Dokumentasi Optimasi Penggunaan RAM Wazuh-Manager pada Penggunaan Skala Kecil

Rozan Gangsar Adibrata 23/521626/TK/57547
Haidar Faruqi Al Ghifari 23/518252/TK/57023

Latar Belakang

Wazuh adalah *open source security platform* yang menyatukan kemampuan *Extended Detection and Response* (XDR) dan *Security Information and Event Management* (SIEM) yang digunakan untuk prevensi ancaman, deteksi, dan respons. Platform ini melindungi beban kerja di lingkungan *on-premise*, virtualisasi, kontainerisasi, dan berbasis *cloud*.

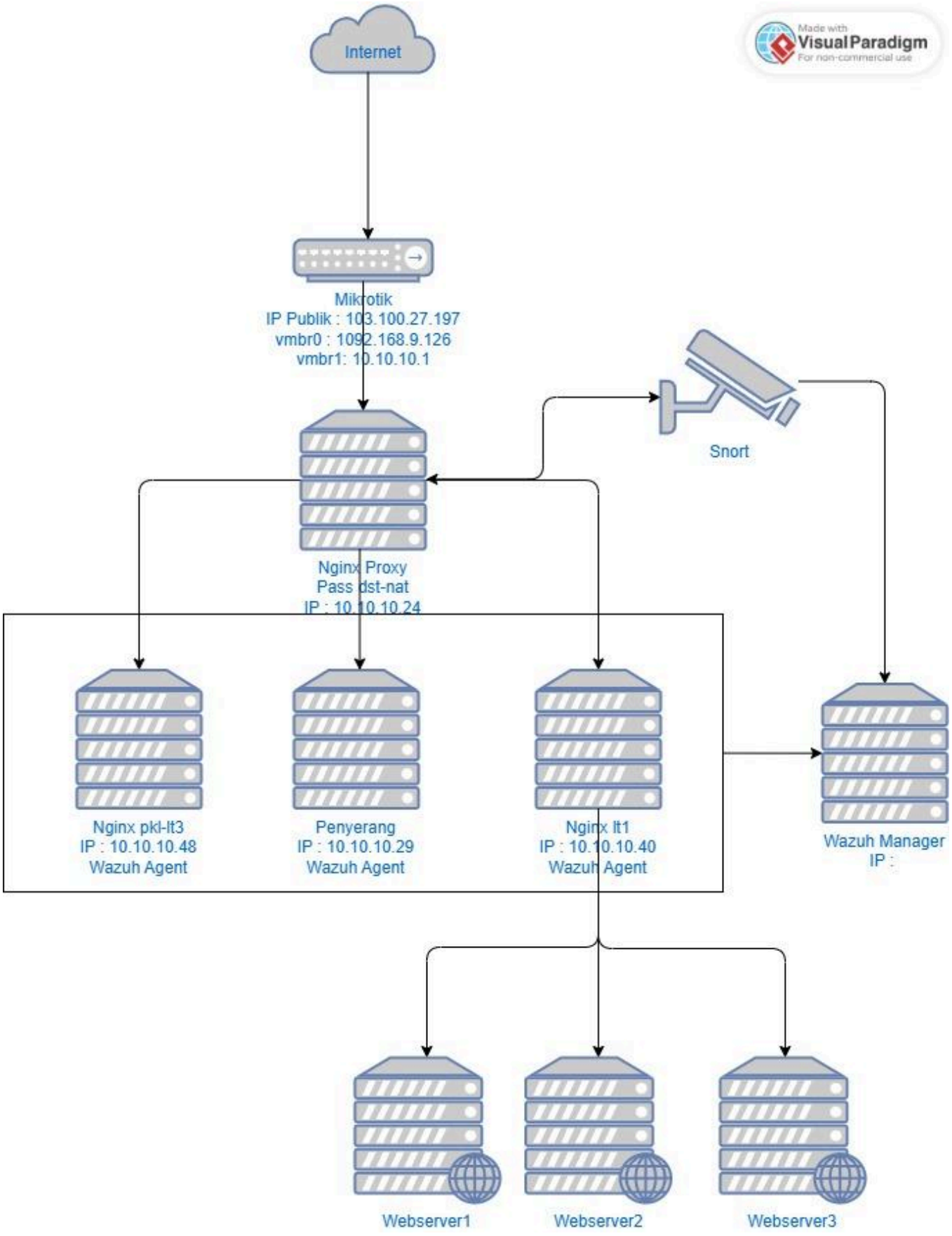
Meski fungsionalitas wazuh sangat berguna, komponen-komponen Wazuh Manager dan Indexer dapat menggunakan sumber daya memori yang signifikan bila dikonfigurasi secara *default*. Pada lingkungan skala kecil atau server/lab dengan RAM terbatas, penggunaan RAM yang tidak dioptimalkan berpotensi menyebabkan penurunan kinerja komputer, latensi pemrosesan log, serta kesulitan dalam bereksperimen dan belajar menggunakan Wazuh.

