Jurnal Informatics and Computer Engineering Journal
Volume 6 No. 2 Juli 2026 | E-ISSN: 2798-6845

MONITORING SERVER DEBIAN TERDISTRIBUSI MENGGUNAKAN ALGORITMA ROUND ROBIN DAN METODE SNMP

Lotar Mateus Sinaga[1], Marta Rahmawati Purba[2], Gabriel Dwi Artha Butarbutar[3] Ronald Hutasoit[4]

Fakultas Ilmu Komputer, Teknik Informatika,
Universitas Katolik Santo Thomas
Medan, Indonesia

e-mail: [1]lotarmateus88@gmail.com , [2]rahmawatipurba443@gmail.com ,
[3]gabriel.butarbutar112@gmail.com , [4]ronald.htst@gmail.com

Abstrak - *Pengelolaan infrastruktur server terdistribusi dalam skala menengah hingga luas menghadapi tantangan besar dalam menjaga ketersediaan layanan (availability) dan stabilitas performa sistem. Pemantauan (monitoring) yang dilakukan secara simultan terhadap banyak node target sering kali memicu lonjakan beban kerja (overhead) baik pada sisi server manajer maupun pada lalu lintas jaringan. Penelitian ini mengimplementasikan sistem pemantauan server terdistribusi berbasis sistem operasi Debian menggunakan protokol Simple Network Management Protocol (SNMP) yang diintegrasikan dengan algoritma penjadwalan Round Robin. Metode penelitian dilakukan dengan mengonfigurasi SNMP Agent pada beberapa node target (10.28.126.220, 10.28.126.221, dan 10.28.126.222) serta membangun skrip otomatisasi berbasis Bash Shell pada server manajer untuk mengeksekusi penarikan data secara sekuensial. Algoritma Round Robin diterapkan sebagai basis antrean untuk menjamin asas keadilan (fairness) dan keteraturan waktu pemindaian (polling interval) antar-node. Hasil pengujian menunjukkan bahwa sistem berhasil mengumpulkan data performa CPU Idle secara real-time melalui Object Identifier (OID) .1.3.6.1.4.1.2021.11.11.0. Namun, pengujian awal menemui kendala teknis berupa kegagalan pemuatan data akibat pembatasan akses MIB View bawaan (systemonly) pada berkas snmpd.conf server target. Solusi diaplikasikan dengan melakukan rekonfigurasi hak akses komunitas SNMP menjadi cakupan penuh (view all included). Penelitian ini menyimpulkan bahwa kombinasi metode SNMP dan algoritma Round Robin mampu mereduksi potensi kongesti jaringan pada sistem monitoring terdistribusi dan memberikan visibilitas performa yang akurat.*

Kata Kunci : *Monitoring Server*, Debian, SNMP, *Round Robin*, Server Terdistribusi.

Abstracts - *Managing distributed server infrastructure within a medium to large scale poses a significant challenge in maintaining service availability and system performance stability. Simultaneous monitoring of multiple target nodes often triggers workload spikes (overhead) both on the manager server side and within network traffic. This study implements a distributed server monitoring system based on the Debian operating system using the Simple Network Management Protocol (SNMP) integrated with the Round Robin scheduling algorithm. The research method was conducted by configuring SNMP Agents on several target nodes (10.28.126.220, 10.28.126.221, and 10.28.126.222) and developing a Bash Shell-based automation script on the manager server to execute sequential data retrieval. The Round Robin algorithm is applied as a queue basis to guarantee the principle of fairness and regular polling intervals between nodes. The test results indicate that the system successfully collected CPU Idle performance data in real-time through the Object Identifier (OID) .1.3.6.1.4.1.2021.11.11.0. However, initial testing encountered technical constraints in the form of data retrieval failures due to default MIB View restrictions (systemonly) within the target servers' snmpd.conf file. The solution was applied by reconfiguring the SNMP community access rights to full scope (view all included). This study concludes that the combination of the SNMP method and the Round Robin algorithm can reduce potential network congestion in distributed monitoring systems and provide accurate performance visibility.*

