PEMANTAUAN DAN ANALISIS PERFORMA SISTEM *HONEYPOT* DENGAN *SIMPLE NETWORK MANAGEMENT PROTOCOL* (SNMP)

Imron Kadafi Hariri, Alif Subardono
Departemen Teknik Elektro dan Informatika
Universitas Gadjah Mada
imron.kadafi.h@mail.ugm.ac.id, alif@ugm.ac.id

Abstract – Installation of the honepot sensor on Raspberry Pi device integrated with Modern Honey Network (MHN) is a DSSDI UGM action to see the threats of existing public Network, so it can be preventive action. But the installed Honeypot sensors sometimes have problems in the form of overload on the System so that data attacks are not recorded. It is necessary to develop a monitoring System on the Honeypot System to monitor and manage System resources and to analyze Honeypot data needs. System monitoring which is one of the Network Management System (NMS) functions can be implemented with Simple Network Management Protocol (SNMP) to monitor the performance of Honeypot sensor devices. From monitoring the Honeypot System shows the Honeypot sensor device overloaded in memory. In addition, from the monitoring is seen the influence of the number of attacks on System performance.

Keywords - Honeypot, Modern Honey Network (MHN), Network Management System (NMS), Simple Network Management Protocol (SNMP)

Intisari – Pemasangan sensor honepot pada perangkat Raspberry Pi yang terintegrasi dengan Modern Honey *Network* (MHN) merupakan tindakan DSSDI UGM guna melihat ancaman-ancaman yang ada jaringan publik, sehingga dapat dilakukan tindakan pencegahan. Namun sensor *Honeypot* yang terpasang kadang mengalami masalah berupa overload pada sistem sehingga data serangan tidak tercatat. Perlu dikembangakan sistem pemantauan pada sistem *Honeypot* tersebut untuk memantau dan mengelola sumber daya sistem juga untuk kebutuhan analisis data *Honeypot*. Pemantauan sistem yang merupakan salah satu fungsi *Network Management System* (NMS) dapat dierapkan dengan *Simple Network Management Protocol* (SNMP) untuk memantau performa dari perangkat sensor *Honeypot*. Dari pemantauan sistem *Honeypot* menunjukkan perangkat sensor *Honeypot* mengalami overload pada memori. Selain itu dari pemantauan tersebut terlihat adanya pengaruh jumlah serangan terhadap performa sistem.

Kata Kunci - *Honeypot*, Modern Honey *Network* (MHN), *Network Management System* (NMS), *Simple Network Management Protocol* (SNMP)

I. PENDAHULUAN

Serangan terhadap sistem informasi dan jaringan saat ini semakin banyak dan bermacam-macam seiring dengan jumlah penggunaan teknologi komputer yang terus meningkat. Adanya ancaman-ancaman yang ada pada sistem informasi membuat Direktorat Sistem dan Sumber Daya Informasi Universitas Gadjah Mada (DSSDI UGM) sebagai penyedia sistem dan layanan informasi di lingkungan UGM harus tetap menjaga sistem jaringan di UGM dari ancaman tersebut. Untuk menjaga kinerja sistem yang berjalan agar bekerja secara semestinya, diperlukan tindakan pencegahan, pemeliharaan dan penanganan terhadap segala ancaman yang ada pada sistem informasi.

DSSDI UGM telah memasang sensor Honepot yang terintegrasi dengan *Modern Honey Network* (MHN) guna melihat ancaman-ancaman yang ada jaringan publik, sehingga dengan data yang diperoleh dari *Honeypot* administrator jaringan dapat melakukan tindakan pencegahan. *Honeypot* merupakan sebuah sistem yang menyerupai sistem asli di mana pemasangan sistem palsu tersebut bertujuan untuk menjebak pengguna yang bertujuan buruk [1]. Kemudian MHN digunakan untuk mengumpulkan data yang telah tercatat pada tiap sensor *Honeypot* yang sudah di integrasikan.

Sensor *Honeypot* yang digunakan oleh DSSDI UGM dipasang pada perangkat Raspberry Pi yang merupakan mini PC. Karena sumber daya yang dimiliki oleh Raspberry Pi ini relatif kecil mengakibatkan perangkat mudah mengalami *overload* sehingga data serangan yang masuk ke sistem *Honeypot* tidak tercatat oleh sensor. Selain masalah pada perangkat sensor *Honeypot*, *server* MHN juga bisa mengalami masalah pada penerimaan data yang tidak tersimpan pada basis data MHN yang disebabkan *storage* pada *server* MHN penuh.

