TUGAS

MANAJEMEN JARINGAN

OLEH :

NAMA : INDAH SARI

NIM : 09011181320011

JURUSAN SISTEM KOMPUTER

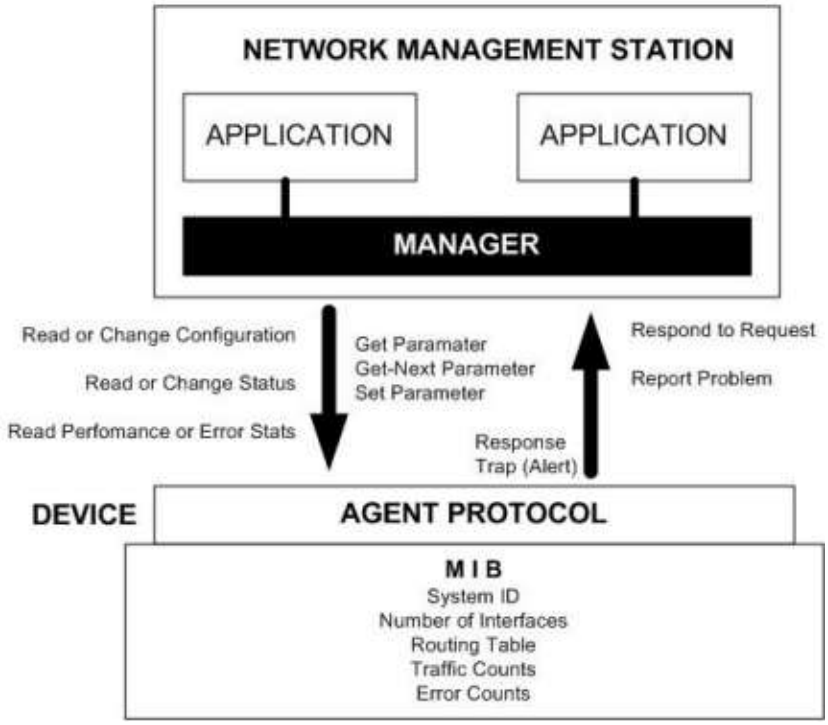
FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2016

Analisa Protokol SNMP dengan Menggunakan Wireshark

Dimana point – point protokol SNMP sesuai dengan gambar interaksi antara manager dan agent dibawah ini:



Gambar interaksi antara manager dan agent

Simple Network Management Protocol (SNMP) adalah, sebuah protocol yang digunakan sebagai standar untuk melakukan pengaturan perangkat-perangkat jaringan. Dengan bantuan tools/ daemon lain dan dapat mengumpulkan dan memanipulasi informasi network dengan mengumpulkan informasi baseline dengan interval waktu tertentu. SNMP dapat digunakan untuk mengonfigurasi device yang jauh, memantau unjuk kerja jaringan, mendeteksi kesalahan jaringan atau akses yang tidak cocok, dan mengaudit pemakaian jaringan. Sedangkan Aplikasi NMS digunakan untuk menjalankan aplikasi yang dapat memonitor dan mengontrol managed device. NMS memberikan resource memory dan prosesor yang dibutuhkan untuk manajemen network. Satu atau lebih NMS harus ada dalam sebuah jaringan yang di manage. Terdapat 3 konsep dasar pada SNMP, yaitu: manager, agent, dan management information based (MIB).

Manager

Sebuah managed device adalah sebuah node di jaringan yang berisi agen SNMP yang berada di jaringan yang dapat di manage. Managed device akan mengumpulkan dan menyimpan informasi manajemen dan membuat informasi ini tersedia bagi NMS menggunakan SNMP. Pada beberapa konfigurasi di titik manager menjalankan suatu software management, dimana perangkat yang dapat dimanage seperti bridges, routers, servers dan workstations yang dapat integrasikan dengan sebuah modul software agent.

