

Laporan Akhir
Membangun Sistem Monitoring untuk Penyimpanan Akses Acak dan
Unit Pemrosesan Utama di Mesin Virtual dengan Grafana dan
Prometheus

Diajukan untuk Memenuhi Tugas Mata Kuliah
”Workshop Administrasi Jaringan”
Dosen: Ferry Astika



Oleh:
Dukhaan Kamimpangan (3122600003)
Rakha Putra Pratama (3122600005)
Diah Aulia Kusuma Putri (3122600008)

2 D4 Teknik Informatika A
Semester 4
D4 Teknik Informatika
Departemen Teknik Informatika dan Komputer
Politeknik Elektronika Negeri Surabaya

DAFTAR ISI

DAFTAR ISI.....	1
A. ABSTRAKSI.....	2
B. PENDAHULUAN.....	2
C. RUANG LINGKUP.....	3
D. DESAIN SISTEM.....	4
E. TIM & TUGAS.....	6
F. TAHAPAN PELAKSANAAN.....	6
G. IMPLEMENTASI.....	7
H. SISTEM TESTING.....	14

A. ABSTRAKSI

Proyek ini bertujuan untuk membangun sebuah sistem monitoring yang efektif untuk penggunaan RAM dan CPU menggunakan Grafana dan Prometheus. Dalam era digital yang terus berkembang, kemampuan untuk memantau kinerja RAM dan penggunaan sumber daya CPU secara real-time menjadi semakin krusial. Dengan menggunakan Prometheus sebagai alat pengumpul data (data collection) dan Grafana sebagai alat visualisasi, sistem ini mampu memberikan informasi yang mendalam dan menyeluruh mengenai kondisi RAM dan penggunaan CPU. Implementasi ini tidak hanya membantu dalam mendeteksi masalah secara dini, tetapi juga memungkinkan pengelolaan sumber daya yang lebih efisien dan optimal. Studi ini mencakup langkah-langkah konfigurasi dan integrasi kedua alat tersebut, serta analisis hasil monitoring untuk pengambilan keputusan yang lebih baik dalam pengelolaan infrastruktur Teknologi Informasi.

Kata kunci: monitoring, grafana, prometheus, visualisasi data.

B. PENDAHULUAN

Perkembangan teknologi informasi yang pesat saat ini, khususnya di bidang telekomunikasi, telah memunculkan beragam layanan baru yang memanfaatkan sistem jaringan terintegrasi. Komunikasi data menjadi semakin mudah dilakukan dengan adanya jaringan komputer yang canggih. Namun, di balik kemajuan tersebut, sistem topologi jaringan komputer tetap memiliki kelemahan yang harus diatasi. Server sebagai perangkat integral dengan spesifikasi hardware dan software tertentu seperti web server, DNS server, dan proxy server, memungkinkan pembangunan jaringan skala besar. Seiring dengan peningkatan ukuran dan jumlah perangkat jaringan, risiko gangguan juga semakin tinggi. Oleh karena itu, manajemen jaringan, khususnya monitoring, menjadi sangat penting.

Monitoring jaringan secara rutin adalah kunci untuk memastikan jaringan selalu dalam kondisi baik dan tersedia bagi penggunaanya. Tugas ini umumnya dilakukan oleh administrator jaringan yang bertanggung jawab mengelola jaringan komputer. Salah satu tantangan dalam monitoring jaringan adalah

kebutuhan untuk terus memantau layar agar mengetahui setiap perubahan yang terjadi, yang sering kali kurang efisien.

Untuk mengatasi masalah ini, sistem monitoring secara real-time menggunakan aplikasi seperti Prometheus dan Grafana menjadi solusi yang efektif. Grafana mampu memberikan informasi kondisi komponen jaringan secara real-time dan mengirimkan pesan kepada administrator jaringan melalui platform seperti Telegram.

