

PENERAPAN PROMETHEUS TERINTEGRASI GRAFANA UNTUK MONITORING VIRTUALISASI BERBASIS PROXMOX VIRTUAL ENVIRONMENT

Sahimin, I Putu Hariyadi, Khairan Marzuki

Universitas Bumigora, Mataram, Indonesia

e-mail: sahimin789@gmail.com

Abstrak

Virtualisasi menjadi salah satu solusi utama dalam pengelolaan sumber daya komputasi. *Proxmox Virtual Environment (PVE)* merupakan salah satu *hypervisor* yang digunakan untuk manajemen virtualisasi. *Monitoring* sumber daya virtualisasi menjadi sangat penting dilakukan secara terus-menerus untuk mengetahui performansi, ketersediaan, dan stabilitas layanan berjalan dengan yang baik. Umumnya *monitoring* dilakukan dimana pengguna atau *administrator* tetap berada di depan layar komputer sehingga menjadi kurang efektif. Penerapan *Prometheus* dan *Grafana* yang diintegrasikan dengan *Telegram* menjadi solusi *monitoring* PVE yang memungkinkan pengguna mendapatkan informasi secara cepat dan akurat meskipun tidak melakukan pemantauan melalui layar komputer yakni melalui notifikasi *Telegram*. Dalam penelitian ini menggunakan metode *Network Development Life Cycle (NDLC)* sebagai pendekatan dalam perancangan dan pengembangan sistem. Terdapat 4 tahapan *NDLC* yang digunakan yaitu Analisis, Desain, *Simulation Prototyping* dan *Monitoring*. Pada tahap analisis dilakukan *proses* pengumpulan dan evaluasi data yang berkaitan dengan kebutuhan sistem. Tahap desain buat rancangan jaringan uji coba dan rancangan pengalaman *IP* serta rancangan sistem termasuk kebutuhan perangkat keras dan lunak. Tahap *simulation prototyping* mencakup *proses* instalasi dan konfigurasi pada PVE, pembuatan *container* dan konfigurasi *Prometheus* serta *Grafana* dilanjutkan dengan pengujian sistem serta analisis terhadap hasil ujicoba tersebut. Pada tahap *monitoring* dilakukan *monitoring* terhadap tiga komponen utama yaitu CPU, memori, dan penyimpanan, untuk menilai kinerja sistem yang telah diterapkan. Penerapan *Prometheus* yang terintegrasi *Grafana* sebagai alat *monitoring* pada virtualisasi berbasis PVE dapat menghasilkan sebuah sistem yang mampu membantu pengguna dalam memantau kondisi *server* secara *efisien* dan *real-time*.

Kata kunci: *Prometheus*, PVE, *Grafana*, *Monitoring*, Virtualisasi, *NDLC*

Abstract

Virtualization is a key solution for managing computing resources. Proxmox Virtual Environment (PVE) is a hypervisor used to manage virtualization. Continuous monitoring of virtualization resources is crucial to ensure proper performance, availability, and stability of services. Monitoring is typically performed while the user or administrator remains in front of a computer screen, making it less effective. The integration of Prometheus and Grafana with Telegram provides a PVE monitoring solution that allows users to obtain fast and accurate information, even without monitoring via a computer screen, through Telegram notifications. This study uses the Network Development Life Cycle (NDLC) approach to system design and development. The NDLC comprises four stages: Analysis, Design, Simulation Prototyping, and Monitoring. The analysis phase involves collecting and evaluating data related to system requirements. The design phase involves designing a trial network, IP addressing, and system design, including hardware and software requirements. The simulation prototyping phase includes the installation and configuration of PVE, container creation, and Prometheus and Grafana configuration, followed by system testing and analysis of the test results. The monitoring phase involves monitoring three main components—CPU, memory, and storage—to assess the performance of the implemented system. The implementation of Prometheus integrated with Grafana as a monitoring tool for PVE-based virtualization can result in a system that helps users monitor server conditions efficiently and in real time.

