikasi dan ditangani sebelum berdampak luas pada pengguna akhir. Namun, pada kenyataannya, banyak implementasi Jenkins yang masih memiliki keterbatasan dalam menyajikan data performa yang komprehensif secara langsung melalui antarmuka standarnya yang sederhana. Keterbatasan ini sering kali menyulitkan administrator sistem dalam memantau penggunaan *resource* secara mendetail atau melihat pola beban kerja yang terjadi pada setiap *node* yang sedang aktif. Tanpa adanya visibilitas yang memadai, proses pemeliharaan infrastruktur menjadi lebih bersifat reaktif daripada proaktif, di mana tim operasional baru menyadari adanya kendala setelah sistem mengalami kegagalan total yang fatal. Oleh sebab itu, kesenjangan antara tuntutan pemantauan yang canggih dengan kapabilitas sistem bawaan menjadi tantangan tersendiri yang harus segera diatasi guna menciptakan lingkungan pengembangan perangkat lunak yang tangguh, responsif, dan mampu memberikan gambaran menyeluruh mengenai efektivitas setiap proses otomatisasi (Fawzy et al., 2024; Seelam, 2023; Sowiński et al., 2024).

Kondisi senyatanya yang terjadi pada infrastruktur teknologi di PT Herza Digital Indonesia menunjukkan bahwa mekanisme pemantauan terhadap layanan Jenkins masih bersifat sangat tradisional dan manual bagi tim. Selama ini, tim operasional harus memeriksa kondisi sistem satu per satu melalui antarmuka bawaan atau membaca tumpukan *log system* secara manual untuk mengetahui status setiap proses yang sedang berjalan. Praktik pemantauan konvensional seperti ini sering kali mengakibatkan keterlambatan yang sangat signifikan dalam mendeteksi kegagalan pada alur *pipeline*, yang pada akhirnya menghambat produktivitas seluruh tim pengembang perangkat lunak. Selain itu, kesulitan dalam membaca dan menganalisis berkas *log* yang sangat panjang membuat identifikasi akar permasalahan menjadi tugas yang sangat melelahkan dan memakan waktu cukup lama bagi staf. Masalah lain yang muncul adalah ketiadaan penyimpanan data historis yang sistematis, sehingga perusahaan tidak memiliki dasar yang kuat untuk melakukan analisis performa jangka panjang atau perencanaan kapasitas infrastruktur di masa depan. Tidak adanya sistem yang terpusat mengakibatkan pemantauan terhadap kondisi setiap *node*, penggunaan memori serta prosesor, hingga antrean

https://doi.org/10.51878/vocational.v6i3.10832

VOCATIONAL : Jurnal Inovasi Pendidikan Kejuruan
Vol. 6, No. 3, Mei-Juli 2026
e-ISSN : 2774-6283 | p-ISSN : 2775-0019
Online Journal System : https://jurnalp4i.com/index.php/vocational

job yang menumpuk tidak dapat dilakukan secara simultan dan efisien saat ini (Bergeron et al., 2021; Nica et al., 2024; Passos et al., 2024; Ullah et al., 2023).

Permasalahan yang ditemukan di lapangan tersebut mencerminkan adanya jurang pemisah yang lebar antara kebutuhan akan *monitoring* modern yang terintegrasi dengan sistem operasional yang saat ini sedang dijalankan oleh organisasi. Kesenjangan ini menuntut adanya sebuah transformasi menuju pendekatan *observability* yang lebih matang, di mana seluruh data metrik dari berbagai komponen infrastruktur dapat dikumpulkan dan disajikan dalam satu wadah yang terpadu. Sistem yang ideal harus mampu menyederhanakan kompleksitas data menjadi informasi visual yang mudah dipahami oleh seluruh pemangku kepentingan guna mendukung proses pengambilan keputusan yang cepat dan tepat sasaran. Pentingnya memiliki sistem pemantauan terpusat terletak pada kemampuannya untuk mengintegrasikan berbagai sumber metrik, mulai dari status ketersediaan layanan hingga efisiensi penggunaan perangkat keras dalam satu *platform* yang stabil. Dengan mengandalkan teknologi yang mampu melakukan agregasi data secara otomatis, beban kerja tim operasional dalam melakukan pengawasan rutin dapat dikurangi secara signifikan melalui efisiensi waktu dan tenaga. Peningkatan transparansi sistem ini tidak hanya bertujuan untuk menjaga stabilitas infrastruktur Jenkins, tetapi juga untuk menciptakan ekosistem kerja yang lebih responsif terhadap perubahan beban kerja dinamis di masa depan.