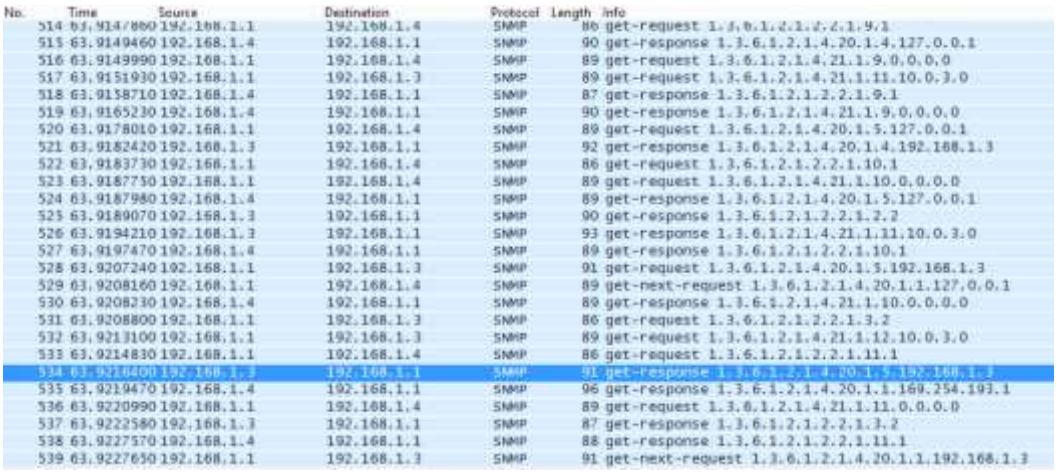
Agent

Agent pertanggung jawab untuk menyediakan akses ke lokal MIB dari object resources dan aktivitas node tersebut. Agen tersebut juga akan bereaksi terhadap perintah manager untuk mendapat kembali nilai-nilai dari MIB dan untuk menetapkan nilai nilai di dalam MIB. Satu contoh dari suatu obyek didapat kembali dari suatu perhitungan dari banyaknya paket-paket pengirim dan penerima pada sebuah node. Manager dapat memonitor nilai yang diload pada jaringan tersebut. Software Agent berada pada di devices tersebut, beberapa agent menerima pesan yang masuk dari manager, pesan permintaan tersebut di baca atau ditulis pada data device tersebut. Agent akan mengirimkan kembali respon yang diterima, dimana agent tidak harus menunggu untuk bertanya tentang sebuah informasi. Namun pada beberapa kasus tertentu agent akan mengirimkan sebuah pesan notifikasi untuk melakukan trap ke satu atau lebih manager. Software Management pada sebuah station management akan mengirimkan pesan request ke agent dan menerima respon dan trap dari agent. Protocol UDP yang biasa digunakan sebagai pembawa paket tersebut dengan karakteristiknya yang hemat dengan bandwidth, namun protocol pembawa lainnya juga dapat digunakan.

MIB (Management Information Base)

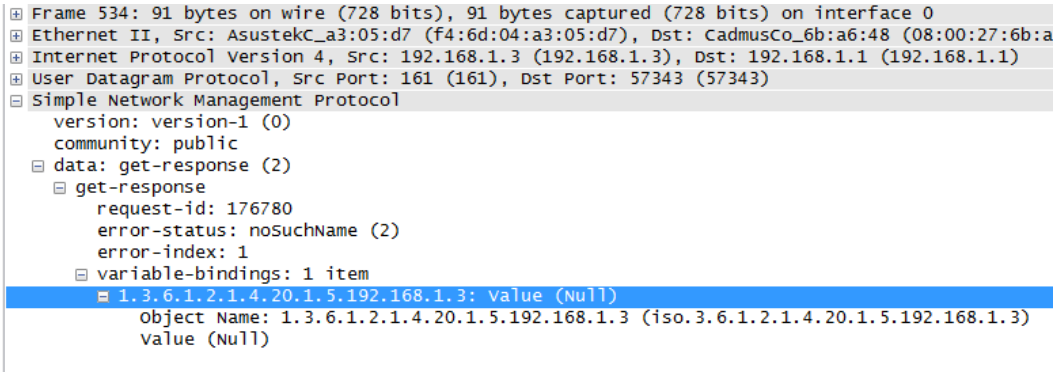
Management Information Base, merupakan struktur basis data variabel dari elemen jaringan yang dikelola. Struktur ini bersifat hierarki dan memiliki aturan sedemikian rupa sehingga informasi setiap variabel dapat dikelola atau ditetapkan dengan mudah. MIB di akses menggunakan protokol network-manajemen seperti SNMP. MIB terdiri dari managed objek dan di identifikasi oleh object identifier (pengidentifikasi objek). Sebuah managed object, kadang kala di sebut sebagai MIB object, objek, atau MIB, adalah satu dari banyak karakteristik spesifik dari peralatan yang di manage.

Wireshark PCAP SNMP yang Dianalisa:



No.	Time	Source	Destination	Protocol	Length	Info
514	63.914780	192.168.1.1	192.168.1.4	SNMP	86	get-request 1.3.6.1.2.1.2.2.1.9.1
515	63.9149460	192.168.1.4	192.168.1.1	SNMP	90	get-response 1.3.6.1.2.1.4.20.1.4.127.0.0.1
516	63.9149990	192.168.1.1	192.168.1.4	SNMP	89	get-request 1.3.6.1.2.1.4.21.1.9.0.0.0.0
517	63.9151930	192.168.1.1	192.168.1.3	SNMP	89	get-request 1.3.6.1.2.1.4.21.1.11.10.0.3.0
518	63.9138710	192.168.1.4	192.168.1.1	SNMP	87	get-response 1.3.6.1.2.1.2.2.1.9.1
519	63.9165230	192.168.1.4	192.168.1.1	SNMP	90	get-response 1.3.6.1.2.1.4.21.1.9.0.0.0.0
520	63.9178010	192.168.1.1	192.168.1.4	SNMP	89	get-request 1.3.6.1.2.1.4.20.1.5.127.0.0.1
521	63.9182420	192.168.1.3	192.168.1.1	SNMP	92	get-response 1.3.6.1.2.1.4.20.1.4.192.168.1.3
522	63.9183730	192.168.1.1	192.168.1.4	SNMP	86	get-request 1.3.6.1.2.1.2.2.1.10.1
523	63.9187750	192.168.1.1	192.168.1.4	SNMP	89	get-request 1.3.6.1.2.1.4.21.1.10.0.0.0.0
524	63.9187980	192.168.1.4	192.168.1.1	SNMP	89	get-response 1.3.6.1.2.1.4.20.1.5.127.0.0.1
525	63.9189070	192.168.1.3	192.168.1.1	SNMP	90	get-response 1.3.6.1.2.1.2.2.1.2.2
526	63.9194210	192.168.1.3	192.168.1.1	SNMP	93	get-response 1.3.6.1.2.1.4.21.1.11.10.0.3.0
527	63.9197470	192.168.1.4	192.168.1.1	SNMP	89	get-response 1.3.6.1.2.1.2.2.1.10.1
528	63.9207240	192.168.1.1	192.168.1.3	SNMP	91	get-request 1.3.6.1.2.1.4.20.1.5.192.168.1.3
529	63.9208160	192.168.1.1	192.168.1.4	SNMP	89	get-next-request 1.3.6.1.2.1.4.20.1.1.127.0.0.1
530	63.9208230	192.168.1.4	192.168.1.1	SNMP	89	get-response 1.3.6.1.2.1.4.21.1.10.0.0.0.0
531	63.9208800	192.168.1.1	192.168.1.3	SNMP	86	get-request 1.3.6.1.2.1.2.2.1.3.2
532	63.9213100	192.168.1.1	192.168.1.3	SNMP	89	get-request 1.3.6.1.2.1.4.21.1.12.10.0.3.0
533	63.9214830	192.168.1.1	192.168.1.4	SNMP	86	get-request 1.3.6.1.2.1.2.2.1.11.1
534	63.9216400	192.168.1.3	192.168.1.1	SNMP	91	get-response 1.3.6.1.2.1.4.20.1.5.192.168.1.3
535	63.9219470	192.168.1.4	192.168.1.1	SNMP	96	get-response 1.3.6.1.2.1.4.20.1.1.169.254.193.1
536	63.9220990	192.168.1.1	192.168.1.4	SNMP	89	get-request 1.3.6.1.2.1.4.21.1.11.0.0.0.0
537	63.9222580	192.168.1.3	192.168.1.1	SNMP	87	get-response 1.3.6.1.2.1.2.2.1.3.2
538	63.9227570	192.168.1.4	192.168.1.1	SNMP	88	get-response 1.3.6.1.2.1.2.2.1.11.1
539	63.9227650	192.168.1.1	192.168.1.3	SNMP	91	get-next-request 1.3.6.1.2.1.4.20.1.1.192.168.1.3

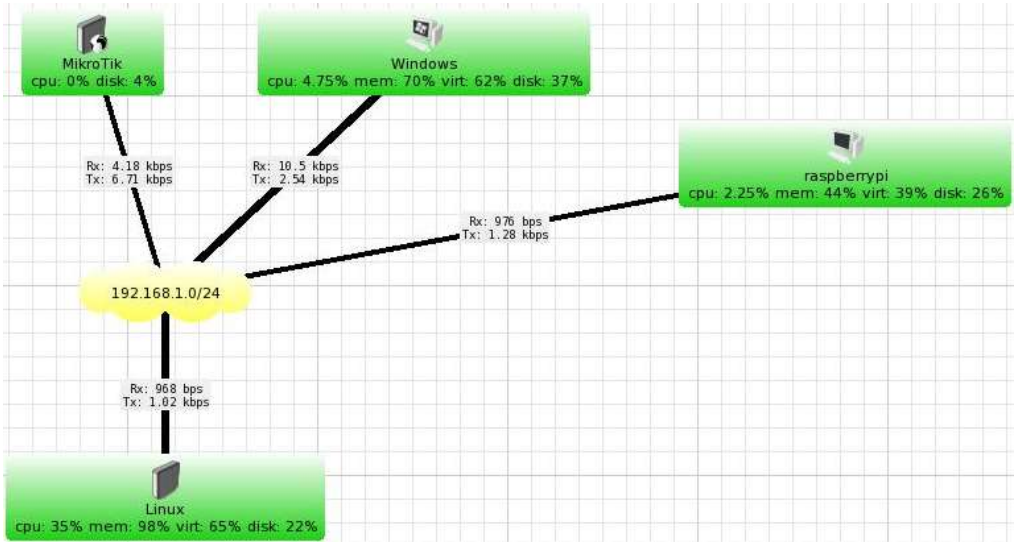
Pada pcap diatas menggunakan protocol SNMP dimana timer: 63.9216400, IP Source 192.168.1.3 dan IP Destinationnya 192.168.1.1



```
⊞ Frame 534: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface 0
⊞ Ethernet II, Src: AsustekC_a3:05:d7 (f4:6d:04:a3:05:d7), Dst: CadmusCo_6b:a6:48 (08:00:27:6b:a
⊞ Internet Protocol Version 4, Src: 192.168.1.3 (192.168.1.3), Dst: 192.168.1.1 (192.168.1.1)
⊞ User Datagram Protocol, Src Port: 161 (161), Dst Port: 57343 (57343)
⊟ Simple Network Management Protocol
    version: version-1 (0)
    community: public
  ⊟ data: get-response (2)
    ⊟ get-response
        request-id: 176780
        error-status: noSuchName (2)
        error-index: 1
      ⊟ variable-bindings: 1 item
        ⊟ 1.3.6.1.2.1.4.20.1.5.192.168.1.3: value (Null)
            Object Name: 1.3.6.1.2.1.4.20.1.5.192.168.1.3 (iso.3.6.1.2.1.4.20.1.5.192.168.1.3)
            Value (Null)
```