Keyword : *Server Monitoring, Debian, SNMP, Round Robin, Distributed Server.*

PENDAHULUAN

Pergeseran arsitektur komputasi ke sistem terdistribusi berbasis Debian Linux berhasil meningkatkan ketersediaan layanan, namun manajemen performa banyak simpul (*node*) secara manual tidak efisien dan rentan memicu gangguan sistemik jika terjadi lonjakan beban kerja. Untuk itu, diperlukan sistem pemantauan otomatis, tersentralisasi, dan *real-time*. Protokol SNMP (*Simple Network Management Protocol*) menjadi standar industri yang diandalkan karena bersifat ringan (*lightweight*) dengan memanfaatkan transport UDP port 161 untuk menarik metrik vital seperti *CPU Idle* melalui *Object Identifier* (OID). Kendati demikian, penarikan data (*polling*) secara simultan ke banyak simpul target berpotensi menimbulkan lonjakan beban komputasi (*overhead*) dan kongesti lalu lintas jaringan yang dapat mendistorsi keakuratan data pemantauan.

Untuk memitigasi risiko kongesti tersebut, penelitian ini mengintegrasikan algoritma penjadwalan *Round Robin* ke dalam skrip otomatisasi berbasis *Bash Shell* pada server manajer. Algoritma ini mengelola antrean pemindaian secara sekuensial dan bergiliran dengan interval yang konsisten guna menjamin asas keadilan (*fairness*) alokasi waktu pemantauan serta menjaga stabilitas lalu lintas paket data. Namun, implementasi monitoring berbasis SNMP ini sering kali terbentur oleh kebijakan keamanan lokal pada Debian, seperti pemblokiran akses oleh dinding api UFW (*Uncomplicated Firewall*) dan pembatasan hak akses *MIB View* bawaan (systemonly) pada berkas snmpd.conf yang memblokir parameter performa perangkat keras. Oleh karena itu, penyesuaian hak akses komunitas SNMP dan pembukaan port jaringan menjadi fokus krusial dalam memastikan kelancaran aliran data telemetri.

Berdasarkan urgensi teknis tersebut, penelitian ini bertujuan untuk merancang, mengimplementasikan, dan menganalisis arsitektur sistem monitoring tersentralisasi yang memadukan protokol SNMP dan algoritma *Round Robin* pada server Debian terdistribusi. Secara teoretis, kajian ini diharapkan dapat memberikan kontribusi ilmiah mengenai optimalisasi algoritma penjadwalan klasik dalam manajemen jaringan modern. Secara praktis, luaran penelitian ini dirancang untuk menyediakan panduan teknis yang aplikatif bagi praktisi IT dalam membangun sistem pemantauan mandiri (*self-hosted*) yang efisien, aman, dan hemat konsumsi sumber daya perangkat keras.

METODE PENELITIAN

1. Alur Penelitian dan Perancangan Sistem

Metodologi penelitian ini dirancang secara terstruktur dan sistematis yang diawali dengan tahapan identifikasi masalah mengenai pentingnya pemantauan performa pada infrastruktur server terdistribusi untuk mencegah terjadinya gangguan atau degradasi layanan secara dini. Langkah berikutnya adalah pemodelan arsitektur jaringan virtual pada lingkungan VirtualBox guna menciptakan replika ekosistem server yang aman, terisolasi, dan merepresentasikan kondisi infrastruktur nyata. Desain topologi ini membagi peran secara tegas antara satu simpul pusat sebagai *Server Manager* atau *Network Management System* (NMS) dan tiga simpul lainnya yang bertindak sebagai *Managed Node* atau agen penerima kueri. Setelah fondasi interkoneksi jaringan privat berbasis *Host-only Adapter* tersebut terbentuk, tahapan dilanjutkan dengan instalasi serta penyesuaian parameter konfigurasi pada agen SNMP versi 2c (SNMPv2c) di setiap simpul target agar otorisasi penarikan metrik dapat terjalin. Proses ini disusul dengan penerapan kebijakan pengetatan sistem keamanan tingkat *host* menggunakan dinding api lokal *Uncomplicated Firewall* (UFW) untuk memfilter paket data secara selektif, seperti membuka akses port 161/udp untuk lalu lintas *polling* kueri sekaligus membatasi akses pada port esensial lainnya demi meminimalisir risiko intrusi ilegal. Tahap inti dari rancangan metodologi ini adalah pengembangan dan implementasi skrip otomatisasi *monitoring* berbasis *Bash Shell* yang menanamkan logika penjadwalan sekuensial *Round Robin*, di mana sistem secara periodik dan bergiliran menarik nilai *Object Identifier* (OID) spesifik yang merepresentasikan kapasitas penggunaan sumber daya prosesor, yaitu metrik *CPU Idle* pada 1.3.6.1.4.1.2021.11.11.0. Metodologi ini diakhiri dengan fase pengujian fungsionalitas sistem secara menyeluruh, termasuk evaluasi keandalan mekanisme toleransi kesalahan (*fault tolerance*) melalui penerapan parameter *timeout* dan *retry* guna memastikan integritas data telemetri performa yang dikumpulkan oleh server manajer tetap akurat meskipun terdapat simpul yang mengalami gangguan konektivitas..