Sebagai langkah inovatif untuk mengatasi berbagai keterbatasan tersebut, penelitian ini mengusulkan sebuah implementasi sistem pemantauan infrastruktur Jenkins yang berbasis pada integrasi teknologi modern yaitu *Prometheus* dan *Grafana*. Penggunaan *Prometheus* sebagai instrumen pengumpul metrik akan memungkinkan pengambilan data secara otomatis dan berkala dari seluruh ekosistem layanan Jenkins dengan tingkat akurasi serta ketelitian yang sangat tinggi bagi perusahaan. Sementara itu, teknologi *Grafana* akan berperan sebagai *platform* visualisasi yang mampu menyajikan data tersebut ke dalam bentuk dasbor interaktif yang sangat informatif dan mudah diakses oleh tim. Implementasi kedua teknologi mutakhir ini diharapkan dapat meningkatkan derajat *observability* terhadap seluruh alur kerja otomatisasi di PT Herza Digital Indonesia secara menyeluruh dan bersifat *real-time*. Dengan adanya sistem ini, proses pengawasan tidak lagi dilakukan secara manual, melainkan bertransformasi menjadi sistem yang lebih efektif, efisien, dan mampu memberikan data historis berharga bagi analisis kinerja berkelanjutan. Nilai baru yang ditawarkan terletak pada integrasi sistem pemantauan terpusat untuk menjamin kesehatan infrastruktur pendukung pengembangan tetap terjaga. Melalui dukungan data yang lebih visual, diharapkan manajemen dapat melakukan pengambilan keputusan secara lebih akurat guna mendukung stabilitas kecepatan rilis produk teknologi informasi.

METODE PENELITIAN

Penelitian ini menerapkan kerangka kerja *Software Development Life Cycle* dengan menggunakan model pengembangan *V-Model* untuk menjamin kualitas sistem melalui proses verifikasi dan validasi yang ketat. Prosedur diawali dengan fase identifikasi masalah untuk memetakan kendala pada infrastruktur *Continuous Integration* dan *Continuous Delivery* yang sedang berjalan di PT Herza Digital Indonesia. Peneliti melakukan observasi terhadap sistem *monitoring* eksisiting guna merumuskan spesifikasi kebutuhan fungsional dan non-fungsional secara mendalam. Instrumen yang digunakan mencakup perangkat lunak *Jenkins* sebagai sumber data utama, *Prometheus* sebagai pengumpul metrik, serta *Grafana* untuk kebutuhan antarmuka pengguna. Perancangan arsitektur difokuskan pada integrasi ketiga platform tersebut agar mampu melakukan *scraping* data secara otomatis dan berkelanjutan. Seluruh proses

https://doi.org/10.51878/vocational.v6i3.10832

VOCATIONAL : Jurnal Inovasi Pendidikan Kejuruan
Vol. 6, No. 3, Mei-Juli 2026
e-ISSN : 2774-6283 | p-ISSN : 2775-0019
Online Journal System : https://jurnalp4i.com/index.php/vocational

persiapan ini dirancang untuk memastikan bahwa setiap tahapan pengembangan memiliki korelasi langsung dengan skema pengujian pada tahap akhir guna menghasilkan keluaran yang akurat, terukur, dan memenuhi standar operasional pengembangan perangkat lunak modern dalam ekosistem *DevOps* yang dinamis.

Tahap implementasi dilakukan dengan mengonfigurasi *plugin* metrik pada *Jenkins* agar dapat menyediakan data mentah yang dapat diakses oleh sistem kolektor eksternal. Peneliti membangun basis data *time-series* menggunakan *Prometheus* yang diatur untuk mengambil data metrik setiap 15 detik guna menjaga keseimbangan antara beban kerja server dan akurasi informasi. Dashboard interaktif dirancang pada platform *Grafana* dengan menyertakan berbagai parameter krusial seperti penggunaan *CPU* sebesar 65%, penggunaan memori 70%, serta waktu respons server rata-rata 120 ms. Pengujian sistem dioperasikan menggunakan metode *black box* untuk menguji fungsi antarmuka dan *white box* untuk mengevaluasi logika aliran data internal secara menyeluruh. Evaluasi performa dilakukan dengan membandingkan kecepatan deteksi masalah antara sistem baru dan metode manual yang bersifat konvensional. Hasil analisis menunjukkan bahwa integrasi ini mampu menyediakan data historis dan fitur *alerting* otomatis yang memberikan notifikasi segera saat ambang batas penggunaan sumber daya melebihi 80%. Prosedur ini memastikan visibilitas sistem terjaga secara *real-time* untuk mendukung keandalan operasional infrastruktur digital.