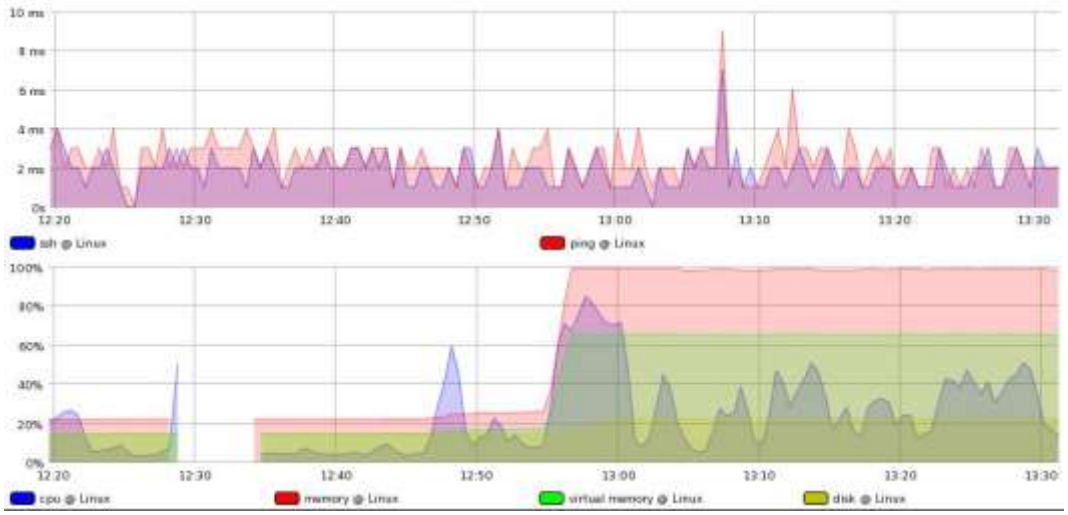
Pada simple network terdapat data get response, dimana didalamnya terdapat request – id : 176780, error – status: noSuchName, error – index nya: 1, dan variabel – bindingsnya memiliki 1 item yaitu: 1.3.6.1.2.1.4.20.1.5.192.168.1.3 : value (null)

Topologi SNMP Antara Linux, Mikrotik, Windows, dan Raspberry Pi

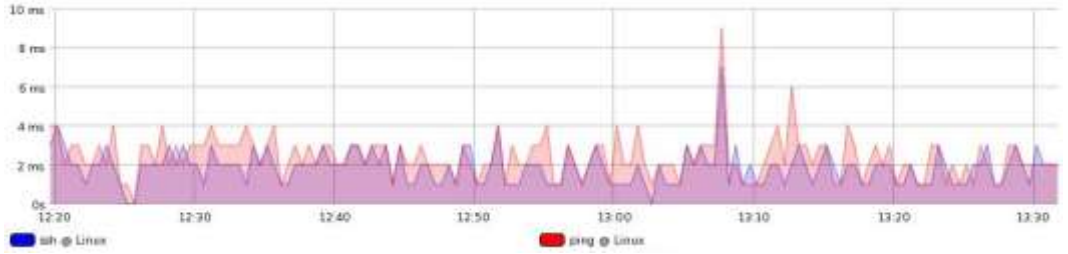


Pada gambar topologi diatas dapat kita lihat antara Linux, Mikrotik, Windows8, dan Raspberry py terhubung dengan menggunakan satu IP address yaitu 192.168.1.0/24 dimana **Linux** memiliki cpu (35%), mem (98%), virtual (65%), disk (22%), kecepatan menerima data (Rx): 968 bps, dan kecepatan mengirim data (Tx): 1.02 Kbps. **Mikrotik** memiliki cpu (0%), disk (4%), kecepatan menerima data (Rx): 4.18 Kbps, dan kecepatan mengirim data (Tx): 6.71 Kbps. **Windows8** memiliki cpu (4.75%), mem (70%), virtual (62%), disk (37%), kecepatan menerima data (Rx): 10.5 Kbps, dan kecepatan mengirim data (Tx): 2.54 Kbps. Sedangkan pada **Raspberry pi** memiliki cpu (2.25%), mem (44%), virtual (39%), disk (26%), kecepatan menerima data (Rx): 9.76 bps, dan kecepatan mengirim data (Tx): 1.28 Kbps. Dari gambar topologi diatas dapat dilihat urutan tercepat dalam menerima data (Rx): Windows8 (10.5 Kbps), Mikrotik (4.18 Kbps), Raspberry py (9.76 bps), dan Linux (968 bps). Sedangkan dalam mengirim data (Tx) urutan tercepat: Mikrotik (6.71 Kbps), Windows8 (2.54 Kbps), Raspberry py (1.28 Kbps), dan Linux (1.02 Kbps).

Gambar Trafik Pada Linux



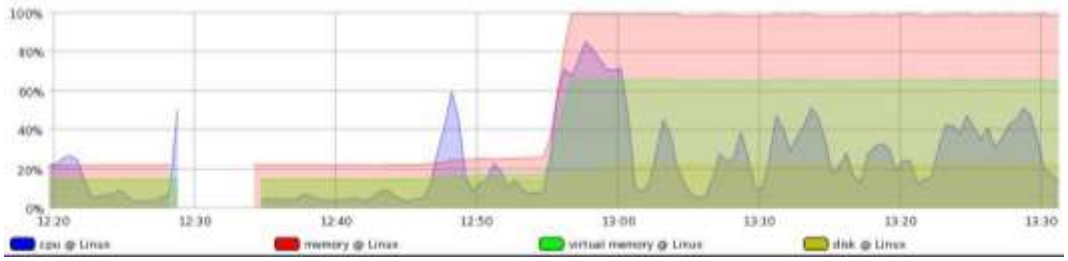
Trafik 1 pada linux



Pada gambar trafik1 Linux diatas dapat dilihat warna biru untuk statistik ssh dan merah untuk statistik ping.

