JITEL (Jurnal Ilmiah Telekomunikasi, Elektronika, dan Listrik Tenaga) p-ISSN: 2774-7972
Vol. 6, No. 1, Maret 2026, pp. 1-12 e-ISSN: 2775-6696
DOI: https://doi.org/10.35313/jitel.v6.i1.2026.1-12

Implementasi *simple network management protocol* untuk monitoring perangkat jaringan di sekolah menengah kejuruan

Muhammad Azzam Khalif Hasanudin[1*], Rifa Hanifatunnisa[2], Usman B. Hanafi[3]
[1,2,3]Jurusan Teknik Elektro, Politeknik Negeri Bandung,
Jl. Gegerkalong Hilir Ds. Ciwaruga, Kabupaten Bandung Barat, Indonesia
[1*]azzamkhalif10@gmail.com, [2]rifahani@polban.ac.id, [3]usmanb@polban.ac.id

ABSTRAK

Sekolah menengah kejuruan (SMK) sebagai institusi pendidikan vokasi yang menerapkan teknologi informasi dalam kegiatan operasional dan pembelajaran memiliki kebutuhan akan pengelolaan perangkat jaringan yang memadai, efektif, dan efisien. Namun, saat ini seperti di SMKN 4 Bandung, pengelolaan perangkat jaringan masih secara manual terhadap masing-masing perangkat. Pada penelitian kali ini dirancang sebuah sistem monitoring perangkat jaringan secara terpusat, dengan memanfaatkan sistem *simple network management protocol* (SNMP) yang ada pada perangkat untuk dikirimkan ke server Zabbix sebagai pusat pengumpulan data perangkat jaringan. Data tersebut akan diambil oleh server Grafana untuk divisualisasikan agar lebih menarik untuk ditampilkan, serta dilengkapi dengan bantuan notifikasi bot Telegram apabila terdapat indikasi perangkat bermasalah. Hasil sistem monitoring melalui SNMP telah membuktikan seluruh perangkat dapat berhasil dimonitoring secara terpusat dan *real-time*, menerima notifikasi saat perangkat terdeteksi mengalami masalah, dan dapat membantu administrator jaringan di sekolah untuk mengurangi dampak *downtime* perangkat, serta membantu evaluasi dan perencanaan pengelolaan infrastruktur jaringan berdasarkan data kinerja perangkat yang terekam.

Kata kunci: monitoring, perangkat jaringan, SNMP, Zabbix, Grafana

ABSTRACT

Vocational high schools (SMKs), as educational institutions that implement information technology in both operational activities and learning processes, require adequate, effective, and efficient network device management. However, in practice—such as at SMKN 4 Bandung—network devices are still managed manually on a per-device basis. This study proposes the design of a centralized network device monitoring system by utilizing the Simple Network Management Protocol (SNMP) available on network devices, which sends data to a Zabbix server as the central data collection point. The collected data is then retrieved by a Grafana server for visualization, making it more informative and visually appealing. Additionally, a Telegram bot notification system is integrated to provide alerts when potential device issues are detected. The results show that the SNMP-based monitoring system successfully enables centralized and real-time monitoring of all network devices, provides notifications when devices experience issues, and assists network administrators in reducing downtime. Furthermore, it supports evaluation and planning of network infrastructure management based on recorded device performance data.

Keywords: *monitoring, network devices, SNMP, Zabbix, Grafana*

1. PENDAHULUAN

Jaringan komputer telah menjadi infrastruktur penting dalam mendukung berbagai aktivitas, termasuk di sektor pendidikan. SMKN 4 Bandung merupakan sekolah menengah kejuruan yang berfokus pada pendidikan di bidang teknologi dan rekayasa memiliki kebutuhan akan pengelolaan jaringan yang memadai. Dengan banyaknya perangkat jaringan seperti *router*, *switch*, dan *access point*, pengelolaan manual dianggap masih belum efektif terhadap waktu dan sulit dalam kondisi menganalisa masalah perangkat jaringan yang bermasalah. Sehingga, dikhawatirkan terjadi *downtime* yang dapat menyebabkan hilangnya *traffic* pengguna dan dapat mengganggu aktivitas operasional [1]. Kasus serupa pernah terjadi saat pelaksanaan Ujian Nasional Berbasis Komputer (UNBK) di SMK Negeri Sungailiat, di mana gangguan jaringan terjadi secara tiba-tiba menyebabkan penundaan ujian dan mengganggu kelancaran proses pendidikan [2].

Naskah diterima tanggal 28 November 2025, disetujui tanggal 10 Maret 2026
*E-mail korespondensi

