

Homework Sesi 11 Bootcamp DevOps Engineer DigitalSkola

Install Node Exporter & Prometheus pada server, carilah memory usage dari server tersebut menggunakan metrik yang kalian dapatkan setelah menginstall node exporter & Prometheus

1. Instalasi Node Exporter

Tujuan Node Exporter: Untuk mengumpulkan data kinerja sistem komputer.

Langkah-langkah Instalasi Node Exporter:

Langkah 1: Membuat Pengguna Node Exporter

```
ubuntu_admin@server2:~$ sudo groupadd node_exporter
ubuntu_admin@server2:~$ sudo useradd -g node_exporter --shell /usr/sbin/nologin node_exporter
```

- **sudo groupadd node_exporter:** Membuat grup sistem bernama node_exporter.
- **sudo useradd -g node_exporter --shell /usr/sbin/nologin node_exporter:** Membuat pengguna sistem bernama node_exporter dan memasukkannya ke grup node_exporter. Opsi --shell /usr/sbin/nologin menonaktifkan login shell untuk pengguna ini.

Langkah 2: Mengunduh Node Exporter

```
ubuntu_admin@server2:~$ wget https://github.com/prometheus/node_exporter/releases/download/v1.7.0/node_exporter-1.7.0.linux-amd64.tar.gz
```

- **wget:** Perintah untuk mengunduh file dari internet.
- **https://github.com/prometheus/node_exporter/releases/download/v1.7.0/node_exporter-1.7.0.linux-amd64.tar.gz:** URL unduhan Node Exporter versi 1.7.0 (per artikel).

Langkah 3: Mengekstrak Node Exporter

```
ubuntu_admin@server2:~$ tar xvf node_exporter-*.linux-amd64.tar.gz
node_exporter-1.7.0.linux-amd64/
node_exporter-1.7.0.linux-amd64/LICENSE
node_exporter-1.7.0.linux-amd64/node_exporter
node_exporter-1.7.0.linux-amd64/NOTICE
```

- **tar xvf node_exporter-*.linux-amd64.tar.gz:** Mengekstrak isi file arsip .tar.gz.

Langkah 4: Memindahkan *Binary* Node Exporter

```
ubuntu_admin@server2:~$ sudo mv node_exporter-1.7.0.linux-amd64/node_exporter /usr/local/bin/
ubuntu_admin@server2:~$ sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter
```

- **sudo mv node_exporter-1.7.0.linux-amd64/node_exporter /usr/local/bin/:** Memindahkan file binary node_exporter ke direktori /usr/local/bin/ agar dapat diakses secara sistem luas.

- **sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter:** Mengubah kepemilikan file node_exporter di /usr/local/bin/ menjadi pengguna dan grup node_exporter

Langkah 5: Membuat *Service File* Node Exporter

```
ubuntu_admin@server2:~$ sudo nano /etc/systemd/system/node_exporter.service
```

Buat file node_exporter.service menggunakan editor nano dan masukkan konten berikut:

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

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target
```

- **[Unit]:** Bagian definisi unit systemd.
 - **Description:** Deskripsi layanan.
 - **Wants=network-online.target, After=network-online.target:** Memastikan layanan dimulai setelah jaringan aktif.
- **[Service]:** Bagian konfigurasi layanan.
 - **User=node_exporter, Group=node_exporter:** Menjalankan layanan sebagai pengguna dan grup node_exporter.
 - **Type=simple:** Jenis layanan sederhana.
 - **ExecStart=/usr/local/bin/node_exporter:** Perintah untuk menjalankan Node Exporter.
- **[Install]:** Bagian instalasi.
 - **WantedBy=multi-user.target:** Layanan akan dimulai saat sistem mencapai target multi-user.target (mode operasi normal)

Langkah 6: Memuat Ulang Systemd dan Memulai Node Exporter

```
ubuntu_admin@server2:~$ sudo systemctl daemon-reload
ubuntu_admin@server2:~$ sudo systemctl daemon-reload
ubuntu_admin@server2:~$ sudo systemctl start node_exporter
ubuntu_admin@server2:~$ sudo systemctl enable node_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
```