Adanya masalah yang berkaitan dengan sumber daya pada sistem perlu dilakukan manajemen dan pemantauan sumber daya yang digunakan untuk menjaga sistem tetap berjalan sebagaimana mestinya. *Network Management System* (NMS) yang bermanfaat untuk memantau dan mengelola jaringan bisa menjadi solusi untuk masalah tersebut. Ada banyak cara untuk menerapkan NMS, salah satunya adalah dengan menggunakan protokol *Simple Network Management Protocol* (SNMP). Protokol tersebut bisa memenuhi kebutuhan pemantauan sistem. Selain berguna dalam pemantauan dan manajemen sistem, NMS dengan SNMP juga bisa digunakan untuk kebutuhan analisis data performa sistem. Penerapan NMS dengan SNMP ini akan sangat bermanfaat bagi administrator jaringan karena dapat membantu dan mempermudah dalam mengelola sistem jaringan.

II. TEORI PENDUKUNG

2.1. Keamanan Jaringan

Keamanan jaringan adalah segala aktivitas yang ditujukan untuk melindungi fungsi dan integritas data dan jaringan. Hal tersebut mencakup segala teknologi perangkat keras dan perangkat lunak yang terpasang. Pengelolaan akses ke jaringan dengan menargetkan berbagai ancaman dan mencegah serangan masuk atau menyebar pada jaringan merupakan tujuan dari keamanan jaringan [2].

Keamanan jaringan pada suatu jaringan dapat diuji dengan metode *Information Systems Security Assessment Framework* (ISSAF). Metode tersebut memiliki struktur yang jelah dan intuitif sehingga dapat digunakan untuk menguji keamanan sistem jaringan secara optimal. Dari hasil pengujian keamanan sistem jaringan administrator dapat mengetahui celah keamanan yang ada dan kemudian

melakukan tindakan antisipasi untuk menghadapi ancaman yang ada [3].

Open Source Security Information Management (OSSIM) merupakan sistem yang menggabungkan *tool* keamanan dalam satu paket. OSSIM yang dipasang pada jaringan digunakan untuk mengolah dan menganalisa data *traffic* atau aktivitas yang berbahaya pada jaringan. Dari hasil pengolahan data tersebut, didapat informasi yang bisa membantu admin dalam mengamankan jaringan [4].

2.2. *Honeypot*

Honeypot adalah suatu sistem palsu atau layanan palsu yang sengaja dibentuk untuk menjebak pengguna yang mempunyai tujuan buruk atau mendeteksi adanya usahausaha yang dapat merugikan sistem atau layanan. Biasanya *Honeypot* terdiri dari komputer, aplikasi, dan data yang menyerupai perilaku sistem nyata yang tampak menjadi bagian dari jaringan tetapi sebenarnya terisolasi dan terpantau [1].

Honeypot dipasang dengan tujuan mencatat setiap serangan yang masuk ke dalam sistem *Honeypot*. Kegiatan tersebut pasti menghasilkan *file log*, *file log* tersebut pasti akan semakin bertambah banyak seiring dengan semakin banyaknya serangan yang masuk dan akan memenuhi storage dari sistem *Honeypot*. Apabila log sampai memenuhi *resource* maka sistem tidak bisa berjalan sebagaimana mestinya. *File log* yang berukuran besar juga sulit untuk dianalisis dan memakan banyak waktu. Untuk mengatasi hal tersebut diperlukan pengelolaan file log yang efektif dan efisien [5].

2.2.1. Low Interaction *Honeypot*

Honeypot ini adalah *Honeypot* yang dibuat menyerupai sistem atau layanan pada *server* nyata, biasanaya hanya meyerupai layanan tertentu. Peretas atau penyerang hanya bisa memeriksa satu atau beberapa bagian saja pada jaringan tersebut dan tidak dapat berinteraksi langsung dengan sistem operasi yang digunakan, namun dengan interaksi secara tidak lansung maka informasi yang didapat cukup terbatas. Sistem ini bersifat seperti IDS yang hanya mendeteksi serangan masuk. Contoh dari jenis *Honeypot* ini misalnya Dionaea, Honeyd, Kippo, Glastopf, Snort dan lain-lain.

2.2.1.1. Dionaea

Dionaea merupakan salah satu *Honeypot* yang sering digunakan dan termasuk dalam tipe low interaction *Honeypot*, *Honeypot* ini bertujuan untuk mendeteksi serngan berupa malware yang disusupkan oleh penyerang.

