



Implementasi Sistem Monitoring Jaringan Rumah Sakit Menggunakan Blackbox Exporter, Prometheus, Grafana, dan Alertmanager

Raharja Adhi Putrayana ^{1)*}, Iqmal Akbar ²⁾¹⁾, Wasis Haryono ³⁾

^{1,2,3}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang

*Corresponding author. raharjaadhi0254@gmail.com

ARTICLE INFO

Keywords: Sistem Monitoring, Blackbox Exporter, Prometheus, Grafana, Alertmanager

How to cite: Raharja Adhi Putrayana, Iqmal Akbar, Wasis Haryono, (2025). Implementasi Sistem Monitoring Jaringan Rumah Sakit Menggunakan Blackbox Exporter, Prometheus, Grafana, dan Alertmanager.



ABSTRACT

Rumah Sakit Asyisyifaa Leuwiliang merupakan sebuah institusi kesehatan yang beroperasi di bidang pelayanan medis. Dalam mendukung operasionalnya, staf IT secara berkala melakukan maintenance terhadap perangkat jaringan yang mengalami gangguan. Oleh karena itu, dibutuhkan sebuah sistem monitoring yang mampu memantau seluruh perangkat jaringan di lingkungan rumah sakit. Keberadaan sistem monitoring ini dapat meningkatkan produktivitas staf IT, karena mereka dapat memantau status up atau down dari setiap perangkat secara real-time. Dengan memanfaatkan kombinasi seperti Blackbox Exporter, Prometheus, Grafana, dan Alertmanager, kondisi perangkat jaringan dapat dipantau secara efektif. Selain itu, sistem ini juga mampu memberikan notifikasi otomatis melalui Telegram apabila terjadi perubahan status pada salah satu perangkat, baik ketika perangkat mati maupun kembali aktif.

1. Pendahuluan

Rumah sakit sebagai institusi pelayanan kesehatan memiliki ketergantungan tinggi terhadap ketersediaan dan stabilitas jaringan komputer dalam mendukung sistem informasi manajemen rumah sakit (SIMRS). Kegagalan jaringan, meskipun hanya berlangsung dalam waktu singkat, dapat menghambat proses pelayanan, memperlambat akses data penting, serta berdampak langsung terhadap kualitas layanan kesehatan yang diberikan kepada pasien.

Pada praktiknya, pemeliharaan jaringan di lingkungan rumah sakit umumnya masih bersifat reaktif, di mana tim IT baru akan melakukan perbaikan setelah menerima laporan adanya gangguan. Pendekatan ini tidak hanya tidak efisien, tetapi juga menyulitkan tim IT dalam memantau kondisi perangkat jaringan secara menyeluruh, terutama ketika terdapat banyak perangkat yang tersebar di berbagai ruangan dan gedung. Beberapa penelitian tentang eksploitasi keamanan termasuk pemantauan serangan menggunakan alat tertentu untuk mengetahui potensi kerentanan sistem jaringan [3]. Oleh karena itu, dibutuhkan suatu sistem monitoring jaringan yang mampu memberikan informasi kondisi perangkat secara real-time dan

terpusat. Protokol yang digunakan dalam pengiriman dan penerimaan lalu lintas data pesan teks adalah protokol TCP dan UDP dengan enam parameter sebagai pembanding kedua protokol tersebut, sehingga diperoleh hasil protokol mana yang lebih baik dalam pengiriman dan penerimaan lalu lintas data pesan teks [4]. Pemantauan kesehatan secara real-time melalui aplikasi nirkabel dapat mengoptimalkan alur kerja klinis dan memungkinkan perawatan medis yang lebih responsif [6]. Kegiatan monitoring jaringan sangat penting untuk memastikan jaringan komputer di suatu lembaga berjalan lancar tanpa gangguan, sehingga komunikasi menjadi lebih efektif dan efisien [2]. Kegunaan merupakan suatu konteks penggunaan aplikasi dengan ukuran efektivitas, efisiensi dan kepuasan sehingga dapat mengukur sejauh mana suatu aplikasi dapat digunakan dengan mudah [1]. Sistem ini harus mampu mendeteksi perubahan status perangkat, baik ketika terjadi pemutusan koneksi maupun saat perangkat kembali aktif. Dengan demikian, tim IT dapat merespon permasalahan jaringan lebih cepat dan akurat.