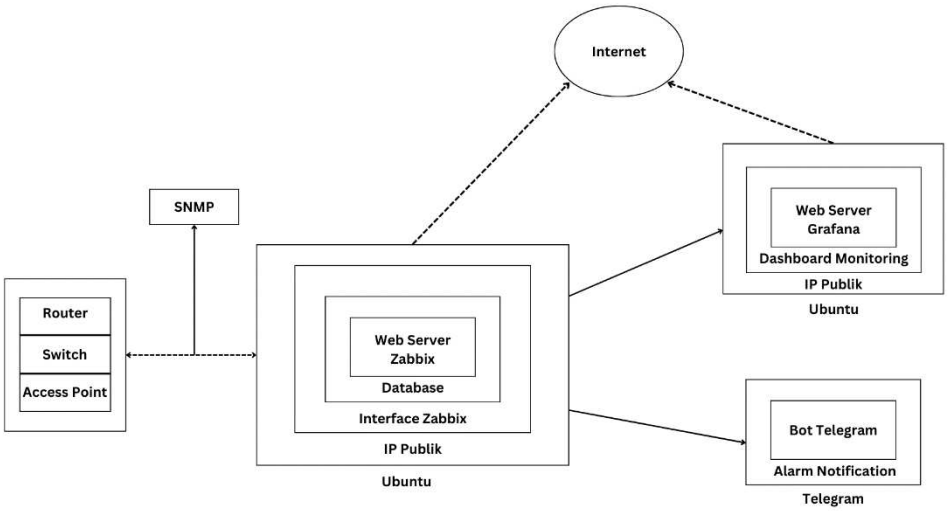
Salah satu solusi yang bisa diterapkan adalah menggunakan *simple network management protocol* atau SNMP. Protokol ini dirancang khusus untuk membantu memantau dan mengelola perangkat jaringan dengan lebih mudah dan efisien. Dengan SNMP, data penting seperti penggunaan *bandwidth*, kondisi CPU, memori, dan status perangkat bisa dikumpulkan dan diakses secara terpusat melalui aplikasi yang dapat membantu penggunaan protokol SNMP agar setiap perangkat jaringan dapat mengirimkan data ke server monitoring [3]. Pada penggunaan SNMPv1 dan SNMPv2c masih menggunakan *community-based security model* (CSM) yang hanya mengandalkan nama *community* sebagai kata sandi tanpa enkripsi. Hal ini menjadi kelemahan utama karena data dikirim dalam teks terbuka, sehingga rentan disadap dan dimanfaatkan oleh pihak tidak berwenang [4]. Sedangkan, SNMPv3 sudah memiliki keunggulan pada keamanan karena sudah dilengkapi autentikasi, enkripsi, dan kontrol akses. Autentikasi memastikan pesan benar-benar dikirim oleh sumber yang sah, enkripsi menjaga data tetap aman selama pengiriman, dan kontrol akses membatasi siapa saja yang bisa melihat atau mengubah informasi tertentu dalam jaringan [5]. Menurut penelitian [6], Zabbix dapat dimanfaatkan sebagai pusat pengumpulan data dari perangkat jaringan melalui integrasi dengan SNMP, sehingga proses monitoring menjadi lebih terpusat dan efisien bagi administrator. Fungsi Zabbix memang merupakan alat monitoring jaringan yang dirancang untuk mengumpulkan, menyimpan, dan menganalisis data perangkat secara otomatis. Sistem ini juga dilengkapi fitur notifikasi yang dapat memberi peringatan saat terjadi kondisi di luar batas normal [7]. Salah satu implementasi sistem monitoring yang memanfaatkan SNMP adalah integrasi Zabbix dengan notifikasi Telegram, yang secara langsung mengirimkan notifikasi secara *real-time* kepada administrator tanpa memerlukan pemantauan layar secara terus-menerus, meskipun belum dilengkapi dengan fitur visualisasi tambahan [8]. Zabbix dapat diintegrasikan dengan Telegram untuk mengirim notifikasi otomatis saat terjadi gangguan atau masalah jaringan ketika mencapai nilai ambang batas. Integrasi ini memerlukan token bot sebagai autentikasi dan ID grup sebagai tujuan pesan, yang dikonfigurasi melalui menu *media types* di Zabbix [9]. Menurut buku Monitoring Cloud-Native Applications, Grafana merupakan *tools* visualisasi yang dapat diintegrasikan dengan Zabbix untuk menyajikan data monitoring dalam bentuk yang lebih informatif dan mudah dipahami. Dengan berbagai pilihan tampilan seperti *gauge*, grafik, statistik, dan diagram batang, Grafana dapat memudahkan analisis performa perangkat jaringan secara lebih detail [10]. Implementasi lain yang lebih lengkap ditunjukkan melalui integrasi Zabbix dengan Grafana dan Telegram untuk memantau *traffic* dan log gangguan pada perangkat jaringan secara berkala, sekaligus menyajikan visualisasi yang memudahkan analisis serta notifikasi cepat agar permasalahan segera ditangani [11]. Pemanfaatan data dari *management information base* (MIB) dapat juga membantu sistem monitoring mengambil tindakan otomatis, seperti menonaktifkan *port* yang melewati ambang batas penggunaan *bandwidth*, sehingga pengelolaan jaringan menjadi lebih responsif dan efisien [12].

Sebagai upaya mengatasi permasalahan monitoring perangkat jaringan di SMKN 4 Bandung, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem monitoring jaringan berbasis protokol SNMP yang terpusat di lingkungan SMKN 4 Bandung. Sistem ini dikembangkan agar administrator jaringan dapat memantau kondisi perangkat seperti *router*, *switch*, dan *access point* secara *real-time*, serta menerima notifikasi otomatis melalui Telegram apabila terjadi gangguan, tanpa perlu terus-menerus memantau sistem. Visualisasi data melalui *dashboard* monitoring Grafana juga digunakan agar informasi performa jaringan dapat disajikan secara informatif dan mudah dianalisis. Meskipun sistem ini belum dapat melakukan tindakan penanganan secara otomatis, informasi yang dikirimkan secara *real-time* diharapkan dapat membantu administrator dalam proses analisa dan *troubleshooting* dengan lebih cepat dan efisien untuk meminimalisir dampak *downtime*.

2. METODE PENELITIAN

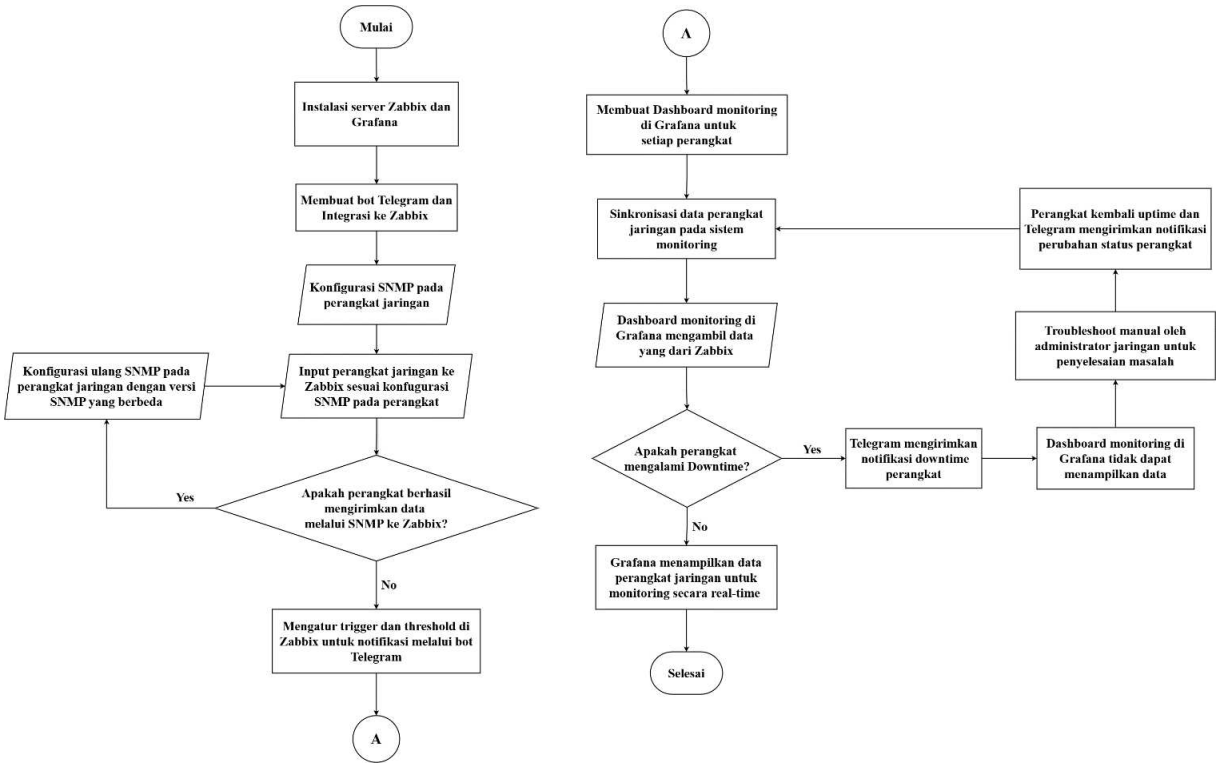
2.1 Ilustrasi Sistem

Tahap pertama dibuat ilustrasi sistem monitoring untuk memberikan gambaran umum mengenai alur kerja dan keterhubungan seluruh perangkat jaringan dengan server monitoring. Setiap perangkat jaringan akan mengirimkan data performa melalui protokol SNMP yang kemudian dikumpulkan dan diproses secara terpusat untuk memudahkan pemantauan kondisi jaringan secara *real-time.*



