 ISSN:2302-8149
e-ISSN:2540-9719

Integrasi Sistem Monitoring Aplikasi Uptime Kuma dan Router MikroTik menggunakan Prometheus dan Grafana

Integration of Uptime Kuma Application Monitoring and MikroTik Router System using Prometheus and Grafana

[1]Refalia Defani*, [2]Danur wijayanto
[1,2]Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Aisyiyah Yogyakarta
[1,2]Jl. Siliwangi (Ring Road Barat) No. 63 Nogotirto, Gamping, Sleman, Yogyakarta, Indonesia
*e-mail: 2111501010@student.unisayogya.ac.id

(***received*:** 18 May 2025, ***revised*:** 12 July 2025, ***accepted*:** 13 July 2025)

Abstrak

Sistem *monitoring* jaringan memegang peranan penting dalam memastikan ketersediaan layanan dan performa infrastruktur jaringan secara optimal. Penelitian ini bertujuan untuk mengintegrasikan sistem *monitoring* aplikasi Uptime Kuma dan perangkat *router* MikroTik menggunakan Prometheus dan Grafana pada perusahaan Life Media. Sistem *monitoring* yang dikembangkan dirancang untuk memberikan pemantauan layanan dan perangkat jaringan secara *real-time* melalui satu *dashboard* terpusat yang lebih efisien dan informatif. Proses pengembangan sistem dilakukan dengan pendekatan metode PPDIOO yang mencakup tahapan *prepare, plan, design, implement, operate, dan optimize*. Hasil implementasi menunjukkan bahwa sistem *monitoring* terintegrasi memberikan peningkatan signifikan dibandingkan sistem sebelumnya, baik dari sisi stabilitas akses, kelengkapan visualisasi pemantauan, maupun informasi teknis yang lebih detail. Data teknis yang berhasil ditampilkan meliputi penggunaan *bandwidth*, beban *CPU*, penggunaan *memori*, *latency* layanan, dan tingkat *availability* layanan. Sistem juga memanfaatkan *interval scraping* setiap 10 detik untuk memperoleh data secara berkala, yang membantu mempercepat identifikasi potensi gangguan konektivitas. Penelitian ini memberikan kontribusi dalam pengembangan sistem *monitoring* jaringan berbasis *open-source* yang adaptif, karena mampu menyesuaikan pemantauan terhadap berbagai perangkat dan layanan yang digunakan, serta efisien karena tidak memerlukan perangkat tambahan dan dapat menampilkan data teknis secara *real-time* melalui satu *dashboard* terpusat tanpa ketergantungan pada sistem komersial. Ke depan, sistem ini masih dapat dikembangkan lebih lanjut melalui fitur notifikasi dan *alerting* otomatis untuk memperkuat kapabilitas pemantauan secara menyeluruh.

Kata Kunci: grafana, mikrotik, monitoring, prometheus, uptime kuma

Abstract

A network monitoring system plays a crucial role in ensuring service availability and optimizing network infrastructure performance. This study aims to integrate the Uptime Kuma application monitoring system and MikroTik router devices using Prometheus and Grafana at Life Media Company. The developed monitoring system is designed to provide real-time monitoring of services and network devices through a centralized dashboard that is more efficient and informative. The system was developed using the PPDIOO methodology, which includes the phases of Prepare, Plan, Design, Implement, Operate, and Optimize. The implementation results indicate that the integrated monitoring system offers significant improvements over the previous system in terms of access stability, monitoring visualization completeness, and the availability of more detailed technical information. The technical data successfully displayed include bandwidth usage, CPU load, memory usage, service latency, and service availability levels. The system also utilizes a 10-second scraping interval to collect data periodically, enabling faster identification of potential connectivity issues. This study contributes to the development of adaptive, open-source network monitoring systems by demonstrating that it can flexibly monitor various devices and services, while also being efficient—requiring no additional hardware and capable of displaying real-time technical data through a centralized dashboard without relying on commercial systems.

 ISSN:2302-8149
e-ISSN:2540-9719

In the future, the system can be further enhanced by implementing automated notifications and alerting features to strengthen its overall monitoring capabilities.

***Keywords**: grafana, mikrotik, monitoring, prometheus, uptime kuma*

1 Pendahuluan

Seiring berkembangnya teknologi, kebutuhan akan pemantauan sistem jaringan menjadi semakin penting[1]. Proses data yang dikumpulkan dari berbagai sumber daya dalam bentuk data *real-time* disebut dengan sistem *monitoring*[2]. *Monitoring* jaringan berfungsi untuk mengamati, mencatat dan mengevaluasi setiap aktivitas yang terjadi pada perangkat jaringan guna memastikan kinerjanya tetap optimal[3]. Sistem *monitoring* dibutuhkan oleh berbagai organisasi, salah satunya dapat diterapkan di perusahaan yang menyediakan layanan sistem dan jaringan untuk memastikan kinerja aplikasi dan jaringan agar tetap berjalan secara optimal.

Salah satu perusahaan yang memiliki kebutuhan terhadap sistem *monitoring* adalah PT Sarana Insan Muda Selaras (*Life media*), yang merupakan perusahaan yang bergerak di bidang penyedia layanan internet (*Internet Service Provider*/ISP). Perusahaan ini menyediakan solusi *broadband* untuk layanan internet, televisi kabel, suara dan lainnya menggunakan teknologi *Fiber Optic To the Home (FTTH)*[4]. Life Media mengoperasikan berbagai aplikasi, salah satunya adalah aplikasi pemantauan dan juga infrastruktur jaringan guna menjamin kualitas layanan kepada pelanggan.

Life Media memanfaatkan Uptime Kuma sebagai sistem *monitoring* perangkat *Closed-Circuit Television* (*CCTV)* yang tersebar di berbagai titik di wilayah Yogyakarta dan beberapa layanan pendukungnya. Namun, Uptime Kuma belum mendukung pemantauan perangkat lain seperti *router* MikroTik. Selain keterbatasan tersebut, Uptime Kuma juga sering mengalami gangguan konektivitas seperti *lost connection* secara tiba tiba, data *monitoring* yang hilang sementara, dan keterlambatan dalam pembaruan data. Hal ini menyebabkan efisiensi *monitoring* menjadi terganggu dan tidak mendukung pemantauan secara *real-time* dengan optimal.

Selain perangkat *CCTV*, Life Media mengelola perangkat jaringan, salah satunya adalah *router* MikroTik. Hasil wawancara dengan manajer Life Media menunjukkan bahwa MikroTik digunakan sekitar 80% dari total perangkat jaringan yang dikelola oleh Life Media. Perangkat MikroTik seperti *switch* dan *router* digunakan di pusat data *(data center)* maupun pada sisi pelanggan, dimana perangkat tersebut berperan mendistribusikan koneksi internet utama ke seluruh jaringan.

Berdasarkan permasalahan dalam sistem *monitoring* yang ada, Life Media membutuhkan sistem *monitoring* yang terintegrasi dan terpusat dalam satu *dashboard* antara perangkat *CCTV* dan *router* MikroTik yang memungkinkan pemantauan secara *real-time*. Selain itu, hasil diskusi dengan manajer teknis Life Media, Menunjukkan bahwa perusahaan memiliki kebutuhan untuk menyederhanakan proses *monitoring* melalui satu *dashboard* terpusat.

Saat ini pemantauan dilakukan dengan menggunakan beberapa sistem yang terpisah, seperti Uptime Kuma untuk CCTV dan PRTG untuk perangkat jaringan. Pendekatan ini dinilai kurang praktis karena informasi *monitoring* tersebar di beberapa antarmuka, sehingga menyulitkan dalam proses analisis secara menyeluruh. Sebagaimana pentingnya integrasi sistem *monitoring* dalam meningkatkan efisiensi dan mengurangi fragmentasi data[5]. Oleh karena itu, perusahaan membutuhkan solusi *monitoring* yang mampu menyatukan seluruh perangkat dalam satu platform yang terintegrasi, sehingga proses pemantauan menjadi terpusat dan mudah diakses.

Penelitian ini menggunakan metode PPDIOO yang dikembangkan oleh Cisco [6] Metode ini mencakup enam Fase, yaitu persiapan (*Prepare*), perencanaan (*Plan*), perancangan (*Design*), implementasi (*Implement*), operasional (*Operate*), dan optimasi (*Optimize*)[7]. Dengan menerapkan metode ini, Sistem *monitoring* terintegrasi yang dikembangkan diharapkan dapat memudahkan perusahaan dalam memantau perangkat jaringan secara *real-time*, serta memudahkan dalam pengambilan keputusan secara tepat dan cepat.

2 Tinjauan Literatur

Beberapa penelitian sebelumnya telah memanfaatkan platform seperti Prometheus dan Grafana sebagai solusi *monitoring*. Penelitian [8] menunjukkan bahwa penerapan Prometheus dan Grafana efektif untuk memantau *interface* aktif, penggunaan *CPU*, memori, dan *traffic* jaringan. Hasilnya,

 ISSN:2302-8149
e-ISSN:2540-9719

sistem *monitoring* ini membantu administrator jaringan untuk mengetahui kondisi jaringan tanpa harus melakukan pengecekan manual, sehingga meningkatkan efektivitas kerja. Penelitian [9] berhasil memanfaatkan Prometheus sebagai penyimpan data dalam bentuk metrik dan Grafana sebagai visualisasi grafik. Sistem *monitoring* server Kubernetes yang dikembangkan dalam penelitian tersebut mampu menampilkan data performa seperti penggunaan *CPU*, memori, dan *bandwidth* jaringan secara *real-time*. Pengujian menunjukkan adanya perubahan grafik seiring dengan peningkatan jumlah pengguna, serta menegaskan pentingnya koneksi stabil dalam proses pemantauan.