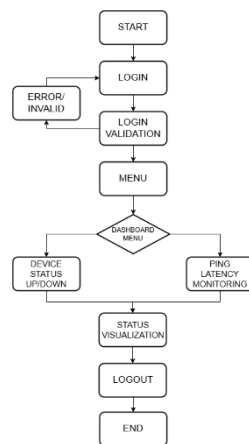
2. Tinjauan Pustaka

Penelitian ini bertujuan untuk mengimplementasikan sistem monitoring jaringan di Rumah Sakit Asyfyfaa Leuwiliang dengan menggunakan VirtualBox OS Ubuntu serta memanfaatkan Blackbox Exporter, Prometheus, Grafana, dan Alertmanager. Selain itu, sistem juga dilengkapi dengan fitur alerting yang terintegrasi dengan aplikasi Telegram, sehingga pemberitahuan terkait perangkat yang mengalami gangguan dapat langsung diterima oleh tim IT secara otomatis. Prometheus adalah alat open-source yang sangat populer yang dirancang untuk mengumpulkan data deret waktu tentang kinerja sistem, Prometheus berjalan seiring dengan Grafana, platform visualisasi open-source [15]. Kombinasi Prometheus dan Grafana memberdayakan operator dengan visibilitas komprehensif ke dalam metrik sistem, memungkinkan mereka untuk dengan cepat mengidentifikasi dan mengatasi masalah melalui pengambilan keputusan yang ditingkatkan dan pemecahan masalah yang efisien [12]. Integrasi Prometheus dan Grafana memungkinkan visibilitas waktu nyata ke dalam Kesehatan sistem, yang memungkinkan administrator untuk secara proaktif mendeteksi dan menanggapi anomali jaringan, secara signifikan mengurangi waktu henti dan meningkatkan keandalan layanan [8]. Dalam penerapan observabilitas berbasis Prometheus dan Grafana, visualisasi real-time memungkinkan tim operasi untuk mengidentifikasi anomali performa seperti lonjakan waktu respons atau error rate secara proaktif [14]. Monitoring yang efektif menjadi esensial untuk menjamin reliabilitas dan performa layanan dalam arsitektur sistem terdistribusi. Prometheus dan Grafana terbukti menjadi solusi monitoring yang kuat, fleksibel, dan scalable, dengan kemampuan mengumpulkan metrik, membangun visualisasi, dan memberikan sistem alert yang dapat disesuaikan sesuai kebutuhan organisasi [11]. Sistem ini terbukti mampu melakukan pemantauan waktu nyata pada fasilitas produksi yang tersebar luas, termasuk pengumpulan data sensorik dan visualisasi historis maupun dinamis secara efektif [13]. Pendekatan ini diharapkan dapat meningkatkan efektivitas dan efisiensi dalam pengelolaan infrastruktur jaringan rumah sakit.

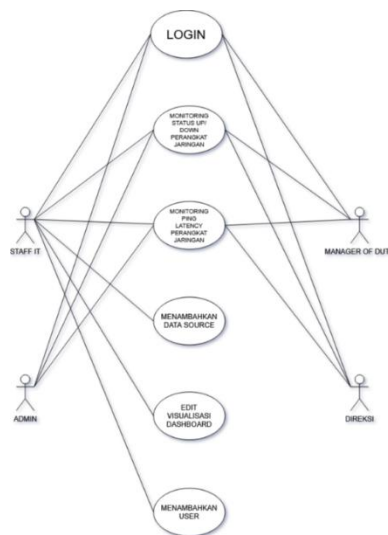
3. Bahan dan Metode Pelaksanaan

Berdasarkan perancangan yang telah dibuat seperti Activity Diagram agar memudahkan proses alur sistem bekerja. Selain itu juga ditambahkan Use Case dengan beberapa actor yang akan