Gambar 1. Diagram blok sistem monitoring

Gambar 1 menampilkan sebuah diagram blok terkait alur sistem monitoring jaringan di SMKN 4 Bandung yang memanfaatkan protokol SNMP untuk mengirim data dari perangkat jaringan seperti *router*, *switch*, dan *access point* ke server Zabbix. Data dikumpulkan secara *real-time* dan disimpan dalam *database* yang terintegrasi dengan Zabbix. Selanjutnya, data ini diambil oleh server Grafana untuk menampilkan visualisasi dalam bentuk *dashboard* monitoring yang lebih informatif. Ketika perangkat terdeteksi bermasalah, seperti mengalami *downtime* atau melebihi ambang batas pemakaian yang telah ditentukan pada Zabbix, sistem akan secara otomatis mengirimkan notifikasi melalui bot Telegram ke grup administrator sebagai bentuk peringatan. Baik server Zabbix maupun Grafana dapat diakses secara *online* melalui internet karena sudah di-*hosting* melalui IP Publik dan domain, sehingga memudahkan administrator dalam melakukan monitoring secara fleksibel.



Gambar 2. Diagram alir perancangan dan cara kerja sistem

Gambar 2 menggambarkan alur kerja sistem monitoring mulai dari instalasi server Zabbix dan Grafana dalam satu VM berbasis Ubuntu yang dijalankan menggunakan Docker Compose. Setelah server siap, bot Telegram dikonfigurasi dan diintegrasikan ke Zabbix untuk mengirimkan notifikasi otomatis. Selanjutnya, perangkat jaringan dikonfigurasi dengan SNMP dan diinput ke Zabbix. Jika perangkat gagal mengirim data, konfigurasi ulang SNMP dilakukan dengan *security* atau versi SNMP yang berbeda. Setelah koneksi berhasil, *threshold* ditentukan di Zabbix untuk memicu notifikasi ketika parameter seperti CPU atau *bandwidth* melewati nilai ambang batas. Selain itu, data divisualisasikan oleh *dashboard* monitoring Grafana secara *real-time*. Jika terjadi *downtime* atau masalah jaringan, sistem akan mengirimkan notifikasi melalui Telegram, sehingga administrator akan segera melakukan *troubleshoot*. Setelah perangkat kembali normal, monitoring akan berfungsi kembali seperti semula.

2.2 Perancangan Sistem Monitoring

Realisasi sistem monitoring dilakukan pada komputer server milik SMKN 4 Bandung dengan memanfaatkan platform virtualisasi VMware. VMware merupakan perangkat lunak virtualisasi yang membantu memberikan kemampuan untuk satu server fisik menjalankan beberapa *virtual machine* (VM) secara bersamaan. Setiap VM dapat memiliki sistem operasi dan fungsi yang berbeda, namun tetap berjalan secara masing-masing di atas perangkat keras yang sama. Dalam implementasi ini, salah satu VM digunakan khusus untuk membangun sistem monitoring menggunakan Zabbix dan Grafana. Penggunaan *virtual machine* memberikan beberapa keuntungan, seperti efisiensi penggunaan daya listrik, penghematan ruang penempatan server, dan kemudahan dalam proses pemeliharaan [13].

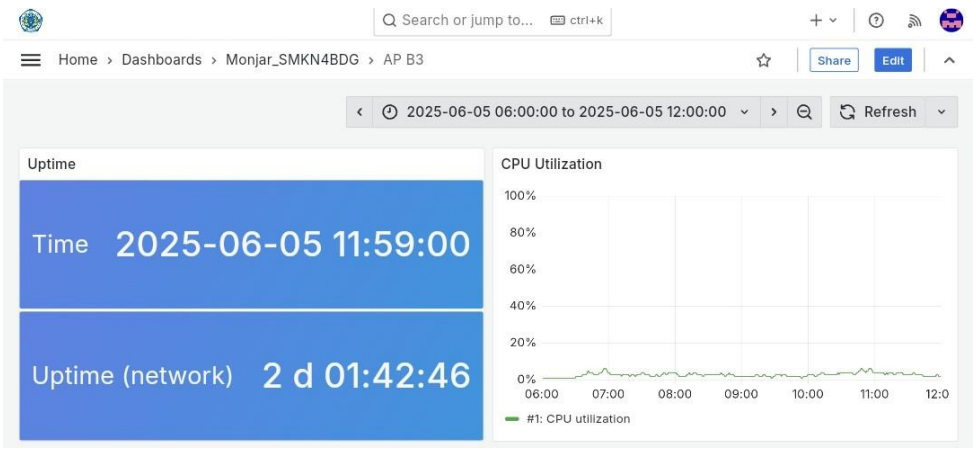
Dalam perancangan sistem monitoring ini, Docker Compose digunakan untuk mempermudah proses *deploy* dua layanan utama, yaitu Zabbix dan Grafana, ke dalam satu VM. Dengan satu *file* konfigurasi, kedua layanan dapat dijalankan secara otomatis dan saling terhubung tanpa harus melakukan instalasi dan konfigurasi secara manual satu per satu. Hal ini membuat proses *set-up* menjadi lebih cepat, efisien, dan mudah untuk direplikasi atau dipindahkan ke server lain jika dibutuhkan. Docker Compose sendiri adalah *tools* yang memungkinkan dapat menjalankan beberapa *container* secara bersamaan dalam satu file saja. Semua pengaturan layanannya ditulis dalam satu *file* YAML, sehingga lebih mudah dikelola, terutama untuk sistem yang kompleks. Komponen penting dalam Docker Compose meliputi *services* untuk mendefinisikan tiap layanan, *image* untuk menentukan *image* Docker yang digunakan, *volume* untuk menyimpan data agar tetap ada meskipun *container* dimatikan, *network* untuk mengatur koneksi antar *container*, dan *environment variables* untuk mengatur konfigurasi penting seperti *password*, API *key*, dan pengaturan lainnya [14].