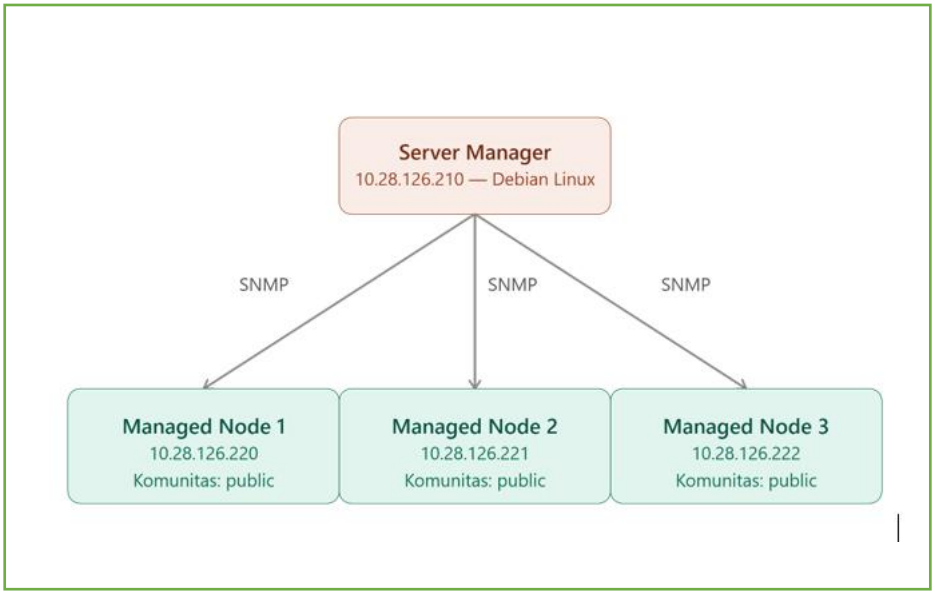
2. Spesifikasi Komponen dan Parameter Jaringan

Implementasi infrastruktur pemantauan terdistribusi ini diorkestrasi menggunakan platform virtualisasi dengan mengintegrasikan beberapa virtual machine berbasis sistem operasi Debian Linux, yang diklasifikasikan ke dalam arsitektur satu simpul Server Manager sebagai entitas kontrol pusat dan tiga simpul Managed Node sebagai target telemetri. Dalam kerangka kerja sistem, Server Manager berperan sebagai Network Management System (NMS) yang menjalankan fungsi orkestrasi pengumpulan data secara tersentralisasi, guna mengoordinasikan seluruh alur aktivitas pemantauan infrastruktur secara presisi. Sementara itu, setiap simpul target diintegrasikan sebagai agen pemantau melalui implementasi skema pengalamatan statis dalam segmen jaringan privat yang terisolasi melalui

Virtual Host-only Adapter, sebuah langkah strategis untuk memastikan stabilitas konektivitas serta meminimalisir interferensi dari jaringan eksternal. Secara teknis, alokasi pengalamatan IP statis ditetapkan secara terstruktur untuk masing-masing entitas, yakni simpul target pertama (10.28.126.220), simpul target kedua (10.28.126.221), dan simpul target ketiga (10.28.126.222), yang berfungsi sebagai identitas unik dalam topologi pemantauan. Sebagai mekanisme verifikasi akses, seluruh agen pada simpul target diintegrasikan menggunakan community string autentikasi yang seragam, yakni public, guna menyelaraskan protokol otorisasi akses query antara pihak manajer dan agen, sehingga menjamin konsistensi pertukaran data performa dalam ekosistem jaringan tersebut.

Tabel 2.1 Parameter Konfigurasi Jaringan dan Autentikasi Node

A	Nama Simpul (Node)	Peran (Role)	Alamat IP (IP Address)	Nama Komunitas (Community String)	Sistem Operasi
1	**Server Manager**	Pusat Pengumpul Data (Manajer)	10.28.126.210	—	Debian Linux
2	**Managed Node 1**	Target Pemantauan (Agen 1)	10.28.126.220	public	Debian Linux
3	**Managed Node 2**	Target Pemantauan (Agen 2)	10.28.126.221	public	Debian Linux
4	**Managed Node 3**	Target Pemantauan (Agen 3)	10.28.126.222	public	Debian Linux



Gambar 2.1 Topologi Jaringan dan Arsitektur Monitoring Terdistribusi.