- Pada ssh dari jam 12.20 s/d 13.00 kecepatan statistiknya dari 0s sampai 4ms, pada jam 13.00 s/d 13.10 kecepatan statistiknya naik dari 0s sampai ±6ms, sedangkan pada jam 13.10 s/d 13.30 kecepatan statistiknya dari 0s sampai <4ms.
- Pada ping di linux dari jam 12.20 s/d 13.00 kecepatan statistiknya dari 0s sampai 4ms, pada jam 13.00 s/d 13.10 kecepatan statistiknya naik dari 0s sampai ±8ms, pada jam 13.10 s/d >13.20 kecepatan statistiknya dari 0s sampai 6ms, sedangkan pada jam 13.20 s/d 13.30 kecepatan statistiknya dari 0s sampai >4ms

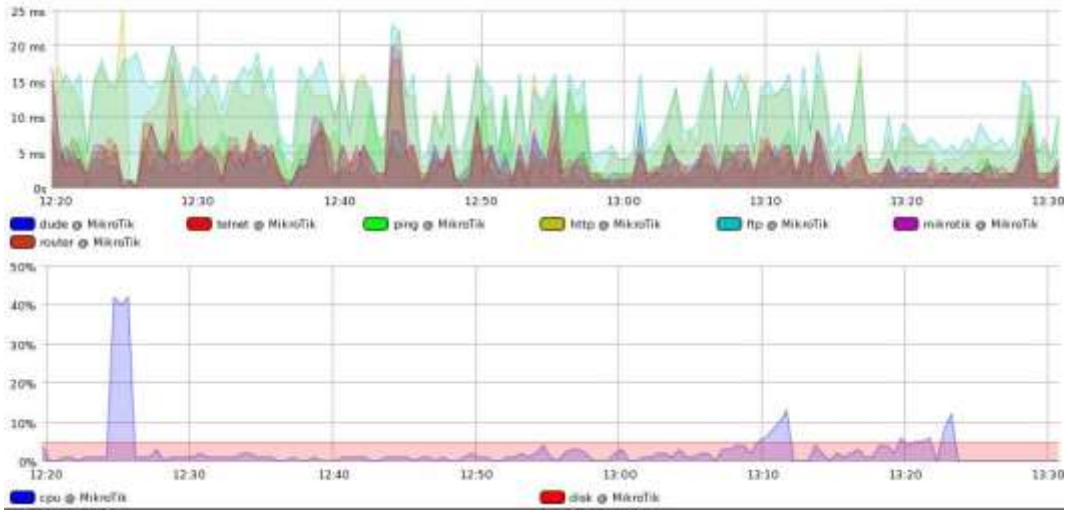
Trafik 2 pada linux



Pada gambar trafik2 Linux diatas dapat dilihat warna biru untuk statistik cpu, merah untuk statistik memory, hijau untuk statistik virtual memory, dan kuning untuk statistik disk

- Pada cpu dari jam 12.20 s/d 12.40 statistiknya dari 0% sampai ±40%, pada jam 12.40 s/d 12.50 statistiknya dari 0% sampai >60%, pada jam 12.50 s/d 13.00 statistiknya dari 0% sampai ±80%, sedangkan pada jam 13.00 s/d 13.30 statistiknya dari 0% sampai ±40%
- Pada memory dari jam 12.20 s/d 12.50 statistiknya dari 0% sampai ±20%, sedangkan pada jam kurang dari 13.00 s/d 13.30 statistiknya naik dari 0% sampai 100%
- Pada virtual memory dari jam 12.20 s/d 12.50 statistiknya dari 0% sampai <20%, sedangkan pada jam kurang dari 13.00 s/d 13.30 statistiknya naik dari 0% sampai ±60%
- Pada disk dari jam 12.20 s/d 12.50 statistiknya dari 0% sampai <20%, sedangkan pada jam 13.00 s/d 13.30 statistiknya naik dari 0% sampai ±20%

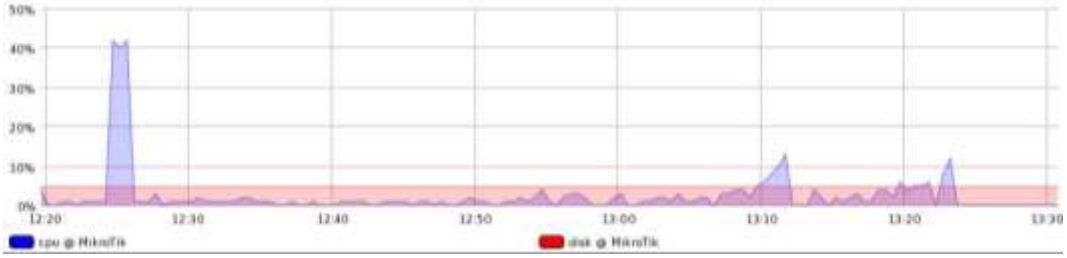
Gambar Trafik Pada Mikrotik



Trafik 1 pada Mikrotik

Pada gambar trafik1 Mikrotik diatas dapat dilihat warna biru untuk statistik dude, merah untuk statistik telnet, hijau untuk statistik ping, kuning untuk statistik http, biru toska untuk statistik ftp, ungu untuk statistik mikrotik, dan orange untuk statistik router. Statistik trafik 1 pada mikrotik dapat dilihat pada gambar diatas.