Keywords: Prometheus, PVE, Grafana, Monitoring, Virtualization, NDLC

1. Pendahuluan

Virtualisasi *server* kini menjadi bagian yang krusial dalam infrastruktur teknologi informasi. Virtualisasi adalah teknologi yang memungkinkan konversi entitas fisik menjadi bentuk *virtual* berbasis perangkat lunak [1]. Melalui teknologi ini, organisasi dapat mengoperasikan berbagai sistem operasi dan aplikasi secara bersamaan di satu perangkat keras fisik, sehingga pengelolaan sumber daya menjadi lebih efisien dan fleksibel. Salah satu *platform open-source* yang populer untuk virtualisasi adalah *Proxmox Virtual Environment (PVE)*[2]. Menurut [3] PVE merupakan salah satu manajemen *server* berbasis virtualisasi yang menawarkan keunggulan berupa lisensi gratis karena bersifat *open source*. Platform ini menggunakan distribusi *Linux Debian* dengan *kernel Ubuntu LTS* yang telah dimodifikasi, sehingga mendukung penerapan dan pengelolaan mesin virtual maupun *container*. PVE memiliki fitur manajemen *server virtual* terpusat, mendukung *High Availability (HA)*, dan dilengkapi antarmuka pengguna yang ramah [4].

Namun, seiring dengan semakin banyaknya *server virtual* dan kompleksitas sistem yang dikelola, kebutuhan untuk melakukan pemantauan performa infrastruktur secara *real-time* menjadi sangat penting. *Monitoring* tidak hanya bertujuan memastikan sistem berfungsi dengan baik [5], tetapi juga mendeteksi gangguan sedini mungkin, mengoptimalkan kinerja, dan merencanakan kapasitas sistem secara efektif. Menurut [6] *Monitoring* merupakan kegiatan yang bertujuan untuk mengatur dan mengelola sistem jaringan pada wilayah atau topologi tertentu. Dengan adanya sistem pemantauan ini, teknisi dan *administrator* dapat lebih mudah melakukan pengawasan langsung terhadap kondisi jaringan di lapangan.

Terdapat beberapa penelitian terdahulu terkait *monitoring* pada *server*. Penelitian yang dilakukan oleh [7] mengembangkan dan merancang sistem visualisasi data untuk memantau *server* yang dilakukan dengan memanfaatkan teknologi *chat bot* menggunakan bahasa pemrograman *Python* serta metode *agentless*. Sistem ini mampu menampilkan hasil *monitoring server Linux* baik dalam bentuk teks maupun grafik visualisasi data. Sistem tersebut juga dirancang dengan menggunakan tiga komponen utama, yaitu *engine* untuk penerimaan pesan, *engine* untuk pemrosesan pesan, dan *engine* untuk pengiriman pesan. Sedangkan penelitian yang dilakukan oleh [8] merancang dan menerapkan sistem *monitoring* sumber daya *server* yang telah diuji pada penggunaan *hard disk*, *CPU*, *RAM*, serta status lalu lintas jaringan *ingress* dan *egress*. Selain itu, sistem mampu mendeteksi permasalahan pada *resource server* dan secara otomatis mengirimkan notifikasi melalui *Telegram*. Sebaliknya penelitian yang dilakukan oleh [9] mengimplementasikan sistem *monitoring* jaringan dan *server* menggunakan *zabbix* yang terintegrasi dengan *Grafana* dan *Telegram*. Hasil pengujian pada penelitian tersebut menunjukkan bahwa penerapan sistem *monitoring* jaringan dan *server* menggunakan *Zabbix* di PT Cyber Network Indonesia telah membantu *network administrator* dalam memantau gangguan jaringan secara efektif sehingga dapat segera diatasi. Informasi mengenai gangguan jaringan ditampilkan melalui integrasi dengan *Grafana*, dan notifikasi juga dikirimkan secara otomatis melalui *telegram*. *Grafana* merupakan perangkat lunak *open-source* yang digunakan untuk menampilkan data pemantauan dalam bentuk grafik dan diagram. Platform ini mendukung berbagai jenis sumber data seperti *Graphite*, *Prometheus*, *Elasticsearch*, *OpenTSDB*, dan *InfluxDB* [10]. *Telegram* merupakan salah satu aplikasi perpesanan instan yang banyak digunakan untuk mengirim pesan secara cepat. Salah satu fitur unggulannya adalah *Telegram Bot*, yang dapat dimanfaatkan oleh pengguna untuk mengotomatisasi berbagai proses dalam suatu sistem. Melalui *Application Programming Interface (API)* yang disediakan, *Telegram Bot* menawarkan berbagai fungsi yang berguna. Beberapa di antaranya termasuk kemampuan untuk bertukar pesan antara pengguna dan *bot*, mengirim pengingat kepada pengguna, hingga menyampaikan pesan ke banyak penerima sekaligus [11].