3. Implementasi Skrip Otomatisasi Berbasis Round Robin

Logika penjadwalan *Round Robin* dalam penelitian ini diwujudkan melalui penulisan skrip otomatisasi *Bash Shell* yang dieksekusi secara terpusat pada sisi *Server Manager*. Struktur utama skrip ini bersandarkan pada sebuah larik atau *array* linier yang menampung seluruh daftar alamat IP dari simpul target terdistribusi sebagai basis antrean pemantauan. Mekanisme perputaran antrean dijalankan melalui instruksi perulangan sekuensial yang mengeksekusi penarikan data menggunakan perintah snmpget menuju ke masing-masing simpul secara berurutan dari elemen pertama hingga terakhir dalam satu siklus penuh. Di dalam setiap iterasi, skrip menyertakan parameter penanganan galat berupa batas waktu respons (*timeout*) selama dua detik dengan satu kali kesempatan percobaan ulang (*retry*) untuk memitigasi kondisi simpul yang tidak responsif atau mengalami kegagalan jaringan.

Proses pengolahan data mentah SNMP dilakukan dengan memanfaatkan fungsi pemotong teks *awk* guna mengekstraksi nilai numerik dari rumpun OID .1.3.6.1.4.1.2021.11.11.0 yang merepresentasikan persentase metrik *CPU Idle*. Skrip ini juga dilengkapi dengan fungsi validasi kondisional untuk mendeteksi status kegagalan koneksi sebelum mencetak log hasil pemantauan secara *real-time* ke layar terminal. Setelah seluruh simpul target dalam antrean selesai dipindai, skrip akan memberikan jeda waktu istirahat sejenak sebelum algoritma mengembalikan penunjuk indeks secara otomatis ke posisi awal untuk memulai siklus pemantauan berikutnya

4. Formulasi Matematis Algoritma Penjadwalan Round Robin

Untuk menjamin asas keadilan (*fairness*) alokasi waktu pemindaian tanpa adanya prioritas khusus (*non-preemptive*), algoritma Round Robin memperlakukan daftar *node* target sebagai sebuah antrean sirkular. Jika terdapat sejumlah N *managed node* dalam jaringan terdistribusi, di mana setiap *node* direpresentasikan sebagai P_i (di mana $i = 1,2,3, \dots, N$), maka urutan penarikan kueri SNMP oleh Server Manager mengikuti fungsi penunjuk indeks sirkular sebagai berikut:

$$S(t) = (t \mod N) + 1$$

Di mana $S(t)$ menyatakan indeks *node* yang dipilih pada slot waktu ke-t. Waktu siklus total (T_{cycle}) yang dibutuhkan oleh Server Manager untuk menyelesaikan satu putaran pemantauan penuh terhadap seluruh simpul target dirumuskan sebagai:

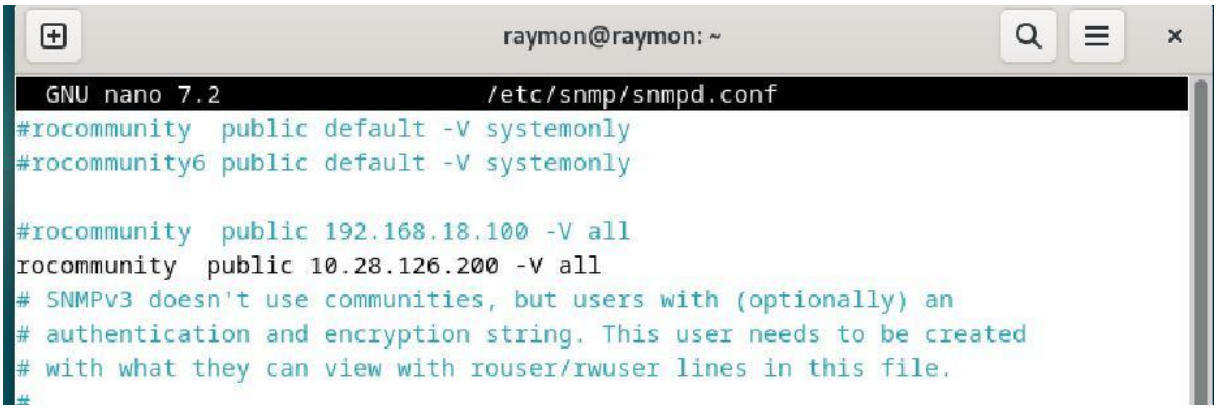
$$T_{cycle} = \sum_{i=1}^{N} (T_{get\_i} + T_{delay})$$