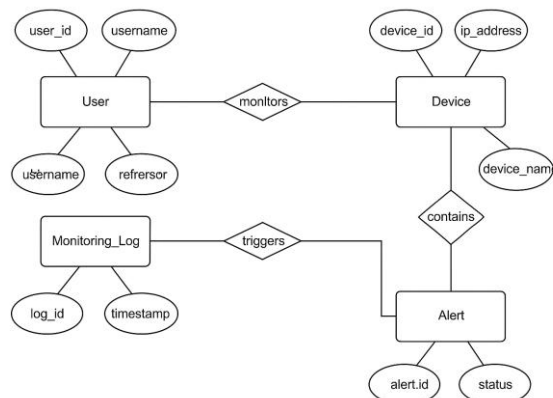
menggunakan aplikasi ini, dan Entitas Relasi Diagram untuk mengetahui database apa saja yang dimasukkan ke dalam sistem.



Gambar 1. Activity Diagram



Gambar 2. Use Case



Gambar 3. Entitas Relasi Diagram

Tujuan dari penelitian ini adalah membangun sistem monitoring jaringan yang dapat memantau status perangkat secara real-time dan memberikan notifikasi otomatis apabila terjadi perubahan

status (down/up) menggunakan kombinasi Prometheus, Blackbox Exporter, Grafana, serta sistem alerting berbasis Telegram di dalam VirtualBox.

1. Bahan

- VirtualBox & Ubuntu

Sistem operasi yang digunakan adalah Ubuntu 22.04 dengan menggunakan VirtualBox minimal 2 GB RAM dan 2 processor CPU. Pemilihan Ubuntu digunakan sebagai sistem operasi didasarkan pada kestabilannya, dukungan komunitas yang luas, serta kompatibilitas tinggi terhadap perangkat lunak monitoring seperti Prometheus, Grafana, Blackbox Exporter, dan Alertmanager. Ubuntu menyediakan lingkungan yang ringan dan fleksibel sehingga mendukung proses instalasi dan konfigurasi komponen sistem monitoring secara efisien. Sementara itu, penggunaan VirtualBox memungkinkan pembuatan lingkungan uji coba yang terisolasi dari sistem utama. Perancangan sistem yang akan dilakukan menggunakan virtualisasi dengan Oracle VM VirtualBox sistem tersebut diimplementasikan pada VirtualBox setelah itu dilakukan pengujian sistem terhadap akses data ke sistem dengan komputer [16]. Hal ini memberikan keuntungan dalam hal keamanan eksperimen, kemudahan replikasi skenario pengujian, serta meminimalkan risiko terhadap sistem produksi.

- Blackbox Exporter

Blackbox Exporter digunakan sebagai komponen utama dalam sistem monitoring didasarkan pada kemampuannya untuk melakukan pemantauan eksternal (black-box probing) terhadap berbagai jenis endpoint jaringan tanpa memerlukan akses langsung ke sistem target. Blackbox Exporter mendukung ICMP sehingga memungkinkan pemantauan ketersediaan dan responsivitas perangkat jaringan secara menyeluruh. penggunaan ICMP melalui Blackbox Exporter sangat relevan untuk memantau status perangkat jaringan seperti router, switch, dan server. Dengan pendekatan ini, sistem dapat mengukur waktu respons (latency), tingkat ketersediaan (uptime), serta mendeteksi kegagalan konektivitas secara real-time.

- Prometheus

Prometheus merupakan sistem monitoring open-source berbasis time-series database yang dirancang untuk mengumpulkan dan menyimpan data metrik secara efisien. Sistem ini mendukung bahasa kueri PromQL yang memungkinkan pengguna melakukan analisis metrik secara mendalam, serta memiliki integrasi yang kuat dengan berbagai exporter dan sistem visualisasi. Prometheus memungkinkan arsitektur yang fleksibel dan berskala untuk pemantauan metrik waktu nyata, terutama bila diintegrasikan dengan alat visualisasi seperti Grafana [7].