Peneliti terdahulu melakukan *monitoring* menggunakan *Zabbix*, *chat bot* dan *Grafana*. Mendorong ketertarikan penulis untuk melakukan penelitian terkait *monitoring* pada platform virtualisasi yaitu *Proxmox Virtual Environment (PVE)* yang masih jarang menjadi fokus dari penelitian. Sistem *monitoring* yang dibuat menggunakan *Prometheus* yang berfungsi untuk mengumpulkan metrik dari *host* dan *Virtual Machine (VM)* dari PVE serta ditampilkan secara visual melalui *Grafana*. Fokus penelitian ini adalah untuk menganalisis penerapan *Prometheus* dan *Grafana* dalam *monitoring server* berbasis PVE, serta mengevaluasi efektivitasnya dalam membantu manajemen dan pengawasan sistem virtualisasi. *Prometheus* adalah sebuah perangkat lunak *open source* yang digunakan untuk keperluan *monitoring* dan sistem *alerting*. Karena bersifat sumber terbuka, pengguna memiliki kebebasan untuk menggunakan, memodifikasi, dan menyesuaikan kode sumber sesuai dengan kebutuhan masing-masing [12]. *Prometheus* mampu mengumpulkan metrik berupa data sumber daya server. Data yang berhasil

dikumpulkan tersebut dapat divisualisasikan menggunakan aplikasi pendukung. Salah satu aplikasi yang umum digunakan untuk menampilkan data metrik dari hasil pemantauan jaringan adalah Grafana[13].

2. Metode Penelitian

Metode penelitian yang penulis gunakan dalam penyusunan skripsi ini adalah metodologi *Network Development Life Cycle (NDLC)*. Menurut [14] *NDLC* merupakan fondasi utama dalam tahapan penyusunan rancangan jaringan komputer. *NDLC* adalah sebuah model yang menggambarkan siklus proses perancangan sistem jaringan komputer. Dalam penelitian ini, tahapan yang dilakukan mengikuti metode pengembangan jaringan, pada gambar 1. Metode *NDLC* memiliki enam fase yaitu *Analysis, Design, Simulation Prototyping, Implementation, Monitoring, dan Management*.



Gambar 1. Metode NDLC [14]

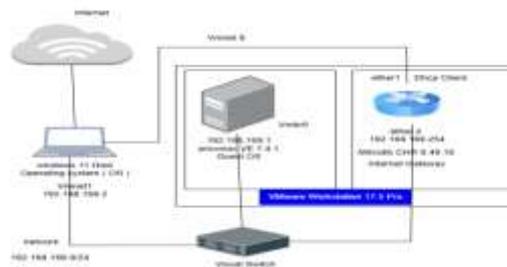
Dari enam tahapan yang ada pada metodologi *NDLC* tersebut, penulis hanya menggunakan empat tahapan yaitu *analysis, design, simulation prototyping* dan *monitoring*.

A. Analysis

Pada tahap analisis ini, peneliti mengumpulkan data melalui studi literatur, dengan memanfaatkan jurnal, buku, skripsi, dan artikel ilmiah lainnya yang berkaitan dengan topik penelitian. Setelah data terkumpul, data tersebut kemudian dianalisis.

B. Design

Tahap ini melibatkan pembuatan rancangan yaitu rancangan jaringan uji coba, rancangan pengalaman IP, rancangan sistem monitoring dan kebutuhan perangkat keras dan perangkat lunak. Pada tahap rancangan uji coba, simulasi dilakukan melalui virtualisasi menggunakan *VMWare Workstation* sebagai *Host* pada sebuah *notebook* dengan sistem operasi *Windows 11*. Di dalam *VMWare Workstation*, dibuat *guest Virtual machine* yang menggunakan sistem operasi *PVE 7.4.1* terlihat pada gambar 2.