```
ubuntu_admin@server2:~$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-03-06 05:09:51 UTC; 27s ago
     Main PID: 9054 (node_exporter)
        Tasks: 5 (limit: 1134)
       Memory: 2.5M
      CGroup: /system.slice/node_exporter.service
              └─9054 /usr/local/bin/node_exporter
```

- **sudo systemctl daemon-reload:** Memuat ulang konfigurasi systemd.
- **sudo systemctl start node_exporter:** Memulai layanan Node Exporter.
- **sudo systemctl enable node_exporter:** Mengaktifkan layanan agar otomatis dimulai saat boot.
- **sudo systemctl status node_exporter:** Memeriksa status layanan Node Exporter. Pastikan statusnya "active (running)".

Langkah 7: Mengakses Metrik Node Exporter

Buka browser dan akses:

http://<IP_SERVER>:9100/metrics

Server yang dipakai disini adalah vm dari gcp. Maka dari itu, harus dibuat suatu firewall rule untuk membuka atau membolehkan port 9100 dari server tersebut untuk diakses. Port 9090 dan port 3000 untuk prometheus dan grafana yang akan disetup nanti juga dibuka.

[← Firewall rule details](#)

allow-monitoring-system

Description

allow node exporter, prometheus, grafana.

Source filters

IP ranges 0.0.0.0/0

Protocols and ports

tcp:3000, 9090, 9100

Link yang berhasil diakses tersebut melihatkan halaman teks berisi metrik.

```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
```

2. Instalasi Prometheus

Tujuan Prometheus: Prometheus adalah sistem monitoring dan time-series database untuk mengumpulkan, menyimpan, dan memproses metrik.

Langkah-langkah Instalasi Prometheus:

Langkah 1: Membuat Pengguna dan Direktori Prometheus

```
ubuntu_admin@server2:~$ sudo groupadd prometheus
ubuntu_admin@server2:~$ sudo useradd -g prometheus --shell /usr/sbin/nologin prometheus
ubuntu_admin@server2:~$ sudo mkdir /etc/prometheus
ubuntu_admin@server2:~$ sudo mkdir /var/lib/prometheus
ubuntu_admin@server2:~$ sudo chown prometheus:prometheus /etc/prometheus
ubuntu_admin@server2:~$ sudo chown prometheus:prometheus /var/lib/prometheus
ubuntu_admin@server2:~$ sudo chmod 775 /etc/prometheus
ubuntu_admin@server2:~$ sudo chmod 775 /var/lib/prometheus
```

- Sama seperti langkah Node Exporter, perintah ini membuat pengguna dan grup prometheus, direktori konfigurasi (/etc/prometheus), direktori data (/var/lib/prometheus), dan mengatur kepemilikan serta izin akses.
- **sudo chmod 775 /etc/prometheus** dan **sudo chmod 775 /var/lib/prometheus:** Memberikan izin baca, tulis, dan eksekusi untuk pemilik dan grup, serta baca dan eksekusi untuk lainnya.

Langkah 2: Mengunduh Prometheus

```
ubuntu_admin@server2:~$ wget https://github.com/prometheus/prometheus/releases/download/v2.47.0/prometheus-2.47.0.linux-amd64.tar.gz
```

wget: Perintah unduh.

<https://github.com/prometheus/prometheus/releases/download/v2.47.0/prometheus-2.47.0.linux-amd64.tar.gz>: URL unduhan Prometheus versi 2.47.0.

Langkah 3: Mengekstrak Prometheus