- Grafana

Grafana digunakan untuk terintegrasi dengan sumber data seperti Prometheus, dengan integrasi ini, Grafana dapat menampilkan metrik seperti latensi jaringan, status perangkat, penggunaan CPU, dan uptime dalam bentuk grafik. Grafana juga mendukung pembuatan dashboard yang dinamis dan dapat dikustomisasi sesuai kebutuhan monitoring. Penggunaan Grafana sebagai antarmuka visualisasi terbukti mampu menyederhanakan sistem monitoring yang sebelumnya menggunakan sistem khusus. Dengan integrasi arsitektur pemrosesan data berbasis open-source, sistem monitoring menjadi lebih mudah dikembangkan, skalabel, dan terotomatisasi, sesuai dengan kebutuhan pengelolaan data real-time yang kompleks [10].

- Alertmanager

Alertmanager memungkinkan konfigurasi aturan notifikasi yang fleksibel, seperti pengelompokan berdasarkan sumber dan jenis perangkat. Notifikasi dapat dikirim melalui saluran komunikasi seperti Telegram, menggunakan Telegram Bot sehingga memastikan bahwa tim teknis menerima informasi secara real-time dan dapat segera mengambil tindakan korektif.

Dengan dukungan notifikasi otomatis, sistem mampu memberikan peringatan langsung kepada pengguna terkait perubahan status dari perangkat yang dipantau [5].

2. Metode Pengumpulan Data

Metode yang digunakan dalam penelitian ini meliputi observasi lapangan, wawancara, dan studi dokumen. Ketiga metode ini dipilih untuk mendapatkan data yang akurat dan relevan, yang bertujuan untuk mendukung proses sistem informasi manajemen yang ada di Rumah Sakit Asyasyifaa.

- Observasi Langsung

Peneliti melakukan observasi langsung ke Rumah Sakit Asyasyifaa untuk menganalisis kebutuhan sistem yang dapat mendukung seperti perangkat jaringan dan topologinya. Agar nantinya sistem yang akan diimplementasikan bisa berjalan dengan baik. Hal ini penting untuk merancang sistem yang dapat meningkatkan efisiensi dan memberikan kontribusi secara maksimal.

- Wawancara

Wawancara dilakukan dengan direksi dan juga tim IT untuk memperoleh wawasan yang lebih mendalam tentang kebutuhan sistem yang dapat membantu tim IT.

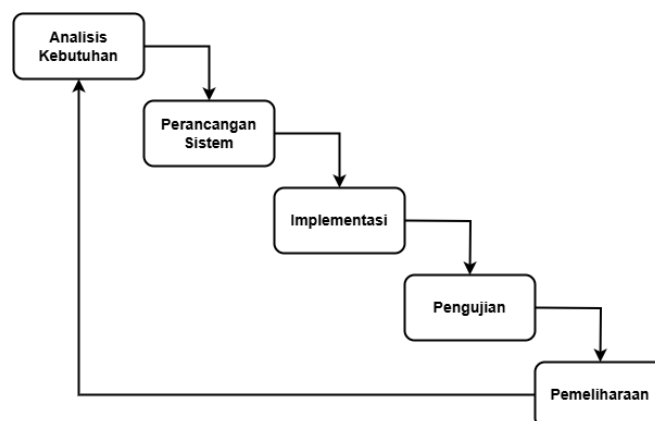
- Studi Dokumen

Peneliti mengkaji literatur terkait teknologi monitoring jaringan, seperti penggunaan ICMP, Prometheus, Grafana, dan Alertmanager. Studi ini mencakup jurnal ilmiah, dokumentasi resmi perangkat lunak, serta penelitian terdahulu yang relevan sebagai dasar teoritis dan pembanding.