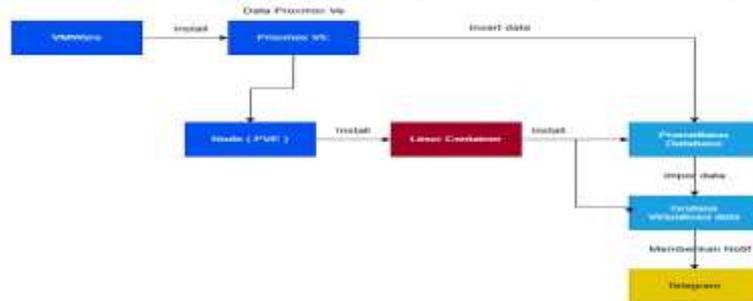
Gambar 2. Rancangan Jaringan Uji Coba

Terlihat alamat jaringan yang digunakan adalah 192.168.169.0/24 dengan alokasi pengalaman IP 192.168.169.2 untuk *interface windows 11* dan IP 192.168.169.1 untuk *interface vmbr0* di *Guest Os Proxmox* dan *Mikrotik Chr* memiliki 2 *interface* yaitu *ether1* dengan jenis *Nat* dan *ether2* dengan menggunakan alamat IP 192.168.169.254 dengan jenis koneksi *host-only*. Sedangkan *ether1* dialokasikan secara dinamis sehingga bertindak sebagai *DHCP client*. Sedangkan untuk rancangan pengalaman IP dari jaringan uji coba ini menggunakan *Network Class C* dengan 1 alamat *network* yaitu 192.168.169.0/24 dengan alokasi IP meliputi 192.168.169.1 untuk *interface Vmbr0* di *OS Proxmox* dan 192.168.169.2 untuk *interface vlnet1 Windows 11*. Untuk keterangan pengalaman yang digunakan pada jaringan uji coba ini dapat dilihat pada tabel 1.

Tabel 1. Rancangan Pengalamatan IP

No	Perangkat	Interfaces	IP Address	Gateway
1	Proxmox Ve	Vmbr0	192.168.169.1/24	192.168.169.254
2	Mikrotik CHR	Ether 1	Dhcp client	
3	Windows 11	Ether 2	192.168.169.254/24	
4	CT Prometheus dan Grafana	Vmnet1	192.168.169.2/24	
		Eth2	192.168.169.100/24	192.168.169.254

Sedangkan rancangan sistem *monitoring* pada penelitian ini, seperti terlihat pada gambar 3.



Gambar 3. Rancangan Sistem *Monitoring*

Terlihat pada gambar 3 pada *Proxmox* akan di instalasi dan konfigurasi *Container* Dengan sistem operasi *Debian 11* dengan IP 192.168.169.100. yang akan digunakan sebagai tempat instalasi *Prometheus* dan *Grafana*, dimana *Prometheus* sebagai menampung data dari *proxmox* dan *Grafana* sebagai virtualisasi. Konfigurasi *Grafana* notifikasi juga dilakukan untuk bisa terhubung ke *smartphone* menggunakan aplikasi *telegram* untuk mengetahui penggunaan data pada *proxmox*. Untuk menunjang rancangan uji coba yang telah disusun, diperlukan perangkat keras dan perangkat lunak yang mampu mendukung implementasi dari rancangan tersebut.

a. Kebutuhan Perangkat Keras

Adapun kebutuhan perangkat keras dalam penelitian ini meliputi:

- **BIOS** : AMCN29WW (Versi 1.12)
- **Prosesor** : AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx (4 CPU) ~2.6GHz
- **RAM** : 8 GB DDR4 (8192MB)
- **Sistem Operasi** : Windows 11 Pro 64-bit (Build 22631)
- **Disk** : WDC PC SN520 SDAPMUW-512G-1101 (SSD (NVMe) 477 GB (476 GB formatted)
- **Memori Virtual** : : Total: 12284MB, Terpakai: 3014MB

b. Kebutuhan Perangkat Lunak

Adapun perangkat lunak yang dibutuhkan dalam pembangunan sistem ini meliputi:

- *VMWare*. Perangkat Lunak untuk Virtualisasi
- *Debian 11*. Sistem operasi yang digunakan untuk instalasi *prometheus* dan *Grafana*.
- *Grafana* dan *Prometheus*. Perangkat lunak yang akan menampilkan virtualisasi data *proxmox*
- *Chrome*. *Web browser* untuk mengakses *web* dari *proxmox*, *Prometheus* dan *Grafana*

C. Simulation Prototyping

Tahap ini terdiri dari 4 bagian yaitu instalasi dan konfigurasi pada *Container*, uji coba dan analisa hasil uji coba. Tahap instalasi dan konfigurasi terdiri dari empat bagian, yaitu instalasi dan konfigurasi *Container* pada *Proxmox*, *Grafana* dan *Prometheus* yang digunakan untuk virtualisasi.