Penelitian [10]menghasilkan sistem pemantauan server secara *real-time* dengan memanfaatkan Prometheus dan Grafana, serta berjalan di atas Ubuntu Server 18.04. Sistem ini mampu memberikan notifikasi kepada administrator jika terjadi gangguan pada *CPU*, memori, maupun service seperti Apache dan MySQL. Sistem juga dilengkapi dengan fitur alert di Grafana yang terhubung ke Telegram, memungkinkan pemberitahuan dikirim secara otomatis ketika terjadi kegagalan layanan atau penggunaan sumber daya melebihi ambang batas yang ditentukan. Sistem ini dibangun dengan mempertimbangkan spesifikasi minimum instance seperti *CPU*, *RAM*, dan *storage*, dan berhasil menjalankan *monitoring* sesuai ekspektasi. Penelitian [11] juga memperkuat temuan tersebut dengan mengembangkan sistem *monitoring* server berbasis Prometheus dan Grafana yang mendukung pemantauan secara remote dan mampu memberikan informasi akurat terkait status *CPU*, memori, koneksi jaringan, dan *storage*. Fitur notifikasi melalui Telegram terbukti meningkatkan respons administrator dalam menangani masalah.

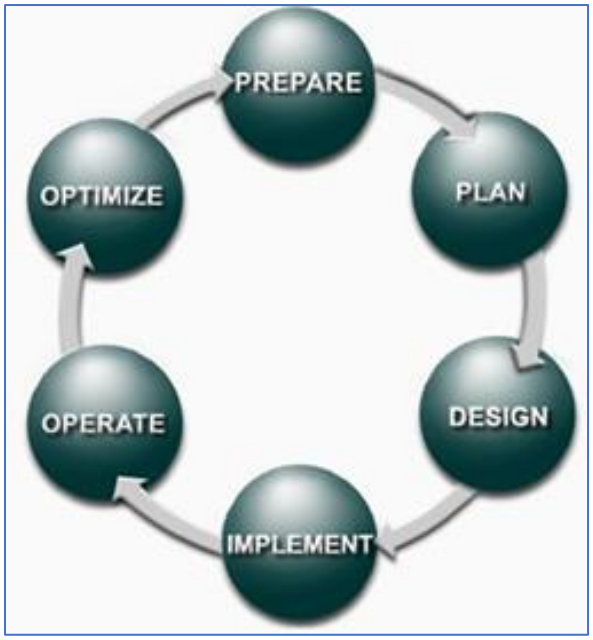
Penelitian [12] menerapkan sistem *monitoring* untuk basis data MongoDB menggunakan Prometheus dan Grafana. Sistem ini meningkatkan keakuratan pembacaan data dan mempermudah administrator dalam memantau performa basis data secara otomatis tanpa perlu pengecekan manual. Keunggulan Prometheus yang menyimpan data dalam bentuk *time-series* dan fleksibilitas visualisasi Grafana menjadi kombinasi efektif dalam *monitoring* basis data. Sifat *open-source* dari kedua platform juga memberi ruang kontribusi luas dari pengembang. Penelitian [13] menambahkan aspek keamanan dengan mengintegrasikan Prometheus, Grafana, dan HashiCorp Vault dalam *monitoring* infrastruktur cloud. Hasilnya menunjukkan bahwa kombinasi ini mampu meningkatkan ketahanan sistem dan perlindungan data secara signifikan.

Berdasarkan tinjauan terhadap penelitian sebelumnya, penggunaan Prometheus dan Grafana telah diterapkan dalam berbagai sistem pemantauan jaringan. Namun, mayoritas penelitian yang telah dilakukan hanya berfokus pada pemantauan satu jenis perangkat atau aplikasi secara terpisah, seperti *monitoring* server, *database*, atau antarmuka jaringan secara individual. Belum ditemukan penelitian yang secara spesifik mengintegrasikan pemantauan Uptime Kuma sebagai *monitoring* layanan dan MikroTik sebagai *monitoring* perangkat jaringan dalam satu sistem berbasis Prometheus dan Grafana. Oleh karena itu Penelitian ini mengusulkan pendekatan baru yang menyederhanakan proses *monitoring* di Life Media dengan mengintegrasikan pemantauan layanan dan perangkat dalam satu *dashboard*. Selain itu, sistem ini dirancang untuk memungkinkan deteksi gangguan lebih cepat dibandingkan metode sebelumnya, serta menyajikan informasi *monitoring* yang lebih lengkap dan mudah dipahami melalui visualisasi yang terpusat dalam satu *dashboard*.

3 Metode Penelitian

Pada Penelitian ini digunakan metode PPDIOO (*Prepare, Plan, Design, Implement, Operate, Optimize).* PPDIOO dikembangkan oleh Cisco sebagai pendekatan siklus hidup jaringan[14],Metode PPDIOO di pilih karena dapat mengurangi TCO *(Total cost Ownership)*[15], yaitu penurunan biaya kepemilikan jaringan, yang menjadi faktor penting dalam pengelolaan teknologi informasi. Dengan mengidentifikasi kebutuhan teknologi sejak awal, melakukan perencanaan infrastruktur yang matang, serta mengembangkan desain yang selaras dengan tujuan bisnis metode ini dapat membantu mengoptimalkan pengeluaran dan menghindari biaya yang tidak perlu [16], Gambar 1 tahapan metode PPDIOO.

 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 1 Tahapan metode PPDIOO

a. Tahap *Prepare*

Tahap *Prepare* merupakan Proses awal untuk mengevaluasi kebutuhan perusahaan dalam menganalisis jaringan yang sedang beroperasi[17], serta mengidentifikasi kebutuhan sistem *monitoring* yang akan dikembangkan. Tahap ini dilakukan analisis kebutuhan sistem *monitoring* pada perusahaan Life Media. Tahap ini dimulai dengan mempelajari infrastruktur jaringan yang ada, seperti perangkat MikroTik dan aplikasi Uptime Kuma yang sedang digunakan. Selanjutnya dilakukan identifikasi terhadap parameter-parameter penting yang perlu dipantau, seperti kinerja jaringan, penggunaan *bandwidth*, dan status perangkat yang nantinya akan diwujudkan dalam bentuk metrik seperti *health summary*, *device info*, *traffic* jaringan dan *status uptime*. Selain itu, dilakukan pemilihan perangkat pendukung yang akan digunakan, yaitu Prometheus, kumpulan alat *open source* yang berfungsi untuk memantau sistem melalui pengumpulan metrik secara berkala. Prometheus dipilih karena kemampuannya dalam mengumpulkan data metrik secara *real-time* dari berbagai sumber serta fleksibilitas integrasinya dengan berbagai platform *monitoring* lainnya [18] serta Grafana, sebuah platform visualisasi data *time-series* yang mendukung berbagai sumber data seperti Prometheus dan memungkinkan pembuatan *dasboard* interaktif untuk pemantauan jaringan secara *real-time* [19].

b. Tahap *Plan*

Pada tahap *Plan* dilakukan perencanaan dan analisis terkait spesifikasi kebutuhan sistem dari segi perangkat keras dan perangkat lunak[20]. Tabel 1 adalah analisis kebutuhan perangkat keras, sedangkan Tabel 2 menunjukkan adalah kebutuhan perangkat lunak.

Tabel 1 Analisis kebutuhan perangkat keras

Nama Perangkat	Spesifikasi
Laptop	*Device name* : LAPTOP-9HQTQER3 *Processor*: Intel(R) Core(TM) i3-1005G1 *CPU* @ 1.20GHz 1.20 GHz *Installed RAM* : 4.00 GB (3.75 GB usable) *System type*: 64-bit *operating system*, x64-*based processor*

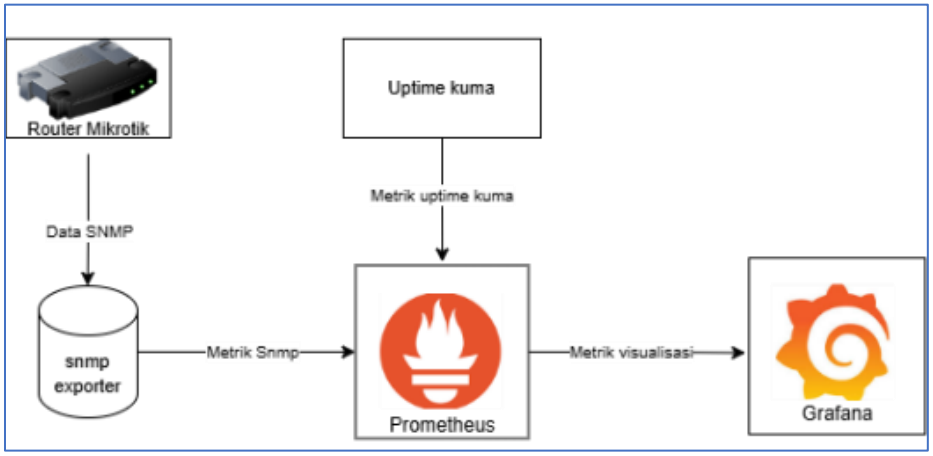
 ISSN:2302-8149
e-ISSN:2540-9719

Tabel 2 Analisis kebutuhan perangkat lunak

Nama Perangkat	Keterangan	Alasan Pemilihan
1. Centos server	Sistem operasi utama untuk *server monitoring*.	Ringan, stabil, *open-source*, cocok untuk *server production*.
2. MobaXterm	Manajemen terminal untuk akses *server* secara *remote*.	Mempermudah akses *SSH* ke *server* dengan *GUI* yang *user-friendly*
3. Prometheus	Pengumpulan dan penyimpanan metrik dari sistem (misalnya, *CPU Usage*, *Traffic*,dll.)	Mampu mengelola *time-series* data, kompatibel dengan SNMP Exporter.
4. Grafana	Menampilkan metrik dalam bentuk visualisasi *dashboard*	Visualisasi powerful, mendukung Prometheus, *user-friendly*
5. Uptime Kuma	Platfrom *monitoring* yang digunakan untuk memantau *uptime* perangkat jaringan	Sudah digunakan perusahaan, cocok untuk pemantauan *service*.
6. Winbox	Aplikasi *GUI* untuk konfigurasi dan manajemen perangkat MikroTik.	Aplikasi resmi MikroTik, untuk konfigurasi.