3. Metode Pengumpulan Data

Model pengembangan yang digunakan dalam penelitian ini adalah Waterfall. Supaya tujuan dari pembangunan sistem ini tercapai, maka diperlukan tahapan-tahapan yang harus dilakukan seperti yang ada dalam model waterfall [9]. Yaitu pendekatan linear dan berurutan yang terdiri dari lima tahap utama:

- Analisis kebutuhan: Mengidentifikasi kebutuhan pemantauan perangkat jaringan dan kebutuhan fitur alerting oleh tim IT rumah sakit.
- Perancangan sistem: Menyusun arsitektur sistem, topologi jaringan, serta konfigurasi tools yang digunakan (Prometheus, Blackbox Exporter, Grafana, dan Alertmanager).
- Implementasi: Proses instalasi dan konfigurasi tools pada VirtualBox, penambahan target pemantauan, dan penghubungan notifikasi ke Telegram.
- Pengujian: Simulasi perangkat up/down untuk menguji keakuratan deteksi status, kecepatan notifikasi, dan visualisasi di dashboard. Serta menguji kecepatan PING dengan PING Latency
- Pemeliharaan: Melakukan penyempurnaan konfigurasi, dan meninjau kinerja sistem dalam jangka waktu tertentu.



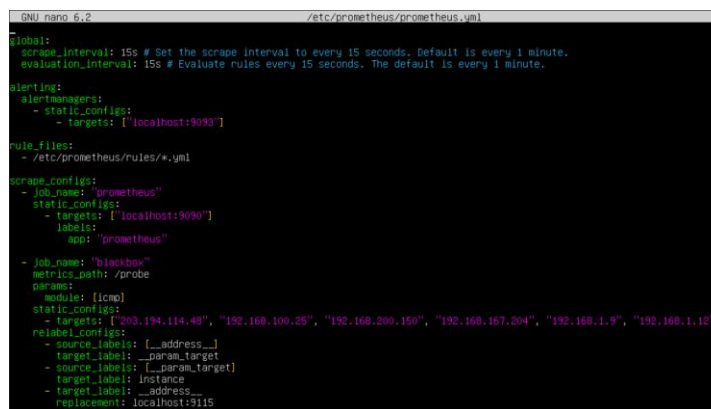
Gambar 4. Model Perancangan Sistem Waterfall

4. Hasil dan Pembahasan

Sistem monitoring berhasil diimplementasikan pada VirtualBox dengan sistem operasi Ubuntu 22.04. Seluruh komponen sistem dapat berjalan dengan baik dan saling terintegrasi. Hasil utama dari implementasi ini mencakup:

1. Prometheus berhasil melakukan scraping data dari Blackbox Exporter secara berkala.
2. Blackbox Exporter mampu memantau berbagai target (IP address perangkat jaringan) melalui protokol ICMP.
3. Grafana menampilkan status perangkat secara real-time dalam bentuk panel visual yang mudah dibaca.
4. Alertmanager mampu mengirim notifikasi ke Telegram saat perangkat tidak dapat dijangkau atau ketika kembali aktif.