a. Konfigurasi Container proxmox

Konfigurasi pada *Container Proxmox* mencakup pengaturan IP address pada antarmuka *Container Proxmox* serta penambahan repository yang diperlukan untuk instalasi *Grafana* dan *Prometheus*.

b. Konfigurasi Prometheus

Konfigurasi pada *Prometheus* meliputi pembuatan *user name*, *password*, organisasi, dan *bucket* yang bertujuan untuk mengintegrasikan *database Proxmox* dan *Grafana* guna mendukung virtualisasi.

c. Konfigurasi *Proxmox*

Konfigurasi pada *Proxmox* mencakup penambahan *metrik server Prometheus* dengan mengisi kotak dialog yang berisi informasi seperti *bucket*, dan *server Prometheus* yang sebelumnya telah dibuat pada *konfigurasi Prometheus*.

d. Konfigurasi *Grafana*

Konfigurasi pada *Grafana* meliputi penambahan sumber data *Prometheus* ke dalam *Grafana* serta pemasangan *template dashboard* virtualisasi yang telah diunduh dari situs resmi *Grafana*.

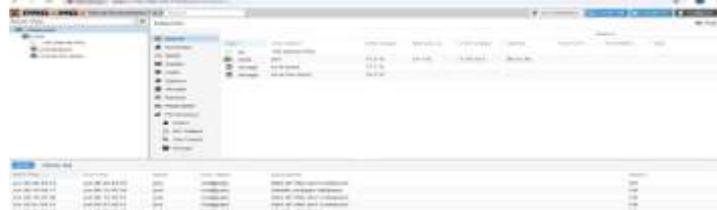
Pada skenario pengujian terdapat dua skenario pengujian, yaitu pengujian penambahan *Container* pada *Proxmox* agar *prometheus* dapat mengambil data dari *proxmox* dan *Grafana* dapat memantau data *Container* tersebut, serta pengujian notifikasi *Grafana* melalui Telegram. Pengujian notifikasi *Grafana* pada Telegram dilakukan dengan tiga kondisi, jika penggunaan CPU, Memori dan Disk berada dalam rentang 15%-50% data berhasil di kumpulkan oleh *Prometheus* dan di visualisasikan grafan kemudian mengirimkan notifikasi ke telegram dengan isi pesan “penggunaan rendah”, sedangkan 51%-75% notifikasi telegram dengan isi pesan “Penggungan sedang”, 76%-100% notifikasi telegram dengan isi pesan “Penggungan tinggi”

D. Monitoring

Pada tahap *monitoring* yang dilakukan, ada 3 objek yang dipantau yaitu *CPU*, *RAM* dan *Disk*.

3. Hasil dan Pembahasan

Pada bagian ini membahas tentang hasil dari instalasi dan konfigurasi yang dilakukan pada setiap perangkat, ujicoba dan analisa terhadap uji coba yang dilakukan. Hasil dari instalasi PVE yang dilakukan pada VM di *hypervisor VMware Workstation*, seperti terlihat pada gambar 4.



Gambar 4. Dashboard berbasis Web dari PVE

Terlihat terdapat beberapa menu untuk membantu untuk aktivitas manajemen *proxmox* seperti mengunggah *file image template* linux *Container* atau *file iso*, pembuatan *Virtual machine (Create VM)* atau *Container (Create CT)* dan lain sebagainya. Selanjutnya hasil dari instalasi dan konfigurasi *grafana* pada *container* tersebut, seperti terlihat pada gambar 5.



Gambar 5. Hasil Intalasi dan Konfigurasi Grafana

Melalui *dashboard* dari *grafana* dilakukan konfigurasi agar dapat memvisualisasikan *monitoring* penggunaan *CPU*, *RAM* dan *Disk* dari sumber daya yang ada di PVE. Selanjutnya hasil instalasi *prometheus* dan *node exporter*, seperti terlihat pada gambar 6.



Gambar 6. Hasil Konfigurasi Prometheus dan Node Exporter

Node exporter mengumpulkan data metrik yang diambil dari PVE kemudian disimpan pada prometheus serta divisualisasikan menggunakan Grafana. Grafana memiliki fitur untuk mengirimkan notifikasi hasil monitoring ke Telegram. Hasil ujicoba pengiriman notifikasi dari Grafana ke Telegram, seperti terlihat pada gambar 7.