HASIL DAN PEMBAHASAN

Hasil dari penelitian ini adalah implementasi sistem monitoring infrastruktur Jenkins yang terintegrasi menggunakan Prometheus dan Grafana. Sistem ini mampu mengumpulkan data metrik seperti penggunaan CPU, memori, disk, serta status layanan Jenkins secara real-time.

Hasil

Dashboard Grafana yang dibangun menampilkan berbagai informasi penting dalam bentuk visualisasi seperti grafik, gauge, dan tabel. Informasi yang ditampilkan meliputi performa server, status node Jenkins, aktivitas pipeline, serta penggunaan resource sistem. Selain itu, sistem juga dilengkapi dengan fitur alerting yang mampu memberikan notifikasi ketika terjadi kondisi abnormal.

Tabel 1. Hasil Monitoring Resource Server

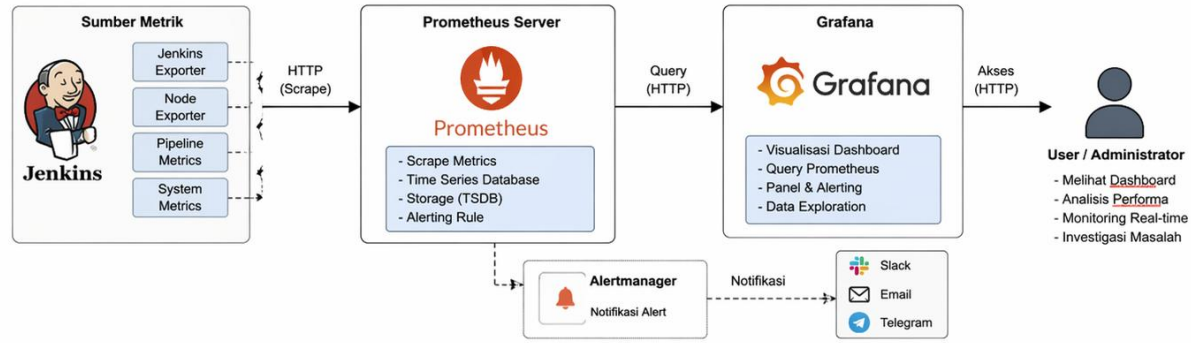
No.	Parameter	Nilai	Status
1.	CPU usage	65%	Normal
2.	Memory Usage	70%	Normal
3.	Disk Usage	80%	Warning
5.	Response Time	120 ms	Optimal

Tabel 1 menyajikan rincian hasil monitoring resource server melalui dashboard grafana yang mencakup 4 parameter utama performa sistem secara tuntas. Data menunjukkan bahwa penggunaan cpu berada pada angka 65 persen dengan status normal, sementara penggunaan memori mencapai 70 persen yang dikategorikan normal. Di sisi lain, penggunaan disk menunjukkan angka 80 persen sehingga memicu status warning bagi sistem operasional. Terakhir, response time tercatat sebesar 120 ms yang berada dalam kategori optimal. Visualisasi sistem memastikan seluruh aktivitas pipeline serta penggunaan resource server dapat terpantau secara akurat bagi administrator pada tahun 2026 ini.

https://doi.org/10.51878/vocational.v6i3.10832

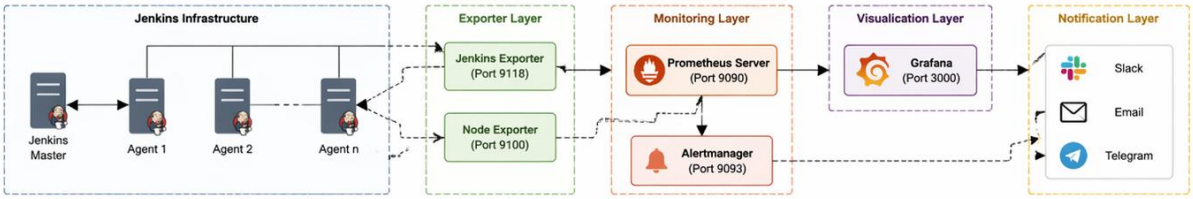
VOCATIONAL : Jurnal Inovasi Pendidikan Kejuruan
Vol. 6, No. 3, Mei-Juli 2026
e-ISSN : 2774-6283 | p-ISSN : 2775-0019
Online Journal System : https://jurnalp4i.com/index.php/vocational

