

## SISTEM *MONITORING SERVER* DENGAN *SNMP* DAN *RRD TOOLS*

Wahyu Dwinanto<sup>1\*</sup>, Dolly Virgiana Shaka Yudha Sakti<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta Selatan, Indonesia

Email: <sup>1</sup>\*2211510785@student.budiluhur.ac.id, <sup>2</sup>dolly.virgianshaka@budiluhur.ac.id

(\* : corresponding author)

**Abstrak**-Pada era *digital* yang semakin maju, ketersediaan dan performa server merupakan elemen kritis dalam infrastruktur teknologi informasi. *Monitoring server* secara efektif menjadi penting untuk memastikan layanan tetap berjalan optimal dan gangguan dapat diidentifikasi serta ditangani dengan cepat. Penelitian ini bertujuan mengimplementasikan *system monitoring server* yang memanfaatkan *Simple Network Management Protocol (SNMP)* dan *Round Robin Database (RRD) Tools* dengan antarmuka *Graphical User Interfaces (GUI)* menggunakan *Python*. Metode penelitian ini melibatkan beberapa tahapan. Pertama, pengumpulan data dari *server* menggunakan protokol *SNMP* yang mampu mendukung berbagai versi (*v1, v2c, v3*). Kedua, penyimpanan data dalam format *RRD* yang menyediakan mekanisme penyimpanan data yang ringkas dan berkelanjutan. Ketiga, penyajian data melalui antarmuka grafis (*GUI*). Sistem dirancang untuk mengumpulkan dan menampilkan informasi penting, seperti penggunaan *CPU, RAM, dan Disk*, serta status operasional *server* secara *real-time*. Data yang terkumpul diolah dan disimpan dalam *database RRD* untuk divisualisasikan dalam bentuk grafik yang informatif dan mudah dipahami. Hasil implementasi menunjukkan bahwa sistem mampu melakukan monitoring secara *real-time* dan memberikan notifikasi saat terjadi anomali atau kegagalan pada *server*. Penggunaan *SNMP* memungkinkan pengumpulan data yang efisien dan standar, sementara *RRD Tools* menyediakan mekanisme penyimpanan data yang ringkas dan berkelanjutan. Antarmuka *GUI* yang dibangun menggunakan *Python* memberikan kemudahan akses informasi dan analisis kinerja *server*. Kesimpulan penelitian ini adalah bahwa sistem *monitoring server* berbasis *SNMP* dan *RRD Tools* dengan antarmuka *GUI* dapat diimplementasikan secara efektif untuk mendukung pengelolaan infrastruktur TI yang lebih baik. Sistem ini tidak hanya membantu dalam pemantauan kinerja *server* tetapi juga dalam pengambilan keputusan cepat saat terjadi masalah, sehingga meningkatkan ketersediaan dan keandalan layanan TI.

**Kata Kunci:** *Monitoring Server, SNMP, RRD Tools, GUI, Python*

## *SERVER MONITORING SYSTEM WITH SNMP AND RRD TOOLS*

**Abstract**-In the increasingly advanced digital era, server availability and performance are critical elements in information technology infrastructure. Effective server monitoring is important to ensure services continue to run optimally and disruptions can be identified and handled quickly. This research aims to implement a server monitoring system that utilizes *Simple Network Management Protocol (SNMP)* and *Round Robin Database (RRD) Tools* with a *Graphical User Interfaces (GUI)* interface using *Python*. This research method involves several stages. First, data collection from the server uses the *SNMP* protocol which is capable of supporting various versions (*v1, v2c, v3*). Second, data storage in *RRD* format which provides a concise and sustainable data storage mechanism. Third, presenting data through a graphical interface (*GUI*). The system is designed to collect and display important information, such as *CPU, RAM, and Disk usage*, as well as server operational status in *real-time*. The collected data is processed and stored in the *RRD* database to be visualized in the form of graphs that are informative and easy to understand. The implementation results show that the system is capable of monitoring in *real-time* and providing notifications when anomalies or failures occur on the server. The use of *SNMP* enables efficient and standardized data collection, while *RRD Tools* provides a compact and sustainable data storage mechanism. A *GUI* interface built using *Python* provides easy access to information and server performance analysis. The conclusion of this research is that a server monitoring system based on *SNMP* and *RRD Tools* with a *GUI* interface can be implemented effectively to support better *IT infrastructure* management. This system not only helps in monitoring server performance but also in making quick decisions when problems occur, thereby increasing the availability and reliability of *IT services*.

**Keywords:** *Server Monitoring, SNMP, RRD Tools, GUI, Python*

## 1. PENDAHULUAN

Dalam era *digital* ini, keberlangsungan operasional sistem informasi sangat bergantung pada kinerja *server* yang handal dan efisien. *Server* berperan sebagai tulang punggung dalam penyimpanan, pengolahan, dan distribusi data [1]. Oleh karena itu, memantau kondisi dan kinerja *server* secara terus-menerus merupakan hal yang esensial untuk memastikan bahwa sistem tetap berjalan dengan optimal tanpa gangguan [2].