Gambar 7. Hasil Konfigurasi Notifikasi Telegram

pemberitahuan real-time kepada administrator melalui aplikasi Telegram sehingga administrator memperoleh notifikasi melalui ke saluran Telegram yang telah ditentukan[15].sehingga pemantauan sistem PVE dapat dilakukan melalui perangkat handphone.

3.1 Hasil Uji Coba

Ujicoba notifikasi dilakukan pada CPU, Memori dan Disk untuk penggunaan rendah, sedang dan tinggi.

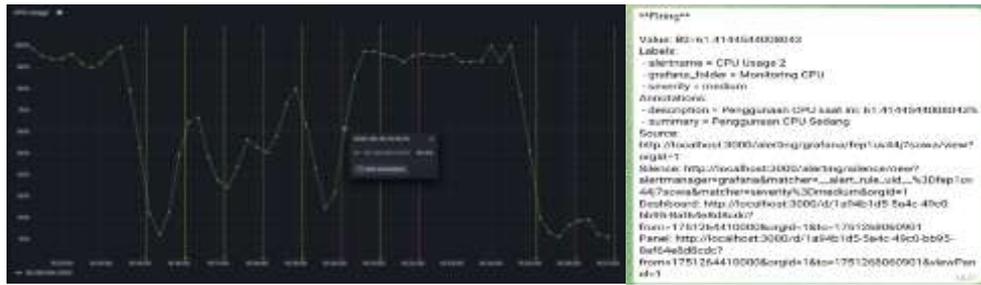
a. Penggunaan CPU

Uji coba dilakukan pada penggunaan CPU dengan hasil notifikasi telegram dan monitoring virtualisasi di grafana dengan penggunaan CPU rendah terlihat pada gambar 8.



Gambar 8. Tampilan Penggunaan Cpu Rendah di Grafana dan Notifikasi Telegram

Pada gambar 8 hasil pengujian di grafana dan telegram dengan pengiriman file ke root proxmox menggunakan aplikasi Winscp sebesar 200MB tapi tidak dapat memicu alert di grafana sehingga di kirimkan lagi file sebesar 2.1GB sehingga memicu alert di grafana, dengan menunjukkan bahwa Grafana mendeteksi penggunaan CPU mencapai 18% sehingga grafana mengirimkan notifikasi ke telegram ketika CPU berada pada angka 18% dan pesan yang diterima melalui Telegram. Selanjutnya hasil notifikasi telegram dan monitoring virtualisasi di grafana penggunaan CPU sedang terlihat pada gambar 9.



Gambar 9. Tampilan Penggunaan Cpu Sedang di Grafana dan Notifikasi Telegram

Terlihat Pada gambar 9 hasil pengujian di *grafana* dan *telegram* dengan pengiriman *file* ke *root proxmox* menggunakan aplikasi *Winscp* sebesar 200MB, 2.1GB dan 9 GB tapi tidak dapat memicu *alert* di *grafana* sehingga di kirimkan lagi *file* sebesar 1.5GB sehingga memicu *alert* di *grafana*, dengan menunjukkan bahwa Grafana mendeteksi penggunaan CPU mencapai 61% sehingga grafana mengirimkan notifikasi ke *telegram* ketika CPU berada pada angka 61% dan pesan yang diterima melalui *telegram*. Hasil notifikasi *telegram* dan *monitoring* virtualisasi di *grafana* penggunaan CPU tinggi terlihat pada gambar 10.