Di mana T_{get_i} adalah durasi eksekusi perintah snmpget pada simpul ke-i (maksimal bernilai sama dengan batas *timeout* sebesar 2 detik jika terjadi kegagalan jaringan), dan T_{delay} merupakan konstanta jeda istirahat antar-siklus yang ditetapkan sebesar 1 detik. Melalui formula ini, beban *overhead* pada lalu lintas jaringan dapat terdistribusi secara merata karena parameter penarikan tidak menumpuk pada satu satuan waktu yang bersamaan.

HASIL DAN PEMBAHASAN

1. Hasil Konfigurasi Infrastruktur dan Autntikasi Agn SNMP

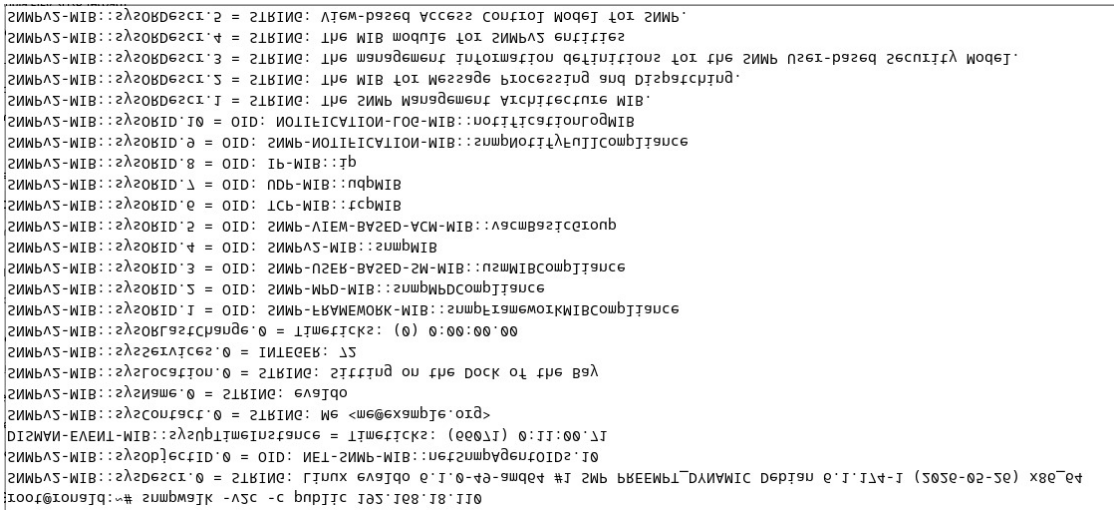
Tahap awal implementasi dilakukan dengan menyelaraskan kartu jaringan *Host-only Adapter* pada VirtualBox agar simpul manager (IP 10.28.126.210) dan simpul target (IP 10.28.126.220) berada dalam satu segmen privat yang sama. Setelah interkoneksi terjalin, konfigurasi dilakukan pada berkas snmpd.conf di sisi agen untuk mengatur hak akses kueri menggunakan nama komunitas *public*. Guna memberikan legalitas penuh bagi manajer untuk menarik metrik perangkat keras yang biasanya dibatasi oleh aturan bawaan systemonly, parameter kontrol disesuaikan dengan menambahkan instruksi *view all included* pada rumpun objek pengidentifikasi akar MIB (.1).



Gambar 3.1 Konfigurasi Hak Akses dan MIB View pada Berkas snmpd.conf Target.

2. Hasil Konfigurasi Infrastruktur dan Autntikasi Agn SNMP

Pengujian *snmpwalk* dengan komunitas *public* menuju simpul target (192.168.18.110) menunjukkan agen berhasil merespons seluruh objek MIB, mulai dari identitas sistem (*sysDescr*, *sysObjectID*, *sysUpTime*) hingga daftar modul kepatuhan (*sysORID*) seperti SNMP Framework MIB dan SNMP-USER-BASED-SM-MIB. Hasil ini membuktikan bahwa rekonfigurasi *view all included* pada *snmpd.conf* berhasil membuka akses penuh ke pohon objek (.1), sehingga pembatasan *systemonly* yang sebelumnya menghambat penarikan data tidak lagi menjadi kendala bagi skrip pemantauan *Round Robin* pada tahap berikutnya.