Seringkali, masalah pada *server* tidak terdeteksi hingga menimbulkan dampak yang signifikan pada operasional, seperti *downtime* yang berkepanjangan atau kerusakan data [3]. Dalam konteks ini, implementasi sistem *monitoring server* menjadi sangat penting. Sistem *monitoring server* yang baik tidak hanya mampu

mendeteksi masalah secara *real-time*, tetapi juga memberikan informasi historis tentang kinerja *server* sehingga memudahkan analisis dan perencanaan kapasitas di masa mendatang.

*Simple Network Management Protocol* (SNMP) adalah protokol jaringan yang digunakan secara luas untuk mengelola dan memantau perangkat jaringan [4]. *SNMP* memungkinkan *administrator* jaringan untuk mengumpulkan informasi dari perangkat yang dikelola dan mengubah konfigurasi mereka. Dengan menggunakan *SNMP*, kita dapat memantau berbagai parameter kinerja server seperti penggunaan *CPU*, *Memory*, dan *Disk* [5].

*RRDTool* (*Round Robin Database Tool*) adalah alat yang efektif untuk menyimpan dan memvisualisasikan data kinerja dalam bentuk grafik. *RRDTool* memungkinkan penyimpanan data historis dengan cara yang efisien sehingga penggunaan ruang penyimpanan tetap minimal dan informasi historis tetap dapat diakses dengan mudah [2], [6], [7]. *RRDTool* merupakan standar industri *OpenSource*, pencatatan data berkinerja tinggi dan sistem grafik untuk data deret waktu. *RRDtool* dapat dengan mudah diintegrasikan dalam *skrip shell*, *perl*, *python*, *ruby*, *lua* atau aplikasi *tcl*.

Data yang dikumpulkan dan disimpan saja tidak cukup. Penggunaan antarmuka GUI (*Graphical User Interface*) yang intuitif dan *user-friendly* sangat penting agar informasi yang diperoleh dapat dengan mudah diinterpretasikan dan dimanfaatkan oleh pengguna. Dalam hal ini, bahasa pemrograman Python menawarkan berbagai pustaka yang memungkinkan pengembangan antarmuka GUI yang interaktif dan menarik. Penelitian ini bertujuan untuk mengimplementasikan sistem *monitoring server* menggunakan *SNMP* dan *RRDTool* dengan antarmuka GUI berbasis *Python*. Sistem ini diharapkan dapat membantu *administrator* jaringan umumnya dalam memantau kinerja *server* secara *real-time* dan historis, serta dapat mengambil tindakan proaktif untuk mencegah gangguan operasional.

Ruang lingkup sistem *monitoring* ini hanya akan memantau server yang berada dalam jaringan lokal (LAN) dan tidak mencakup *server* yang berada di luar jaringan dan hanya parameter kinerja dasar yang akan dimonitor, seperti penggunaan *CPU*, *Memory*, dan *Disk*. Parameter lain seperti suhu, kecepatan kipas, dan status daya tidak termasuk dalam lingkup penelitian ini.

Teknologi yang digunakan untuk pengumpulan data adalah *Simple Network Management Protocol* (SNMP) dengan mekanisme penyimpanan dan visualisasi data yang digunakan adalah *Round Robin Database* (RDB) *Tool*. Sedangkan untuk bahasa pemrograman yang digunakan untuk pengembangan antarmuka GUI adalah *Python*.

Antarmuka GUI yang dikembangkan hanya menyediakan fitur untuk memonitor dan menampilkan data kinerja *server* (*CPU*, *Memory* dan *Disk*). Fitur tambahan seperti notifikasi otomatis atau integrasi dengan sistem lain belum diterapkan. Antarmuka akan dirancang tidak mencakup desain responsif untuk perangkat *mobile*.

Perbedaan penelitian sebelumnya dengan penelitian sekarang adalah tidak adanya fitur notifikasi otomatis, integrasi sistem, desain *mobile* dan aplikasi tidak berbasis *web*. Pengujian sistem hanya akan dilakukan dalam lingkungan jaringan yang telah dikonfigurasi sebelumnya dan tidak mencakup uji coba dalam kondisi jaringan yang tidak stabil atau dengan beban kerja yang sangat tinggi. Begitu pula dengan pengujian keamanan sistem di luar ruang lingkup penelitian ini dan tidak akan dibahas. Tujuan yang ingin dicapai dengan membangun sistem ini adalah mengembangkan sistem *monitoring server* yang menggunakan *SNMP* (mengumpulkan data), implementasi *RRDTool* (menyimpan data dan menampilkan grafik), dan membangun antarmuka GUI.