Gambar 3. Zabbix sudah terkoneksi dengan perangkat melalui SNMP

Gambar 3 menunjukkan *web interface* Zabbix yang telah berhasil menerima data dari beberapa perangkat *access point* (AP) yang memiliki status "*Enabled*" dan "*Available*" pada kolom SNMP, yang menandakan bahwa konfigurasi SNMP antara perangkat dan server Zabbix sudah berjalan dengan baik. Terdapat juga kolom *Items*, *Triggers*, dan *Graphs* yang telah aktif, menunjukkan bahwa data monitoring berhasil ditarik dan ditampilkan oleh sistem Zabbix. Adapun data yang dikirim oleh perangkat melalui protokol SNMP berbentuk *Object Identifier* (OID), yaitu serangkaian angka yang merepresentasikan informasi spesifik dari perangkat, seperti status *interface*, penggunaan *bandwidth*, kondisi *storage*, dan lainnya. OID ini dibaca dan diterjemahkan oleh Zabbix sesuai dengan *template* atau *query* yang sudah ditentukan, sehingga data mentah dari perangkat dapat diubah menjadi informasi yang bisa dimonitor secara visual dan *real-time* [15].

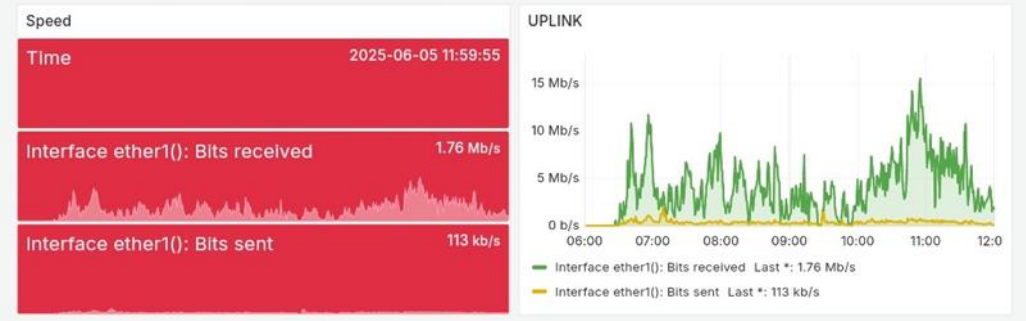
Sedangkan, *dashboard* monitoring akan divisualisasikan melalui Grafana yang memiliki keunggulan dalam menyajikan data secara visual dan menarik, sehingga memudahkan administrator dalam memantau dan menganalisa kondisi perangkat jaringan secara *real-time*. Dengan fitur analisis dan

visualisasi data seperti grafik dan tabel, Grafana membantu mengidentifikasi pola, tren, maupun anomali pada performa jaringan [16]. *Dashboard* monitoring berfungsi sebagai tampilan utama untuk menampilkan data perangkat jaringan yang dimonitor secara *real-time*. Dalam *dashboard* ini, administrator dapat memantau berbagai parameter penting seperti penggunaan *bandwidth*, performa CPU, kapasitas memori, serta status dan durasi *uptime* perangkat untuk membantu administrator dalam mengawasi kondisi jaringan secara menyeluruh dan mengambil tindakan cepat jika sistem mendeteksi perangkat mengalami masalah [17].



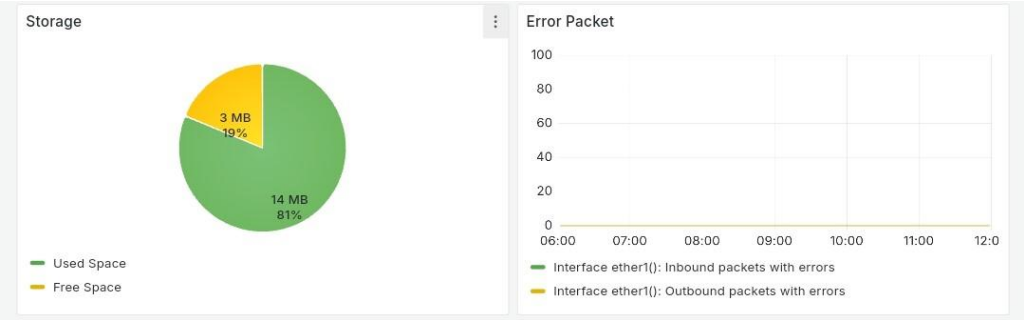
Gambar 4. Tampilan *dashboard* monitoring bagian 1

Gambar 4 menampilkan panel monitoring yang menunjukkan berapa lama perangkat telah aktif melalui informasi *uptime*, seperti waktu aktif total dan durasi koneksi jaringan (*network uptime*). Selain itu, grafik penggunaan CPU secara *real-time* digunakan untuk memantau performa prosesor, mendeteksi beban kerja berlebih (*overload*), serta aktivitas tidak normal yang dapat memengaruhi kestabilan atau menandakan kerusakan pada perangkat. Data ini penting untuk evaluasi kestabilan sistem dan mencegah penggunaan perangkat secara berlebihan (*overused*).



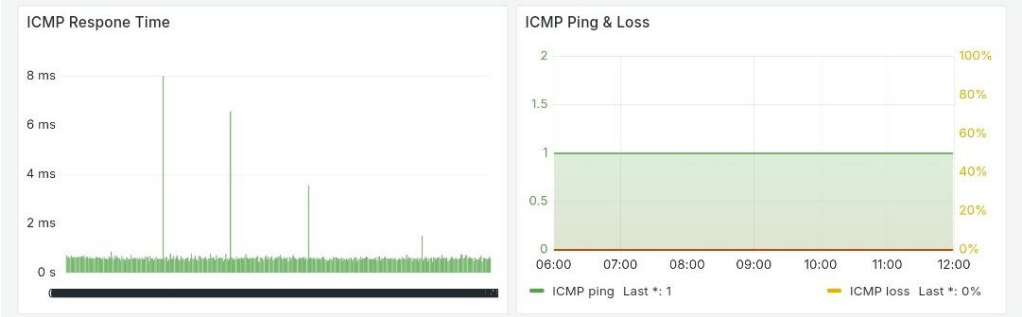
Gambar 5. Tampilan *dashboard* monitoring bagian 2

Gambar 5 yang menampilkan panel kecepatan *interface* jaringan dan grafik *uplink* yang menunjukkan aktivitas *upload* dan *download* secara *real-time*. Informasi ini digunakan untuk memantau seberapa besar penggunaan *bandwidth* pada jaringan, serta mendeteksi apakah jaringan berjalan normal atau mengalami beban berlebih. Grafik ini juga membantu mengidentifikasi gangguan koneksi yang menyebabkan penurunan kecepatan, serta mendukung analisis lonjakan *traffic* yang bisa memengaruhi performa perangkat. Dengan demikian, administrator dapat merencanakan kapasitas jaringan dan melakukan penanganan masalah dengan lebih cepat dan tepat.