Tampilan dashboard Grafana dapat menunjukkan status perangkat sebagai “Up” (jika probe_success = 1) dan “Down” (jika probe_success = 0).



```
gnll nano 6.2 /etc/prometheus/prometheus.yml
# Job name
job_name: 'blackbox'
scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.

# Alerting
alerting:
  alertmanagers:
    - static_configs:
      - targets: ['localhost:9093']

# Rule files
rule_files:
  - /etc/prometheus/rules/*.yml

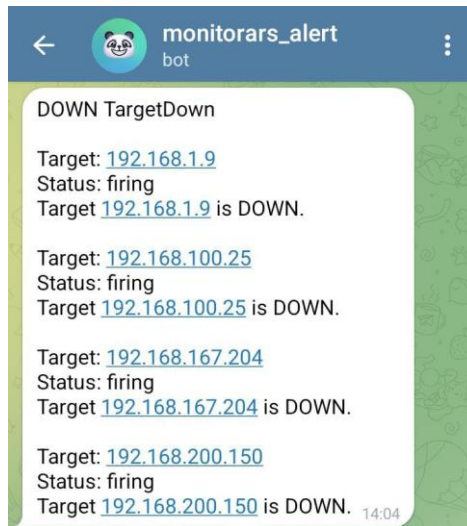
# Scrape configs
scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']
    labels:
      sep: 'prometheus'

  - job_name: 'blackbox'
    metrics_path: /probe
    params:
      module: [icmp]
    static_configs:
      - targets: ['193.194.114.48', '192.168.100.25', '192.168.200.150', '192.168.167.204', '192.168.1.9', '192.168.1.12']
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: localhost:9115
```

Gambar 5. Konfigurasi Prometheus.yml



Gambar 6. Tampilan Dashboard Grafana



Gambar 7. Notifikasi Telegram

1. Pengujian Kondisi Target

Seluruh pengujian memberikan hasil yang sesuai harapan. Setiap IP Target yang dimasukkan menunjukkan statusnya (up/down) serta kecepatan PING.

Table 1. Pengujian Kondisi Target:

IP Target	Status Uji	Hasil Grafana
203.194.114.48	Aktif	1
192.168.200.150	Non-aktif	0
192.168.167.204	Non-aktif	0
192.168.100.25	Non-aktif	0
192.168.1.9	Non-aktif	0
192.168.1.12	Aktif	1

5. Kesimpulan

Implementasi sistem monitoring ini menunjukkan bahwa kombinasi open-source tools dapat menyediakan solusi yang efisien dan real-time dalam memantau status perangkat jaringan di lingkungan rumah sakit. Tidak hanya mampu mengurangi keterlambatan respons dari tim IT terhadap gangguan jaringan, sistem ini juga dapat menyesuaikan skalanya dengan menambah target perangkat tanpa memerlukan perubahan besar pada infrastruktur yang ada.

Salah satu keunggulan sistem ini adalah kemudahan dalam konfigurasi ulang dan penambahan fitur alerting. Penerapan Telegram Bot juga terbukti sangat efektif karena mayoritas staf IT menggunakan aplikasi tersebut dalam komunikasi sehari-hari.

Namun, sistem ini masih memiliki keterbatasan, seperti:

1. Tidak dapat membaca detail trafik jaringan karena hanya menggunakan Blackbox Exporter, bukan Node Exporter dan SNMP Exporter.
2. Membutuhkan PC aktif 24 jam agar monitoring berjalan terus-menerus.

6. Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada dosen pembimbing yang telah memberikan arahan dalam penyusunan penelitian ini, khususnya kepada Bapak Wasis Haryono, S.Kom., M.Kom.

Ucapan terima kasih juga disampaikan kepada tim IT Muhammad Daffan Al Farizi, S.T. selaku Kepala IT dan Abdul Qodir Jaelani, S.T. beserta jajaran Direksi Rumah Sakit Asyisyifaa Leuwiliang atas izin dan kerja samanya dalam proses implementasi sistem di lingkungan jaringan rumah sakit.