Terdapat beberapa manfaat dengan pembuatan sistem ini adalah adanya solusi *monitoring server* yang dalam memantau kinerja *server*, meminimalisasikan risiko *downtime* dengan deteksi awal melalui *monitoring real-time* dan menyediakan informasi historis untuk analisa dan perencanaan kapasitas *server*. Dalam melakukan penelitian, penulis menggunakan metode perbandingan untuk memberikan gambaran komprehensif mengenai bagaimana penelitian ini berbeda atau melengkapi studi yang telah dilakukan sebelumnya. Di bawah ini adalah beberapa hasil penelitian sebelumnya:

- Implementasi *Simple Network Management Protocol* (SNMP) Pada Aplikasi *Monitoring* Jaringan Berbasis *Website* (Studi Kasus Universitas Muhammadiyah Bengkulu) [6]. Untuk mengetahui membandingkan sistem *monitoring* berbasis *web* dan *desktop* dapat diimplementasikan dengan menggunakan *SNMP* sebagai protokol pengumpul *data monitoring* dan *database Round Robin* (*RRDtool*) untuk analisis data *monitoring* dan menampilkan data hasil *monitoring* dalam bentuk grafik.
- Sistem *Monitoring Server* Dengan Menggunakan *SNMP* [8]. Membandingkan sistem yang mampu melakukan pemantauan segala aktivitas di dalam *server* yang dapat memonitor dan mendeteksi apabila terjadi permasalahan pada *server* yang digunakan.
- Network Monitoring System Application Based On Website* [9]. Membandingkan cara melakukan *monitoring* untuk mengamati kondisi *traffic*, *delay* dan *jitter* pada *server*.
- Sistem *Monitoring Server & Perangkat Jaringan* Pada *Enterprise* Menggunakan *Protocol ICMP & SNMP* [10]. Membandingkan metode *monitoring* yang menggunakan *SNMP* dan *ICMP*. Berbeda dengan sistem yang penulis bangun menggunakan *SNMP* dan *RRDTools*.

- e. *RRDTool : A Round Robin Database for Network Monitoring* [7]. Membandingkan target *monitoring* yang dilakukan.
- g. Perancangan Dan Implementasi *Monitoring Server Owncloud* [11]. Penelitian ini melakukan perancangan dan implementasi *monitoring server owncloud* menggunakan *software Cacti*.
- h. Laporan Utilisasi dan Kontrol *Dashboard* Perangkat Berbasis *Cacti* [12]. Sistem menggunakan *software Cacti* dalam menghasilkan laporan.

## 2. METODE PENELITIAN

### 2.1. Sumber Data

- a. Observasi Dilakukan untuk memahami kondisi awal, mengidentifikasi parameter kinerja yang relevan, serta mengamati bagaimana data performa dikumpulkan dan dianalisis.
- b. Studi Pustaka. Mempelajari beberapa teori literatur dan buku referensi, serta situs penyedia layanan yang berhubungan dengan objek penelitian.
- c. Studi Literatur Penelitian. mengumpulkan referensi-referensi dari hasil penelitian, seperti jurnal-jurnal hasil penelitian, bahan seminar yang dijadikan sebagai acuan dalam pembuatan sistem.

### 2.2. Kebutuhan

- a. Perangkat keras yang terdiri dari 1 (satu) unit *server* berbasis *virtualisasi*.

Tabel 1. Kebutuhan Perangkat Keras

Fungsi	Network	CPU	Memory	Disk
Monitoring Server	Internal Network /Host-Only Adapter	2vCPU	2 GB	10vGB

Kebutuhan terhadap perangkat keras ini digunakan dalam membuat *server virtualisasi* yang menjalankan sistem monitoring dengan menggunakan *protokol SNMP* dan *RRDTools*.

- b. Perangkat lunak yang dibutuhkan guna sistem ini adalah :

Tabel 2. Kebutuhan Perangkat Lunak

Fungsi	Network
Monitoring Server	a. Windows Server 2016 atau 2012 b. SNMP Service c. ( <i>RRDTool</i> ) Python 3.9.10
Virtualization	a. Oracle VM VirtualBox 7.0

Kebutuhan terhadap perangkat lunak ini digunakan untuk *service SNMP Agent*, *Virtualization software*, *RRDTools*, dll.

Data yang dikumpulkan dalam sistem ini bersumber dari:

- 1) Data kinerja *processor*, seperti: *CPU Utilization*.
- 2) Data kinerja *memory*, seperti: *Total Memory*, *Memory Used*, *Memory Free* dan *Swap Usage*.
- 3) Data kinerja *disk*, seperti: *Total Disk*, *Disk Used*, dan *Disk Free*

### 2.3. Metode Penerapan

- a. Pembuatan *Round Robin Database (RRD)* untuk menyimpan data yang dikumpulkan dari *server* menggunakan *SNMP*. Algoritma sebagai berikut:

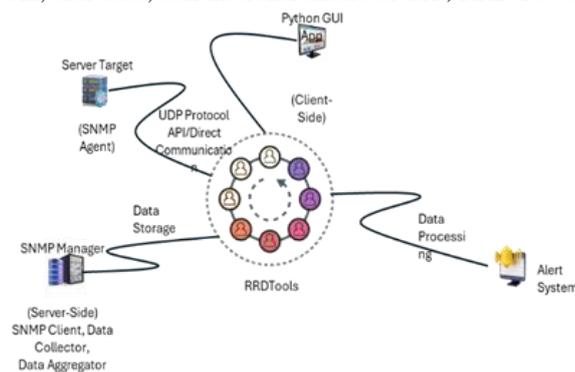
Inisialisasi dan buat *database*

```
def create_rrd(filename, step=300, heartbeat=600):
    rrdtool.create(
        filename,
        "--step", str(step),
        "DS:cpu_usage:GAUGE:" + str(heartbeat) + ":0:100",
        "DS:memory_usage:GAUGE:" + str(heartbeat) + ":0:100",
        "DS:disk_usage:GAUGE:" + str(heartbeat) + ":0:100",
        "RRA:AVERAGE:0.5:1:288", # Menyimpan data per 5 menit selama 1 hari (288 data points)
        "RRA:AVERAGE:0.5:6:336" # Menyimpan data per 30 menit selama 1 minggu (336 data points)
    )
```