Laporan ini berjudul "Membangun Sistem Monitoring untuk Penyimpanan Akses Acak dan Unit Pemrosesan Utama di Mesin Virtual dengan Grafana dan Prometheus." Proyek ini bertujuan membangun sistem monitoring yang efektif untuk penggunaan RAM dan CPU. Di era digital yang terus berkembang, kemampuan memantau kinerja RAM dan penggunaan sumber daya CPU secara real-time sangat krusial. Dengan menggunakan Prometheus sebagai alat pengumpul data dan Grafana sebagai alat visualisasi, sistem ini mampu memberikan informasi mendalam mengenai kondisi RAM dan penggunaan CPU. Implementasi ini tidak hanya membantu mendeteksi masalah secara dini, tetapi juga memungkinkan pengelolaan sumber daya yang lebih efisien dan optimal. Studi ini mencakup langkah-langkah konfigurasi dan integrasi kedua alat tersebut serta analisis hasil monitoring untuk pengambilan keputusan yang lebih baik dalam pengelolaan infrastruktur Teknologi Informasi.

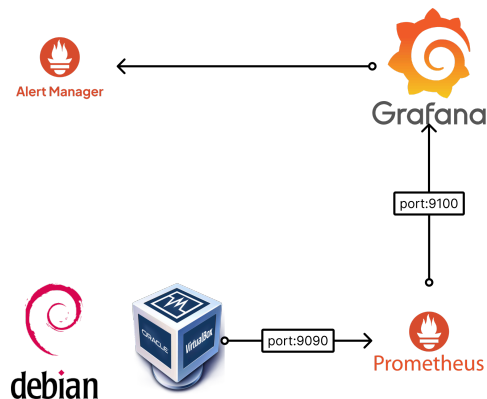
C. RUANG LINGKUP

Ruang lingkup laporan ini mengarah pada pembangunan sistem monitoring untuk penyimpanan akses acak (RAM) dan unit pemrosesan utama (CPU) di mesin virtual menggunakan Grafana dan Prometheus, yaitu sebagai berikut:

1. Penelitian ini dibatasi pada lingkungan mesin virtual yang digunakan dalam infrastruktur Teknologi Informasi.
2. Penelitian menggunakan metode implementasi dan integrasi alat monitoring Prometheus untuk pengumpulan data dan Grafana untuk visualisasi data.

3. Fokus utama adalah pada pemantauan kinerja RAM dan CPU secara real-time guna mendeteksi masalah sejak dini dan mengelola sumber daya secara efisien.

D. DESAIN SISTEM



1. Debian (Virtual Machine)
 - Sistem operasi Debian berjalan pada mesin virtual. Mesin ini adalah target yang akan dimonitor.
2. Prometheus
 - Prometheus berfungsi sebagai alat pengumpul data (data collection). Prometheus diinstal pada mesin virtual atau server terpisah.
 - Prometheus dikonfigurasi untuk mengumpulkan metrik dari mesin virtual melalui port 9090.
3. Node Exporter
 - Node Exporter adalah agen yang berjalan pada mesin virtual Debian. Ia mengumpulkan data metrik seperti penggunaan CPU dan RAM dan mengirimkannya ke Prometheus melalui port 9100.
4. Grafana
 - Grafana adalah alat visualisasi data yang mengambil data dari Prometheus.

- Dengan Grafana, administrator jaringan dapat membuat dashboard yang menampilkan metrik penggunaan RAM dan CPU secara real-time.

5. Alert Manager

- Alert Manager bertanggung jawab untuk mengelola dan mengirimkan notifikasi jika terjadi kondisi abnormal pada metrik yang dipantau.
- Dalam desain ini, Grafana dapat diintegrasikan dengan Alert Manager untuk mengirimkan pesan kepada administrator melalui platform komunikasi seperti Telegram.

Alur Kerja Sistem

1. Pengumpulan Data oleh Node Exporter

- Node Exporter mengumpulkan data metrik dari mesin virtual Debian, termasuk informasi tentang penggunaan RAM dan CPU.
- Data metrik ini kemudian dikirimkan ke Prometheus melalui port 9100.

2. Pengumpulan dan Penyimpanan Data oleh Prometheus

- Prometheus mengumpulkan data metrik dari Node Exporter pada interval waktu yang telah ditentukan dan menyimpannya dalam database internal.