Daftar Pustaka

- [1] W. Haryono, "Usability Engineering and Evaluation of Usability in District Tourism and Culture Information Systems," *Journal of Computing and Applied Informatics (JoCAI)*, vol. 3, no. 2, pp. 96–103, 2019.
- [2] T. Thoyyibah and W. Haryono, *Monitoring Jaringan*, Tangerang Selatan: Eureka Media Aksara, 2023.
- [3] T. Thoyyibah, W. Haryono, and T. Hardi, "Security System Exploitation on Domains and Subdomains of UIN Syarif Hidayatullah Jakarta," *Piksel: Penelitian Ilmu Komputer, Sistem Embedded and Logic*, vol. 11, no. 2, pp. 383–392, Sep. 2023.
- [4] W. Haryono, "Comparative Analysis of TCP and UDP Protocol Performance in Sending Text Messages Using Chatting Applications," *Jurnal Inovasi Teknologi dan Rekayasa*, vol. 10, no. 1, pp. 41–48, Jan.–June 2025, doi: 10.31572/inotera.Vol10.Iss1.2025.ID433.
- [5] D. Susanto, R. A. C. Wardana, D. A. Aditia, dan W. Haryono, "Pengembangan Aplikasi Berbasis Web untuk Monitoring dan Pengelolaan Stok ATK dengan Notifikasi Otomatis dan Sistem Barcode di Bank Mandiri, Commercial Banking 5&6," *Jurnal Informatika dan Bisnis (JIBS)*, vol. 2, no. 3, pp. 493–496, Okt.–Des. 2024.
- [6] M. Adi et al., "Real-Time Wireless Health Monitoring App," *International Journal of Computer Applications*, vol. 175, no. 12, pp. 1–5, 2020.
- [7] F. M. Rawoof, M. Tajammul, dan F. Jamal, "On-Premise Server Monitoring with Prometheus and Telegram Bot," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, vol. 6, no. 4, pp. 1–5, Apr. 2022.
- [8] X. Chen, P. Xian, dan S. Liu, "Implementing an Effective Infrastructure Monitoring Solution with Prometheus and Grafana," *Int. J. Comput. Appl.*, vol. 186, no. 38, pp. 34–39, 2024.
- [9] N. Hidayati and Sismadi, "Application of Waterfall Model In Development of Work Training Acceptance System," *INTENSIF*, vol. 4, no. 1, pp. 75–89, Feb. 2020, doi: <https://doi.org/10.29407/intensif.v4i1.13575>.
- [10] T. Beermann, A. Alekseev, D. Baberis, S. Crépe-Renaudin, J. Elmsheuser, I. Glushkov, M. Svatos, A. Vartapetian, P. Vokac, and H. Wolters, "Implementation of ATLAS Distributed Computing monitoring dashboards using InfluxDB and Grafana," *EPJ Web of Conferences*,

vol. 245, p. 03031, 2020. [Online]. Available:
<https://doi.org/10.1051/epjconf/202024503031>

- [11] Y. Jani, "Unified Monitoring for Microservices: Implementing Prometheus and Grafana for Scalable Solutions," *Journal of Artificial Intelligence, Machine Learning and Data Science*, vol. 2, no. 1, pp. 848–852, Mar. 2024.
- [12] M. Bajpai, "Automating Monitoring and Incident Management with Prometheus, Grafana, and Google Cloud Pub/Sub," *International Journal of Science and Research (IJSR)*, vol. 11, <https://dx.doi.org/10.21275/SR24829151754>
- [13] M. Lobur and M. Malyar, "Selecting a Monitoring Technology for a Control System of Distributed Oil Production Facilities," **Energy Engineering and Control Systems**, vol. 10, no. 1, pp. 28–34, 2024, doi: 10.23939/jeecs2024.01.028.
- [14] B. Madupati, "Observability in Microservices Architectures: Leveraging Logging, Metrics, and Distributed Tracing in Large-Scale Systems," *European Journal of Advances in Engineering and Technology*, vol. 10, no. 11, pp. 24–31, 2023.
- [15] J. Hallur, "From Monitoring to Observability: Enhancing System Reliability and Team Productivity," **International Journal of Science and Research (IJSR)**, vol. 13, no. 10, pp. 602–606, Oct. 2024. DOI: 10.21275/SR241004083612. [Online]. Available: <https://www.ijsr.net/getabstract.php?paperid=SR241004083612>
- [16] A. S. Manalu dan S. S. Sitanggang, "Perancangan dan Implementasi Private Cloud Storage dengan OwnCloud pada Jaringan Lokal Menggunakan VirtualBox," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 1, no. 2, pp. 60–71, Jul. 2019.