- b. Pengumpulan data menggunakan *SNMP*. Algoritmanya sebagai berikut:
- ```
def collect_snmp_data(ip, community='public'):
    session = Session (hostname=ip, community=community, version=2)
    # OID untuk CPU, memory, dan disk
    oid_cpu = '1.3.6.1.4.1.2021.11.9.0' # OID untuk CPU load
    oid_memory = '1.3.6.1.4.1.2021.4.5.0' # OID untuk total memory
    oid_disk = '1.3.6.1.4.1.2021.9.1.6.1' # OID untuk total disk
    # Mengambil data SNMP
    cpu_usage = session.get(oid_cpu).value
    memory_usage = session.get(oid_memory).value
    disk_usage = session.get(oid_disk).value
    # Konversi ke tipe data yang sesuai (misalnya, integer)
    cpu_usage = int(cpu_usage)
    memory_usage = int(memory_usage)
    disk_usage = int(disk_usage)
    # Kembalikan data yang diambil
    return cpu_usage, memory_usage, disk_usage
```
- d. Memasukkan data ke *Round Robin Database (RRD)*.
- ```
def update_rrd(cpu_usage, memory_usage, disk_usage):
    # Tentukan nama file database RRD
    rrd_filename = "server_monitoring.rrd"
    # Dapatkan timestamp saat ini
    timestamp = int(time.time())
    # Format data untuk RRD
    data = f"{timestamp}:{cpu_usage}:{memory_usage}:{disk_usage}"
    # Masukkan data ke RRD
    rrdtool.update(rrd_filename, data)
```

### 2.3. Rancangan Arsitektur Sistem

Rancangan arsitektur sistem yang dibuat merupakan diagram untuk menggambarkan rancangan hubungan antar komponen sistem, alur data, dan interaksi antara *SNMP*, *RRD Tools*, dan antarmuka *GUI*.



Gambar 1. Rancangan Arsitektur Sistem

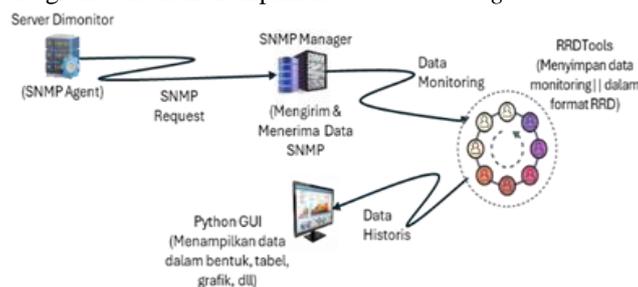
Gambar 1 dapat dijelaskan sebagai berikut :

- a. *Client-Side (User Interface)*
  - 1) *Python GUI Application*. Antarmuka pengguna yang dibangun menggunakan *library Tkinter* untuk menampilkan informasi *monitoring server*.
  - 2) *User Input*. Pengguna dapat menambahkan *server* yang akan dimonitor, mengatur *interval polling*, dan melihat statistik *server*.
- b. *Server-Side*
  - 1) *SNMP Manager*.
    - a) *SNMP Client*. Modul yang berfungsi untuk mengirimkan permintaan *SNMP* ke *server* yang dimonitor.
    - b) *Data Collector*. Mengumpulkan data dari *server* target menggunakan *SNMP* (seperti penggunaan *CPU, memory, status server*).

2. *Round Robin Database (RRD) Tool.*  
*RRDTool.* Digunakan untuk menyimpan data dalam format *Round Robin Database*, untuk penyimpanan data *time-series* dan mendukung kompresi otomatis data lama.
3. *Data Processing*
  - a) *Data Aggregator.* Mengolah data mentah dari *SNMP* menjadi format yang bisa disimpan dalam *RRD*.
  - b) *Alert System.* Mengirimkan notifikasi kepada pengguna jika ada anomali (misalnya *server down* atau penggunaan *CPU* tinggi).
4. *Communication Layer*
  - a) *SNMP Protocol.* Protokol yang digunakan untuk komunikasi antara *SNMP manager* dan *SNMP agent* di *server* yang dimonitor.
  - b) *Network Interface.* Mendukung komunikasi antara *client-side* dan *server-side*, serta antara *server-side* dengan *server* yang dimonitor.
5. *Monitored Servers*  
*SNMP Agents,* Aplikasi atau *daemon* yang berjalan di *server* yang dimonitor untuk menyediakan data *monitoring* kepada *SNMP manager*.

## 2.4. Rancangan Desain Sistem

- a. *Server* dengan *SNMP Agent.* *Server* yang akan dimonitor, dilengkapi dengan *SNMP Agent* yang mengirimkan data *monitoring*.
- b. *SNMP Manager.* Aplikasi yang mengirimkan permintaan *SNMP* dan menerima respons dari *SNMP Agent*.
- c. *RRD Tools:* Alat untuk menyimpan dan mengelola data *monitoring* dalam format *RRD*.
- d. *Python GUI.* Antarmuka grafis untuk menampilkan data *monitoring*.