Gambar 6. Tampilan *dashboard* monitoring bagian 3

Gambar 6 menampilkan panel penggunaan *storage* dan grafik *error packet* pada *interface* perangkat jaringan. Panel *storage* menunjukkan kapasitas memori yang terpakai dan tersisa untuk memastikan sistem tetap berjalan optimal. Sementara grafik *error packet* memantau jumlah paket data yang gagal dikirim atau diterima, berguna untuk mendeteksi gangguan koneksi, kesalahan konfigurasi, atau potensi kerusakan perangkat. Pemantauan ini penting untuk menjaga performa jaringan tetap stabil dan mempercepat proses *troubleshooting* jika terjadi masalah.



Gambar 7. Tampilan *dashboard* monitoring bagian 4

Gambar 7 menampilkan grafik ICMP *Response Time* serta ICMP *Ping & Loss*. Grafik ini digunakan untuk memantau kecepatan respons perangkat terhadap permintaan ping dan mendeteksi kestabilan koneksi. Nilai ping di representasikan dengan "1" menandakan perangkat merespons, sedangkan "0" menunjukkan perangkat tidak merespons atau *unreachable*. Sementara itu, ICMP *Loss* menunjukkan adanya paket data yang hilang selama pengiriman. Panel ini penting untuk mendeteksi gangguan jaringan secara dini dan memastikan kualitas komunikasi setiap perangkat tetap stabil.

2.3 Skenario Pengujian

Dari total 69 perangkat jaringan yang terkoneksi dengan sistem monitoring, pengujian dilakukan dengan menggunakan metode *stress testing* terhadap perangkat *access point* yang terhubung secara nirkabel dengan laptop yang dianggap sebagai client. *Stress testing* adalah metode pengujian sistem di bawah kondisi ekstrem untuk mengevaluasi ketahanan dan performa perangkat, seperti saat terjadi lonjakan *traffic* atau gangguan teknis [18]. Pengujian dilakukan dengan membebani perangkat *access point* hingga batas kapasitas maksimal, yaitu 100 Mbps guna memicu lonjakan CPU. Pada percobaan ini penulis menggunakan aplikasi Mikrotik Bandwidth Test untuk membantu proses pengujian. Jika ambang batas CPU tercapai, maka Zabbix akan mengirimkan notifikasi melalui bot Telegram. Selama pengujian, diamati juga kestabilan koneksi ICMP dan kemungkinan *packet loss* untuk memastikan sistem monitoring mampu mendeteksi perubahan kondisi perangkat secara *real-time*. Pengujian juga bertujuan untuk melihat stabilitas jaringan dan efektivitas sistem monitoring dalam mendeteksi perubahan kondisi perangkat. Seiring meningkatnya jumlah pengguna internet, beban *traffic* pun meningkat dan dapat memengaruhi performa jaringan. Monitoring diperlukan untuk memastikan jaringan tetap stabil dan aliran data berjalan lancar [19].

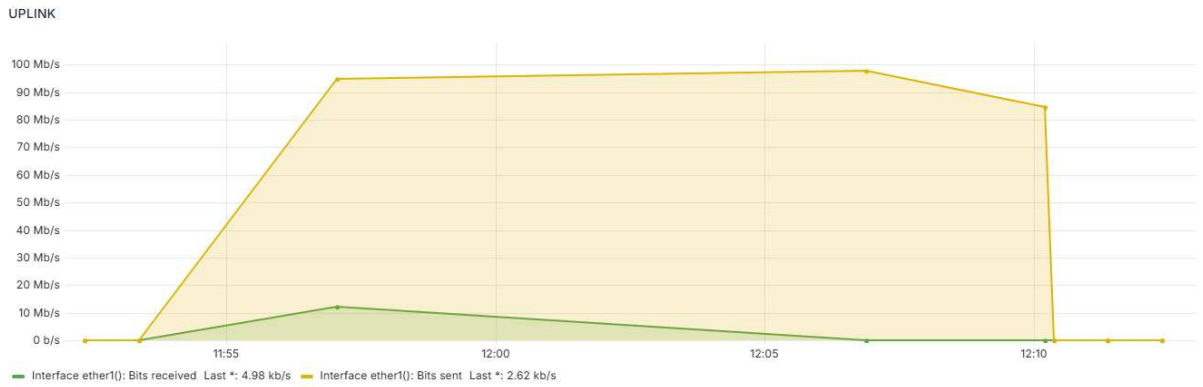
3. HASIL DAN PEMBAHASAN

Pengujian sistem monitoring dilakukan dengan mengamati perubahan kondisi pada salah satu perangkat jaringan, yaitu *access point*. Pengamatan dimulai dari kondisi penggunaan normal, kemudian perangkat diberi beban *traffic* hingga kembali ke kondisi stabil seperti semula. Rentang waktu yang diamati berlangsung dari pukul 11:52 hingga 12:11 WIB, dan seluruh data diambil langsung melalui *dashboard* monitoring Grafana. Selama periode tersebut, beberapa parameter diamati, seperti penggunaan *traffic upload* dan *download* dari proses pengujian *stress test* terhadap lonjakan beban *traffic* yang sengaja dimaksimalkan untuk mencapai batas performa kinerja perangkat. Kemudian, ketika perangkat mengalami kenaikan *traffic* secara signifikan, persentase kinerja CPU terpaksa melonjak akibat pemrosesan data yang semakin berat dan intensif. Bahkan kondisi ICMP ping dan *loss* turut diamati karena dapat merepresentasikan stabilitas jaringan ketika perangkat menerima beban tinggi, misalnya munculnya *delay* atau kehilangan paket dalam transmisi yang menjadi indikator performa jaringan terganggu. Serta diamati juga notifikasi yang dikirim otomatis oleh bot Telegram ketika perangkat jaringan terdeteksi bermasalah atau melampaui nilai ambang batas yang telah ditentukan. Data-data tersebut menjadi dasar dalam menganalisa kemampuan sistem monitoring dalam merespons suatu perubahan kondisi perangkat secara *real-time*, serta membuktikan kemampuan sistem monitoring dalam memberikan data informasi yang akurat dan cepat kepada administrator jaringan.

Tabel 1. Hasil pengujian *uplink* untuk *upload* dan *download*

No	Waktu	*Interface ether1(Bits received)*	*Interface ether1(Bits sent)*
1	11:52:22	11,9 kb/s	5,18 kb/s
2	11:53:22	7,61 kb/s	3,62 kb/s
3	11:57:03	12,2 Mb/s	94,8 Mb/s
4	12:06:53	8,53 kb/s	97,6 Mb/s
5	12:10:11	7,97 kb/s	84,6 Mb/s
6	12:10:22	19,5 kb/s	20,9 kb/s
7	12:11:22	4,89 kb/s	2,37 kb/s

Tabel 1 menunjukkan data hasil pengamatan yang dilakukan dimulai ketika kondisi perangkat dalam penggunaan normal pada pukul 11:52 hingga 15:53 sebelum dilakukan pengujian. Setelah itu, pengujian terhadap pembebanan *traffic* dilakukan, namun baru terdeteksi lonjakan *traffic* pada pukul 11:57 hingga 12:10 secara signifikan pada *traffic* penggunaan *upload*. Setelah pukul 12:10 pengujian dihentikan, lalu kondisi perangkat kembali stabil seperti penggunaan normal hingga pukul 12:11.