```
ubuntu_admin@server2:~$ tar xvf prometheus-*.linux-amd64.tar.gz
prometheus-2.47.0.linux-amd64/
prometheus-2.47.0.linux-amd64/console_libraries/
prometheus-2.47.0.linux-amd64/console_libraries/menu.lib
prometheus-2.47.0.linux-amd64/console_libraries/prom.lib
prometheus-2.47.0.linux-amd64/prometheus.yml
prometheus-2.47.0.linux-amd64/promtool
prometheus-2.47.0.linux-amd64/LICENSE
prometheus-2.47.0.linux-amd64/prometheus
prometheus-2.47.0.linux-amd64/NOTICE
prometheus-2.47.0.linux-amd64/consoles/
prometheus-2.47.0.linux-amd64/consoles/prometheus-overview.html
prometheus-2.47.0.linux-amd64/consoles/node.html
prometheus-2.47.0.linux-amd64/consoles/node-cpu.html
prometheus-2.47.0.linux-amd64/consoles/index.html.example
prometheus-2.47.0.linux-amd64/consoles/prometheus.html
prometheus-2.47.0.linux-amd64/consoles/node-overview.html
prometheus-2.47.0.linux-amd64/consoles/node-disk.html
```

- **tar xvf prometheus-*.linux-amd64.tar.gz:** Mengekstrak arsip.

Langkah 4: Menyalin *Binary* dan File Konfigurasi Prometheus

```
ubuntu_admin@server2:~$ cd prometheus-2.47.0.linux-amd64/
ubuntu_admin@server2:~/prometheus-2.47.0.linux-amd64$ ls -l
total 236864
-rw-r--r-- 1 ubuntu_admin ubuntu_admin      11357 Sep  6  2023 LICENSE
-rw-r--r-- 1 ubuntu_admin ubuntu_admin       3773 Sep  6  2023 NOTICE
drwxr-xr-x 2 ubuntu_admin ubuntu_admin       4096 Sep  6  2023 console_librari
es
drwxr-xr-x 2 ubuntu_admin ubuntu_admin       4096 Sep  6  2023 consoles
-rwxr-xr-x 1 ubuntu_admin ubuntu_admin 124167859 Sep  6  2023 prometheus
-rw-r--r-- 1 ubuntu_admin ubuntu_admin        934 Sep  6  2023 prometheus.yml
-rwxr-xr-x 1 ubuntu_admin ubuntu_admin 118339410 Sep  6  2023 promtool

ubuntu_admin@server2:~/prometheus-2.47.0.linux-amd64$ sudo cp prometheus /usr/local/bin/
ubuntu_admin@server2:~/prometheus-2.47.0.linux-amd64$ sudo cp promtool /usr/local/bin/
ubuntu_admin@server2:~/prometheus-2.47.0.linux-amd64$ sudo cp -r consoles/ console_librar
ies/ /etc/prometheus/
ubuntu_admin@server2:~/prometheus-2.47.0.linux-amd64$ sudo cp prometheus.yml /etc/prometh
eus/
ubuntu_admin@server2:~/prometheus-2.47.0.linux-amd64$ sudo chown prometheus:prometheus /u
sr/local/bin/prometheus
ubuntu_admin@server2:~/prometheus-2.47.0.linux-amd64$ sudo chown prometheus:prometheus /u
sr/local/bin/promtool
ubuntu_admin@server2:~/prometheus-2.47.0.linux-amd64$ sudo chown -R prometheus:prometheus
/etc/prometheus/consoles
ubuntu_admin@server2:~/prometheus-2.47.0.linux-amd64$ sudo chown -R prometheus:prometheus
/etc/prometheus/console_libraries
ubuntu_admin@server2:~/prometheus-2.47.0.linux-amd64$ sudo chown prometheus:prometheus /e
tc/prometheus/prometheus.yml
```

- **cd prometheus-2.47.0.linux-amd64/**: Pindah ke direktori hasil ekstraksi.
- **sudo cp prometheus /usr/local/bin/**: Menyalin binary Prometheus (prometheus) ke /usr/local/bin/.
- **sudo cp promtool /usr/local/bin/**: Menyalin command-line tool (promtool) ke /usr/local/bin/.
- **sudo cp -r consoles/ /etc/prometheus/**: Menyalin direktori consoles ke /etc/prometheus/ secara rekursif (-r).
- **sudo cp -r console_libraries/ /etc/prometheus/**: Menyalin direktori console_libraries (untuk web console) ke /etc/prometheus/ secara rekursif (-r).
- **sudo cp prometheus.yml /etc/prometheus/**: Menyalin file konfigurasi utama prometheus.yml ke /etc/prometheus/.
- Baris **sudo chown ...**: Mengubah kepemilikan file dan direktori ke pengguna dan grup prometheus. -R untuk rekursif pada direktori.

Langkah 5: Membuat *Service File* Prometheus