c. Tahap *Design*

Pada tahap *Design*, dilakukan penyusunan arsitektur sistem *monitoring*. Desain ini mencakup skema alur data dari perangkat MikroTik dan Uptime Kuma yang akan dikumpulkan oleh Prometheus, serta cara data tersebut akan divisualisasikan melalui Grafana.



Gambar 2 Arsitektur sistem *monitoring*

Gambar 2 menunjukkan arsitektur sistem *monitoring* yang digunakan dalam penelitian ini. *Router* MikroTik menghasilkan data jaringan melalui protokol SNMP. Data ini dikumpulkan SNMP Exporter, Kemudian di proses oleh Prometheus sebagai metrik SNMP. Sementara itu Uptime Kuma digunakan untuk memantau status Uptime layanan. Metrik yang dihasilkan Uptime Kuma juga dikumpulkan oleh Prometheus. Seluruh data yang dikumpulkan oleh Prometheus. kemudian di visualisasikan melalui Grafana untuk memudahkan analisis dan pemantauan sistem.

d. Tahap *Implement*

Pada tahap *implement*, dilakukan konfigurasi dan integrasi antara Uptime Kuma dan MikroTik. Tahapan ini mencakup konfigurasi Prometheus sebagai data collector serta pengaturan SNMP Exporter untuk mengambil metrik dari perangkat MikroTik. SNMP (Simple Network Management Protocol) merupakan protokol standar untuk *monitoring* perangkat jaringan berbasis IP[21], dengan data yang disimpan dalam Management Information Base (MIB) dan diakses menggunakan Object Identifiers (OID)[22]. SNMP

 ISSN:2302-8149
e-ISSN:2540-9719

Exporter berperan dalam mengubah data SNMP menjadi format yang dapat dibaca oleh Prometheus[23]. Selain itu, dilakukan integrasi data dari Uptime Kuma yang telah tersedia sebelumnya di perusahaan sebagai sumber metrik tambahan. Penyesuaian file konfigurasi dan penentuan target *monitoring* juga dilakukan agar semua data metrik dapat dikumpulkan dan divisualisasikan secara *real-time* melalui Grafana sebagai platform visualisasi deret waktu yang mendukung Prometheus sebagai sumber data[19].

e. Tahap *Operate*
Pada Tahap *Operate*, sistem *monitoring* yang telah diimplementasikan mulai dijalankan secara aktif. Tahapan ini bertujuan untuk memastikan sistem berfungsi dengan baik. Status job pada Prometheus digunakan untuk memastikan seluruh target berada dalam kondisi *UP*. Data metrik seperti *health summary*, Informasi perangkat, lalu lintas jaringan dan status *uptime*, berhasil dikumpulkan serta divisualisasikan melalui *dashboard* Grafana. Hasil visualisasi ini menunjukkan bahwa sistem telah berjalan sesuai dengan fungsinya.

f. Tahap *Optimize*
Pada tahap *Optimize* dilakukan proses evaluasi dan penyempurnaan sistem *monitoring* yang telah diimplementasikan. Evaluasi dilakukan untuk menilai efektivitas dan efisiensi tampilan *dashboard* serta kelengkapan data yang ditampilkan. Jika ditemukan kekurangan pada visualisasi atau informasi yang disajikan, maka akan dilakukan penyesuaian seperti pengaturan ulang panel atau penambahan informasi yang dibutuhkan. Tahap ini bertujuan untuk memastikan sistem *monitoring* dapat mendukung kebutuhan pemantauan secara optimal dan berkelanjutan.

4 Hasil dan Pembahasan

Bab ini menyajikan hasil implementasi metode PPDIOO yang telah dijelaskan pada Bab Tiga. Fokus pembahasan berada pada tiga fase akhir siklus, yaitu *Implement*, *Operate*, dan *Optimize*, karena fase *Prepare*, *Plan*, dan *Design* telah dilakukan pada tahap perencanaan dan dijabarkan pada bagian metodologi.

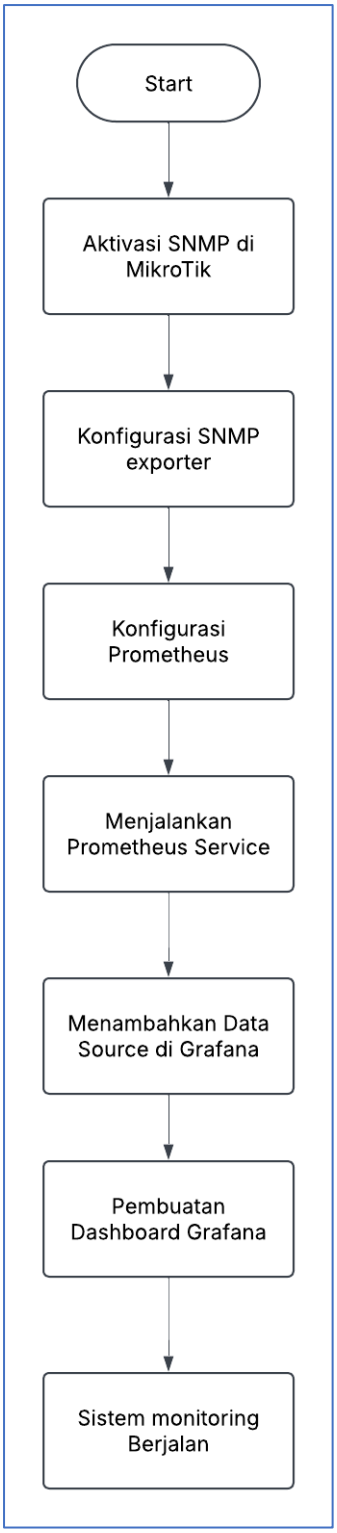
4.1 Integrasi Uptime Kuma dengan Prometheus dan Grafana

Integrasi sistem *monitoring* dalam penelitian ini dirancang untuk menggabungkan fungsi pemantauan layanan dan perangkat jaringan ke dalam satu sistem terpusat dan real-time. Tiga komponen utama yang diintegrasikan yaitu Uptime Kuma sebagai pemantau status layanan, SNMP Exporter untuk pengambilan data dari perangkat MikroTik, serta Prometheus dan Grafana sebagai sistem utama pengumpulan dan visualisasi metrik. Dalam penelitian ini,. Uptime Kuma tidak dikonfigurasi ulang karena sudah digunakan sebelumnya oleh perusahaan. Uptime Kuma menyediakan metrik status uptime dari layanan seperti CCTV dan endpoint lainnya melalui endpoint bawaan /*metrics*. Metrik ini kemudian ditarik secara berkala oleh Prometheus melalui konfigurasi job khusus, dengan pengaturan scrape_interval sebesar 10 detik agar pembaruan data dapat dilakukan secara hampir real-time. Untuk perangkat MikroTik, *monitoring* dilakukan melalui protokol SNMP yang diaktifkan pada perangkat, dan data dikumpulkan melalui SNMP Exporter. Konfigurasi pada file snmp.yml diatur untuk menarik metrik seperti penggunaan CPU, suhu perangkat, trafik jaringan, dan parameter relevan lainnya. Seluruh metrik dari Uptime Kuma dan MikroTik kemudian diproses oleh Prometheus dan divisualisasikan menggunakan Grafana sebagai antarmuka visual utama. *Dashboard* Grafana yang dikembangkan menampilkan berbagai panel, termasuk health summary perangkat, trafik jaringan, serta status dan performa layanan yang dipantau oleh Uptime Kuma. Dengan integrasi ini, sistem *monitoring* yang sebelumnya terpisah kini dapat dikelola dalam satu *dashboard* terpusat, sehingga proses pemantauan menjadi lebih sederhana, respons terhadap gangguan lebih cepat, dan pengambilan keputusan teknis lebih efektif dalam lingkungan operasional perusahaan.

4.2 Tahap *Implement*

Tahap Implementasi *Implement* merupakan Tahap perancangan integrasi sistem *monitoring* untuk aplikasi Uptime Kuma dan *Router* MikroTik. Implementasi dilakukan pada perusahaan Life Media dengan mengintegrasikan aplikasi Uptime Kuma sebagai *monitoring* CCTV dan layanan pendukungnya dan *router* MikroTik juga dipantau untuk memastikan stabilitas jaringan pada perusahaan.