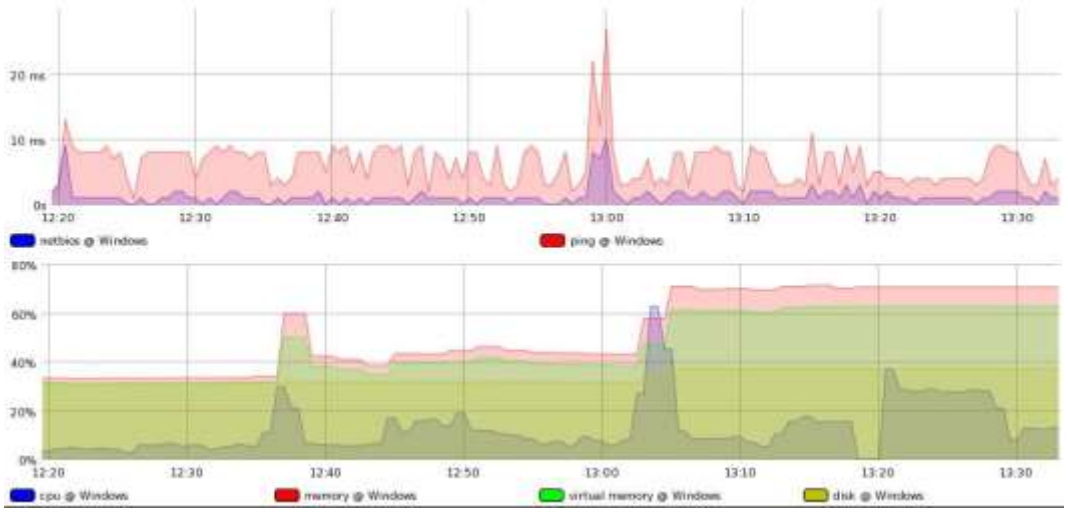
Trafik 2 pada Mikrotik



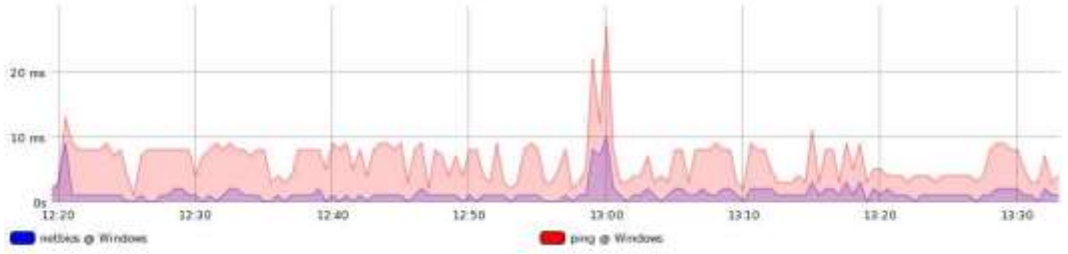
Pada gambar trafik2 Mikrotik diatas dapat dilihat warna biru untuk statistik cpu, dan merah untuk disk

- Pada cpu diantara jam 12.20 dan sebelum jam12.30 statistiknya naik dari 0% sampai ±40%, pada jam 12.30 s/d 13.10 statistiknya dari 0% sampai ±2%, setelah jam 13.10 statistiknya naik dari 0% sampai ±10%, sebelum jam 13.20 statistinya menurun kembali dan pada jam 13.20 lebih statistiknya dari 0% sampai ±10% lagi, sedangkan pada jam 13.30 statistiknya 0%
- Pada disk dari jam 12.20 s/d 13.30 statistiknya netral tidak naik dan tidak menurun dari 0% sampai <10%

Gambar Trafik Pada Windows8



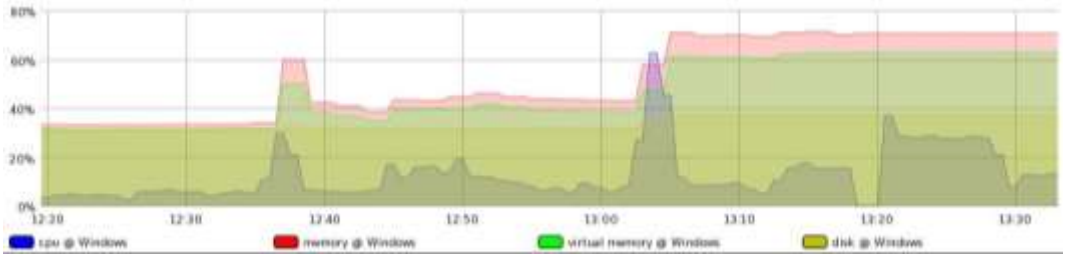
Trafik 1 pada windows8



Pada gambar trafik1 windows8 diatas dapat dilihat warna biru untuk statistik netbios dan merah untuk statistik ping.