Dionaea memiliki log yang mencatat aktivitas serangan pada jaringan seperti informasi asal alamat serangan, port tujuan yang diserang protokol yang layanan yang diserang. Selain itu Dionaea juga akan menyimpan berkas malware yang disusupkan oleh penyerang. Berkas malware yang didapat dapat dianalisa tool lain untuk mengetahui jenis dan tujuan malware tersebut [6]. Dionaea juga dapat digunakan untuk mendeteksi port scanning pada suatu jaringan, dari pola port scanning yang diperoleh administrator dapat memberikan tindakan keamanan lebih efisien [7].

2.2.1.2. Kippo

Honeypot yang khusus digunakan untuk mendeteksi serangan pada protokol SSH. Biasanya Kippo digunakan untuk mendeteksi adanya brute force pada suatu sistem. Tidak hanya mencatat serangan, Kippo juga dapat mempelajari pola serangan dengan bantuan tool lain. *Honeypot* ini dapat diterapkan menggunakan perangkat berspesifikasi rendah seperti Raspberry Pi. Kippo akan mencatat alamat IP sumber serangan dan juga mencatat user dan password yang digunakan untuk percobaan akses ke layanan SSH Kippo [8].

2.2.1.3 Glastopf

Ancaman yang ada pada layanan web dapat dideteksi dengan memanfaatkan Glastopf. Glastopf adalah *Honeypot* dengan tinkat interaksi rendah [9]. Implementasi Glastopf yang dikombinasikan dengan HIHAT dapat dimanfaatkan untuk mengetahui tujuan dan parameter pada HTTP *request*. Halaman web dan sistem informasi palsu yang dimiliki Glastopf dan HIHAT dugunakan untuk menjebak penyerang dan melihat *request* yang dikirim ke sistem palsu tersebut oleh penyerang [10].

2.2.2. High Interactions *Honeypot*

Honeypot yang dibuat agar penyerang berinteraksi langsung dengan sistem operasi atau layanan serta tidak ada batasan dalam interaksi tersebut. Karena interaksi langsung antara penyerang dengan sistem menyebabkan *Honeypot* ini memiliki risiko yang sangat tinggi, namun diisamping risiko tersebut bisa didapat banyak informasi tentang serangan yang masuk. Dari risiko yang ada pada penggunaan *Honeypot* ini maka perlu perlakuan ekstra dalam pengelolaanya. Hal tersebut bisa diatasi dengan subuah jail, sanbox atau VMware box karena dengan s*oft*ware ini akan mengisolasi *Honeypot* tersebut. Contoh dari *Honeypot* ini adalah Honeynet.

2.3. Modern Honey *Network*

Modern Honey *Network* (MHN) merupakan *server* terpusat untuk manajemen dan pengumpulan data *Honeypot*. MHN mempermudah dan mempercepat proses pemasangan sensor *Honeypot* karena didalam MHN sendiri sudah terdapat sekrip untuk pemasangan sensor *Honeypot* seperti Snort, Dionaea, Glastopf, Kippo dan lain-lain [11].

MHN diintegrasikan dengan beberapa sensor *Honeypot* seperti Dionaea, Kippo, Snort, Glastopf dan lain sebagainya. Sensor *Honeypot* dipasang pada perangakat RasberryPi yang terhubung dengan jaringa publik. Data serangan yang didapat dari sensor kemudian ditampilkan pada *interface* MHN. Dari pemasangan MHN tersebut telah didapat data serangan seperti asal serangan, tujuan serangan, port tujuan serangan, user dan password yang digunakan untuk menyerang serta jumlah serangan yang dilakukan [12]

2.4. Raspberry Pi

Raspberry Pi merupakan mini PC seukuran kartu kredit yang dapat langsung di hubungkan dengan monitor dan perangkat keyboard dan mouse. Selain ukurannya yang kecil, harga dari perangkat ini juga lebih murah dibanding dengan komputer pada umumnya. Sudah banyak proyek digital yang menggunakan perangkat ini. Salah satu seri yang banyak digunakan Raspberry Pi 3 Model B. Seri tersebut memiliki sepesifikasi CPU Quad Core 1.2 GHz, 1

GB RAM, 1 port FastEthernet, port HDMI dan beberapa fitur lain [13].

Perangkat Raspberry Pi dapat dimanfaatkan untuk penggunaan komputasi kecil sampai sedang. Seperti penggunaan Raspberry Pi sebagai IDS pada sebuah jaringan. Perangkat Raspberry Pi digunakan untuk membandingkan performa anatara Snort dan Suricata. Dari pengujian yang dilakukan menujukkan bahwa perangkat Raspberry Pi model B+ dapat menangani lebih dari 10000 rules [14].