ISSN:2302-8149
e-ISSN:2540-9719



Gambar 2 *Flowchart* alur Konfigurasi Integrasi *monitoring*

Gambar 2 Merupakan flowchart alur konfigurasi sistem *monitoring* yang digunakan pada penelitian ini. Tahapan dimulai dengan Aktivasi SNMP pada perangkat MikroTik agar dapat mengirimkan data metrik, kemudian dilanjut dengan konfigurasi SNMP Exporter agar data dapat dibaca oleh Prometheus, kemudian Prometheus dikonfigurasi dan dijalankan untuk menarik data dari SNMP Exporter. Kemudian Grafana dikonfigurasi dengan menambahkan Prometheus sebagai data source, dilanjutkan dengan pembuatan *dashboard* visualisasi. Setelah itu seluruh konfigurasi selesai maka *monitoring* dapat berjalan dengan baik dan memberikan informasi kondisi jaringan secara *real-time*.

 ISSN:2302-8149
e-ISSN:2540-9719

4.2.1 Konfigurasi Prometheus

Pada tahap konfigurasi Prometheus, diperlukan penentuan interval pengambilan data serta definisi target yang akan di monitor yang ditunjukkan pada Gambar 3. Dengan konfigurasi yang tepat, Prometheus secara teratur mengambil data metrik dari target yang telah ditentukan.

```
global:
  scrape_interval: 10s
scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']
  - job_name: 'uptime-kuma-jitv'
    scrape_interval: 30s
    metrics_path: /metrics
    scheme: "http"
    basic_auth:
      username: 'jitv'
      password: 'jitv123'
    static_configs:
      - targets: ['103.255.15.54:3001']
  - job_name: 'mikortik-snmp'
    static_configs:
      - targets: ['202.169.224.45','111.92.166.1']
    metrics_path: /snmp
    params:
      auth: [public_v2]
      module: [mikrotik]
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: 202.169.224.10:9116  # The SNMP exporter's real hostname:port
```

***Gambar* 3 Konfigurasi *job* prometheus**

Pada bagian Global konfigurasi, terdapat pengaturan *scrape_interval*: 10s, yang berarti Prometheus akan mengambil data setiap 10 detik. Secara default, Prometheus menggunakan *scrape_interval* sebesar 1 menit (1m), tetapi interval ini dapat disesuaikan sesuai kebutuhan *monitoring* [24] Kemudian pada bagian *scrape_config*, terdapat beberapa konfigurasi untuk setiap sumber data (*job*) yang ingin di-*scrape*. *Job* pertama adalah Prometheus, yang mengambil data dari server Prometheus lokal dengan target di localhost:9090. *Job* kedua, yaitu up: 'uptime-kuma-jitv', berfokus pada aplikasi uptime-kuma, dengan jalur metrik '/*metrics'* dan targetnya. Uptime Kuma merupakan sistem *monitoring* yang sebelumnya sudah digunakan oleh perusahaan, sehingga pada penelitian ini tidak dilakukan proses instalasi ulang. Kemudian *Job* ketiga adalah MikroTik-snmp, yang digunakan untuk mengumpulkan data dari perangkat MikroTik melalui SNMP. Targetnya untuk *job* diatas sesuai dan terdapat pengaturan tambahan di *params* dan *relabel_config* yang mengatur modul pengambilan data ke MikroTik serta mendefinisikan label alamat sumber dan target alamat dan *port* untuk SNMP Exporter diatur. Dengan konfigurasi ini Prometheus akan menarik data dari berbagai sumber yang akan digunakan sebagai dasar dalam visualisasi performa sistem *monitoring*.

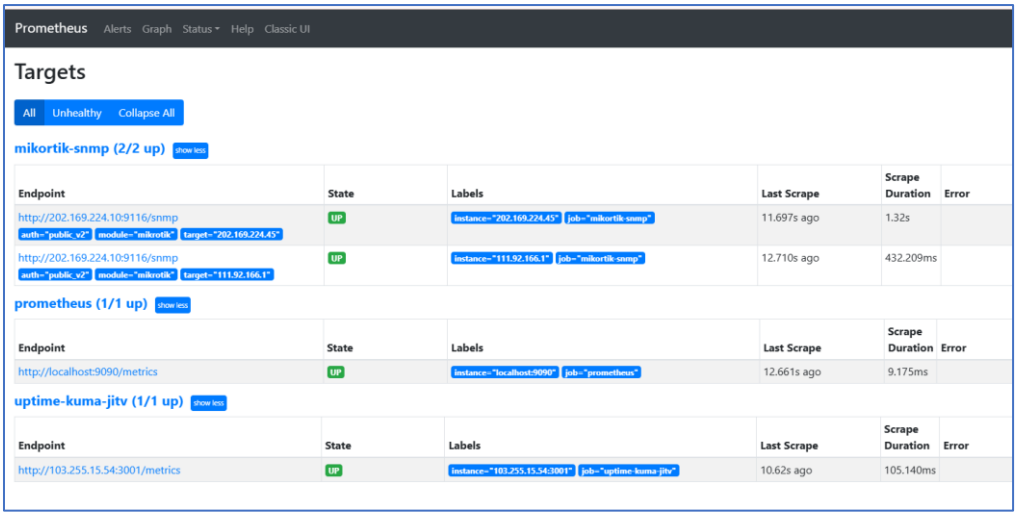
4.2.2 Konfigurasi SNMP Exporter

Konfigurasi SNMP Exporter bertujuan untuk mengambil metrik dari perangkat MikroTik dan mengirimkannya ke Prometheus. proses diawali dengan mengaktifkan fitur SNMP pada perangkat MikroTik melalui perintah */snmp set enable=yes*. Tahapan ini penting untuk memastikan bahwa perangkat dapat merespon permintaan SNMP dari SNMP Exporter. Setelah snmp diaktifkan tahap berikutnya adalah menentukan komunitas snmp agar dapat diakses oleh SNMP Exporter. Selanjutnya SNMP Exporter diunduh melalui repository resmi Prometheus, kemudian file binary yang di peroleh diekstrak dan konfigurasi agar dapat berjalan secara otomatis sebagai layanan melalui *systemd*. Untuk menyesuaikan kebutuhan *monitoring*, dilakukan pemeriksaan pada file konfigurasi snmp.yml yang berisi informasi mengenai modul, OID, dan parameter lainnya. Pada implementasi ini, file snmp.yml menggunakan konfigurasi default karena sudah memenuhi kebutuhan *monitoring* perangkat MikroTik, sehingga tidak diperlukan penyesuaian tambahan.

4.2.3 Hasil Pengumpulan Metrik

Setelah konfigurasi Prometheus selesai, kemudian *service* Prometheus di jalankan ulang untuk menerapkan konfigurasinya. Kemudian sistem mulai melakukan Pengumpulan metrik dari setiap sumber data/ target yang telah di konfigurasi. Proses ini di pantau melalui *dashboard* Prometheus pada menu status target yang akan menampilkan daftar *job* yang sedang di monitor dengan statusnya, jika *job* berstatus *Up* maka Prometheus berhasil menarik data dari target tersebut, Seperti yang ditunjukkan pada Gambar 4.

 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 4 Tampilan *dashboard* prometheus pada menu *Targets*

Dari masing masing target dengan status *Up*, seperti yang ditunjukkan pada Gambar 4. terdapat endpoint metrik yang dapat diakses melalui browser ataupun *Promql Query* di *dashboard* Prometheus. Kemudian metrik yang dikumpulkan dari MikroTik dan Uptime Kuma dapat di visualisasikan dengan menggunakan Grafana untuk analisis performa sistem. Gambar 5 merupakan tampilan *output* metrik dari *endpoint* SNMP Exporter. Kemudian, Gambar 6 menunjukkan tampilan *output* metrik dari *endpoint* Uptime Kuma.

```
# HELP hrDeviceDescr A textual description of this device, including the device's manufacturer and revision, and optionally, its serial number. - 1.3.6.1.2.1.25.3.2.1.3
# TYPE hrDeviceDescr gauge
hrDeviceDescr{hrDeviceDescr="",hrDeviceIndex="1"} 1
hrDeviceDescr{hrDeviceDescr="",hrDeviceIndex="2"} 1
hrDeviceDescr{hrDeviceDescr="",hrDeviceIndex="3"} 1
hrDeviceDescr{hrDeviceDescr="",hrDeviceIndex="4"} 1
# HELP hrDeviceErrors The number of errors detected on this device - 1.3.6.1.2.1.25.3.2.1.6
# TYPE hrDeviceErrors counter
hrDeviceErrors{hrDeviceIndex="1"} 0
hrDeviceErrors{hrDeviceIndex="2"} 0
hrDeviceErrors{hrDeviceIndex="3"} 0
hrDeviceErrors{hrDeviceIndex="4"} 0
# HELP hrDeviceID The product ID for this device. - 1.3.6.1.2.1.25.3.2.1.4
# TYPE hrDeviceID gauge
hrDeviceID{hrDeviceID="0.0",hrDeviceIndex="1"} 1
hrDeviceID{hrDeviceID="0.0",hrDeviceIndex="2"} 1
hrDeviceID{hrDeviceID="0.0",hrDeviceIndex="3"} 1
hrDeviceID{hrDeviceID="0.0",hrDeviceIndex="4"} 1
# HELP hrDeviceIndex A unique value for each device contained by the host - 1.3.6.1.2.1.25.3.2.1.1
# TYPE hrDeviceIndex gauge
hrDeviceIndex{hrDeviceIndex="1"} 1
hrDeviceIndex{hrDeviceIndex="2"} 2
hrDeviceIndex{hrDeviceIndex="3"} 3
hrDeviceIndex{hrDeviceIndex="4"} 4
# HELP hrDeviceStatus The current operational state of the device described by this row of the table - 1.3.6.1.2.1.25.3.2.1.5
# TYPE hrDeviceStatus gauge
hrDeviceStatus{hrDeviceIndex="1"} 2
hrDeviceStatus{hrDeviceIndex="2"} 2
hrDeviceStatus{hrDeviceIndex="3"} 2
hrDeviceStatus{hrDeviceIndex="4"} 2
# HELP hrDeviceType An indication of the type of device - 1.3.6.1.2.1.25.3.2.1.2
# TYPE hrDeviceType gauge
hrDeviceType{hrDeviceIndex="1",hrDeviceType="1.3.6.1.2.1.25.3.1.3"} 1
hrDeviceType{hrDeviceIndex="2",hrDeviceType="1.3.6.1.2.1.25.3.1.3"} 1
hrDeviceType{hrDeviceIndex="3",hrDeviceType="1.3.6.1.2.1.25.3.1.3"} 1
hrDeviceType{hrDeviceIndex="4",hrDeviceType="1.3.6.1.2.1.25.3.1.3"} 1
# HELP hrMemorySize The amount of physical read-write main memory, typically RAM, contained by the host. - 1.3.6.1.2.1.25.2.2
# TYPE hrMemorySize gauge
```