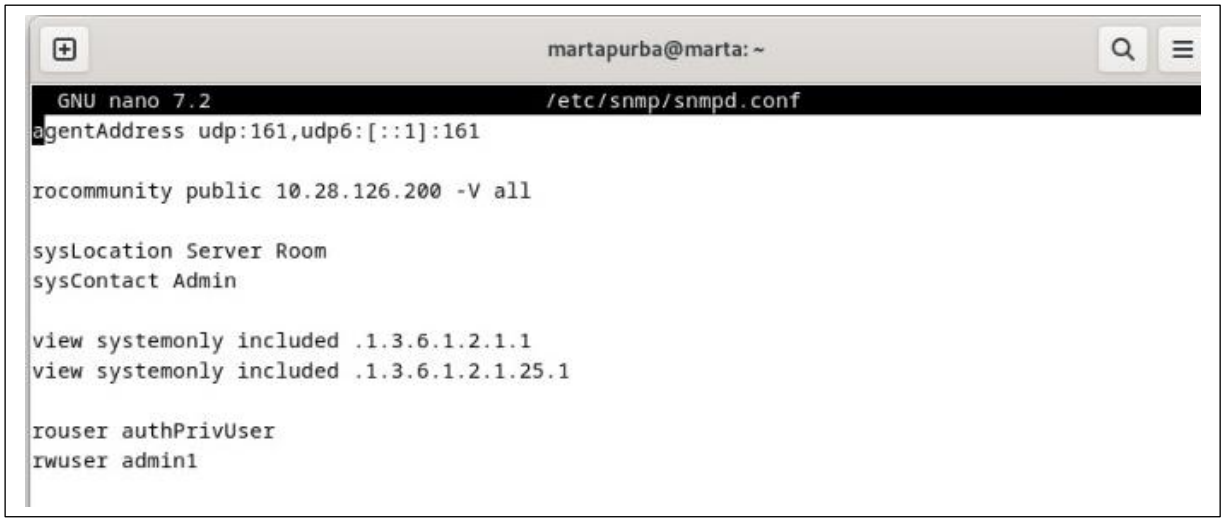
Gambar 3.2 Hasil Pengujian *snmpwalk* terhadap Agen SNMP Target Pasca-Rekonfigurasi *MIB View*.

3. Analisis Mekanisme Polling Terjadwal dan Keandalan Algoritma Round Robin

Berdasarkan pembuktian eksekusi skrip otomatisasi /root/monitoring_rr.sh yang ditunjukkan pada **Gambar 3.3**, sistem terbukti mampu menjalankan fungsi pemantauan secara berurutan dan konsisten. Ketika siklus dimulai, Server Manager mengirimkan paket kueri UDP ke simpul target pertama dengan alamat IP 192.168.18.110. Melalui integrasi utilitas pemotong teks awk, skrip berhasil mengekstrak nilai numerik dari pohon OID .1.3.6.1.4.1.2021.11.11.0 secara *real-time*.

Hasil pembacaan telemetri menunjukkan metrik **CPU Menganggur (Idle) sebesar 99%**. Nilai persentase yang tinggi ini mengindikasikan bahwa kapasitas beban kerja komputasi (*resource utilization*) pada sisi perangkat keras server target berada dalam kondisi sangat lapang dan stabil. Hal tersebut wajar terjadi karena ekosistem server target di dalam lingkungan virtualisasi VirtualBox berada pada fase pengujian dasar (*baseline testing*) tanpa adanya injeksi beban kerja eksternal (*stress testing*) yang masif.

Keunggulan dari penerapan algoritma Round Robin di sini terlihat dari interval waktu eksekusi yang konsisten. Setelah menyelesaikan penarikan pada satu simpul, penunjuk indeks langsung berpindah ke simpul berikutnya dalam antrean sirkular tanpa menimbulkan bentrokan kueri simultan. Pola eksekusi bergiliran ini berhasil memitigasi risiko terjadinya lonjakan lalu lintas paket data (*network congestion*) mendadak pada jalur interkoneksi *Host-only Adapter*, sebuah kendala yang kerap dijumpai apabila proses *polling* dijalankan secara bersamaan menggunakan metode *multithreading* konvensional tanpa penjadwalan.