2.5. *Network Management System* (NMS)

Network Management System (NMS) merupakan aplikasi atau sistem yang memungkinkan administrator jaringan mengelola komponen independen jaringan di dalam kerangka kerja manajemen jaringan yang lebih besar. NMS dapat digunaakan untuk memantau baik perangkat lunak maupun komponen perangkat keras yang ada dalam suatu jaringan. Biasanya digunakan untuk mencatat data dari jaringan yang dipantau kemudian data tersebut diteruskan ke sistem administrator [15].

Manfaat utama dari NMS adalah memungkinkan pengguna untuk memantau atau mengelola seluruh sistem operasi jaringan menggunakan perangkat komputer yang terpusat. NMS sangat berguna untuk kebutuhan deteksi perangkat, pemantauan, analisis performa, manajemen dan notifikasi pada jaringan.

International Organizationn for Standardization (ISO) mendefinisikan bahwa terdapat lima area fungsional dalam manajemen jaringan. Lima area tersebut manajement tersebut adalah sebagai berikut [16] :

1. *Fault Management* – manajemen untuk deteksi, isolasi, peringatan dan perbaikan kesalahan yang ada pada jaringan.
2. *Configuration Management* – manajemen pada aspek konfigurasi perangkat jaringan seperti manajemen file konfigurasi, manajemen inventaris, dan manajemen perangkat lunak.
3. *Performance Management* — Pantau dan ukur berbagai aspek kinerja sehingga keseluruhan kinerja dapat dipertahankan pada tingkat yang dapat diterima.
4. *Security Management* — Menyediakan akses ke perangkat jaringan dan sumber daya perusahaan untuk individu yang berwenang.
5. *Accounting Management* — Informasi penggunaan sumber daya jaringan.

Monitoring pada sistem jaringan sangatlah penting untuk dilakukan karena pentingnya menjaga kualitas dan ketersediaan jaringan. Salah satu cara melakukan hal tersebut adalah dengan monitoring pada jaringan adalah dengan menggunakan protokol *Simple Network Management Protocol* (SNMP), seperti penelitian yang berjudul "Implementasi dan Analisis Sistem Monitoring Performance Jaringan dengan Parameter *Quality Of Service* (QoS)". Pada penelitian tersebut dilakukan pemantauan pada jaringan untuk melihat kualitas performa jaringan berdasarkan parameter QoS, pengambilan data besar *troughput* jaringan, *jitter*, *latency* dan packet loss digunakan untuk menentukan kualitas dari QoS jaringan [17]. Yuli Sholikatin juga melakukan penelitian tentang pemantaua jaringan dengan menggunakan SNMP guna kebutuhan *Fault Management*, dari penelitian tersebut didapat kesimpulan bahwa NMS dengan SNMP dapat menampilkan informasi penggunaan resource perangakat yang dipantau dan dapat memberi pemberitauan tentang kejadian error pada sistem [18].

2.6. *Simple Network Management System* (SNMP)

Simple Network Management Protocol (SNMP) merupakan sebuah protokol yang dugunakan untuk mengkoleksi dan mengatur informasi pada perangkat jaringan yang dikelola. SNMP sering digunakan pada manajemen jaringan untuk memantau jaringan. SNMP dapat digunakan pada jaringan yang besar dan kompleks. Secara manual dan individual pencatatan dan pemantauan banyak perangkat akan memakan waktu yang cukup lama, namun dengan menggunakan SNMP administrator jaringan dapat mengelola dan memantau sermua node jaringan dari satu antarmuka [19].

Pada SNMP terdapat tiga komponen pokok yang diperlukan yaitu sebagai berikut (Wilkins, 2011):

1. SNMP *Agent* - program ini berjalan pada node yang dipantau, mengumpulkan data tentang berbagai metrik seperti penggunaan *bandwidth* atau *memory*. Agen akan mengirimkan data pada SNMP manager ketika diminta. Agen juga dapat mengirim peringatan secara proaktif ke manajer ketika terjadi kesalahan.
2. SNMP Manager / *Network Management* Stations (NMS) – SNMP manager akan meminta kepada agen untuk mengirim informasi dari node melalui SNMP secara berkala. NMS akan mengumpulkan dan mengontrol data dari agen secara terpusat.
3. *Management* Information Base (MIB) – merupakan basis data yang berbentuk file teks (.mib) yang merinci dan menjelaskan semua objek yang digunakan oleh perangkat tertentu yang dapat diminta atau dikendalikan menggunakan SNMP. Basis data ini harus dimuat ke NMS sehingga dapat mengidentifikasi dan memantau status dari node pada jaringan. Setiap item MIB diberi pengidentifikasi objek (OID).