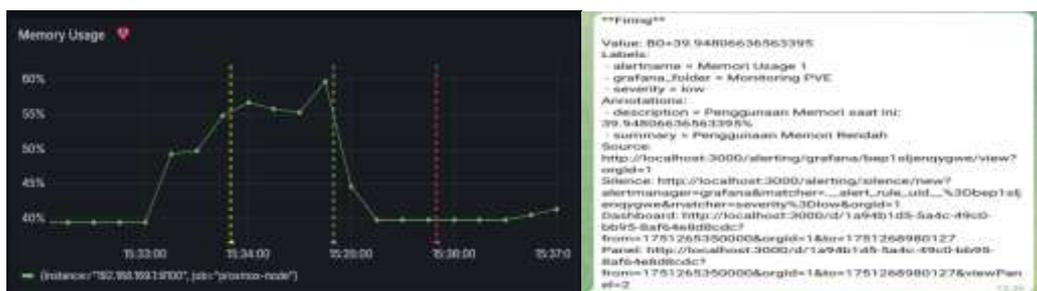


Gambar 10. Tampilan Penggunaan Cpu Tinggi di Grafana dan Notifikasi Telegram

Pada gambar 10 hasil pengujian di *grafana* dan *telegram* dengan pengiriman *file* ke *root proxmox* menggunakan aplikasi *Winscp* sebesar 200MB tapi tidak dapat memicu *alert* di *grafana* sehingga di kirimkan lagi *file* yang lebih besar sebesar 2.1GB sehingga memicu *alert* di *grafana*, dengan menunjukkan bahwa Grafana mendeteksi penggunaan CPU mencapai 98% sehingga grafana mengirimkan notifikasi ke *telegram* ketika CPU berada pada angka 98% dan pesan yang diterima melalui *telegram*.

b. Penggunaan Memori

Ujicoba dilakukan pada penggunaan memori dengan hasil notifikasi *telegram* dan *monitoring* virtualisasi di *grafana* untuk penggunaan memori rendah, seperti terlihat pada gambar 11.



Gambar 11. Tampilan Penggunaan Memori Rendah di Grafana dan Notifikasi Telegram

Peningkatan utilisasi memori dilakukan dengan mengeksekusi perintah “*stress --vm 2 --vm-bytes 512M -time out 120s*” di *container CT100*. Hasil pengujian bahwa bahwa setelah perintah tersebut dieksekusi maka *grafana* mendeteksi penggunaan memori melonjak hingga 39% dan memicu *alert* serta pengiriman notifikasi ke *telegram*. Sedangkan hasil notifikasi *telegram* dan *monitoring* virtualisasi di *grafana* untuk penggunaan memori sedang, seperti terlihat pada gambar 12.

cepat menyebabkan *CPU usage* naik-turun secara singkat dan untuk disk pada proses pengiriman file penyimpan yang tersedia masih banyak sehingga tidak sempat di *proses* sebagai kondisi *alert* di *grafana*. Sedangkan Penggunaan memori pada PVE saat dilakukan pengujian menggunakan perintah *stress*. Uji coba dilakukan dengan menjalankan beban berbeda melalui perintah *stress* pada *container* (CT). Pada saat ujicoba perintah *stress* pada *container* terdapat beberapa notifikasi tidak muncul atau tidak memicu *alert* pada *grafana* menunjukkan bahwa semakin banyak penggunaan *container* untuk ujicoba perintah *stress* pada waktu yang bersamaan akan menyebabkan penggunaan memori semakin tinggi.

4. Kesimpulan

Implementasi *Prometheus* yang terintegrasi *Grafana* pada *Proxmox* berhasil diinstalasi dan dikonfigurasi sehingga menampilkan informasi terkait utilisasi dari sistem yang berjalan yaitu berupa penggunaan *CPU*, memori dan media penyimpanan (*disk*). *Node exporter* yang diinstalasi pada *Proxmox* bertindak sebagai *agent* untuk mengumpulkan metrik utilisasi sistem yang diambil dan disimpan oleh *Prometheus* serta divisualisasikan datanya menggunakan *Grafana*. Pengiriman notifikasi dari *Grafana* ke *Telegram* telah berhasil dilakukan. Sistem yang dibangun mampu memberikan peringatan berupa notifikasi ke *Telegram* saat mendeteksi penggunaan *CPU*, memori, atau media penyimpanan (*disk*) baik dalam kondisi rendah, sedang, maupun tinggi.