3. Visualisasi Data oleh Grafana

- Grafana mengambil data metrik yang telah dikumpulkan oleh Prometheus.
- Administrator dapat membuat berbagai dashboard dan grafik untuk memvisualisasikan data penggunaan RAM dan CPU secara real-time.

4. Pengelolaan Peringatan oleh Alert Manager

- Grafana atau Prometheus dapat diatur untuk mengirimkan alert ke Alert Manager jika metrik tertentu mencapai ambang batas yang telah ditentukan.

- Alert Manager kemudian mengirimkan notifikasi kepada administrator melalui platform komunikasi seperti Telegram untuk tindakan lebih lanjut.

E. TIM & TUGAS

1. Dukhaan Kamimpangan (3122600003)
2. Rakha Putra Pratama (3122600005)
3. Diah Aulia Kusuma Putri (3122600008)

Eksekutor	Task
Dukhaan, Diah	Setup prometheus
Dukhaan, Diah	Setup Node Exporter
Dukhaan, Diah	Integrasi Node Exporter ke Prometheus
Rakha, Dukhaan	Setup Grafana
Rakha, Dukhaan	Menambahkan Dashboard Resource Monitoring di Grafana
Rakha, Dukhaan	Setup Alerting Contact Point dan bot Telegram
Rakha, Dukhaan	Memilih Query Data yang akan di set ke Alerting Rules
Dukhaan, Diah	Setup Alerting Alert Rules
Diah, Rakha	Membuat Laporan Hasil dan PPT

F. TAHAPAN PELAKSANAAN

Tanggal	Keterangan
27 Mei 2024	Merancang Project Chart & Pembagian Task Management
29 Mei 2024	Instalasi & Setup Prometheus dan Layanannya

30 Mei 2024	Instalasi & Setup Grafana dan Layanannya
1 Juni 2024	Implementasi dari Integrasi Resource Monitor CPU dan RAM di VM Menggunakan Prometheus dan Grafana
2 Juni 2024	Pembuatan Laporan & Finalisasi Laporan

G. IMPLEMENTASI

1. Instalasi prometheus dengan mengunduh dan ekstrak prometheus juga node exporter. Jalankan perintah di bawah ini :

- wget
<https://github.com/prometheus/prometheus/releases/download/v2.33.3/prometheus-2.33.3.linux-amd64.tar.gz>
- wget
https://github.com/prometheus/node_exporter/releases/download/v1.8.1/node_exporter-1.8.1.linux-amd64.tar.gz
- tar xvfz prometheus-2.33.3.linux-amd64.tar.gz
- tar xvfz node_exporter-1.3.1.linux-amd64.tar.gz
- cd prometheus-2.33.3.linux-amd64

2. Jalankan prometheus pada background proses dengan systemd dan lakukan hal yang sama untuk node exporter juga. Jalankan perintah di bawah ini:

- sudo useradd --no-create-home --shell /bin/false prometheus
- sudo mv /usr/local/prometheus/prometheus /usr/local/prometheus/promtool /usr/local/bin/
- sudo mv /usr/local/prometheus/consoles /usr/local/prometheus/console_libraries /etc/prometheus/
- sudo mv /usr/local/prometheus/prometheus.yml /etc/prometheus/prometheus.yml
- sudo mkdir /var/lib/prometheus
- sudo chown -R prometheus:prometheus /etc/prometheus /var/lib/prometheus

- `sudo nano /etc/systemd/system/prometheus.service`

Masukkan ini dalam file :

[Unit]

Description=Prometheus

Wants=network-online.target

After=network-online.target

[Service]