- Pada netbios windows8 dari jam 12.20 kecepatan statistiknya dari 0s sampai <10ms, pada jam >12.20 s/d >12.50 kecepatan statistiknya turun dari 0s sampai ±2ms, pada jam 13.00 kecepatan statistiknya naik dari 0s sampai 10ms, sedangkan pada jam 13.10 s/d 13.30 kecepatan statistiknya turun dari 0s sampai ±2ms.
- Pada ping windows8 dari jam 12.20 kecepatan statistiknya dari 0s sampai ±10ms, pada jam >12.20 s/d >12.50 kecepatan statistiknya turun dari 0s sampai <10ms, pada jam 13.00 kecepatan statistiknya naik dari 0s sampai ±20ms, sedangkan pada jam 13.10 s/d 13.30 kecepatan statistiknya turun dari 0s sampai ±10ms.

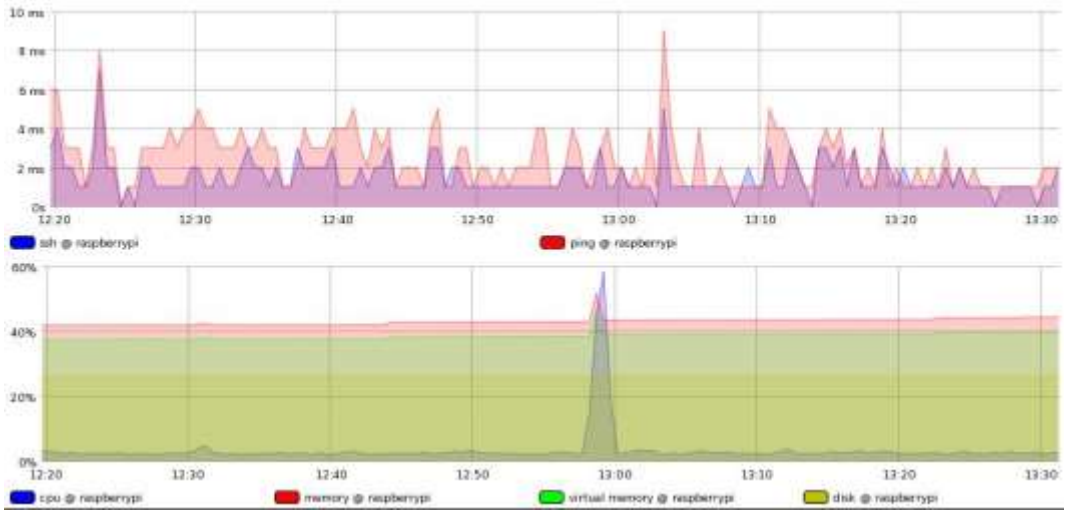
Trafik 2 pada windows8



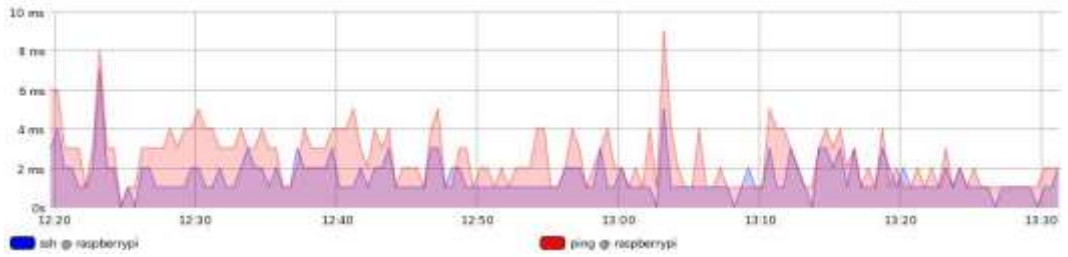
Pada gambar trafik2 windows8 diatas dapat dilihat warna biru untuk statistik cpu, merah untuk statistik memory, hijau untuk statistik virtual memory, dan kuning untuk statistik disk

- Pada cpu dari jam 12.20 s/d 12.30 statistiknya dari 0% sampai <20%, sebelum jam 12.40 statistiknya naik dari 0% sampai ±20%, pada jam 12.40 s/d 13.00 statistiknya dari 0% sampai <20%, sedangkan diantara jam 13.00 s/d 13.10 statistiknya naik dari 0% sampai ±60%
- Pada memory dari jam 12.20 s/d 12.30 statistiknya dari 0% sampai ±20%, sebelum jam 12.40 statistiknya naik dari 0% sampai 60%, pada jam 12.40 s/d 13.00 statistiknya dari 0% sampai ±40%, sedangkan sebelum jam 13.10 s/d 13.30 statistiknya naik dari 0% sampai ±60%,
- Pada virtual memory dari jam 12.20 s/d 12.30 statistiknya dari 0% sampai ±20%, sebelum jam 12.40 statistiknya naik dari 0% sampai ±40%, pada jam 12.40 s/d 13.00 statistiknya dari 0% sampai ±40%, sedangkan setelah jam 13.00 s/d 13.30 statistiknya naik dari 0% sampai ±60%,
- Pada disk dari jam 12.20 s/d 13.00 statistiknya dari 0% sampai ±20%, sedangkan setelah pada jam 13.00 s/d 13.30 statistiknya naik dari 0% sampai hampir mendekati 40%