Daftar Pustaka

- [1] Santoso, N. A., Zakaria, Z., & Kurniawan, R. D. (2022). Analisis Jaringan Komputer Menggunakan Teknologi Virtualisasi. *Jurnal Minfo Polgan*, 11(2), 52–58. <https://doi.org/10.33395/jmp.v11i2.11652>
- [2] Febriansyah, A. A., & Prapanca, A. (2024). Simulasi Implementasi High Availability Server Menggunakan Ceph Pada Proxmox. *Journal of Informatics and Computer Science (JINACS)*, 6(01), 131–136. <https://doi.org/10.26740/jinacs.v6n01.p131-136>
- [3] Putra, R. C. W., Limpraptono, F. Y., & Sotyohadi. (2023). Pengembangan Perangkat Praktikum Sistem Operasi Untuk Pembuatan Mesin Virtual Secara Otomatis Di Proxmox Ve Server Menggunakan Software Ansible. *Magnetika: Jurnal Mahasiswa Teknik Elektro*, 7(2), 337–344. <https://ejournal.itn.ac.id/index.php/magnetika/article/view/8627>
- [4] Sheila, S. (2024). EVALUASI TEKNOLOGI VIRTUALISASI MESIN PROXMOX UNTUK. 04(1), 10–13.
- [5] Prasetyo, E., Dedy Irawan, J., & Ariwibisono, F. X. (2022). Rancang Bangun Sistem Monitoring Server Virtual Berbasis Web Menggunakan Script Monitoring Pada Proxmox Virtual Environment. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(1), 179–185. <https://doi.org/10.36040/jati.v6i1.4550>
- [6] Ramdhany, A. F., Saedudin, R. R., & Septo, U. Y. K. (2022). Perancangan Desain Monitoring Jaringan Komputer Untuk Easy Maintenance Di Telkom University Landmark Tower. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 7(4), 1176–1188. <https://doi.org/10.29100/jupi.v7i4.3215>
- [7] Candra, I. P. W. A., Suarjaya, I. M. A. D., & Sukarsa, I. M. (2023). Rancang Bangun Sistem Visualisasi Data Monitoring Server dengan Teknologi Chatbot. *JITTER : Jurnal Ilmiah Teknologi Dan Komputer*, 4(1), 1589. <https://doi.org/10.24843/jtrti.2023.v04.i01.p04>
- [8] Yulvianda, R., & Ismail, M. (2023). Desain dan Implementasi Sistem Monitoring Sumber Daya Server Menggunakan Zabbix dan Grafana. *Jurnal Informatika Dan Rekayasa Komputer(JAKAKOM)*, 3(1), 322–329. <https://doi.org/10.33998/jakakom.2023.3.1.712>
- [9] Husna, M. A., & Rosyani, P. (2021). Implementasi Sistem Monitoring Jaringan dan Server Menggunakan Zabbix yang Terintegrasi dengan Grafana dan Telegram. *JURIKOM (Jurnal Riset Komputer)*, 8(6), 247. <https://doi.org/10.30865/jurikom.v8i6.3631>
- [10] Ramadoni, Amirudin, M. Z., Rifki Fahmi, Ema Utami, & Mustafa, M. S. (2021). Evaluasi Penggunaan Prometheus dan Grafana Untuk Monitoring Database MongoDB. *Jurnal Informatika Polinema*, 7(2), 43–50. <https://doi.org/10.33795/jip.v7i2.530>
- [11] I Ketut Gede Sugita, Kadek Suar Wibawa, I Wayan Wahyu Ivan Mahendrajaya, Rey Bernard, I Putu Abdi Purnawan, & I Gede Nyoman Ambara Yasa. (2022). Sistem Notifikasi Kepangkatan Dosen Secara Realtime Berbasiskan Telegram Bot API. *Tematik*, 9(2), 202–209. <https://doi.org/10.38204/tematik.v9i2.1061>
- [12] Muhammad Fauzan Rafi, & Eko Marpanaji. (2025). Pengembangan Sistem Monitoring Cloud LMS berbasis Grafana, Prometheus dan Telegram di SMK Negeri 2 Yogyakarta. *Journal of Information Technology and Education (JITED)*, 3(1), 91–100. <https://doi.org/10.21831/jited.v3i1.1052>
- [13] Dira, S. R., Arif, M., & Ridha, F. (2022). Monitoring Kubernetes Cluster Menggunakan Prometheus dan Grafana. *Proceeding Applied Business and Engineering Conference, November*, 17–19

-
- [14] Harjanto, H., & Purnama, G. (2024). Perancangan Dan Simulasi Jaringan Komputer Dengan Metode Pengembangan Network Development Life Cycle (Ndlc) Pada Kantor Cabang Pt. V2 Indonesia. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 8032–8039. <https://doi.org/10.36040/jati.v8i4.10622>
- [15] Rasyidi, B., & Pratama, F. (2024). Sistem Monitoring Server di PT. XYZ Media Indonesia Berbasis Grafana dan Prometheus. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(4), 1456–1465. <https://doi.org/10.57152/malcom.v4i4.1546>