Gambar 5 Tampilan *output* metrik dari endpoint SNMP Exporter

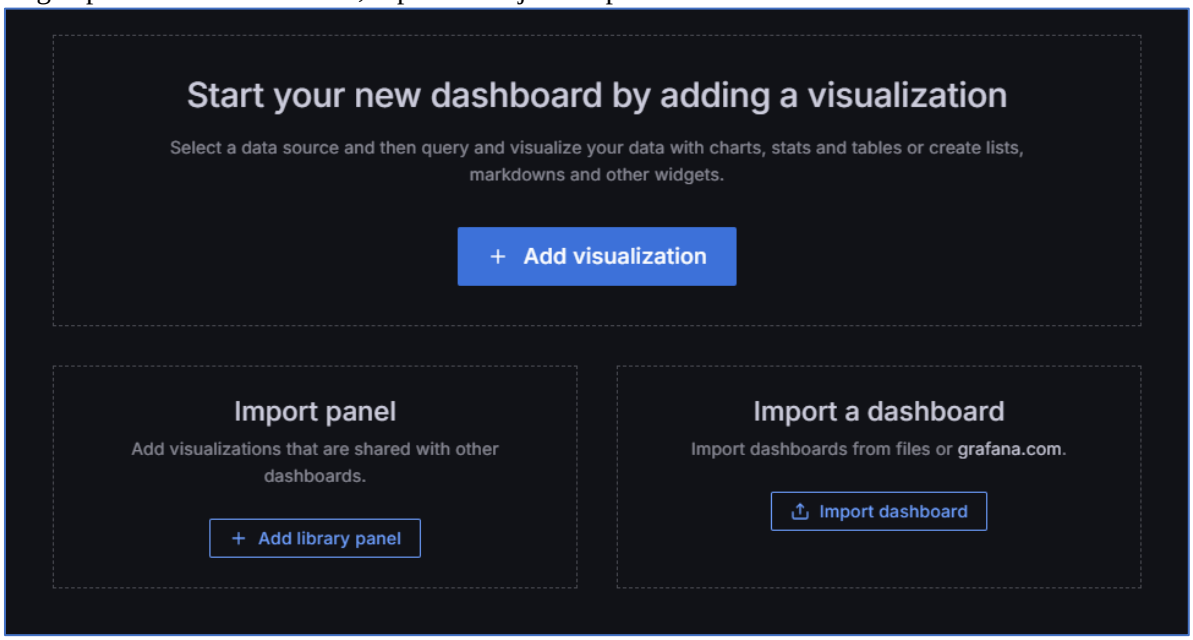
```
# HELP monitor_cert_days_remaining The number of days remaining until the certificate expires
# TYPE monitor_cert_days_remaining gauge

# HELP monitor_cert_is_valid Is the certificate still valid? (1 = Yes, 0= No)
# TYPE monitor_cert_is_valid gauge

# HELP monitor_response_time Monitor Response Time (ms)
# TYPE monitor_response_time gauge
monitor_response_time{monitor_name="JITV",monitor_type="group",monitor_url="https://",monitor_hostname="null",monitor_port="null"} -1
monitor_response_time{monitor_name="GW Multimedia",monitor_type="ping",monitor_url="https://",monitor_hostname="10.200.253.1",monitor_port="null"} 0.392
monitor_response_time{monitor_name="Server Wowza Mulmed",monitor_type="ping",monitor_url="https://",monitor_hostname="10.200.253.2",monitor_port="null"} 1.13
monitor_response_time{monitor_name="VMIX DACEN",monitor_type="ping",monitor_url="https://",monitor_hostname="10.200.253.4",monitor_port="null"} 1.23
monitor_response_time{monitor_name="VMIX MCR",monitor_type="ping",monitor_url="https://",monitor_hostname="10.200.253.7",monitor_port="null"} 0.491
monitor_response_time{monitor_name="Encoder JITV",monitor_type="ping",monitor_url="https://",monitor_hostname="10.200.253.10",monitor_port="null"} 0.536
monitor_response_time{monitor_name="Server MAM",monitor_type="ping",monitor_url="https://",monitor_hostname="10.200.253.5",monitor_port="null"} 0.497
monitor_response_time{monitor_name="Server Playbox",monitor_type="ping",monitor_url="https://",monitor_hostname="10.200.253.3",monitor_port="null"} 0.458
monitor_response_time{monitor_name="Server SAMBA 1",monitor_type="ping",monitor_url="https://",monitor_hostname="10.200.253.14",monitor_port="null"} 0.526
monitor_response_time{monitor_name="Server SAMBA 2",monitor_type="ping",monitor_url="https://",monitor_hostname="10.200.253.12",monitor_port="null"} 0.555
monitor_response_time{monitor_name="Prambanan",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.5",monitor_port="null"} 2.85
monitor_response_time{monitor_name="Paris",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.7",monitor_port="null"} -1
monitor_response_time{monitor_name="Breksi",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.11",monitor_port="null"} 5.92
monitor_response_time{monitor_name="MGM",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.4",monitor_port="null"} 100
monitor_response_time{monitor_name="Simpang Tugu ",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.16",monitor_port="null"} 5.09
monitor_response_time{monitor_name="Server Wowza HCI",monitor_type="ping",monitor_url="https://",monitor_hostname="172.16.16.222",monitor_port="null"} 1.26
monitor_response_time{monitor_name="Garuda",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.18",monitor_port="null"} 0.93
monitor_response_time{monitor_name="Beringharjo",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.20",monitor_port="null"} 1.3
monitor_response_time{monitor_name="Samsat Kota",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.22",monitor_port="null"} 1.65
monitor_response_time{monitor_name="Samsat Bantul",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.24",monitor_port="null"} 1.97
monitor_response_time{monitor_name="Samsat KP",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.25",monitor_port="null"} 2.18
monitor_response_time{monitor_name="Mall",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.19",monitor_port="null"} 1.4
monitor_response_time{monitor_name="Samsat GK",monitor_type="ping",monitor_url="https://",monitor_hostname="10.10.60.26",monitor_port="null"} 2.25
```

Gambar 6. Tampilan *output* metrik dari endpoint uptime kuma

 ISSN:2302-8149
e-ISSN:2540-9719

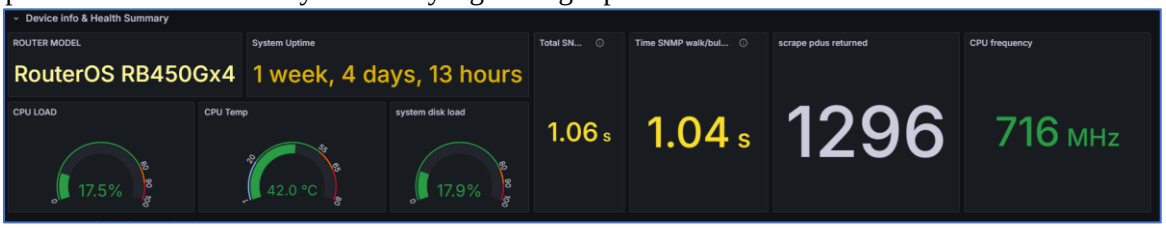
4.2.4 Visualisasi Data *Monitoring* dengan Grafana

Grafana digunakan sebagai alat visualisasi dari data yang telah dikumpulkan oleh Prometheus. Pada awal konfigurasi Grafana menyediakan fitur data source untuk menghubungkan berbagai sumber data, salah satunya Prometheus. Dalam proses ini, jenis data yang di pilih adalah Prometheus yang mengarah dengan alamat URL http://<IP-address>:9090. Setelah konfigurasi berhasil disimpan maka Grafana dapat menarik data metrik dari Prometheus dan menampilkan dalam bentuk panel visualisasi. Kemudian setelah data source berhasil dikonfigurasi langkah selanjutnya adalah membuat *Dashboard* dengan pilihan New dashboard, seperti ditunjukkan pada Gambar 7.