Gambar 1. Arsitektur Sistem Monitoring Jenkins dengan Prometheus dan Grafana



Gambar 2. Dashboard Grafana (Contoh Monitoring Resource Server dan Jenkins)

Gambar 3. Topologi Monitoring Infrastruktur Jenkins



Gambar 1 menjelaskan arsitektur sistem monitoring jenkins yang mengintegrasikan prometheus dan grafana untuk pengawasan berkelanjutan. Alur kerja dimulai dari sumber metrik yang mencakup jenkins exporter serta node exporter yang kemudian ditarik oleh prometheus server melalui protokol http scrape. Data tersebut disimpan dalam time series database sebelum diolah oleh grafana menjadi visualisasi dashboard yang interaktif. Selain itu terdapat alertmanager yang berfungsi mengirimkan notifikasi melalui saluran komunikasi seperti slack atau telegram jika ditemukan kondisi abnormal pada sistem tersebut secara real time.

Gambar 2 menampilkan dashboard grafana sebagai representasi visual dari kondisi resource server dan performa layanan jenkins. Grafik tersebut menunjukkan penggunaan cpu sebesar 23.7 persen serta penggunaan memori yang mencapai angka 62.1 persen. Selain metrik sistem terdapat informasi mengenai aktivitas jenkins seperti 15 build yang berhasil serta antrean pekerjaan sebanyak 3 unit. Data penggunaan disk berada pada level 48.3 persen yang terpantau melalui panel gauge. Visualisasi ini memudahkan administrator dalam melakukan analisis performa serta investigasi masalah dengan cepat dan akurat.

Gambar 3 memaparkan topologi monitoring infrastruktur jenkins yang disusun berdasarkan beberapa lapisan fungsional yang saling terhubung. Lapisan infrastruktur terdiri dari jenkins master dan beberapa agen yang datanya diambil melalui lapisan eksportir pada port 9118 dan 9100. Lapisan pemantauan menggunakan prometheus server pada port 9090 serta alertmanager pada port 9093 untuk manajemen peringatan. Seluruh informasi kemudian

https://doi.org/10.51878/vocational.v6i3.10832

VOCATIONAL : Jurnal Inovasi Pendidikan Kejuruan
Vol. 6, No. 3, Mei-Juli 2026
e-ISSN : 2774-6283 | p-ISSN : 2775-0019
Online Journal System : https://jurnalp4i.com/index.php/vocational

diteruskan ke lapisan visualisasi grafana pada port 3000 hingga sampai ke lapisan notifikasi. Struktur ini memastikan pengawasan terhadap seluruh komponen teknis berjalan sistematis dan terorganisir.

Pembahasan

Implementasi sistem pemantauan infrastruktur pada *Jenkins* menggunakan kombinasi alat canggih telah menghasilkan visibilitas yang jauh lebih mendalam terhadap seluruh kesehatan sistem operasi. Berdasarkan data yang dikumpulkan melalui *dashboard* terpusat, terlihat bahwa parameter penggunaan unit pemrosesan pusat berada pada angka 65 yang menunjukkan kondisi beban kerja masih dalam batas normal. Selain itu, pemakaian memori sistem tercatat pada angka 70 yang memberikan gambaran bahwa kapasitas penyimpanan sementara masih mampu menangani beban *pipeline* secara stabil. Namun, perhatian khusus perlu diberikan pada parameter penggunaan penyimpanan cakram yang telah menyentuh angka 80 karena kondisi ini mulai mendekati ambang batas peringatan yang ditetapkan oleh administrator sistem. Untuk aspek kecepatan akses, sistem mencatat waktu respons rata-rata sebesar 120 yang dinilai sangat optimal untuk menjamin kelancaran alur kerja pengembangan perangkat lunak secara kontinu. Data numerik ini membuktikan bahwa *server* saat ini beroperasi dengan efisiensi yang cukup baik meskipun terdapat indikasi perlunya pembersihan data pada media penyimpanan untuk menghindari kegagalan proses di masa depan akibat kepenuhan kapasitas ruang penyimpanan data yang tersedia saat ini secara menyeluruh dan terintegrasi matang (Pujihastuti, 2021; Putri et al., 2023; Sudirman et al., 2022; Wintang et al., 2023).