SNMP berjalan pada lapisan aplikasi. Semua pesan SNMP dikirim melalui UDP. SNMP *agent* akan menerima *request* pada port 161 UDP. NMS bisa mengirim *request* ke agen melalui port tersebut, kemudian agen akan merespon dengan mengirim informasi sesuai dengan permintaan NMS. NMS juga bisa menerima notifikasi (*Traps* dan *InformRequest*) dari *agent* melalui port 162.

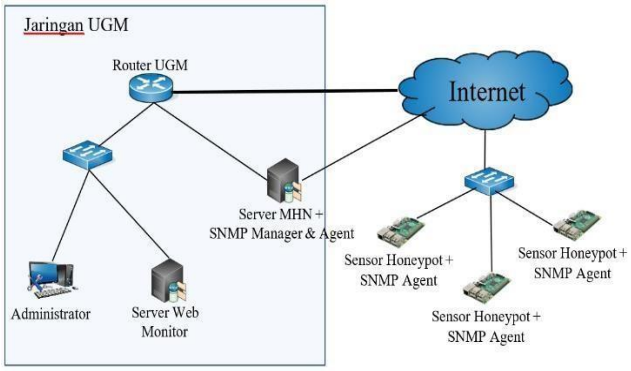
III. METODOLOGI DAN PERCOBAAN

Pada penelitian ini dilakukan dengan beberapa tahap. Tahap pertama pada penelitian ini adalah perancangan topologi dari sistem pemantauan. Topologi yang digunakan menyesuaikan dengan sistem *Honeypot* yang telah terpasang di DSSDI UGM. Selanjutnya adalah instalasi protokol SNMP yang akan digunakan sebagai protokol pemantau sistem. Setelah SNMP sudah dipasang dan diuji, dependensi yang diperlukan seperti web *server*, basis data dan Python konfigurasi pada *server* pemantauan. Kemudian merancang dan mengbangun sistem pemantauan pada *server* yang sudah dipersiapkan sebelumnya.

3.1. Perancangan Topologi

Sistem pemantauan pada penelitian ini dilakukan pada jairngan UGM untuk manajemen sistemnya, kemudian node yang dipantau berada pada jaringan publik seperti Gambar 1. Node yang dipantau adalah perangkat Raspberry Pi yang sudah terpasang sensor *Honeypot*. Dari sistem *Honeypot*

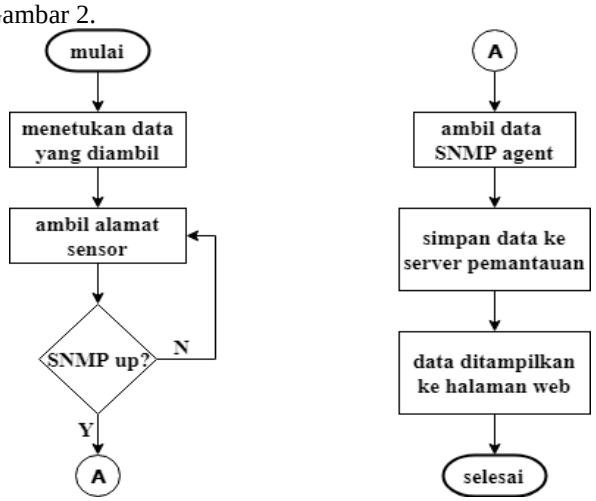
yang sudah ada di DSSDI UGM terdapat dua buah perngkat sensor *Honeypot*, masing-masing perangkat tersebut memiliki alamat IP publik xxx.xxx.92.50 dan xxx.xxx.92.54 sehingga sensor tersebut dapat menerima serangan dari jaringan publik. Dua perangkat sensor tersebut tepasang lebih dari satu *Honeypot*, *Honeypot* yang terpasang pada perangkat tersebut diantaranya Dionaea, Kippo, Glastopf pada sensor "madu-DSSDI-1-UGM" dan Dionaea, Kippo pada sensor "madu-DSSDI-2-UGM".



Gambar 1. Topologi sistem pemantauan

Server MHN yang telah terpasang memiliki dua antarmuka jaringan dimana satu antarmuka dengan alamat IP publik dugunakan untuk komunikasi antara MHN dengan sensor *Honeypot* dan satu antarmuka lainya dugunkan untuk manajemen *server* MHN yang berada pada jaringan UGM dengaan alamat IP xxx.xxx.245.52. Pada penelitian ini menggunakan tambahan *server* yang digunakan untuk *server* pemantau dikarenakan load pada *server* MHN sangat besar. *Server* pemantau tersebut digunakan untuk menampilkan data pemantauan performa sistem *Honeypot* dimana berada pada jaringan UGM dengan alamat IP xxx.xxx.109.68.