Tujuan

Dikutip dari https://documentation.wazuh.com/, mormalnya, hanya untuk menjalankan *Wazuh Manager*, diperlukan setidaknya 4GB RAM untuk *Wazuh Indexer,* 2GB RAM untuk *Wazuh Server,* serta 4GB RAM untuk *Wazuh Dashboard*, atau setidaknya 5GB RAM dan direkomendasikan 8GB RAM untuk menjalankan *all-in-one* dalam penggunaan *lab/server* skala kecil (1-25 *Agent)*.

Eksperimen ini bertujuan untuk mencari tahu berapa banyak penggunaan RAM Wazuh Manager dapat dioptimalkan agar lebih ramah bagi *programmer* pemula yang baru saja ingin belajar Wazuh Manager dan membangun *home-lab* dengan *resource* yang kecil.

Rancangan Arsitektur



Konfigurasi Kontrainer untuk PKL

Type	Name	Content	Proxy status	TTL	Actions
A	api	103.100.27.197	Proxied	Auto	Edit
A	db	103.100.27.197	Proxied	Auto	Edit
A	dump	103.100.27.197	Proxied	Auto	Edit
A	globalintermedia.onli...	103.100.27.197	Proxied	Auto	Edit
A	magang	103.100.27.197	Proxied	Auto	Edit
A	pkl	103.100.27.197	Proxied	Auto	Edit
A	www	103.100.27.197	Proxied	Auto	Edit

```
server {
    listen 80;
    server_name db.globalintermedia.online;

    location / {
        proxy_pass http://10.10.10.49;

        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

```
server {
    listen 80;
    server_name api.globalintermedia.online;

    location / {
        proxy_pass http://10.10.10.48/html/api-presensi;
        index index.html index.php;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Implementasi Wazuh Versi Optimasi

1. Instalasi *all-in-one Wazuh Manager*
    a. *Download Wazuh*
    curl -sO https://packages.wazuh.com/4.14/wazuh-install.sh
    sudo bash ./wazuh-install.sh -a

    b. Jalankan service
    sudo systemctl start wazuh-manager
    sudo systemctl start wazuh-indexer
    sudo systemctl start wazuh-dashboard

    c. akses *Wazuh Dashboard website interface*
    https://<ip_address_wazuh>
    username : admin
    password : <admin_password>

2. Implementasi Optimasi *Wazuh Backend* (*analysisd*)
    a. Clone repo wazuh 2.19.4 versi optimasi
    git clone https://github.com/rozangangsar/Kerja-Praktek-Wazuh.git

    b. Instalasi dependensi *build*
    sudo apt update
    sudo apt install -y git cmake make build-essential gcc g++ make automake autoconf libtool curl python3 policycoreutils pkg-config

    c. build source
    cd ~/Kerja-Praktek-Wazuh/src
    make TARGET=server -j"$(nproc)"

d. *Deploy* optimasi *binary* ke *service*

```
sudo systemctl stop wazuh-manager
sudo cp ~/Kerja-Praktek-Wazuh/src/wazuh-analysisd
/var/ossec/bin/wazuh-analysisd
sudo systemctl start wazuh-manager
```

3. Implementasi Optimasi *Wazuh Backend* (*indexer connector*)
   a. *Generate build system*

```
cd ~/Kerja-Praktek-Wazuh
cmake -S src -B src/build -DCMAKE_BUILD_TYPE=Release
```

   b. *build* target

```
cmake --build src/build --target indexer_connector -j2
```

   c. *Deploy* file .so ke *runtime* Wazuh

```
sudo cp src/build/shared_modules/indexer_connector/libindexer_connector.so
/var/ossec/lib/libindexer_connector.so
sudo systemctl restart wazuh-manager
```