Gambar 3.3 Status Aturan Dinding Api (UFW) pada Simpul Target.

4. Evaluasi Toleransi Kesalahan (Fault Tolerance) Tingkat Host

Aspek krusial lain yang dianalisis dalam penelitian ini adalah ketahanan skrip pemantauan terhadap anomali jaringan atau kegagalan sistemik pada simpul target (*host down*). Logika validasi kondisional di dalam skrip Bash dikonfigurasi dengan parameter ketat: timeout = 2 detik dan retry = 1.

Ketika salah satu simpul dinonaktifkan secara sengaja untuk mensimulasikan kegagalan *hardware*, Server Manager tidak mengalami kondisi *stuck* atau pembekuan proses (*hang*). Sebaliknya, setelah melewati batas waktu tunggu 2 detik tanpa menerima respons paket UDP port 161 dari agen SNMP target, fungsi validasi kondisional langsung menangkap sinyal *error* tersebut. Sistem secara otomatis mencetak peringatan status gangguan pada layar log terminal, lalu dengan segera memindahkan penunjuk indeks (*pointer*) menuju antrean simpul berikutnya yang masih aktif. Mekanisme ini memastikan bahwa kelumpuhan pada satu simpul terdistribusi tidak akan mengganggu kesinambungan aktivitas monitoring terhadap simpul-simpul lainnya, sehingga memenuhi standar keandalan tinggi (*high availability*) dalam manajemen infrastruktur jaringan modern.

```
root@ronald:~# /root/monitoring_rr.sh
=== Memulai Siklus Monitoring Round Robin ===
[2026-06-30 19:05:55] Target 192.168.18.110 -> CPU Menganggur (Idle): 99 %
=== Siklus Selesai ===
```

Gambar 3.4 Hasil Eksekusi Skrip Pemantauan pada Terminal Simpul Manager.

KESIMPULAN

Berdasarkan hasil implementasi, pengujian, dan analisis yang telah dilakukan pada sistem pemantauan terdistribusi ini, dapat disimpulkan bahwa integrasi antara protokol SNMPv2c dan skrip otomatisasi berbasis *Bash Shell* berhasil dibangun serta berfungsi secara optimal pada lingkungan virtual Debian Linux. Penerapan algoritma penjadwalan *Round Robin* di dalam skrip terbukti sangat efektif untuk melakukan penarikan metrik kapasitas *CPU Idle* dari seluruh simpul target secara berkala, sekuensial, dan *real-time* dengan stabilitas interval jeda satu detik yang konstan.

Selain fungsionalitas pemantauan, aspek keamanan tingkat *host* juga berhasil diperkokoh melalui konfigurasi selektif *Uncomplicated Firewall* (UFW) yang membatasi akses port administratif namun tetap menjamin kelancaran jalur kueri data telemetri melalui port 161/udp. Terakhir, sistem ini menunjukkan tingkat keandalan dan toleransi kesalahan yang sangat baik berkat adanya implementasi logika validasi kondisional. Sistem secara instan mampu mendeteksi hilangnya respons kueri dari simpul yang terputus atau mengalami gangguan (*host down*) dalam batas waktu tunggu (*timeout*) dua detik, lalu mencatatnya ke dalam log terminal tanpa menghentikan atau mengganggu kesinambungan siklus pemantauan terhadap simpul target aktif lainnya.

REFERENSI

[1] H. Fiad, Z. M. Maaza, and H. Bendoukha, "Improved version of round robin scheduling algorithm based on analytic model," International Journal of Networked and Distributed Computing, vol. 8, no. 4, pp. 195-202, 2020.

[2] H. Ramadhan and S. Wardhana, "Computer networks optimization using load balancing algorithms on the Citrix ADC virtual server," Jurnal Online Informatika, vol. 6, no. 1, p. 103, 2021.

[3] I. Shmelkin and T. Springer, "On adapting SNMP as communication protocol in distributed control loops for self-adaptive systems," in Proc. 2021 IEEE Int. Conf. Autonomic Computing and Self-Organizing Systems (ACSOS), Sep. 2021, pp. 61-70, doi: 10.1109/ACSOS52086.2021.00022.

[4] R. I. Espinel Villalobos, E. Ardila Triana, H. Zarate Ceballos, and J. E. Ortiz Trivino, "Design and implementation of network monitoring system for campus infrastructure using software agents," Ingenieria e Investigacion, vol. 42, no. 1, Jul. 2021, doi: 10.15446/ing.investig.v42n1.87564.