User=prometheus

Group=prometheus

Type=simple

ExecStart=/usr/local/bin/prometheus \

--config.file /etc/prometheus/prometheus.yml \

--storage.tsdb.path /var/lib/prometheus/ \

--web.console.templates=/etc/prometheus/consoles \

--web.console.libraries=/etc/prometheus/console_libraries

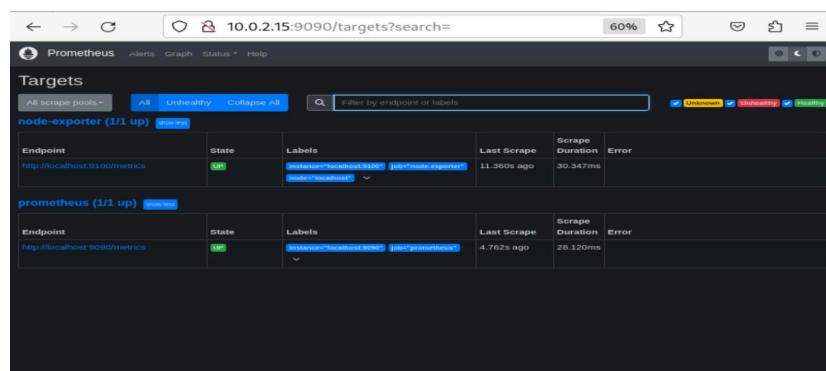
[Install]

WantedBy=multi-user.target

- `sudo systemctl daemon-reload`
- `sudo systemctl start prometheus`
- `sudo systemctl enable prometheus`

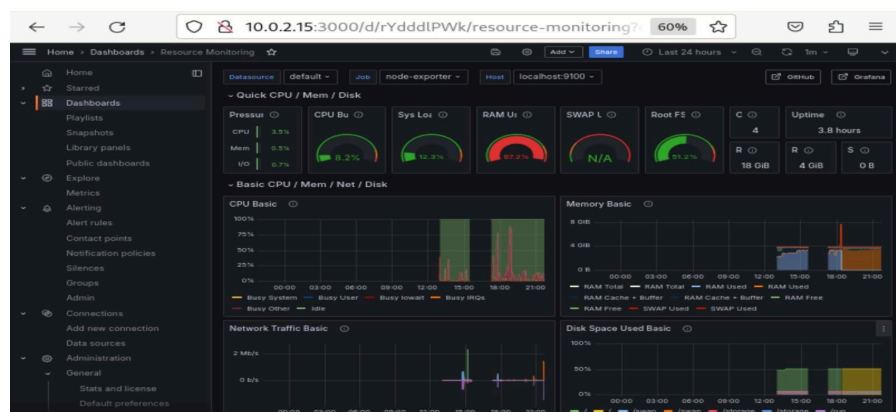
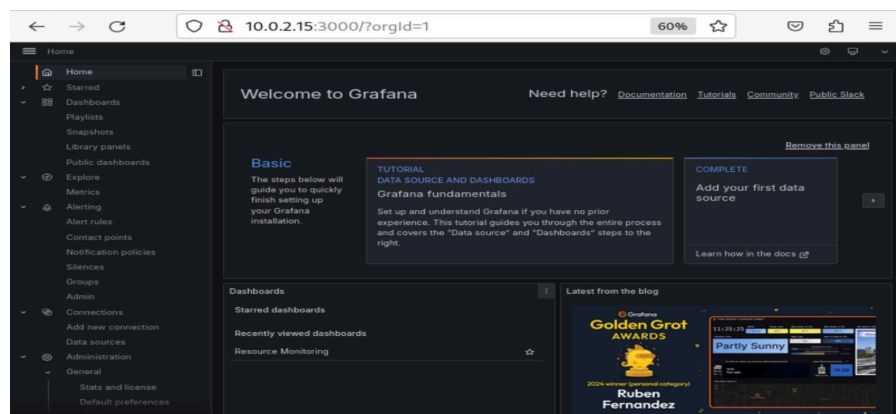
3. Output prometheus pada default port 9090, jalankan perintah di bawah ini:

- `sudo lsof -n -i | grep prometheus`



c. Membuat Dashboard

- Navigasi ke Create -> Dashboard.
- Tambahkan panel dengan query Prometheus untuk metrik yang diinginkan (misalnya, `node_cpu_seconds_total` dan `node_memory_MemAvailable_bytes`).