3.2. Pembuatan Sistem Pemantauan Informasi dari performa dari sensor *Honeypot* dikumpulkan *server* MHN yang terpasang yang menjadi *SNMP manager*. Manajer akan meminta informasi dari sensor *Honeypot* tiap 5 menit. Informasi-informasi yang telah dikumpulkan oleh *server* MHN akan disimpan ke dalam basis data, sebelum masuk ke basis data, informasi tersebut perlu disusun dan disesuaikan dengan kebutuhan agar dalam pemanfaatannya lebih mudah dan efisien. Alur dari sistem pemantauan dapat dilihat pada Gambar 2.



Gambar 2. Diagram alir sistem pemantauan

Server MHN sebagai manajer akan meminta informasi ke agen. Manajer menentukan OID atau MIB yang akan dimintakan datanya ke agen. Agen akan mengirimkan data sesuai dengan MIB atau OID yang diminta.

Daftar alamat sensor diambil dari database MHN. Semua sensor atau *host* yang ada pada daftar tesebut akan diambil informasi performanya dengan SNMP secara satu-persatu. Sebelum data diambil, tiap *host* akan diperiksa terlebih dahulu apakah host tersebut aktif atau tidak. Apabila *host* tidak aktif maka akan melakukan pemeriksaan ke *host* berikutnya dan apabila host tersebut aktif maka manajer akan meminta informasi performa *host*. Setelah data dari tiap *host* diperolah, data tersebut akan disimpan ke dalam basis data MongoDB.

Data yang telah dikumpulkan oleh manajer dan tersimpan akan diolah dan kemudian ditampilkan pada halaman *web* pemantauan. Dari pemantauan perangkat sensor *Honeypot* tersebut, data yang diperoleh akan dapat digunakan untuk untuk melakukan manajemen *resource*. Data pemantauan yang dikombinasikan dengan data serangan *Honeypot* dapt dianalisis secara lebih lanjut untuk mengetahui pengaruh jumlah serangan terhadap performa sistem *Honeypot*.

IV. ANALISIS DAN PEMBAHASAN

Hasil dari pemantauan sistem *Honeypot* menunjukkan sumberdaya yang dimiliki perangkat sensor *Honeypot* dan penggunaannya. Selain itu analisis data dapat dilakukan dengan informasi yang diperoleh dari hasil pemantauan tersebut.

4.1. Hasil Pemantauan Sistem *Honeypot*

Informasi tentang *resource* perangkat dapat dilihat pada sistem pemantauan ini. Pada sistem pemantauan ini akan memperlihatkan keadaan dan penggunaan *resource* dari tiap sensor *Honeypot* yang terpasang. Dari haril pemantauna yang telah dilakukan menunjukkan bahwa terdapat 9 sensor terintegrasi dengan MHN di DSSDI UGM yang terlihat pada halaman web pemantauan seperti pada Gambar 3. Dalam tampilan tersebut terlihat beberapa sensor yang dengan tampilan berwarna hijau yang berarti sensor tersebut aktif, sedangkan sensor yang berwarna merah menandakan bahwa sensor tersebut tidak aktif.

Gambar 3. Indikatr sensor

Tombol yang ada pada pemantauan tersebut memperlihatkan terdapat 2 buah sensor *Honeypot* aktif yang ditunjukkan dengan tombol yang berwarna hijau dan sedangkan lainnya tidak aktif yang ditunjukkan dengan tombol berwarna merah. Dari 2 buah sensor yang aktif masing-masing adalah sensor "madu-DSSDI-1-UGM" yang berada pada alamat IP xxx.xxx.92.50 dan sensor "maduDSSDI-2-UGM" yang berada pada alamat IP xxx.xxx.92.54. Pada Gambar 3 juga memunjukkan sensor-sensor yang tidak aktif, salah satunya "sesnsorFK-1-UGM" yang berada pada alamat IP xxx.xxx.93.214.

Informasi dari penggunaan memori, *storage* dan beban CPU dapat teramati secara detail. Pada pemantauan ini informasi *resource* tiap sensor dapat terpantau, dari hasil pemantauan yang telah diamati didapat informasi yang dipaparkan pada Tabel 1 dan Tabel 2. Tabel tersebut menunjukkan informasi penggunaan resource dari

perangankat sensor *Honeypot* "madu-DSSDI-1-UGM" dan "madu-DSSDI-2-UGM" yang diamati pada tanggal 26 Juli 2018 pukul 11.45.