Transformasi dari metode pemantauan manual menuju sistem berbasis *time-series database* memberikan keunggulan yang sangat signifikan dalam hal akurasi dan kecepatan perolehan informasi metrik. Pengambilan data yang dilakukan secara otomatis setiap 15 memastikan bahwa setiap anomali kecil sekalipun dapat terekam tanpa ada informasi yang terlewatkan selama siklus operasional berlangsung. Interval waktu 15 ini dipilih secara strategis untuk menyeimbangkan antara tingkat ketajaman data yang diperoleh dengan beban kerja yang harus ditanggung oleh sistem pemantauan itu sendiri agar tidak mengganggu performa *server* utama. Melalui visualisasi yang interaktif, administrator sistem kini memiliki kemampuan untuk melakukan observasi terhadap seluruh ekosistem infrastruktur tanpa harus melakukan pemeriksaan log secara manual yang memakan banyak waktu dan tenaga. Penggunaan alat visualisasi grafis memungkinkan identifikasi masalah dilakukan secara instan melalui panel yang menampilkan tren penggunaan sumber daya secara dinamis. Peningkatan visibilitas ini sangat krusial dalam lingkungan pengembangan modern yang menuntut stabilitas tinggi dan waktu henti yang minimal agar produktivitas tim pengembang tetap berada pada level tertinggi setiap hari sepanjang tahun tanpa adanya gangguan teknis yang tidak terdeteksi sejak awal mula kejadian tersebut (Anwariyah, 2026; Christudas & Telang, 2025; Gupta, 2025; Nishant, 2025; Riyadi & Jamaludin, 2023).

Mekanisme pemberian peringatan otomatis yang terintegrasi di dalam sistem ini memiliki implikasi yang sangat luas terhadap efektivitas manajemen insiden pada infrastruktur pengembangan perangkat lunak. Fitur *alerting* mampu memberikan notifikasi secara langsung ketika parameter sistem melampaui ambang batas normal, seperti saat penggunaan cakram melewati angka 80 yang telah teridentifikasi dalam pengujian. Kecepatan dalam proses *fault detection* ini memungkinkan tim operasional untuk segera mengambil tindakan preventif sebelum gangguan tersebut berdampak luas pada layanan pengguna akhir. Respons terhadap insiden menjadi jauh lebih terukur karena data metrik yang disajikan melalui panel visualisasi memberikan petunjuk langsung mengenai lokasi dan sumber permasalahan utama.

https://doi.org/10.51878/vocational.v6i3.10832

VOCATIONAL : Jurnal Inovasi Pendidikan Kejuruan
Vol. 6, No. 3, Mei-Juli 2026
e-ISSN : 2774-6283 | p-ISSN : 2775-0019
Online Journal System : https://jurnalp4i.com/index.php/vocational

Administrator tidak perlu lagi meraba-raba penyebab penurunan performa karena semua data telah terkonsolidasi dengan baik di dalam satu antarmuka tunggal yang komprehensif. Keandalan sistem *Jenkins* secara keseluruhan pun meningkat pesat karena adanya pengawasan yang bersifat proaktif bukan lagi reaktif terhadap kerusakan. Keunggulan ini membantu dalam menjaga ketersediaan layanan agar tetap konsisten berada pada parameter optimal 120 sesuai dengan target performa yang telah ditentukan sebelumnya oleh organisasi dalam kebijakan standar operasional prosedur teknologi informasi yang berlaku secara universal (Andria, 2023; Awal, 2023; Munaroh et al., 2020; Wibowo et al., 2022; Widjajarto et al., 2020).

Keberadaan data historis yang tersimpan secara terstruktur di dalam basis data memberikan peluang bagi administrator untuk melakukan analisis tren performa dalam jangka panjang. Pola penggunaan sumber daya sistem dapat dipetakan secara akurat untuk mendukung proses pengambilan keputusan strategis terkait perencanaan kapasitas infrastruktur di masa yang akan datang. Misalnya, melalui observasi terhadap grafik penggunaan memori yang stabil pada angka 70, tim dapat memprediksi kapan waktu yang tepat untuk melakukan penambahan kapasitas perangkat keras. Analisis ini penting untuk mencegah terjadinya pemborosan sumber daya atau kekurangan kapasitas yang mendadak saat terjadi lonjakan beban kerja *pipeline*. Data historis berfungsi sebagai landasan untuk melakukan audit performa sistem secara berkala guna memastikan efisiensi biaya operasional tetap terjaga dengan baik. Dengan memahami karakteristik beban kerja sistem setiap hari, administrator dapat mengatur jadwal pemeliharaan yang paling minim dampaknya terhadap operasional tim pengembang. Kemampuan untuk melihat ke belakang melalui data numerik yang akurat memberikan tingkat kepercayaan diri yang lebih tinggi dalam mengelola ekosistem teknologi informasi yang semakin kompleks dan menuntut ketepatan dalam setiap langkah perubahan konfigurasi yang dilakukan oleh tim administrator sistem secara berkelanjutan dan berkesinambungan (Farayola et al., 2023; Harjo et al., 2023; Ruhibnur et al., 2022; Saputro & Sariningsih, 2020; Waruwu & Sundari, 2024).