7. Konfigurasi Alerting dan pengiriman Notifikasi

a. Instal alert manager, jalankan perintah dibawah ini

- wget
<https://github.com/prometheus/alertmanager/releases/download/v0.23.0/alertmanager-0.23.0.linux-amd64.tar.gz>
- tar xvfz alertmanager-0.23.0.linux-amd64.tar.gz
- sudo mv alertmanager-0.23.0.linux-amd64 /usr/local/alertmanager

- b. Membuat pengguna alert manager
 - `sudo useradd --no-create-home --shell /bin/false alertmanager`
- c. Setel kepemilikan direktori
 - `sudo chown -R alertmanager:alertmanager /usr/local/alertmanager`
- d. Membuat file konfigurasi alert manager dengan membuat file `alertmanager.yml` dengan isi file sebagai berikut:


```
global:
  resolve_timeout: 5m

route:
  receiver: 'telegram'

receivers:
- name: 'telegram'
  telegram_configs:
  - api_url: 'https://api.telegram.org'
    bot_token: '<YOUR_BOT_TOKEN>'
    chat_id: '<YOUR_CHAT_ID>'
```
- e. Membuat file layanan systemd untuk alert manager, jalankan perintah dibawah ini:
 - `sudo nano /etc/systemd/system/alertmanager.service`
 Masukkan format dibawah ini pada file tersebut:


```
[Unit]
Description=Alertmanager
Wants=network-online.target
After=network-online.target

[Service]
User=alertmanager
Group=alertmanager
Type=simple
ExecStart=/usr/local/alertmanager/alertmanager \
```

--config.file=/usr/local/alertmanager/alertmanager.yml

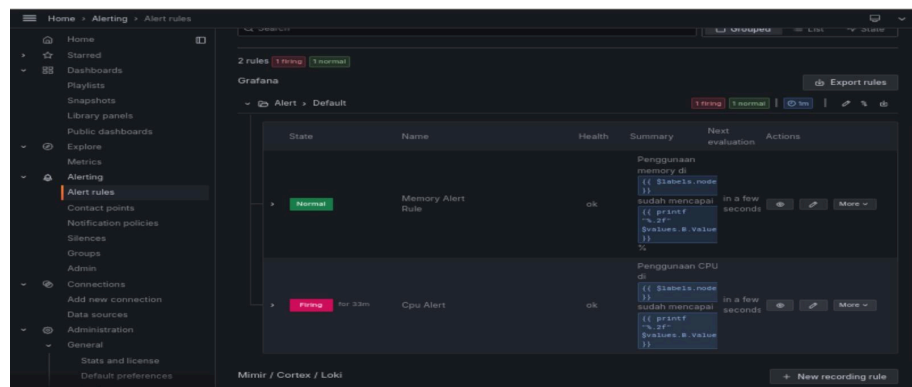
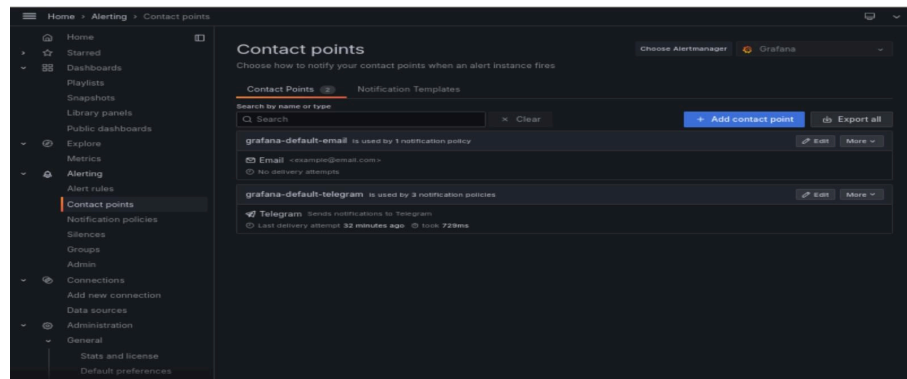
[Install]