Gambar 7 Menu *add* panel pada grafana

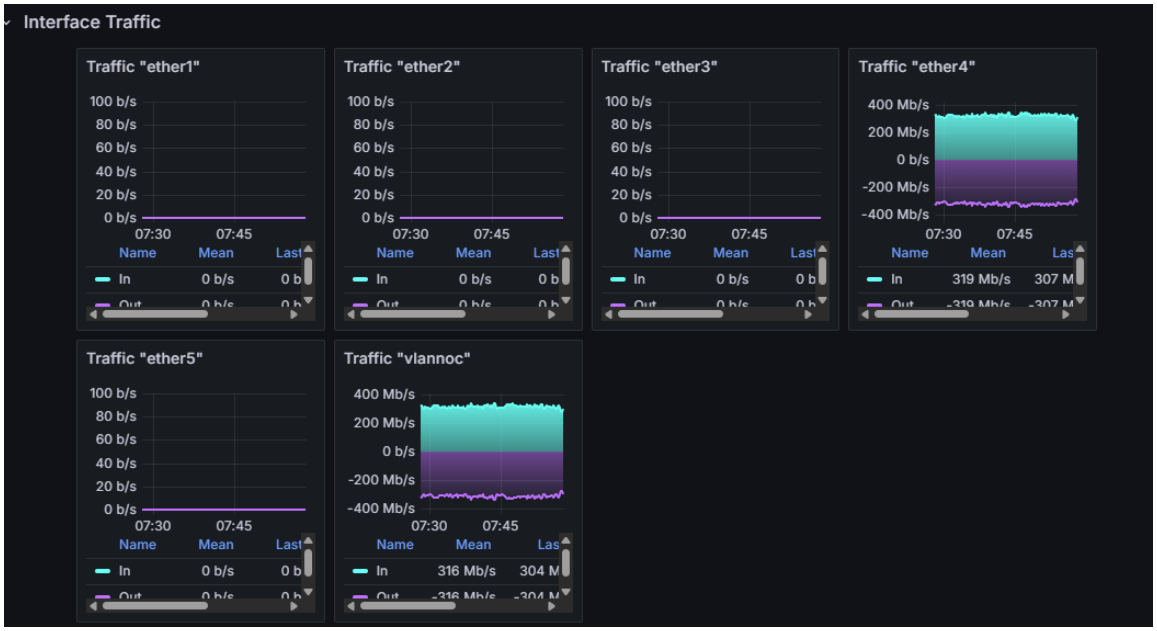
Pembuatan *dashboard* dilakukan sesuai dengan parameter parameter yang sudah di tentukan pada tahapan *prepare*, seperti *row* untuk *health summary* & *device info*, *Interface traffic*, *simple queue in/out/droped/PCQ*, Uptime Kuma CCTV dan *row* untuk setiap monitor yang sedang di pantau. Untuk meningkatkan fleksibilitas dalam pemantauan, *dashboard monitoring* juga dilengkapi dengan *variable filter* yang memungkinkan pengguna untuk memilih informasi spesifik yang ingin ditampilkan. Dengan fitur ini, administrator dapat menyaring tampilan berdasarkan perangkat monitor tertentu, seperti status *router* atau layanan lain yang sedang dipantau.



Gambar 8 *Row* panel *health summary & device info*

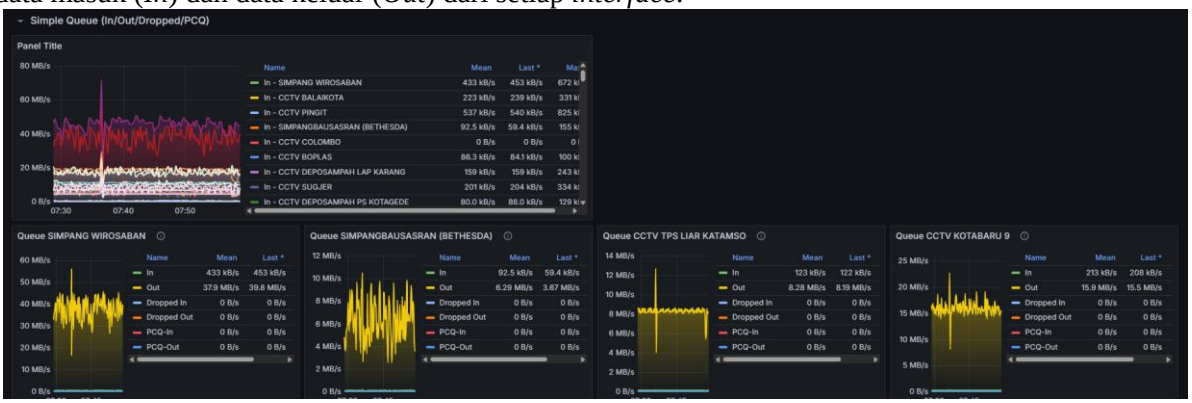
Gambar 8 menunjukkan panel *monitoring* untuk setiap *instance router*. Panel ini menampilkan informasi tipe *router*, *uptime* sistem, CPU *load*, suhu perangkat, penggunaan *disk*, serta parameter SNMP seperti total waktu SNMP, waktu eksekusi SNMP *walk*/*bulkwalk*, dan jumlah *scrape* PDUs *returned*. Informasi ini membantu administrator dalam memantau status operasional *router* dan efektivitas pengambilan data melalui SNMP secara *real-time*.

 ISSN:2302-8149
e-ISSN:2540-9719



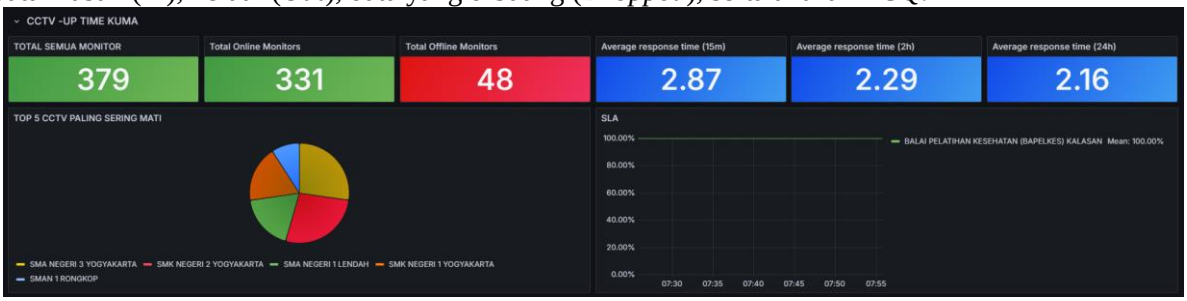
Gambar 9 Panel *interface traffic*

Gambar 9 menunjukkan lalu lintas data jaringan (*traffic*) yang melewati masing-masing interface MikroTik, seperti *ether1, ether2, ether3, ether4, ether5, dan vlannoc*. *Traffic* yang ditampilkan adalah data masuk (*In*) dan data keluar (*Out*) dari setiap *interface*.



Gambar 10 Panel *simple queue (in/out/dropped/PCQ)*

Gambar 10 menampilkan lalu lintas data jaringan (*traffic*) dari berbagai lokasi atau perangkat seperti CCTV yang dimonitor menggunakan *Simple Queue* di MikroTik. Panel ini menampilkan kecepatan data masuk (*In*), keluar (*Out*), data yang dibuang (*Dropped*), serta antrian *PCQ*.

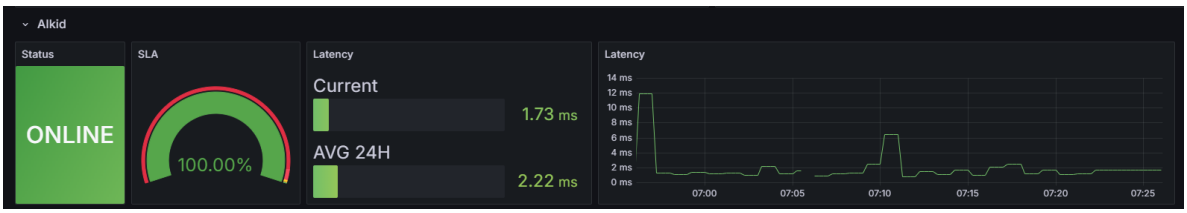


Gambar 11 Panel *summary* total *CCTV*

Gambar 11 menunjukkan panel-panel pada sistem *monitoring* CCTV menggunakan Uptime Kuma yang memberikan informasi mengenai total perangkat yang dipantau, yaitu sebanyak 379 monitor. Monitor dalam konteks ini mengacu pada perangkat atau layanan yang dipantau, seperti CCTV dan layanan pendukungnya. Panel Total *Online Monitor* menunjukkan jumlah perangkat yang saat ini