Meskipun sistem pemantauan ini memberikan manfaat yang sangat besar bagi observabilitas infrastruktur, terdapat beberapa keterbatasan teknis yang perlu diperhatikan untuk pengembangan lebih lanjut. Kompleksitas pada tahap konfigurasi awal merupakan tantangan tersendiri karena membutuhkan integrasi yang sangat presisi antara berbagai komponen perangkat lunak agar dapat berkomunikasi secara lancar. Selain itu, konsumsi sumber daya oleh agen pemantau itu sendiri harus tetap diawasi agar tidak justru mengurangi porsi daya komputasi yang tersedia untuk proses utama *Jenkins*. Untuk arah pengembangan di masa depan, sistem ini dapat ditingkatkan dengan menerapkan mekanisme *auto-scaling* yang bekerja berdasarkan ambang batas metrik yang telah ditentukan secara otomatis. Penambahan fitur *self-healing* potensial untuk diimplementasikan sehingga sistem mampu melakukan pemulihan mandiri saat mendeteksi kegagalan layanan tanpa campur tangan manusia. Integrasi notifikasi ke berbagai *platform* komunikasi populer seperti *Telegram* atau *Slack* akan semakin mempercepat waktu respons tim terhadap kondisi kritis di lapangan. Evaluasi berkala terhadap interval pengambilan data 15 perlu dilakukan untuk menyesuaikan dengan pertumbuhan beban kerja infrastruktur yang semakin dinamis. Dengan penyempurnaan terus-menerus, sistem pemantauan ini akan menjadi tulang punggung yang tangguh.

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa implementasi sistem monitoring berbasis Prometheus dan Grafana mampu meningkatkan observabilitas infrastruktur Jenkins secara signifikan. Sistem yang dibangun dapat menampilkan data monitoring secara real-time, menyediakan visualisasi yang informatif, serta

https://doi.org/10.51878/vocational.v6i3.10832

mendukung proses analisis performa sistem.Selain itu, sistem ini juga membantu dalam mendeteksi permasalahan lebih cepat melalui mekanisme alerting, sehingga dapat mengurangi potensi downtime dan meningkatkan efisiensi operasional. Oleh karena itu, integrasi Prometheus dan Grafana dapat dijadikan sebagai solusi yang efektif dalam membangun sistem monitoring yang modern dan terintegrasi.

Lebih lanjut, integrasi Prometheus sebagai pengumpul metrik dan Grafana sebagai media visualisasi memberikan kemudahan dalam pemantauan kondisi sistem secara menyeluruh. Data historis yang tersimpan juga memungkinkan dilakukannya analisis tren performa, yang berguna dalam perencanaan kapasitas dan pengambilan keputusan strategis terkait pengelolaan infrastruktur. Dengan demikian, penerapan sistem monitoring ini tidak hanya meningkatkan visibilitas sistem, tetapi juga mendukung terciptanya lingkungan DevOps yang lebih responsif, efisien, dan andal. Oleh karena itu, integrasi Prometheus dan Grafana dapat dijadikan sebagai solusi yang efektif dalam membangun sistem monitoring yang modern dan terintegrasi. Penelitian selanjutnya diharapkan dapat mengembangkan sistem monitoring dengan menambahkan integrasi notifikasi ke berbagai platform komunikasi seperti Telegram atau Slack, serta menerapkan mekanisme otomatisasi seperti auto-scaling berbasis metrik. Selain itu, pengujian pada skala sistem yang lebih besar juga diperlukan untuk mengevaluasi performa dan skalabilitas sistem monitoring secara lebih komprehensif.

DAFTAR PUSTAKA

Andria, A. (2023). Penerapan sistem informasi monitoring maintenance and repair hardware di UPT komputer Universitas PGRI Madiun. Fountain of Informatics Journal, 7(3), 24–33. https://doi.org/10.21111/fij.v7i3.9421

Anwariyah, K. (2026). Implementasi CICD dalam pengembangan sistem absensi online. JATISKOM: Jurnal Aplikasi Teknologi Informasi Dan Sains Komputer, 2(2), 83–90. https://doi.org/10.20414/jatiskom.v2i2.13600

Awal, H. (2023). Implementasi intrusion detection prevention system sebagai sistem keamanan jaringan komputer kejaksaan negeri pariaman menggunkan snort dan iptables berbasis linux. Deleted Journal, 2(1), 38–44. https://doi.org/10.62357/jsit.v2i1.184