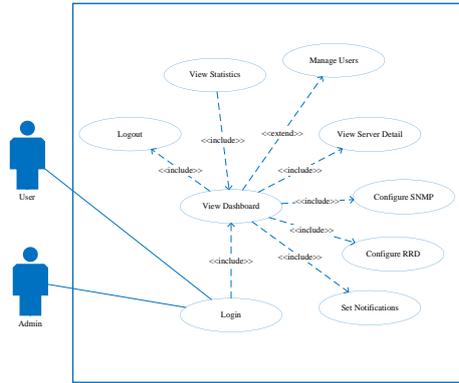
**Gambar 2.** Rancangan Desain Sistem

Gambar 2 dapat dijelaskan sebagai berikut :

- a. *Server* dimonitor (*SNMP Agent*). Data yang Dimonitor. Data seperti *CPU usage*, *memory usage*, dan *disk usage*.
- b. *SNMP Manager.* Fungsi Utama Mengirimkan permintaan *SNMP* ke *server* dan menerima respons dari *SNMP Agent server*.
- c. *RRD Tools.* Menyimpan data *monitoring* dalam format *RRD* untuk penyimpanan data historis dan manipulasi data.
- d. *Python GUI.* Visualisasi Data. Menampilkan data dalam bentuk tabel, grafik, dan visualisasi lainnya.

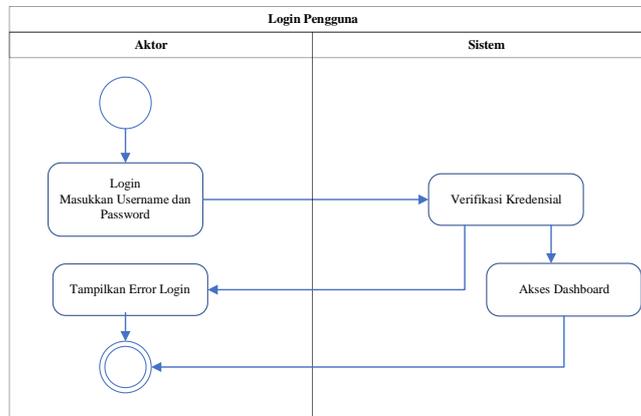
## 2.5. Rancangan Unified Modeling Language (UML)

- a. *Use Case Diagram.* Menggambarkan interaksi antara aktor (pengguna) dan sistem serta fungsionalitas.

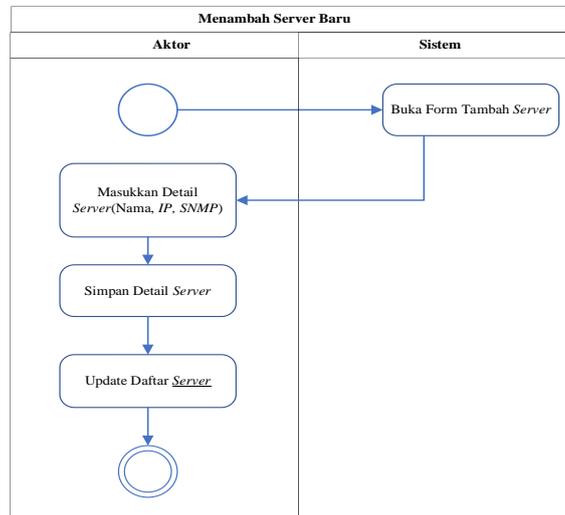


Gambar 3. Use Case Diagram

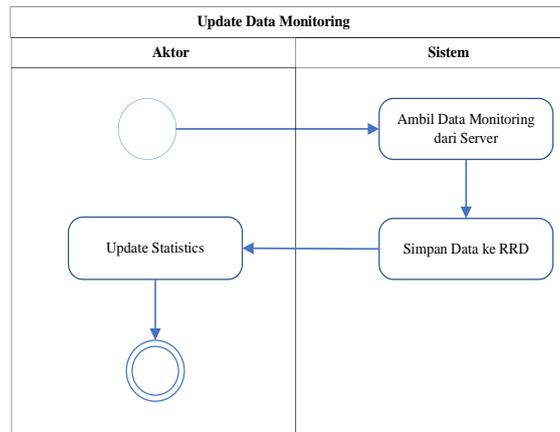
b. Activity Diagram. Menggambarkan alur aktivitas dalam sistem.