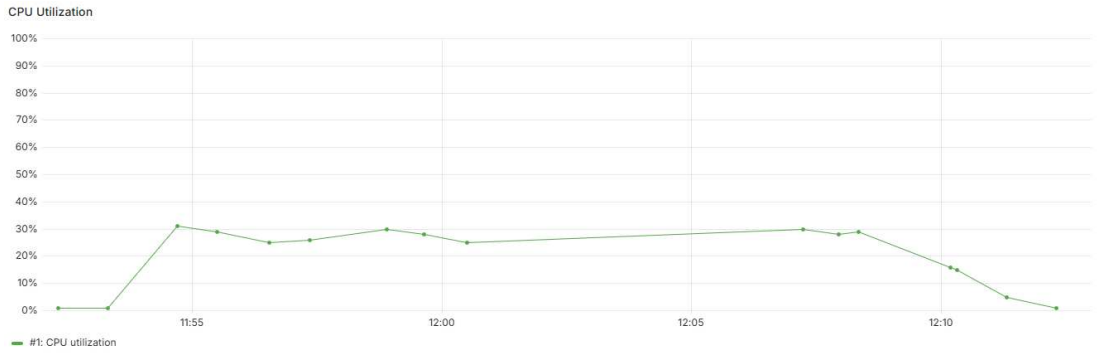
Gambar 8. Grafik pengujian *uplink* untuk *upload* dan *download*

Pada Gambar 8 menunjukkan sebuah grafik terhadap kondisi *traffic* penggunaan *upload* dan *download* yang diambil dari panel tampilan *dashboard* monitoring Grafana. Seperti berdasarkan data yang didapatkan dalam Tabel 1, titik puncak *traffic* penggunaan *upload* mencapai 97,6 Mbps pada pukul 12:06, sedangkan *traffic* penggunaan *download* hanya mencapai 12,2 Mbps dari hasil pengujian yang dilakukan terhadap pembebanan *traffic* yang dilakukan.

Tabel 2. Data persentase kinerja CPU ketika pengujian

No	Waktu	*CPU utilization*	No	Waktu	*CPU utilization*
1	11:52:18	1%	9	12:00:29	25%
2	11:53:18	1%	10	12:07:14	30%
3	11:54:41	31%	11	12:07:56	28%
4	11:55:29	29%	12	12:08:20	29%
5	11:56:32	25%	13	12:10:11	16%
6	11:57:21	26%	14	12:10:18	15%
7	11:58:53	30%	15	12:11:18	5%
8	11:59:38	28%			

Berdasarkan hasil pengamatan, penggunaan CPU mulai meningkat dari kondisi normal sebesar 1% pada pukul 11:52 hingga lonjakan mencapai 31% pada pukul 11:54, bersamaan dengan dimulainya *stress test* yang menyebabkan lonjakan *traffic upload*. Persentase penggunaan CPU tetap berada pada kisaran yang cukup tinggi, yaitu sekitar 25–31%, selama periode pengujian beban jaringan yang dilakukan. Setelah pengujian dihentikan, kondisi persentase penggunaan CPU menurun secara bertahap hingga mencapai 5% pada pukul 12:11 yang menandakan perangkat kembali ke kondisi stabil. Hal ini menunjukkan bahwa sistem monitoring mampu merekam dan menampilkan korelasi antara lonjakan *traffic* dan peningkatan beban kerja perangkat secara *real-time*.



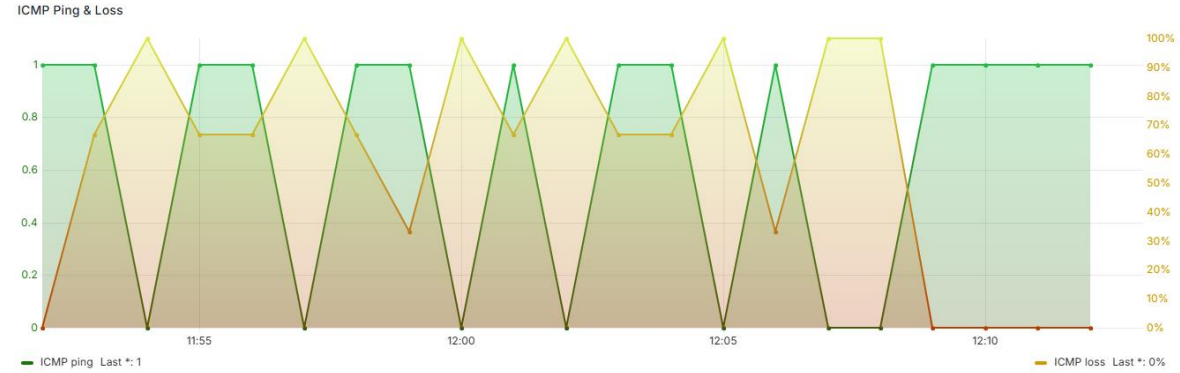
Gambar 9. Grafik persentase kinerja CPU ketika pengujian

Berdasarkan data yang didapatkan konsisten setiap interval satu menit, pada Gambar 9 yang menunjukkan grafik penggunaan CPU dengan pola yang sesuai terhadap peningkatan beban *traffic* jaringan. Pada awal pengamatan, persentase CPU berada dalam kondisi normal. Namun, mulai pukul 11:54, terjadi peningkatan bertahap seiring masuknya beban *traffic* yang terus meningkat. Penggunaan CPU terus berada di kisaran 25–31% selama proses *stress test* berlangsung yang mencerminkan respons perangkat terhadap lonjakan *traffic* yang terjadi. Setelah beban *traffic* dihentikan sekitar pukul 12:10, grafik persentase kinerja CPU menunjukkan tren penurunan secara bertahap dan kembali ke kondisi normal yang menjadi bukti bahwa sistem monitoring mampu merekam perubahan performa perangkat secara *real-time* dan akurat.