 ISSN:2302-8149
e-ISSN:2540-9719

dalam kondisi *online*, yang berarti perangkat tersebut aktif, berfungsi dengan baik, dan dapat diakses melalui sistem *monitoring*. Contohnya adalah CCTV atau endpoint lain yang saat ini dapat dipantau tanpa kendala. Panel Total *Offline Monitors* menampilkan jumlah perangkat yang saat ini dalam kondisi *offline*, yang mengindikasikan bahwa layanan, sistem, atau perangkat seperti CCTV tidak dapat diakses atau tidak memberikan *respons* terhadap sistem *monitoring*. Panel *Average Response Time* menampilkan rata-rata waktu respons (*response time*) dari seluruh perangkat dalam rentang waktu tertentu, yaitu 15 menit terakhir, 2 jam terakhir, dan 24 jam terakhir. Waktu respons ini menunjukkan durasi yang dibutuhkan oleh sistem *monitoring* (Uptime Kuma) untuk menerima respons dari perangkat yang dipantau, seperti CCTV, *endpoint*, atau layanan website. Panel *Top* 5 CCTV Paling Sering Mati mengidentifikasi lima perangkat CCTV atau layanan yang mengalami kondisi *offline* paling sering dalam periode waktu tertentu. Informasi ini ditampilkan dalam bentuk diagram *pie chart* untuk mempermudah analisis perbandingan gangguan pada setiap perangkat. Terakhir, panel *SLA Monitoring* menampilkan persentase tingkat ketersediaan layanan atau perangkat dalam suatu periode tertentu (misalnya, per 24 jam). *SLA* ini dihitung berdasarkan perbandingan antara total waktu perangkat dalam kondisi online terhadap total waktu pemantauan dalam periode tersebut.



Gambar 12 *Row* panel pada setiap monitor

Gambar 12 menunjukkan *row* panel setiap *monitor* yang menyajikan informasi terkait status perangkat, *SLA*, *latency* dan *grafik latency*. Status perangkat merepresentasikan apakah target dalam kondisi *online* atau *offline* berdasarkan hasil *probing*. Jika status menunjukkan *online*, maka sistem berhasil melakukan pemantauan tanpa terdeteksi kendala konektivitas. Selanjutnya, nilai *SLA* digunakan untuk menggambarkan tingkat ketersediaan layanan dalam periode tertentu. Pada contoh gambar, *SLA* ditampilkan sebesar 100%, yang mengindikasikan bahwa target tidak mengalami *downtime* selama durasi pemantauan. *Latency* divisualisasikan dalam dua parameter utama, yaitu *current latency* dan rata-rata *latency* selama 24 jam terakhir (*AVG 24H*). Parameter ini memberikan indikasi terhadap performa koneksi dalam waktu nyata serta dalam rentang waktu yang lebih luas. Selain itu, grafik *latency* ditampilkan dalam format *time-series* untuk memberikan gambaran tren perubahan performa jaringan. Dengan pendekatan ini, pengguna dapat melakukan analisis kondisi jaringan berdasarkan pola yang terbentuk dan mengidentifikasi potensi anomali atau gangguan yang mungkin terjadi.

4.3 Tahap *Operate*

Setelah seluruh proses konfigurasi sistem *monitoring* selesai, tahap selanjutnya adalah proses operasional atau *operate*. Pada tahap ini, sistem *monitoring* beroperasi secara aktif dengan melakukan pengumpulan metrik dari perangkat MikroTik dan Uptime Kuma melalui Prometheus. Data yang diperoleh kemudian divisualisasikan melalui *dashboard* Grafana secara *real-time*. Tahap ini menunjukkan bahwa sistem telah siap digunakan dalam lingkungan operasional, menyediakan informasi kondisi perangkat dan layanan secara berkelanjutan, serta mempermudah proses pemantauan.

4.4 Tahap *Optimize*

Setelah sistem *monitoring* diimplementasikan dan berjalan aktif, dilakukan optimalisasi mencakup penambahan perangkat *router* MikroTik kedua ke dalam konfigurasi Prometheus, serta penyesuaian tampilan *dashboard* Grafana agar informasi lebih mudah dipahami. Penyempurnaan dilakukan pada beberapa panel visualisasi, Termasuk penggantian jenis panel dan pengaturan skema warna, untuk memperjelas perbedaan status perangkat yang dipantau. Optimalisasi ini bertujuan untuk memastikan bahwa sistem *monitoring* tidak hanya berjalan dengan baik, tetapi juga memberikan manfaat bagi pengguna dalam proses pengambilan keputusan.

 ISSN:2302-8149
e-ISSN:2540-9719

4.4.1 Analisis sebelum dan sesudah implementasi

Analisis dilakukan dengan membandingkan kondisi sistem sebelum dan sesudah implementasi sistem *monitoring* menggunakan integrasi SNMP Exporter untuk MikroTik dan Uptime Kuma yang terhubung ke Prometheus dan Grafana. Perbandingan dilakukan berdasarkan aspek-aspek berikut:

1. Kestabilan dan kemudahan akses *dashboard monitoring*.
   Sebelum implementasi sistem *monitoring* terintegrasi, proses *monitoring* layanan dilakukan menggunakan Uptime Kuma. Meskipun tampilan Uptime Kuma cukup sederhana dan memberikan informasi status layanan secara langsung, sistem ini memiliki kelemahan dari sisi kestabilan akses. Berdasarkan hasil pengujian yang dilakukan, selama lima menit pemantauan *dashboard* Uptime Kuma, terjadi *reconnecting* sebanyak enam kali. Gangguan ini menyebabkan tampilan status layanan tidak dapat diakses secara konsisten, sehingga dapat mengganggu proses pemantauan jika terjadi secara berulang dalam jangka waktu yang panjang. Setelah implementasi sistem *monitoring* menggunakan Prometheus dan Grafana, akses *monitoring* menjadi lebih stabil. Data dari Uptime Kuma kini dikumpulkan oleh Prometheus dan divisualisasikan melalui *dashboard* Grafana, sehingga informasi layanan dapat ditampilkan secara *realtime* tanpa adanya gangguan *reconnect* maupun kehilangan data. Selain itu, sistem *monitoring* yang baru ini memungkinkan akses *monitoring* dilakukan secara terpusat dan memberikan visualisasi yang lebih informatif.
2. Kelengkapan Visualisasi *Monitoring*
   Sebelum integrasi sistem, Uptime Kuma hanya menyediakan indikator "*Up*" dan "*Down*" serta grafik sederhana untuk *response time*, yang hanya tersedia pada metode pemantauan tertentu seperti *HTTP(s)* atau *ICMP*. Namun, sistem ini belum mendukung metrik tambahan seperti penggunaan *bandwidth*, *CPU*, memori, atau performa jaringan lainnya, sehingga analisis kondisi layanan menjadi kurang komprehensif. Setelah integrasi dengan Grafana melalui Prometheus, visualisasi *monitoring* menjadi lebih kaya informasi dengan penambahan berbagai metrik yang mencakup *bandwidth*, *CPU usage*, *memory usage*, trafik jaringan, antrean *bandwidth*, *SLA availability*, *latency*, serta daftar *Top* 5 perangkat CCTV yang paling sering mengalami *downtime*. Dengan adanya peningkatan ini, sistem *monitoring* tidak hanya menampilkan status operasional layanan, tetapi juga memberikan informasi yang lebih detail untuk mendukung analisis performa jaringan dan *troubleshooting* yang lebih optimal.
3. Kelengkapan Metriks *Monitoring*
   Sebelum implementasi sistem *monitoring* terintegrasi, informasi yang tersedia masih sangat terbatas. Setelah implementasi sistem *monitoring* terintegrasi, informasi yang tersedia menjadi lebih komprehensif dibandingkan metode sebelumnya yang hanya mengandalkan Uptime Kuma. Sistem *monitoring* baru ini mampu menampilkan berbagai metrik performa perangkat MikroTik dan layanan yang dipantau. Berdasarkan hasil pemantauan Gambar 8, model perangkat yang digunakan adalah RouterOS RB450Gx4, dengan *uptime* sistem mencapai 1 minggu, 4 hari, 13 jam. Dari segi performa perangkat, didapatkan nilai *CPU Load* sebesar 17.5% dan suhu *CPU* 42.0°C, yang menunjukkan bahwa sistem masih beroperasi dalam batas optimal. Selain itu, pemantauan penyimpanan juga menunjukkan bahwa *System Disk Load* hanya sebesar 17.9%, yang mengindikasikan kapasitas disk masih mencukupi untuk operasional sistem. Di sisi lain, sistem *monitoring* yang baru juga menyajikan metrik teknis terkait efisiensi SNMP dalam mengumpulkan data dari perangkat. Hasil pemantauan menunjukkan bahwa waktu yang dibutuhkan untuk melakukan *SNMP Walk* mencapai 1.06 detik, sedangkan *SNMP Bulk Request* dapat diproses dalam 1.04 detik, dengan jumlah data yang dikumpulkan sebanyak 1296 *PDU* dalam satu kali *scraping*. Efisiensi ini memastikan bahwa pengumpulan data dapat dilakukan secara *real-time* tanpa mengalami *delay* yang signifikan. Selain itu, informasi mengenai *CPU Frequency* sebesar 716 MHz memberikan gambaran tentang kondisi operasional *router* dalam menangani beban sistem. Dengan hadirnya metrik yang lebih lengkap, sistem *monitoring* tidak hanya memberikan visibilitas yang lebih baik terhadap performa layanan, tetapi juga memungkinkan analisis performa jaringan secara lebih mendalam, yang dapat digunakan sebagai dasar dalam proses *troubleshooting*.