```
ubuntu_admin@server2:~$ sudo nano /etc/systemd/system/prometheus.service
```

Buat file prometheus.service dan masukkan konten berikut:

[Unit]

Description=Prometheus Time Series Collection and Processing Server

```

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

```

- **[Unit], [Install]:** Mirip dengan service file Node Exporter.
- **[Service]:**
 - **ExecStart:** Perintah menjalankan Prometheus Server.
 - **--config.file:** Menentukan lokasi file konfigurasi.
 - **--storage.tsdb.path:** Direktori penyimpanan data time-series database.
 - **--web.console.templates, --web.console.libraries:** Direktori untuk web console.

Langkah 6: Memuat Ulang Systemd dan Memulai Prometheus

```

ubuntu_admin@server2:~$ sudo systemctl daemon-reload
ubuntu_admin@server2:~$ sudo systemctl start prometheus
ubuntu_admin@server2:~$ sudo systemctl enable prometheus
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.
ubuntu_admin@server2:~$ sudo systemctl status prometheus
● prometheus.service - Prometheus Time Series Collection and Processing Server
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-03-06 06:24:11 UTC; 12s ago

```

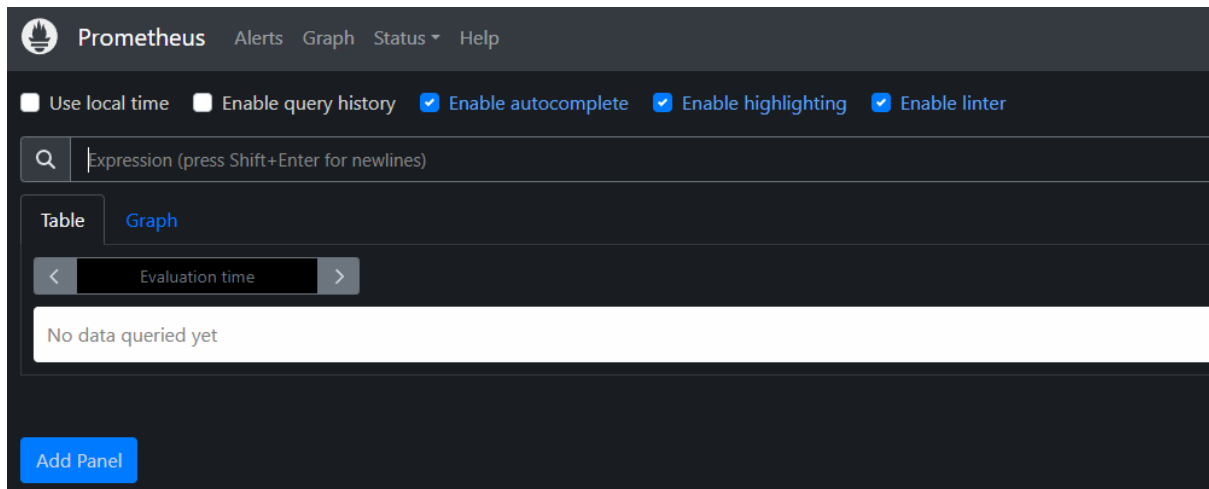
Sama seperti langkah Node Exporter: muat ulang, mulai, aktifkan, dan periksa status layanan Prometheus.

Langkah 7: Mengakses Web UI Prometheus

Buka browser dan akses:

http://<IP_SERVER>:9090

Link yang berhasil diakses tersebut melihatkan Web UI Prometheus.



Langkah 8: Mengkonfigurasi Prometheus untuk *Scrape* Node Exporter