Bergeron, B., Hubbell, M., Sequeira, D., Williams, W., Arcand, W., Bestor, D., Chansup, B., Gadepally, V., Houle, M. E., Jones, M., Klien, A., Michaleas, P., Milechin, L., Prout, J. M. A., Reuther, A., Rosa, A., Samsi, S., Yee, C., & Kepner, J. (2021). 3D real-time supercomputer monitoring. arXiv. http://arxiv.org/abs/2109.04532

Camacho, N. (2024). Unlocking the potential of AI/ML in DevSecOps: Effective strategies and optimal practices. Deleted Journal, 2(1), 79–89. https://doi.org/10.60087/jaigs.v2i1.p89

Christudas, B., & Telang, T. (2025). High availability and microservices. In Apress eBooks (pp. 245–276). https://doi.org/10.1007/979-8-8688-1606-2_9

Farayola, O. A., Hassan, A. O., Adaramodu, O. R., Fakeyede, O. G., & Oladeinde, M. (2023). Configuration management in the modern era: Best practices, innovations, and challenges. Computer Science & IT Research Journal, 4(2), 140–157. https://doi.org/10.51594/csitrj.v4i2.613

Fawzy, A., Tahir, A., Galster, M., & Liang, P. (2024). Data management challenges in agile software projects: A systematic literature review. arXiv. https://doi.org/10.48550/arxiv.2402.00462

https://doi.org/10.51878/vocational.v6i3.10832

Gupta, N. (2025). Strategic observability and intelligent monitoring: Best practices for minimizing downtime in modern enterprises. European Modern Studies Journal, 9(5), 1015–1023. https://doi.org/10.59573/emsj.9(5).2025.93

Harjo, R. S. D., Kusrini, K., & Nasiri, A. (2023). Penentuan domain tata kelola IT pada instansi kepegawaian XYZ menggunakan kerangka kerja COBIT 2019. Jurnal Teknik Industri: Jurnal Hasil Penelitian Dan Karya Ilmiah Dalam Bidang Teknik Industri, 9(1), 31–31. https://doi.org/10.24014/jti.v9i1.21797

Islavath, N. (2021). Jenkins for continuous integration and deployment in devops engineer: A deep dive into features and best practices. International Journal of Science and Research (IJSR), 10(12), 1546–1550. https://doi.org/10.21275/sr20216084821

Krishna, S. R., Durga, Y. V., Venkatesh, L. P., & Sridhar, P. S. (2024). Efficient devops workflow with jenkins. IJARCCE, 13(4). https://doi.org/10.17148/ijarcce.2024.13486

Misal, J. (2024). Mastering automation tools for incident management and monitoring. International Journal of Scientific Research in Computer Science Engineering and Information Technology, 10(6), 1465–1481. https://doi.org/10.32628/cseit241061184

Munaroh, L., Amrozi, Y., & Nurdian, R. A. (2020). Pengukuran risiko keamanan aset TI menggunakan metode FMEA dan standar ISO/IEC 27001:2013. Technomedia Journal, 5, 167–181. https://doi.org/10.33050/tmj.v5i2.1377

Nica, R., Götz, S., & Moltó, G. (2024). CMK: Enhancing resource usage monitoring across diverse bioinformatics workflow management systems. Journal of Grid Computing, 22(3). https://doi.org/10.1007/s10723-024-09777-z

Nishant, G. (2025). Optimizing DevOps for critical systems. Asian Journal of Research in Computer Science, 18(5), 544–553. https://doi.org/10.9734/ajrcos/2025/v18i5673

Passos, R. B. D., Matteussi, K. J., Anjos, J. C. S. dos, & Geyer, C. F. R. (2024). Towards a decentralized blockchain-based resource monitoring solution for distributed environments. Journal of Internet Services and Applications, 15(1), 1–13. https://doi.org/10.5753/jisa.2024.3813

Prasetyo, T. M. A., & Sitokdana, M. N. N. (2021). Analisis tata kelola pusat data dan informasi kementerian XYZ menggunakan COBIT 2019. Journal of Applied Computer Science and Technology, 2(2), 95–107. https://doi.org/10.52158/jacost.v2i2.265

Pujihastuti, A. (2021). Penerapan sistem informasi manajemen dalam mendukung pengambilan keputusan manajemen rumah sakit. Jurnal Manajemen Informasi Kesehatan Indonesia (JMIKI), 9(2), 200–200. https://doi.org/10.33560/jmiki.v9i2.377