 ISSN:2302-8149
e-ISSN:2540-9719

5 Kesimpulan

Penelitian ini bertujuan untuk mengintegrasikan sistem *monitoring* aplikasi Uptime Kuma dan perangkat *router* MikroTik menggunakan Prometheus dan Grafana pada perusahaan Life Media. Sistem yang dikembangkan telah mampu menyediakan pemantauan layanan dan perangkat jaringan secara *real-time* melalui satu *dashboard* terpusat, yang mendukung pemantauan yang lebih terstruktur dan respons yang lebih cepat terhadap gangguan dalam proses pengawasan infrastruktur jaringan. Implementasi dilakukan dengan pendekatan metode PPDIOO, yang meliputi tahapan *prepare, plan, design, implement, operate, dan optimize*. Hasil implementasi menunjukkan bahwa sistem *monitoring* terintegrasi memberikan peningkatan signifikan dibandingkan kondisi sebelumnya. Sistem tidak hanya menyajikan status layanan secara sederhana, tetapi juga menampilkan metrik performa jaringan secara komprehensif. Peningkatan dapat dilihat pada kestabilan akses *monitoring* yang kini lebih konsisten dan tidak mengalami gangguan *reconnect* seperti sebelumnya. Selain itu, visualisasi *monitoring* menjadi lebih informatif melalui *dashboard* Grafana, yang tidak hanya menampilkan status layanan "*Up*" atau "*Down*", tetapi juga menyajikan grafik dan indikator performa layanan secara *real-time*. Sistem juga menunjukkan kelengkapan metrik *monitoring* yang jauh lebih baik, mencakup *bandwidth, CPU usage, memory usage, trafik* jaringan, antrean *bandwidth, SLA availability, latency,* hingga informasi mengenai SNMP dalam mengumpulkan data dari perangkat. Hal ini menunjukkan bahwa sistem *monitoring* terintegrasi memberikan kemampuan analisis performa layanan dan jaringan yang lebih mendalam dibandingkan sebelum implementasi. Selain itu, sistem juga menyajikan informasi layanan yang paling sering mengalami gangguan (*Top* 5 monitor paling sering mati) serta ringkasan status kesehatan layanan pada setiap titik pemantauan. Meski sistem berhasil diimplementasikan, penelitian ini masih memiliki keterbatasan pada fitur *alerting* otomatis yang belum diterapkan. Oleh karena itu, pengembangan selanjutnya dapat difokuskan pada penerapan fitur notifikasi *real-time* dan *alerting* otomatis untuk meningkatkan kemampuan *monitoring* secara menyeluruh dan memastikan respon yang lebih cepat terhadap masalah yang muncul pada jaringan dan layanan.

Referensi

[1] S. Nurjanah, F. Sembiring, and R. R. Ayuningsih, "*Integration of Telegram Bot and Uptime Kuma for Wi-Fi Network Monitoring Using MikroTik*," *Jurnal Pilar Nusa Mandiri*, Vol. 20, No. 2, pp. 118–126, Sep. 2024, doi: 10.33480/pilar.v20i2.5535.

[2] M. A. Husna and P. Rosyani, "Implementasi Sistem Monitoring Jaringan dan *Server* menggunakan *Zabbix* yang Terintegrasi dengan Grafana dan Telegram," *JURIKOM (Jurnal Riset Komputer)*, Vol. 8, No. 6, p. 247, Dec. 2021, doi: 10.30865/jurikom.v8i6.3631.

[3] A. Pradana, I. R. Widiasari, R. Efendi, and T. Informatika, "Implementasi Sistem Monitoring Jaringan menggunakan *Zabbix* berbasis SNMP," *AITI: Jurnal Teknologi Informasi*, Vol. 19, No. Agustus, pp. 248–262, 2022.

[4] Life Media, "*Life Media Broadband Internet, Cable TV, & Voice*." Accessed: Mar. 15, 2025. [Online]. Available: https://lifemedia.id/

[5] M. Aboubakar, M. Kellil, and P. Roux, "*A Review of IoT Network Management: Current Status and Perspectives*," Jul. 01, 2022, *King Saud bin Abdulaziz University*. doi: 10.1016/j.jksuci.2021.03.006.

[6] N. Nugraha and M. Iqbal, "Perancangan dan Simulasi Jaringan Komputer Politeknik Negeri Subang menggunakan *Packet Tracer* Versi 6.2 dengan metode PPDIOO," 2020.

[7] K. Aditama Marpaung and D. Cakra Mudra Wijaya, "Desain dan Manajemen Jaringan (Studi Kasus: Fakultas Ilmu Komputer UPN Veteran Jatim) *Network Design and Management (Case Study: Faculty of Computer Science UPN Veteran East Java)*," Nov. 2021.

[8] R. Mutiara, F. Politeknik, and N. Jakarta, "Implementasi Sistem Monitoring menggunakan *Prometheus* dan Grafana," 2020. [Online]. Available: https://www.researchgate.net/publication/342511231

[9] S. R. Dira, M. Arif, and F. Ridha, "Monitoring Kubernetes *Cluster* menggunakan *Prometheus* dan Grafana," *Proceeding Applied Business and Engineering Conference*, No. November, 2022.

 ISSN:2302-8149
e-ISSN:2540-9719

[10] D. Rahman and H. Amnur, "Monitoring *Server* dengan *Prometheus* dan *Grafana* serta Notifikasi Telegram," 2020. [Online]. Available: http://jurnal-itsi.org
[11] B. Rasyidi and F. Pratama, "Sistem Monitoring *Server* di PT. XYZ Media Indonesia berbasis *Grafana* dan *Prometheus*," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, Vol. 4, No. 4, pp. 1456–1465, Sep. 2024, doi: 10.57152/malcom.v4i4.1546.
[12] Ramadoni, M. Z. Amirudin, R. Fahmi, E. Utami, and M. S. Mustafa, "Evaluasi Penggunaan *Prometheus* dan *Grafana* untuk Monitoring *Database* Mongodb," *Jurnal Informatika Polinema*, Vol. 7, No. 2, pp. 43–50, Feb. 2021.
[13] T. J. Akinbolaji, G. Nzeako, D. Akokodaripon, and A. V. Aderoju, "*Proactive Monitoring and Security in Cloud Infrastructure: Leveraging Tools Like Prometheus, Grafana, and HashiCorp Vault for Robust DevOps Practices*," *World Journal of Advanced Engineering Technology and Sciences*, Vol. 13, No. 2, pp. 074–089, Nov. 2024, doi: 10.30574/wjaets.2024.13.2.0543.
[14] A. Samsudin, A. Pratama, I. Herlambang, and L. O. B. Aksara, "Perancangan Sistem Jaringan berbasis Router OpenWRT pada BPKHTL Wilayah XXII Kendari," 2024.
[15] D. Ryan Hamonangan Sitompul, O. Jaya Harmaja, and E. Indra, "Perancangan Pengembangan Desain Arsitektur Jaringan menggunakan Metode PPDIOO," *Jurnal Sistem Informasi dan Ilmu Komputer Prima*, Vol. 4, No. 2, 2021.
[16] Cisco, "Menganalisis Arsitektur CISCO *Enterprise Campus*." Accessed: Mar. 19, 2025. [Online]. Available: https://www.ciscopress.com/articles/article.asp?p=1608131&seqNum=3
[17] P. Intan Octavia Br Sipayung, V. Purba, P. Studi Sistem Informasi, F. Ilmu Komputer, and J. Timur, "Analisis, Perancangan, dan Simulasi Jaringan VLAN menggunakan Metode PPDIOO (Studi Kasus: SMAS Santo Yusup Surabaya)," Vol. 14, No. 1, pp. 110–118, 2024, doi: 10.36350/jbs.v14i1.
[18] L. Chen, M. Xian, and J. Liu, "*Monitoring System of OpenStack Cloud Platform based on Prometheus*," in *Proceedings - 2020 International Conference on Computer Vision, Image and Deep Learning, CVIDL 2020*, Institute of Electrical and Electronics Engineers Inc., Jul. 2020, pp. 206–209. doi: 10.1109/CVIDL51233.2020.0-100.
[19] M. K. A. P. Chakraborty, "Grafana," in *Monitoring Cloud-Native Applications*, Berkeley, CA: Apress, 2021, pp. 187–200. doi: 10.1007/978-1-4842-6888-9_6.
[20] A. S. Permadi and A. Prihanto, "Simulasi Monitoring Jaringan menggunakan Aplikasi The Dude dengan Notifikasi *Whatsapp*," *Journal of Informatics and Computer Science*, Vol. 05, 2023.
[21] D. R. Mauro and K. J. Schmidt, "*Essential* SNMP," in *Help for System and Network Administrators*, O'Reilly Media, 2005.
[22] P. Roquero and J. Aracil, "*On Performance and Scalability of Cost-Effective SNMP Managers for Large-Scale Polling*," *IEEE Access*, Vol. 9, pp. 7374–7383, 2021, doi: 10.1109/ACCESS.2021.3049310.
[23] Prometheus Team, "*SNMP Exporter for Prometheus*." Accessed: Mar. 19, 2025. [Online]. Available: https://github.com/prometheus/snmp_exporter
[24] Prometheus Authors, "*Configuration | Prometheus*." Accessed: Mar. 26, 2025. [Online]. Available: https://prometheus.io/docs/prometheus/latest/configuration/configuration/#duration