```
ubuntu_admin@server2:~$ sudo nano /etc/prometheus/prometheus.yml
```

Edit file `/etc/prometheus/prometheus.yml` menggunakan editor nano. Cari bagian `scrape_configs` dan tambahkan `job` untuk Node Exporter seperti berikut:

```
global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'node_exporter'
    static_configs:
      - targets: ['localhost:9100']
```

- **scrape_configs:** Daftar konfigurasi scrape.
 - **- job_name:** 'node_exporter': Menambahkan job baru bernama node_exporter.
 - **static_configs:** Konfigurasi target statis.
 - **- targets:** ['localhost:9100']: Target scrape adalah localhost:9100, yaitu alamat Node Exporter. Jika targetnya adalah server lain maka localhost diganti dengan IP Server tersebut.

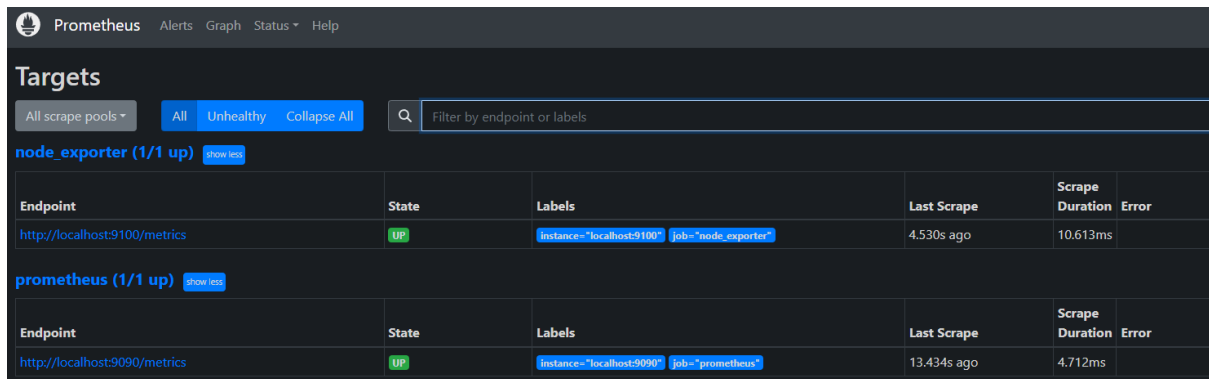
Langkah 9: Memulai Ulang Prometheus