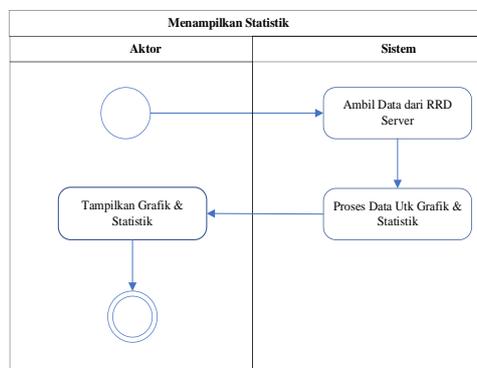
Gambar 4. Activity Diagram Login Pengguna



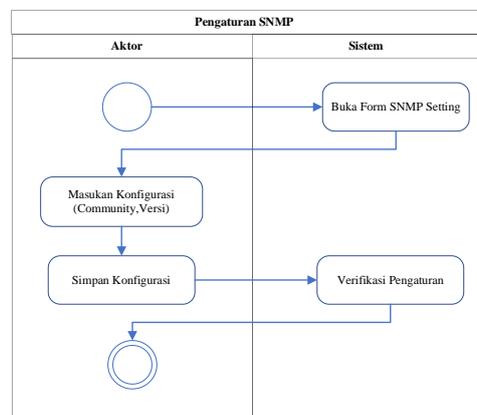
Gambar 5. Activity Diagram Tambah Server



Gambar 6. Activity Diagram Update Data Monitoring



Gambar 7. Activity Diagram Menampilkan Statistik



Gambar 8. Pengaturan SNMP

## 2.6. Rancangan Pengujian

- Pengujian Unit. Uji *SNMP Manager* mengirimkan permintaan *SNMP* dan menerima data dari *SNMP Agent*.
- Pengujian Integrasi. Uji data dari pengumpulan data melalui *SNMP*, penyimpanan di *RRD*, hingga penampilan di GUI.
- Pengujian Fungsional. Uji setiap fitur dari sistem sesuai dengan spesifikasi fungsional.
- Pengujian Kinerja. Uji sistem dengan data *monitoring* dalam jumlah besar dan ukur waktu *respon* dan penggunaan sumber daya.
- Pengujian Keandalan. Uji jalankan sistem secara kontinu selama periode waktu tertentu dan catat setiap kesalahan atau *crash* yang terjadi.
- Pengujian Pengguna. Uji beberapa pengguna menggunakan sistem dan kumpulkan umpan balik mereka.

Untuk rancangan menu mencakup menu utama yang dibutuhkan untuk memonitor *server*, yaitu :

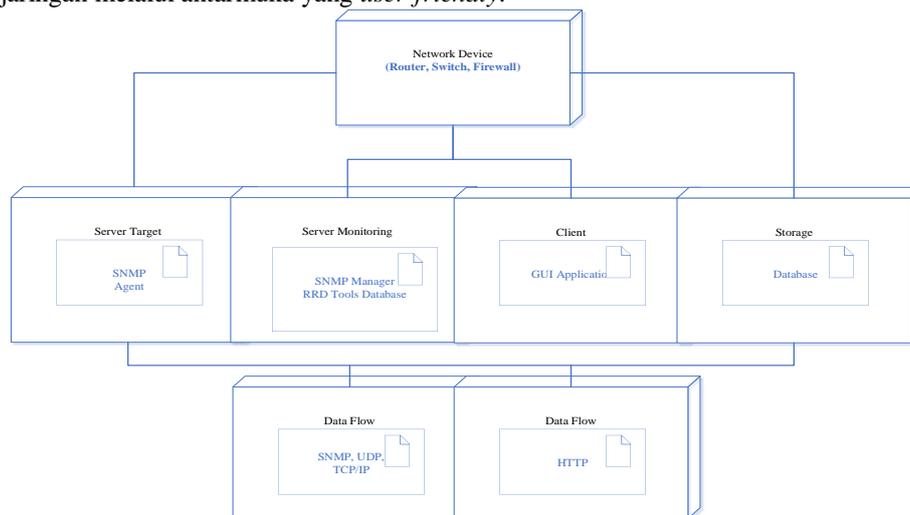
- Dashboard*. Menampilkan ringkasan status *server* dan informasi penting lainnya.
- Monitoring*. Akses ke halaman pemantauan detail untuk setiap *server*.
- Statistics*. Menampilkan grafik dan statistik penggunaan *CPU*, *RAM*, dan *disk*.
- Settings*. Pengaturan untuk konfigurasi sistem, termasuk *SNMP settings* dan pengaturan *RRD*.
- About*. Informasi tentang aplikasi, versi, dan pengembang.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Deployment Diagram

Gambar 9 di bawah dapat dijelaskan sebagai berikut :

- User Device* berinteraksi dengan *Monitoring Server* melalui antarmuka dapat berupa aplikasi *desktop* atau berbasis *web*.
- Pengumpulan Data. *SNMP Daemon* pada *Monitoring Server* mengirimkan permintaan *SNMP* ke perangkat jaringan dan menerima data kinerja.
- Penyimpanan Data. Data yang diterima oleh *SNMP Daemon* disimpan dalam *RRDTool Database* oleh komponen *Backend Logic*.
- Pengolahan dan Penyajian Data. *Backend Logic* mengolah data dan menyediakan *endpoint* untuk diakses oleh *GUI Application* di *Client*.
- Visualisasi Data. *Application* pada *Client* mengambil data dari *Backend Logic* dan menampilkan informasi kinerja jaringan melalui antarmuka yang *user-friendly*.