4. Implementasi Optimasi *Wazuh Indexer* (*Adaptive Query*)
   a. Clone repo *Wazuh Indexer* 2.19.4 versi optimasi

```
git clone https://github.com/rozangangsar/wazuh-indexer.git
cd wazuh-indexer
git checkout 6977d04291c
```

   b. Instalasi dependensi *build*

```
sudo apt update
sudo apt install -y jq openjdk-21-jdk build-essential
```

   c. *build artifact*

```
./gradlew :server:jar --no-daemon
```

   d. *Deploy* optimasi *binary* ke *service*

```
sudo systemctl stop wazuh-indexer
sudo cp /usr/share/wazuh-indexer/lib/opensearch-2.19.4.jar
/usr/share/wazuh-indexer/lib/opensearch-2.19.4.jar.bak-$(date
+%Y%m%d-%H%M%S)
sudo cp
~/wazuh-indexer/server/build/distributions/opensearch-2.19.4-SNAPSHOT.jar
/usr/share/wazuh-indexer/lib/opensearch-2.19.4.jar
```

   e. Beri *permission* dan *restart wazuh*

```
sudo chown root:wazuh /usr/share/wazuh-indexer/lib/opensearch-2.19.4.jar
sudo chmod 0644 /usr/share/wazuh-indexer/lib/opensearch-2.19.4.jar
sudo rm -f /usr/share/wazuh-indexer/lib/opensearch-2.19.4-SNAPSHOT.jar
sudo systemctl daemon-reload
sudo systemctl restart wazuh-indexer
```

Perbandingan Hasil

a. Sebelum Optimasi

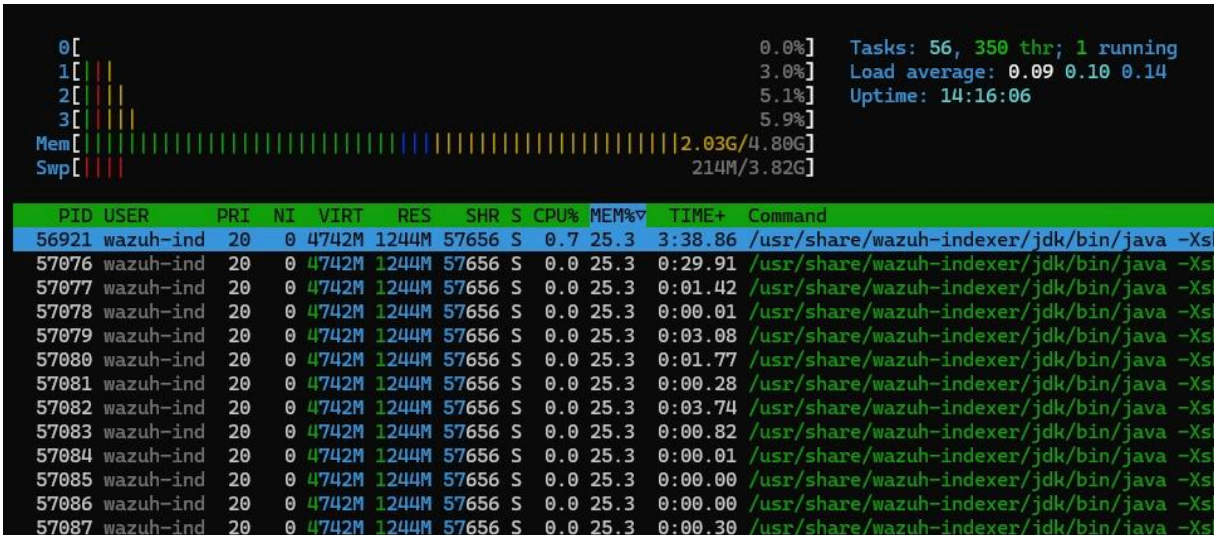
```
wazuh@wazuh-manager:~$ ps -eo pid,cmd,%mem,rss --sort=-rss | head -n 15
    PID CMD                         %MEM   RSS
  73173 /usr/share/wazuh-indexer/jd 30.0 1510364
  73116 /usr/share/wazuh-dashboard/  4.2 213824
  73908 /var/ossec/bin/wazuh-module  2.3 119608
  73538 /var/ossec/framework/python  2.1 110380
  78973 /usr/sbin/netdata -D         1.3 70184
  73539 /var/ossec/framework/python  1.2 65096
  73540 /var/ossec/framework/python  1.2 65088
  73543 /var/ossec/framework/python  1.2 63528
  72995 /usr/share/filebeat/bin/fil  0.9 46156
  73653 /var/ossec/bin/wazuh-analys  0.7 37428
  73602 /var/ossec/bin/wazuh-db      0.7 36940
    670 /usr/sbin/tailscaled --stat  0.6 32332
  73026 /lib/systemd/systemd-journa  0.6 31168
  73178 /usr/lib/snapd/snapd         0.5 29312
wazuh@wazuh-manager:~$
```

```
  0[||                                                    1.9%]   Tasks: 54, 407 thr; 1 running
  1[                                                      0.0%]   Load average: 0.00 0.05 0.07
  2[||||                                                  3.3%]   Uptime: 3 days, 06:37:44
  3[||                                                    2.0%]
Mem[||||||||||||||||||||||||||||||||||||||||||||||||2.33G/4.80G]
Swp[|                                                  58.6M/3.82G]

    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%▽  TIME+  Command
  73173 wazuh-ind  20   0 4954M 1474M 33484 S  0.7 30.0  8:15.95 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
  73467 wazuh-ind  20   0 4954M 1474M 33484 S  0.0 30.0  0:40.88 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
  73468 wazuh-ind  20   0 4954M 1474M 33484 S  0.0 30.0  0:02.25 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
  73469 wazuh-ind  20   0 4954M 1474M 33484 S  0.0 30.0  0:00.00 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
  73470 wazuh-ind  20   0 4954M 1474M 33484 S  0.0 30.0  0:04.83 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
  73471 wazuh-ind  20   0 4954M 1474M 33484 S  0.0 30.0  0:02.07 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
  73472 wazuh-ind  20   0 4954M 1474M 33484 S  0.0 30.0  0:02.15 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
  73473 wazuh-ind  20   0 4954M 1474M 33484 S  0.0 30.0  0:02.22 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
  73483 wazuh-ind  20   0 4954M 1474M 33484 S  0.0 30.0  0:02.67 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
  73484 wazuh-ind  20   0 4954M 1474M 33484 S  0.0 30.0  0:01.09 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
  73485 wazuh-ind  20   0 4954M 1474M 33484 S  0.0 30.0  0:16.18 /usr/share/wazuh-indexer/jdk/bin/java -Xshar
```