```
ubuntu_admin@server2:~$ sudo systemctl restart prometheus
ubuntu_admin@server2:~$ sudo systemctl status prometheus
● prometheus.service - Prometheus Time Series Collection and Processing Server
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: ena
   Active: active (running) since Thu 2025-03-06 06:36:54 UTC; 10s ago
```


Restart layanan Prometheus agar konfigurasi baru diterapkan. Periksa status untuk memastikan tidak ada kesalahan.

Langkah 10: Verifikasi di *Web UI* Prometheus

Buka *Web UI* Prometheus (http://<IP_SERVER>:9090). Pergi ke "Status" -> "Targets". Pastikan **node_exporter** dan **prometheus** berstatus "Up".



The screenshot shows the Prometheus Targets page. At the top, there's a navigation bar with 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below it, the 'Targets' section is active. There are buttons for 'All scrape pools', 'All', 'Unhealthy', and 'Collapse All'. A search bar is present with the text 'Filter by endpoint or labels'. Two target groups are listed: 'node_exporter (1/1 up)' and 'prometheus (1/1 up)'. Each group has a 'show less' button. Below each group is a table with columns: Endpoint, State, Labels, Last Scrape, Scrape Duration, and Error.

| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|-------------------------------|-------|---|-------------|-----------------|-------|
| http://localhost:9100/metrics | UP | instance="localhost:9100" job="node_exporter" | 4.530s ago | 10.613ms | |
| http://localhost:9090/metrics | UP | instance="localhost:9090" job="prometheus" | 13.434s ago | 4.712ms | |

3. Memory usage dari server

Untuk mencari memory usage dari server menggunakan metrik yang dikumpulkan oleh Node Exporter dan Prometheus perlu penggunaan *query* di Prometheus yaitu *Prometheus Query Language* (PromQL). Node Exporter menyediakan berbagai metrik terkait memori yang bisa dimanfaatkan.

Node Exporter memiliki beberapa metrik yang berkaitan dengan memori, yang umumnya diawali dengan awalan `node_memory_`. Beberapa metrik yang relevan untuk menghitung memory usage adalah:

- **node_memory_MemTotal_bytes**: Total memori fisik (RAM) yang terpasang di server, dalam satuan *bytes*.
- **node_memory_MemFree_bytes**: Jumlah memori yang benar-benar kosong dan tidak digunakan, dalam satuan *bytes*.
- **node_memory_Buffers_bytes**: Memori yang digunakan oleh *buffer* kernel, dalam satuan *bytes*. *Buffer* digunakan untuk menyimpan data mentah dari *disk blocks*.
- **node_memory_Cached_bytes**: Memori yang digunakan untuk *page cache* dan *slab cache*, dalam satuan *bytes*. *Cache* digunakan untuk mempercepat akses ke *file* dan data yang sering digunakan.
- **node_memory_Slab_bytes**: Memori yang digunakan oleh *kernel slab cache*, dalam satuan *bytes*. *Slab cache* digunakan untuk mengalokasikan memori untuk struktur data kernel.

- **node_memory_MemAvailable_bytes**: Estimasi jumlah memori yang tersedia untuk memulai aplikasi baru tanpa melakukan *swapping*, dalam satuan *bytes*. Metrik ini lebih berguna daripada *MemFree* untuk mengukur ketersediaan memori dari perspektif aplikasi.

Formula yang umum digunakan adalah menghitung *used memory* dengan mengurangi *MemAvailable* dari *MemTotal*, lalu menghitung persentasenya dari *MemTotal*.

$$\frac{(\text{node_memory_MemTotal_bytes} - \text{node_memory_MemAvailable_bytes})}{\text{node_memory_MemTotal_bytes}} * 100$$

Penjelasan *Query*:

- **node_memory_MemTotal_bytes**: Mengambil nilai metrik total memori.
- **node_memory_MemAvailable_bytes**: Mengambil nilai metrik memori yang tersedia.
- -: Operator pengurangan. **node_memory_MemTotal_bytes - node_memory_MemAvailable_bytes** menghitung perkiraan memori yang sedang digunakan (*used memory*).
- /: Operator pembagian. Hasil pengurangan (*used memory*) dibagi dengan **node_memory_MemTotal_bytes** (total memori). Ini menghasilkan rasio *used memory* terhadap *total memory*.
- * 100: Hasil rasio dikalikan 100 untuk mendapatkan persentase.

Untuk menjalankan *query* adalah sebagai berikut:

Pada homepage Prometheus di **http://<IP_SERVER>:9090**, temukan kotak input di bagian atas yang berlabel "Expression" > masukkan *query* PromQL tersebut > tekan tombol "Execute" (atau tekan Enter). Prometheus akan menjalankan *query* dan menampilkan hasilnya.

Setelah menjalankan *query*, Prometheus akan menampilkan hasil dalam dua format, yaitu "Graph" dan "Table".

"Graph": Menampilkan visualisasi grafik *time-series* dari *memory usage* server. Secara *default*, grafik akan menampilkan data *memory usage* dalam rentang waktu tertentu (misalnya, 1 jam terakhir). Rentang waktu ini dapat diubah menggunakan kontrol waktu di atas grafik.

"Table": Menampilkan nilai *instant* (nilai terakhir) dari *memory usage* server dalam format tabel. Biasanya, tabel akan menampilkan nilai *memory usage* saat ini untuk setiap target Node Exporter yang di-*scrape* oleh Prometheus. Jika hanya memantau satu server, hanya satu baris data yang akan terlihat.