Gambar 9. Deployment Diagram Sistem

#### 3.2. Implementasi Metode

- Pengumpulan Data. *SNMP Manager* mengumpulkan data dari *SNMP agent* secara periodik. Data dapat mencakup metrik seperti penggunaan *CPU*, *memori*, dan *disk*.
  - Pengolahan Data. Data dari *SNMP manager* dapat dimasukkan ke dalam *database RRD* menggunakan *rrdtool update*. Data disimpan dalam format *RRD* dan dapat digunakan untuk pembuatan grafik.
  - Penampilan Data. Grafik dibuat dari data *RRD* menggunakan *rrdtool graph*. Data ditampilkan dalam format grafik di antarmuka pengguna, memperlihatkan tren sumber daya *server*.
- Alur proses Implementasi Metode seperti dalam gambar 10.



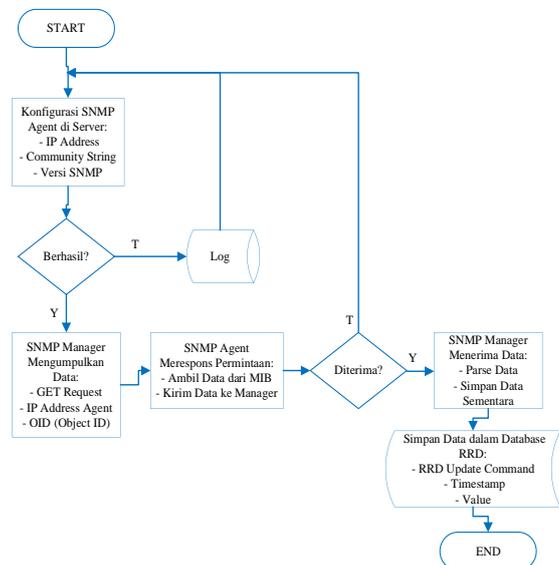
**Gambar 10.** Alur Proses Implementasi Metode

Gambar 10 dapat dijelaskan sebagai berikut:

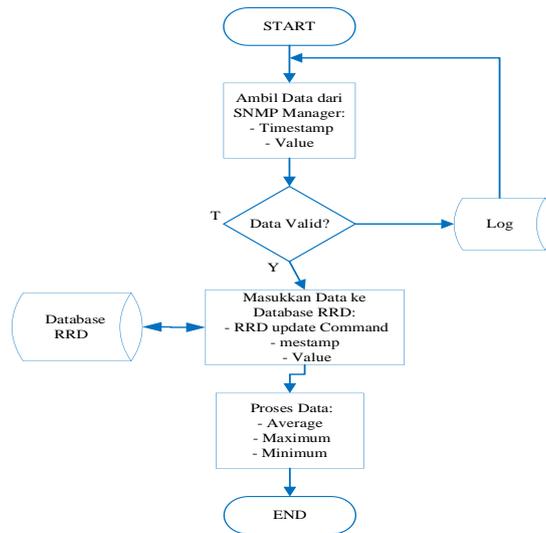
- Server SNMP Agent* yang dipantau/dimonitor mengirimkan data OID ke *Server SNMP Manager*.
- SNMP Manager* mengumpulkan data dari *Server SNMP Agent* dan mengirimkannya ke *Round Robin Database tool* untuk proses penyimpanan dalam retensi waktu yang ditentukan sebelumnya.
- RRDtool* menyimpan dan mengolah data yang selanjutnya menghasilkan grafik.
- Selanjutnya *Python GUI* menampilkan grafik dan data dari hasil proses *RRDtool* kepada *Client*.

Kelebihan penelitian ini adalah pada *Protokol SNMP* yang menggunakan komunikasi *connectionless (UDP)* yang tidak memerlukan sesi yang stabil antara pengirim dan penerima, sehingga tidak memberikan beban tambahan pada jaringan (penghematan bandwidth). Ini membuat monitoring berjalan tanpa mengganggu performa jaringan.

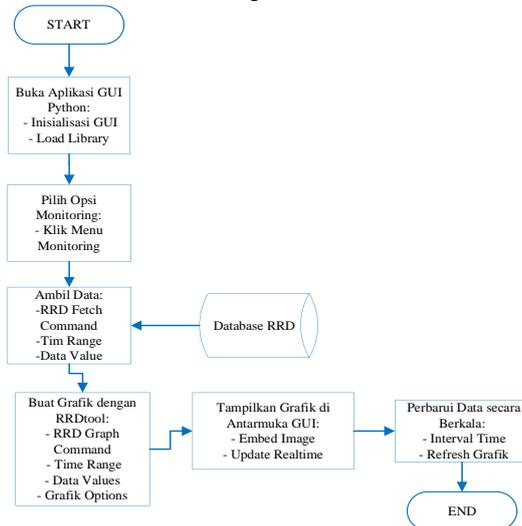
Proses kerja sistem terbagi dalam 4 (empat) alur proses yang saling terkait. Berikut adalah *diagram flowchart* yang menggambarkan alur proses dari sistem ini:



**Gambar 11.** Pengumpulan Data SNMP



Gambar 12. Pengolahan Data RRD



Gambar 13. Tampilkan Data GUI Python

### 3.3. Hasil Pengujian

Tabel 3. Hasil Pengujian

Aspek Pengujian	Metode Pengujian	Target Pengujian	Hasil Pengujian
Pengumpulan Data	Waktu respon an akurasi data	<1 detik, akurasi <5%	Berhasil
Pengolahan Data	Waktu penulisan dan pengambilan	0.5 detik, < 1 detik	Berhasil
Pembaharuan Grafik	Waktu Pembuatan dan pembaharuan	< 2 detik, < 1 detik	Berhasil
Antarmuka Pengguna	Responsivitas GUI dan penggunaan sumber daya	< 0.5 detik, CPU < 5%, <i>memory</i> < 100 MB	Berhasil

## 4. KESIMPULAN

### 4.1. Kesimpulan

- Penggunaan *SNMP* sebagai protokol untuk mengumpulkan data dari *server* terbukti berhasil. *SNMP* memungkinkan pengumpulan data secara *real-time* dari berbagai *server* yang berbeda.
- RRDTool* berhasil digunakan untuk menyimpan dan mengelola data yang dikumpulkan.
- Antarmuka *GUI Python* yang dirancang untuk sistem ini terbukti mudah digunakan dan mampu menyajikan informasi yang dibutuhkan dengan baik.

### 4.2. Saran

- a. Sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur monitoring tambahan.
- b. Untuk meningkatkan aksesibilitas, sistem ini dapat dikembangkan lebih lanjut dengan menyediakan antarmuka berbasis web.
- c. Peningkatan sistem lebih lanjut dapat dilakukan dengan menerapkan standar keamanan yang lebih tinggi, seperti penggunaan protokol komunikasi yang lebih aman (misalnya, *SNMPv3*, *audit log*, dan mekanisme deteksi intrusi).

## DAFTAR PUSTAKA

- [1] H. Harsono, "Faktor-Faktor Yang Mempengaruhi Sistem Informasi Berbasis Komputer: Sistem Operasi, Server, Dan Programmer (Literature Review Executive Support Sistem For Business)," *J. Manaj. Pendidik. DAN ILMU Sos.*, 2022, doi: 10.38035/jmpis.v3i2.1121.
- [2] A. Hardito, "Sistem Pengawasan Kinerja Multi Net- Server Melalui Media On-Line," *Orbith*, vol. 11, no. 1, 2015.
- [3] S. F. Sulaiman and A. H. Priambodo, "Downtime Data Center: Memahami Penyebab, Dampak, dan Solusi Efektif," *Sanskara Manaj. Dan Bisnis*, vol. 2, no. 2, pp. 67–78, 2024, doi: 10.58812/smb.v2i02.297.
- [4] N. Nendi and F. Maulana, "Monitoring Traffic Berbasis SNMP pada Jaringan Perumahan Permata Puri Harmoni 2," *J. Sains dan Teknol.*, vol. 5, no. 3, 2024, doi: 10.55338/saintek.v5i3.1346.
- [5] S. Gee and J. Vianna, "SNMP-Simple Network Management Protocol," *ThousandEyes*. <https://www.thousandeyes.com/learning/techtutorials/snmp-simple-network-management-protocol> (accessed May 03, 2024).
- [6] F. Maulana, "Implementasi Simple Network Management Protocol (Snmp) Pada Aplikasi Monitoring Jaringan Berbasis Website(Studi Kasus Universitas Muhammadiyah Bengkulu)," *J. Inform.*, vol. 16, no. 2, pp. 126–135, 2016.
- [7] M. Singh, H. Mewada, M. Tahilyani, J. Malviya, R. Sharma, and S. S. Shrivastava, "RRDTool: A Round Robin Database for Network Monitoring," *Journal of Computer Science*. 2022. doi: 10.3844/jcssp.2022.770.776.
- [8] Y. N. Wengim, "Rancang Bangun Aplikasi Monitoring Jaringan Dengan Menggunakan SNMP Berbasis Desktop (Smart Campus-USTJ)," *J. Tek. Inform.*, vol. 9, pp. 73–80, 2020.
- [9] R. S. Saputra, Hafidudin, and D. N. Ramadan, "Aplikasi Sistem Monitoring Jaringan Berbasis Website Network," *JETT J. Elektro dan Telekomun. Terap. Univ.*, 2018.
- [10] A. Heryanto, A. Hermansyah, and M. Nizar, "Sistem Monitoring Server dan Perangkat Jaringan pada Enterprise Resource Planning FASILKOM UNSRI Menggunakan Protokol ICMP dan SNMP," *J. Sist.*, vol. 6, no. 2, 2017.
- [11] J. Tafonao, E. Fatkhiyah, and R. K. Yuliana Rachmawati, "Perancangan Dan Implementasi Monitoring Server Owncloud," *J. Jarkom*, 2019.
- [12] D. P. Kuswandono, M. Ryan, W. Hermawansyah, and Suhadi, "Aotomasi Laporan Ulitisasi dan Kontrol Dashboard Perangkat Berbasis Cacti dengan Bash Shell Scripting," *J. Gerbang STMIK Bani Saleh*, vol. 13, no. 1, 2023.
- [13] D. Intern, "Apa itu UML? Beserta Pengertian dan Contohnya," *Dicoding*, 2021. <https://www.dicoding.com/blog/apa-itu-uml/> (accessed Apr. 06, 2024).