b. Setelah Optimasi

```
wazuh@wazuh-manager:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:           4.8Gi       2.0Gi       530Mi       1.0Mi       2.3Gi       2.5Gi
Swap:          3.8Gi       214Mi       3.6Gi
wazuh@wazuh-manager:~$ ps -eo pid,cmd,%mem,rss --sort=-rss | head -n 15
    PID CMD                         %MEM   RSS
  56921 /usr/share/wazuh-indexer/jd 25.3 1274196
    665 /usr/share/wazuh-dashboard/  3.3 168528
   1580 /var/ossec/framework/python  2.2 113004
   1725 /var/ossec/bin/wazuh-analys  2.0 101256
    648 /usr/sbin/netdata -D         1.6 82908
   1581 /var/ossec/framework/python  1.2 61048
   1582 /var/ossec/framework/python  1.2 60948
   1654 /var/ossec/bin/wazuh-db      0.7 36236
    377 /lib/systemd/systemd-journa  0.6 33816
    663 /usr/sbin/tailscaled --stat  0.6 31292
   1585 /var/ossec/framework/python  0.5 28904
    418 /sbin/multipathd -d -s       0.5 27240
    644 /usr/share/filebeat/bin/fil  0.4 22300
    658 /usr/lib/snapd/snapd         0.4 20868
wazuh@wazuh-manager:~$
```



Terlihat bahwa penggunaan RAM *Wazuh-Indexer* sebelum optimasi adalah 30% dari total RAM, yaitu sekitar 1.5GB, sedangkan penggunaan RAM setelah optimasi adalah 25.3% yaitu sekitar 1.265GB. Serta penggunaan total RAM sebelum optimasi berada di 2.33GB sedangkan setelah optimasi adalah 2.03GB. Dengan penurunan penggunaan RAM pada *wazuh-indexer* sebesar 15.67% merupakan hasil yang memuaskan serta membantu *developer* pemula yang ingin mempelajari dan mengimplementasikan *Wazuh Manager*.

Pembahasan

Pada *section* pembahasan ini, bertujuan untuk membahas lebih mendalam mengenai metode/potongan kode yang diubah untuk mengoptimalkan penggunaan RAM oleh *Wazuh Manager*.

1. *Wazuh Backend*
    a. src/analysisd/decoders/syscheck.c
    b. src/shared_modules/indexer_connector/include/indexerConnector.hpp
    c. src/shared_modules/indexer_connector/include/indexerConnector.cpp
2. *Wazuh Indexer*
    a. server/src/main/java/org/opensearch/index/shard/IndexShard.java
    b. server/src/test/java/org/opensearch/index/shard/AdaptiveUsageTrackingQueryCachingPolicyTests.java