WantedBy=multi-user.target

- f. Reload systemd dan jalankan alert manager
 - sudo systemctl daemon-reload
 - sudo systemctl start alertmanager
 - sudo systemctl enable alertmanager
 - g. Konfigurasi alerting pada prometheus pada grafana dengan beberapa case sebagai berikut:
 - Case 1
 - alertname : CPU Alert
 - Threshold : 3
 - Cpu Node Value : 5
 - Result : Firing
 - Case 2
 - alertname : Memory Alert
 - Threshold : 10
 - Cpu Node Value : 100
 - Result : Firing
 - Case 3
 - alertname : CPU Alert
 - Threshold : 100
 - Cpu Node Value : 100
 - Result : Resolved
8. Verifikasi dan monitoring
- a. Verifikasi instalasi
 - sudo systemctl status prometheus
 - sudo systemctl status grafana-server
 - sudo systemctl status alertmanager
 - sudo systemctl status node_exporter

b. Monitoring Dashboard

- Akses dashboard Grafana untuk memantau metrik RAM dan CPU secara real-time.
- Konfigurasi alert di Grafana untuk memanfaatkan Alertmanager.



H. SISTEM TESTING

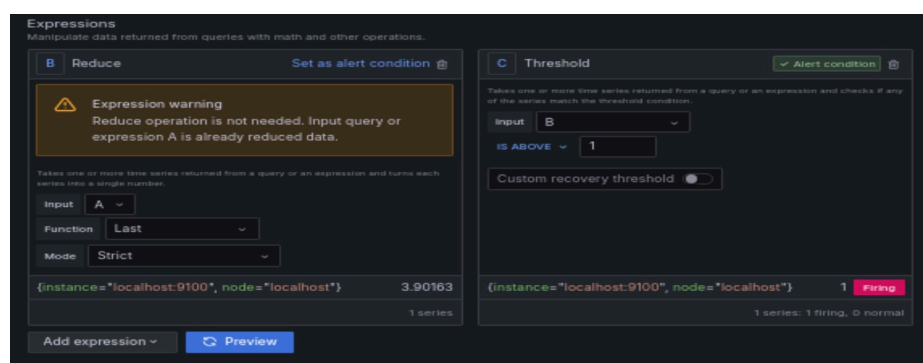
Pada bab ini, kami akan menjelaskan prosedur uji testing yang dilakukan pada aplikasi Grafana dan Prometheus yang telah dibangun. Proses uji testing ini bertujuan untuk memastikan bahwa sistem monitoring berjalan dengan baik dan sesuai dengan yang diharapkan. Berikut adalah rincian dari proses uji testing yang telah dilakukan:

1. Proses Menambahkan Threshold untuk Eksekusi Trigger Bot Telegram

a. **Tujuan:** Memastikan bahwa bot Telegram dapat menerima notifikasi ketika sebuah threshold tercapai.

b. **Langkah-langkah:**

- Konfigurasi threshold pada Prometheus untuk parameter tertentu, seperti penggunaan CPU atau memori.
- Buat aturan alert (alert rule) pada Prometheus yang mendefinisikan kondisi kapan threshold tersebut tercapai.
- Integrasikan alert rule ini dengan bot Telegram menggunakan webhook atau cara lain yang didukung oleh Telegram.
- Uji dengan cara memanipulasi data sehingga threshold tercapai dan pastikan bot Telegram menerima notifikasi



2. Memory Alert Rule Ketika Threshold Di Atas 10%

a. **Tujuan:** Memastikan bahwa alert akan firing ketika penggunaan memori mencapai atau melebihi 10% dari threshold yang ditentukan.

b. Langkah-langkah:

- i. Tentukan threshold penggunaan memori sebesar 10% pada konfigurasi Prometheus.
- ii. Buat aturan alert pada Prometheus yang menyatakan bahwa jika penggunaan memori melebihi 10%, maka alert harus firing.
- iii. Verifikasi aturan alert dengan memantau sistem hingga penggunaan memori melebihi 10%.
- iv. Periksa bahwa alert firing terjadi sesuai dengan kondisi yang telah ditentukan.
- v. Pastikan notifikasi diterima melalui sistem yang telah diintegrasikan, seperti Grafana atau bot Telegram.