Tabel 3. Data hasil pengujian ICMP ping & *loss*

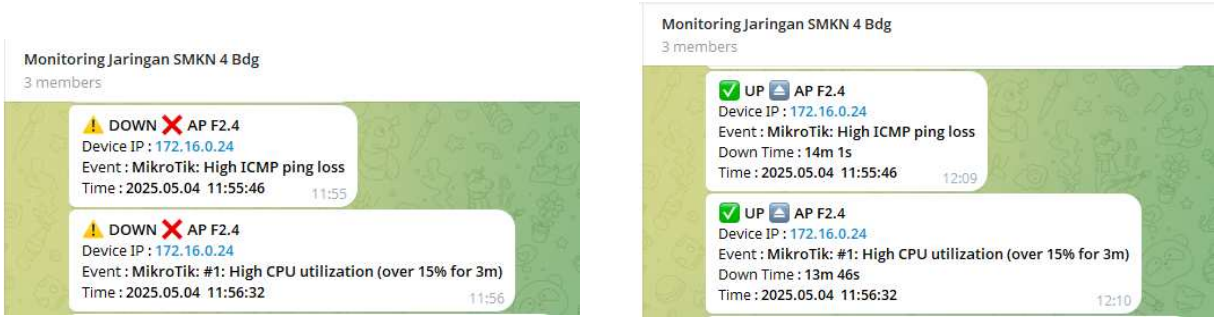
No	Waktu	ICMP ping	ICMP *loss*	No	Waktu	ICMP ping	ICMP *loss*
1	11:52:00	1	0%	11	12:02:00	0	100%
2	11:53:00	1	66,7%	12	12:03:00	1	66,7%
3	11:54:00	0	100%	13	12:04:00	1	66,7%
4	11:55:00	1	66,7%	14	12:05:00	0	100%
5	11:56:00	1	66,7%	15	12:06:00	1	33,3%
6	11:57:00	0	100%	16	12:07:00	0	100%
7	11:58:00	1	66,7%	17	12:08:00	0	100%
8	11:59:00	1	33,3%	18	12:09:00	1	0%
9	12:00:00	0	100%	19	12:10:00	1	0%
10	12:01:00	1	66,7%	20	12:11:00	1	0%

Berdasarkan Tabel 3, data ICMP ping dan loss menunjukkan adanya perubahan stabilitas koneksi jaringan selama proses pengujian berlangsung. Pada awal pengamatan pukul 11:52 perangkat terdeteksi dapat merespons ping dengan baik dan menunjukkan nilai *loss* rendah. Namun, mulai pukul 11:54 hingga sekitar pukul 12:08, terjadi perubahan kondisi nilai ICMP ping serta peningkatan ICMP *loss* yang signifikan, bahkan beberapa kali mencapai 100%, yang menunjukkan adanya gangguan konektivitas selama beban *traffic* berlangsung. Setelah pukul 12:09, kondisi kembali stabil sampai pukul 12:11. Hal ini menandakan bahwa sistem monitoring berhasil merekam perubahan respons terhadap stabilitas koneksi jaringan secara *real-time* melalui parameter ICMP ping dan *loss*.



Gambar 10. Grafik hasil pengujian ICMP ping & *loss*

Gambar 10 menunjukkan grafik ICMP ping dan *loss* selama pengujian. Nilai ICMP ping yang bernilai 1 menunjukkan perangkat merespons (terhubung), sedangkan 0 berarti tidak merespons (tidak terhubung). Sementara itu, ICMP loss menunjukkan persentase paket yang hilang. Jika loss mencapai 100%, berarti semua paket gagal terkirim yang dapat menandakan gangguan konektivitas. Grafik tersebut menunjukkan kondisi jaringan yang tidak stabil saat pengujian, dengan nilai ping dan loss yang naik turun, sebelum akhirnya kembali normal.



(a) **(b)**

Gambar 11. Notifikasi Bot Telegram (a) terdeteksi *down* dan (b) ketika Kembali *up*

Gambar 11 yang menunjukkan notifikasi bot Telegram saat perangkat *access point* terdeteksi bermasalah (a) dan saat kembali normal (b). Pada kondisi *down*, sistem mendeteksi lonjakan CPU di atas ambang 15% selama lebih dari 3 menit dan ICMP loss yang tinggi, sesuai dengan data pada grafik dan tabel sebelumnya. Ketika kondisi membaik, yaitu CPU kembali di bawah ambang batas yang ditentukan pada saat pengujian dan ICMP loss kembali menjadi 0%, sistem secara otomatis mengirim notifikasi bahwa perangkat telah kembali *up* atau kembali berada dalam kondisi normal.

Tabel 4. Standarisasi *packet loss* menurut Tiphon

Kategori Degradasi	*Packet Loss (%)*	*Indeks*
Sangat Bagus	0 s.d 2	4
Bagus	3 s.d 14	3
Sedang	15 s.d 24	2
Jelek	25	1

Berdasarkan hasil pengujian, nilai *packet loss* yang didapatkan oleh ICMP loss beberapa kali menunjukkan persentase hingga 100%, yang menurut standar Tiphon [20] pada Tabel 4 termasuk dalam kategori "Jelek" karena berada di atas 25%. Setelah kondisi jaringan kembali stabil, nilai *packet loss* menurun hingga 0% dan masuk kategori "Sangat Bagus". Hasil ini menunjukkan bahwa sistem monitoring mampu mendeteksi perubahan kualitas jaringan secara tepat dan sesuai dengan standar evaluasi yang berlaku.

Tabel 5. Standarisasi administrator SMKN 4 Bandung untuk monitoring

Parameter	Kategori	*Threshold*	Durasi
CPU Utilization	*Warning*	> 60%	≥ 5 menit
	Downtime	> 80%	≥ 10 menit
ICMP Ping & Loss	*Warning*	> 20%	≥ 3 menit
	Downtime	> 80%	≥ 3 menit
ICMP Ping Unavailable	*Warning*	Tidak ada *respons* ping	≥ 1 menit
	Downtime	Tidak ada *respons* ping	≥ 3 menit
Bandwidth Usage (Interface)	*Warning*	> 80% dari kapasitas	≥ 5 menit
	Downtime	> 95% dari kapasitas	≥ 10 menit
Interface Error Rate	*Warning*	> 250 *error*	≥ 10 menit
	Downtime	> 750 *error*	≥ 10 menit
ICMP Ping Response Time	*Warning*	> 250 ms	≥ 3 menit
	Downtime	> 500 ms	≥ 5 menit
Storage Usage	*Warning*	> 80% kapasitas disk/log	≥ 10 menit
	Downtime	> 90% kapasitas disk/log	≥ 10 menit

Selain mengacu pada standar umum seperti Tiphon, administrator jaringan SMKN 4 Bandung juga menetapkan standarisasi internal pada Tabel 5 sesuai dengan kebutuhan dan kondisi perangkat jaringan yang digunakan. Standar ini mencakup parameter penting seperti CPU *Utilization*, ICMP *Ping & Loss*, *Bandwidth Usage*, *Interface Error Rate*, ICMP *Response Time*, dan *Storage Usage* yang diklasifikasikan ke dalam kategori *Warning* dan *Downtime* dengan ambang batas dan durasi tertentu. Penetapan ini bertujuan agar sistem monitoring dapat mendeteksi potensi gangguan secara lebih akurat, memberikan notifikasi tepat waktu, serta mendukung pengambilan keputusan cepat guna menjaga stabilitas jaringan, kelangsungan pembelajaran, dan dapat membantu evaluasi infrastruktur jaringan di lingkungan SMKN 4 Bandung.