Tabel 1. Pemantauan resource "madu-DSSDI-1-UGM"

Pengamatan		Pemakaian	Kapasitas	Persentase (%)
CPU (%)		1	100	1,0
Memori (MB)	Fisik	914,91	927,16	98,7
	Virtual	964,13	1027,15	93,9
	Buffer	70,71	927,16	7,6
	Cache	135,06	135,06	100,0
	Swap	4392,21	100	4392,2
	/	0	7458,86	0,0
	/Dev	2,37	114,85	2,1
	/Boot	0,75	6,99	10,7
Strorage (MB)		17568,8	29835,44	58,9

Tabel 2. Pemantauan resource "madu-DSSDI-2-UGM"

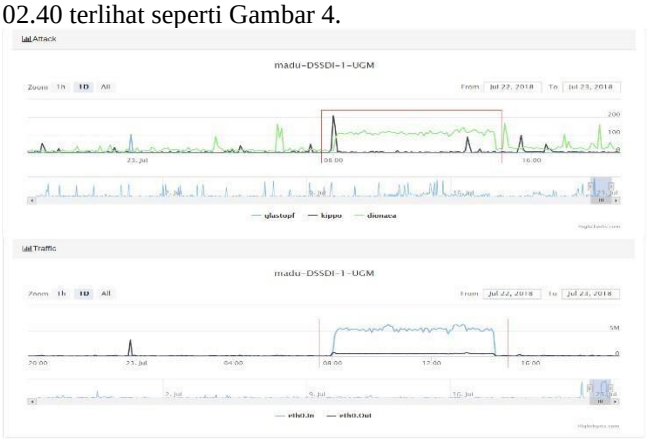
Pengamatan		Pemakaian	Kapasitas	Persentase (%)
CPU (%)		1	100	1,0
Memori (MB)	Fisik	901,91	927,16	97,3
	Virtual	910,47	1027,15	88,6
	Buffer	66,74	927,16	7,2
	Cache	101,05	101,05	100,0
	Swap	2340,41	100	2340,4
	/	0	7458,86	0,0
	/Dev	2,37	114,85	2,1
	/Boot	0,75	6,99	10,7
Strorage (MB)		9361,65	29835,44	31,4

Berdasarkan informasi pada Tabel 1 yang menampilkan informasi perangkat sensor "madu-DSSDI-1-UGM" tersebut, menunjukkan penggunaan memori fisik dan virtual pada perangkat sangat tinggi yang mencapai 98.7% dan 93.9%. Beban CPU pada perangkat sensor tidak begitu besar hanya mencapai 1%. Pada penggunaan *storage* masih tersisa banyak ruang, *storage* sudah terisi sebesar 58,9%.

Informasi resource pada perangkat sensor "madu-DSSDI2-UGM" yang ditampilkan pada Tabel 2 telihat penggunaan resource yang hampir sama dengan sensor "madu-DSSDI-1UGM". Hal tesebut ditunjukkan dengan nilai dari beban CPU sebesar 1%, memori fisik 97,3%. Penggunaan *resource* yang mencapai batas maksimal dapat mengakibatkan sistem berjalan lambat dan bahkan dapat menyebabkan kegagalan layanan. Apabila penggunaan *resource* mencapai batas maksimal secara terus menerus, perlu dilakukan pengelolaan *resource*.

4.2. Analisis Pengaruh Jumlah Serangan pada *Traffic*

Pengamatan pengaruh jumlah serangan terhadap *traffic* antarmuka jaringan sensor *Honeypot* dilakukan dengan melihat grafik jumlah serangan dan grafik *traffic*. Adanya peningkatan atau penurunan pada grafik diamati pada kedua grafik tersebut dengan melihat pada waktu yang sama. Grafik yang yang diamati pada tanggal 24 Juli 2018 pukul 02.40 terlihat seperti Gambar 4.