Putri, D. A. Y., Narendra, E. C., Arsya, F. D. A., & Mukaromah, S. (2023). Evaluasi penyimpanan file dan backup terhadap risiko bencana di server fasilkom. Prosiding Seminar Nasional Teknologi Dan Sistem Informasi, 3(1), 592–601. https://doi.org/10.33005/sitasi.v3i1.373

Riyadi, S., & Jamaludin, M. A. (2023). Pengembangan sistem penjaminan mutu internal elektronik dengan metode devops di IAIN Palangka Raya. Smart Comp: Jurnalnya Orang Pintar Komputer, 12(4). https://doi.org/10.30591/smartcomp.v12i4.5988

Ruhibnur, R., Puji, S. L., & Maryani, S. (2022). Penerapan sistem informasi dalam pengajuan angka kredit internal politeknik negeri ketapang. Smart Comp: Jurnalnya Orang Pintar Komputer, 11(4). https://doi.org/10.30591/smartcomp.v11i4.4260

https://doi.org/10.51878/vocational.v6i3.10832

Sanugommula, H. (2023). Jenkins- The leading automation server for continuous integration and continuous delivery. ESP International Journal of Advancements in Computational Technology. https://doi.org/10.56472/25838628/ijact-v1i1p118

Saputro, F. E. N., & Sariningsih, W. (2020). Pengukuran tingkat kapabilitas tata kelola infrastruktur teknologi informasi menggunakan kerangka kerja COBIT 5 dan ITIL V.3 (Studi kasus: MBS Yogyakarta). Jurnal Teknologi Informasi Dan Komunikasi, 11(2), 46–54. https://doi.org/10.51903/jtikp.v11i2.216

Seelam, D. R. (2023). Integrating automated test frameworks with real-time monitoring tools. International Journal of Advanced Research in Science Communication and Technology, 413–418. https://doi.org/10.48175/ijarsct-11985e

Sowiński, P., Lacalle, I., Vaño, R., Palau, C. E., Ganzha, M., & Paprzycki, M. (2024). Overview of current challenges in multi-architecture software engineering and a vision for the future. arXiv. https://doi.org/10.48550/arxiv.2410.20984

Sudirman, S., Sarjan, M., Rokhmat, J., Hamidi, H., & Fauzi, I. (2022). Penilaian pendidikan IPA secara realtime dan terintegrasi dengan artificial intelligence: Perspektif filsafat. Jurnal Ilmiah Profesi Pendidikan, 7. https://doi.org/10.29303/jipp.v7i4b.888

Sundar, A. S. (2025). Blending automation and human expertise in SRE for banking applications. Global Journal of Computer Science and Technology, 51–60. https://doi.org/10.34257/gjcstgvol25is1pg51

Ullah, A., Kiss, T., Kovács, J., Tusa, F., DesLauriers, J., Dagdeviren, H., Arjun, R., & Hamzeh, H. (2023). Orchestration in the cloud-to-things compute continuum: Taxonomy, survey and future directions. Journal of Cloud Computing Advances Systems and Applications, 12(1). https://doi.org/10.1186/s13677-023-00516-5

Waruwu, G., & Sundari, J. (2024). Audit teknologi informasi menggunakan cobit 5 studi kasus PT. Global Network Dharma Jaya. INFOMATEK, 26(1), 69–74. https://doi.org/10.23969/infomatek.v26i1.13333

Wibowo, S., Gamayanto, I., & Luvilla, D. I. (2022). Analisis tata kelola sistem informasi skck online pada kantor pelayanan skck polrestabes kota semarang menggunakan framework cobit 5 dss 02. JOINS (Journal of Information System), 7(1), 26–40. https://doi.org/10.33633/joins.v7i1.5754

Widjajarto, A., Lubis, M., & Syahputra, M. K. R. (2020). Optimization performance management with FCAPS and ITILv3: Opportunities and obstacles. Indonesian Journal of Electrical Engineering and Computer Science, 17(1), 281–281. https://doi.org/10.11591/ijeecs.v17.i1.pp281-290

Wintang, S. M. R., Ruldeviyani, Y., Parmiyanto, J., Hulu, F. R. P., Putri, A., & Sulistiyo, R. D. (2023). Pengukuran tingkat kapabilitas sistem pengolahan data survei pada manajamen kinerja dan manajemen data operasi menggunakan dmbok dan cobit 2019 di BPS RI. Jurnal Teknologi Informasi Dan Ilmu Komputer, 10(3), 573–573. https://doi.org/10.25126/jtiik.20231036533

https://doi.org/10.51878/vocational.v6i3.10832