4. KESIMPULAN

Berdasarkan hasil perancangan dan implementasi sistem monitoring perangkat jaringan di SMKN 4 Bandung, dapat disimpulkan bahwa pemanfaatan protokol SNMP melalui Zabbix berhasil mengintegrasikan terhadap seluruh perangkat jaringan yang terpasang di lingkungan SMKN 4 Bandung secara *real-time*, dilengkapi dengan visualisasi data melalui Grafana dan notifikasi otomatis menggunakan bot Telegram. Sistem ini mampu menampilkan data performa perangkat seperti status *uptime*, pengunaan *bandwidth*, kondisi CPU, pemakaian *storage* dan kondisi ICMP dalam *dashboard* monitoring yang informatif serta dapat merekam histori kinerja perangkat untuk kebutuhan evaluasi. Adapun untuk pengembangan lebih lanjut, disarankan agar sistem monitoring dibuat lebih responsif dengan meminimalkan delay pengambilan data dan notifikasi, membuat pusat pengupulan data perangkat dan tampilan monitoring dalam satu *web interface* yang sama, serta menambahkan fitur otomatisasi tindakan seperti *restart* atau *disable interface* menggunakan SNMP *write access* sebagai upaya mitigasi awal sebelum administrator jaringan melakukan tindakan, sehingga potensi dampak gangguan dapat diminimalisir lebih cepat.

UCAPAN TERIMA KASIH

Ucapan terima kasih diberikan kepada Politeknik Negeri Bandung yang telah memberikan dana bantuan penelitian hingga dapat dilaksanakan dan terpublikasi, serta kepada SMKN 4 Bandung yang telah memberikan kesempatan untuk bekerja sama dalam penelitian ini hingga dapat terwujud.

REFERENSI

[1] A. ripai, "Apa itu *Downtime*? Yuk Kenali Penyebabnya!," herza.id, 7 Maret 2023. [Online]. Available: https://herza.id/blog/apa-itu-downtime/.

[2] K. Kasmono, "UNBK 208 siswa SMP 5 Sungailiat tertunda," babel.antaranews.com, 23 April 2018. [Online]. Available: https://babel.antaranews.com/berita/75378/unbk-208-siswa-smp-5-sungailiat-tertunda?utm_source.

[3] S. Surono and M. B. Hanif, "Penerapan Protokol *Simple Network Management Protocol* Monitoring LIBRE NMS Pada Jaringan Internet," *Journal of Informatics Education,* vol. 7, no. 2, pp. 273-284, 2024.

[4] E. Gamess and S. Hernandez, "Performance Evaluation of SNMPv1/2c/3 using Different Security Models on Raspberry Pi," *(IJACSA) International Journal of Advanced Computer Science and Applications,* vol. 12, no. 11, pp. 1-9, 2021.

[5] A. H. Mohamad and F. M. Ahmed, "*Development of an SNMP v3 agent for user modelling in LAN environments for thin films by using Dynamic systems modelling* (preprint)," 5 Juni 2023. [Online]. Available: https://www.researchsquare.com/article/rs-2948744/v1.

[6] R. A. Nugroho and P. Rosyani, "Implementasi Monitoring Perangkat *Environment* Menggunakan Zabbix pada Data Center Pusat Data Sarana Informasi (PDSI)," *OKTAL : Jurnal Ilmu Komputer dan Science,* vol. 2, no. 7, pp. 1846-1873, 2023.

[7] R. Olups, A. D. Vacche and P. Uytterhoeven, Zabbix: *Enterprise Network Monitoring Made Easy*, Birmingham: Packt Publishing Ltd., 2017.

[8] A. Mardiyono, W. Sholihah and F. Hakim, "*Mobile-based Network Monitoring System Using Zabbix and Telegram*," in *IEEE*, Yogyakarta, 2020.

[9] M. R. A. Arsandi and A. Syaripudin, "Perancangan Sistem Monitoring Jaringan berbasis Web Server Terintegrasi Zabbix dan Notifikasi Telegram Pada PT Time Excelindo," *OKTAL : Jurnal Ilmu Komputer dan Science,* vol. 3, no. 6, pp. 1553-1561, 2024.

[10] M. Chakraborty and A. P. Kundan, *Monitoring Cloud-Native Applications*, New York: Apress Media, 2021.

[11] 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. 247−255, 2021.

[12] A. H. Alhilali, A. A. Farawn and A. Y. Mjhool, "*Design and implement a real-time network traffic management system using SNMP protocol*," *Eastern-European Journal of Enterprise Technologies,* vol. 5, no. 9, p. 35–44, 2023.

[13] S. N. Khasanah and S. J. Kuryanti, "Rancangan Virtualisasi Server Menggunakan VMWare Vsphare," *Jurnal Evolusi,* vol. 7, no. 1, pp. 42-46, 2019.

[14] S. Wulandari, "Docker Compose: Pengertian, Kegunaan, dan Kelebihannya," dibimbing.id, 12 Agustus 2023. [Online]. Available: https://dibimbing.id/blog/detail/docker-compose-%20pengertian-kegunaan-dan-kelebihannya.

[15] A. Hizriadi, R. Shiddiq, I. Jaya and S. Prayudani, "*Network Device Monitoring System based on Geographic Information System and Simple Network Management Protocol*," *JITE (Journal Of Informatics And Telecommunication Engineering),* vol. 3, no. 2, pp. 216-223, 2020.

[16] G. Purnama, B. Yuliadi, R. L. S. Putra, A. Supriyadi and M. J. Saputra, "Optimasi Pengambilan Keputusan Akademik Perguruan Tinggi Menggunakan Visualisasi Data dan Analisis Performa dengan Implementasi *Dashboard* Grafana," *JSAI: Journal Scientific and Applied Informatics,* vol. 7, no. 3, pp. 647-654, 2024.

[17] A. Pradana, I. R.Widiasari and R. Efendi, "Implementasi Sistem Monitoring Jaringan Menggunakan Zabbix Berbasis SNMP," *AITI: Jurnal Teknologi Informasi,* vol. 19, no. 2, pp. 248-262, 2022.

[18] Firstian, "*Stress Testing* : Mengenal Definisi, Tipe dan *Tools Stress Testing*," www.wowrack.com, 16 Oktober 2023. [Online].

[19] Nendi and F. Maulana, "Monitoring *Traffic* Berbasis SNMP pada Jaringan Perumahan Permata Puri Harmoni 2," *Jurnal Sains dan Teknologi,* vol. 5, no. 3, pp. 735-740, 2024.

[20] H. A. S. A. Nugroho, Sonhaji and A. C. Prasetyo, "Evaluasi Kinerja Jaringan WiFi Mahasiswa: Analisis *Throughput, Delay, Jitter, dan Packet loss*," *Jurnal BATIRSI,* vol. 8, no. 1, pp. 23-27, 2024.