Gambar Trafik Pada Raspberry Pi



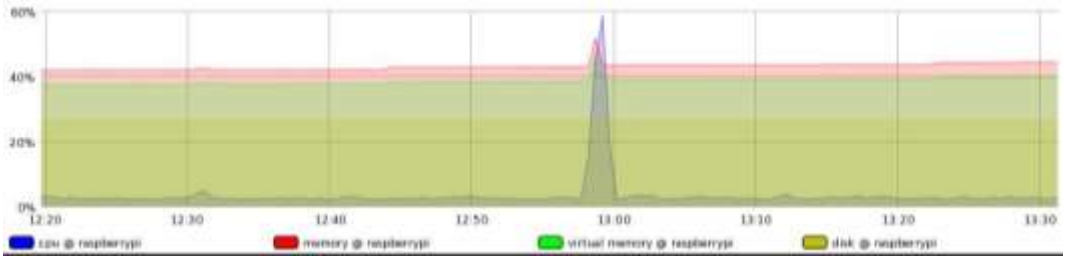
Trafik1 pada Raspberry pi



Pada gambar trafik1 Raspberry pi diatas dapat dilihat warna biru untuk statistik ssh dan merah untuk statistik ping.

- Pada ssh ±12.50 kecepatan statistiknya naik dari 0% sampai ±6ms, pada jam 12.30 s/d 13.00 kecepatan statistiknya dari 0% sampai ±2ms, diantara jam 13.00 dan 13.10 kecepatan statistiknya naik dari 0% sampai ±4ms, sedangkan pada jam 13.10 s/d 13.30 kecepatan statistiknya dari 0% sampai ±2ms
- Pada ping saat jam 12.20 kecepatan statistiknya naik dari 0% sampai ±6ms, dan pada jam 1ebih dari 12.20 sebelum pada jam 12.30 kecepatan statistiknya naik lagi dari 0% sampai 8ms, pada jam 12.30 s/d 13.00 kecepatan statistiknya turun dari 0% sampai ±4ms, diantara jam 13.00 s/d 13.10 kecepatan statistiknya naik lagi dari 0% sampai ±8ms, sedangkan pada jam 13.10 s/d 13.30 kecepatan statistiknya turun lagi dari 0% sampai ±4ms hingga kurang dari 2ms

Trafik2 pada Raspberry pi



Pada gambar trafik2 Raspberry pi diatas dapat dilihat warna biru untuk statistik cpu, merah untuk statistik memory, hijau untuk statistik virtual memory, dan kuning untuk statistik disk

- Pada cpu dari jam 12.20 s/d $\pm$12.50 statistiknya dari 0% sampai kurang dari 20%, pada jam $\leq$13.00 statistiknya naik dari 0% sampai $<$60%, sedangkan pada jam 13.00 s/d 13.30 statistiknya turun dari 0% sampai kurang dari 20
- Pada memory dari jam 12.20 s/d $\pm$12.50 statistiknya dari 0% sampai 40%, jam $\leq$13.00 statistiknya naik dari 0% sampai $\pm$40%, sedangkan pada jam 13.00 s/d 13.30 statistiknya turun dari 0% sampai kurang dari $\pm$40%,
- Pada virtual memory dari jam 12.20 s/d 13.30 statistiknya netral tidak naik dan tidak turun dari 0% sampai $<$40%
- Pada disk dari jam 12.20 s/d 13.30 statistiknya netral tidak naik dan tidak turun dari 0% sampai $\pm$20%

Kesimpulan:

Simple Network Management Protocol (SNMP) adalah, sebuah protocol yang digunakan sebagai standar untuk melakukan pengaturan perangkat-perangkat jaringan. Dengan bantuan tools/ daemon lain dan dapat mengumpulkan dan memanipulasi informasi network dengan mengumpulkan informasi baseline dengan interval waktu tertentu. SNMP dapat digunakan untuk mengonfigurasi device yang jauh, memantau unjuk kerja jaringan, mendeteksi kesalahan jaringan atau akses yang tidak cocok, dan mengaudit pemakaian jaringan. Sedangkan Aplikasi NMS digunakan untuk menjalankan aplikasi yang dapat memonitor dan mengontrol managed device. NMS memberikan resource memory dan prosesor yang dibutuhkan untuk manajemen network. Satu atau lebih NMS harus ada dalam sebuah jaringan yang di manage. Terdapat 3 konsep dasar pada SNMP, yaitu: manager, agent, dan management information based (MIB).

Pada pcap diatas menggunakan protocol SNMP dimana timer: 63.9216400, IP Source 192.168.1.3 dan IP Destinationnya 192.168.1.1 Pada simple network terdapat data get response, dimana didalamnya terdapat request – id : 176780, error – status: noSuchName, error – index nya: 1, dan variabel – bindingsnya memiliki 1

item yaitu: 1.3.6.1.2.1.4.20.1.5.192.168.1.3 : value (null). Dari gambar interaksi antara manager dan agent sebelumnya menggunakan SNM di mana didalamnya memiliki dua aplikasi, pada tugas ini menggunakan aplikasi wireshark yang diatur dan dimonitor oleh manager, dimana manager dan agent protokol saling merespon mengirim dan menerima data dengan menggunakan IP Source dan IP Destination yang digunakan tadi.