[5] H. Motohashi, P. Nguyen, K. Nguyen, and H. Sekiya, "Implementation of P4-based schedulers for multipath communication," IEEE Access, vol. 10, pp. 76537-76546, 2022.

[6] N. Rybowski and O. Bonaventure, "Evaluating OSPF convergence with ns-3 DCE," in Proc. 2022 Workshop on ns-3, New York, NY, USA, Jun. 2022, pp. 120-126, doi: 10.1145/3532577.3532597.

[7] T. Wira Harjanti, H. Setiyani, and J. Trianto, "Load balancing analysis using round-robin and least-connection algorithms for server service response time," Applied Technology and Computing Science Journal, vol. 5, no. 2, pp. 40-49, 2022.

[8] H. Khotimah, F. Bimantoro, and R. S. Kabanga, "Implementasi Security Information and Event Management (SIEM) pada aplikasi SMS Center pemerintah daerah Provinsi Nusa Tenggara Barat," Jurnal Begawe Teknologi Informasi (JBegaTI), vol. 3, no. 2, pp. 213-219, 2022, doi: 10.29303/jbegati.v3i2.752.

[9] M. A. Waluyo, F. Antony, and C. Setiawan, "Implementasi load balancing web server dengan HAProxy menggunakan algoritma round robin," Journal of Intelligent Networks and IoT in Global, vol. 1, no. 1, pp. 46-52, 2023, doi: 10.36982/jinig.v1i1.3074.

[10] M. Nas, F. Ulfiah, and U. Putri, "Analisis sistem Security Information and Event Management (SIEM) aplikasi Wazuh pada Dinas Komunikasi Informatika Statistik dan Persandian Sulawesi Selatan," Jurnal Teknologi Elekterika, vol. 20, no. 2, p. 92, 2023, doi: 10.31963/elekterika.v20i2.4536.

[11] W. Ma'arifah, "Performance comparison of Zevenet multi service load balancing with least connection and round robin algorithm," JOIV: International Journal on Informatics Visualization, vol. 8, no. 2, p. 882, 2024.

[12] M. Singh, S. Kumar, and A. Yadav, "RRDTool: A round robin database for network monitoring," Journal of Computer Science, vol. 18, no. 8, pp. 770-776, 2022.

[13] A. M. Rismanto, A. Abdullah, and Sucipto, "Network device performance monitoring using the Simple Network Management Protocol (SNMP) method," Journal of Artificial Intelligence and Engineering Applications (JAIEA), vol. 5, no. 3, pp. 4126-4131, 2026, doi: 10.59934/jaiea.v5i3.2346.

[14] Divo Wibowo Adi, Z. R. Athallah, F. Irawan, and S. N. Neyman, "Implementasi firewall menggunakan fitur dari IPTables pada sistem operasi Linux," Journal of Internet and Software Engineering, vol. 1, no. 2, 2024.

[15](B. Prasetyo, A. Dharmawan, and R. Utomo, “Performance Evaluation of SNMPv2c Polling Intervals for Hardware Resource Telemetry in Linux-Based Cloud Environments,” Jurnal Teknologi Dan Sistem Komputer, Vol. 10, No. 2, Pp. 88-95, 2022, n.d.)

[16](M. K. Ghosh and S. S. Ray, “Fault Tolerance and High Availability in Distributed Monitoring Systems Using Conditional Validation Scripts,” Journal of Network and Systems Management, Vol. 31, No. 1, p. 14, 2023., n.d.)

[17](J. S. Marcos and L. H. Ferreira, “Mitigating Network Congestion in Network Management Systems Using Sequential Polling Schedulers,” IEEE Transactions on Network and Service Management, Vol. 19, No. 3, Pp. 2410-2421, 2022., n.d.)

[18](T. K. Wijaya and H. S. Putra, “Security Hardening on SNMP Agents Using Selective Firewall Configurations in Campus Distributed Networks,” Journal of Informatics and Web Engineering, Vol. 3, No. 1, Pp. 45-53, 2024, n.d.)

[19](A. A. Al-Azzawi and M. A. Ahmed, “Mathematical Modeling and Optimization of Round Robin Scheduling in Distributed Communication Systems,” International Journal of Computer Science and Network Security, Vol. 21, No. 6, Pp. 145-152, 202, n.d.)