Gambar 4. Perbandingan grafik serangan dan *traffic* 24 juli 02.40

Pada Gambar 4 yang merupakan tampilan grafik jumlah serangan dan *traffic* dari sensor *Honeypot* "madu-DSSDI1UGM". Grafik tersebut menunjukan adanya kenaikan jumlah serangan yang signifikan dan kenaikan terjadi juga pada *traffic* antarmuka jaringan. Kenaikan jumlah serangan yang signifikan terjadi pada pukul 15.05 tanggal 23 Juli 2018. Kenaikan jumlah serangan tersebut juga diikuti kenaikan pada *traffic* sensor *Honeypot*. Kenaikan jumlah serangan tersebut bermula pada pukul 15.00 dimana terdapat jumlah serangan pada sensor Dionaea 11 serangan, Kippo 208 serangan dan Glastopf 0 serangan, dari serangan tersebut diikuti dengan kenaikan pada *traffic* antarmuka dengan nilai *traffic* masuk 635.068 Byte dan *traffic* keluar sebesar 286.438. detail dari kenaikan jumlah serangan dan *traffic* dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 3. Peningkatan jumlah serangan

Waktu	Dionaea	Kippo	Glastopf
14.50	9	3	0
14.55	9	20	0
15.00	11	208	0
15.05	77	136	0
15.10	104	5	0
15.15	109	9	0
15.20	110	2	1

Tabel 4. Peningkatan *traffic*

Waktu	*Traffic* Masuk (Byte)	*Traffic* Keluar (Byte)
14.50	18.899	30.128
14.55	37.934	75.388
15.00	286.438	635.065
15.05	3.965.887	714.208
15.10	4.962.856	454.885
15.15	5.399.166	506.529
15.20	5.433.565	528.003

Dari detail data tersebut bahwa adanya peningkatan jumlah serangan pada sesnsor Kippo dan Dionaea. Pada saat peningkatan jumlah serangan pada Kippo juga diikuti

peningkatan pada *traffic* masuk dan keluar yaitu pada pukul 14.55,15.00 dan 15.05. Peningkatan serangan yang signifikan terjadi pada sensor Dionaea mengakibatkan peningkatan juga pada traffic, kenaikan *traffic* terlihat jelas pada *traffic* masuk. Kenaikan *traffic* tertinggi terjadi pada pukul 15.05 yang mencapai angka 3.68 MB untuk traffic masuk, jumlah serangan yang meningkat drastis pada waktu tersebut adalah jumlah serangan dari sensor Dionaea yaitu dari 11 serangan menjadi 77 serangan dalam jangka waktu 5 menit. Peningkatan *traffic* keluar tertinggi tercatat pada pukul 15.00 dengan kenaikan *traffic* keluar sebesar 559.68 KB dimana jumlah serangan pada sensor Kippo mencapai 208 serangan yang sebelumnya hanya 20 serangan.

Traffic antarmuka jairngan tidak hanya dipengaruhi jumlah serangan saja, perlu diamati jumlah serangan pada tiap sensor secara seksama. Pada beberapa sampel yang diambil pada waktu tertentu dapat diamati tingkat pengaruh jumlah serangan terhadap traffic yang ada pada antarmuka jaringan. Perhitungan pengaruh tersebut dilakukan dengan cara membagi kenaikan yang terjadi pada traffic dengan kenaikan jumlah serangan. Perhitungan ini merupakan cara yang masih kasar dan perlu dilakukan penelitian yang lebih lanjut. Pada Tabel 5, Tabel 6 dan Tabel 7 akan menunjukkan tingkat pengaruh dari masing masing sensor *Honeypot* terhadap traffic antarmuka jaringan yang diambil pada tanggal 14 Juli 18.00-15 Juli 17.55.

Tabel 5. Pengaruh Glastopf pada Traffic

Serangan	*Traffic In*	*Traffic Out*	In (Byte / Serangan)	Out (Byte / Serangan)
41	71618	1502831	1219,20	35818,76
42	79439	1677929	7821,00	175098,00
34	67385	1156317	1506,75	65201,50
90	107966	2775807	927,96	30405,32
103	98333	3175909	677,55	30390,21
Rata-rata pengaruh			2430,49	67382,76

Dari Tabel 5 di atas dapat dilihat bahwa rata-rata pengaruh dari tiap serangan Glastopf akan mengakibatkan penambahan *traffic* data kurang lebih sebesar 67 KB dan pada traffic keluar dan 2 KB pada *traffic* masuk. Pengarunh Glastopf pada *traffic* keluar disebabkan karenaa Glastopf merupakan *Honeypot* yang dapat memberikan respon ke penyerang sehingga akan ada data yang dikirim keluar pada *traffic* jaringan sebagai bentuk respon dari Glastopf

Tabel 6. Pengaruh Kippo pada Traffic

Serangan	*Traffic In*	*Traffic Out*	In (Byte / Serangan)	Out (Byte / Serangan)
83	123411	255752	1221,21	2643,17
94	70412	141022	467,7	1040,09
100	254802	317965	25121,2	7901,6
76	152205	216140	1734,2	2383,42
39	58256	122011	949,17	2163,22
Rata-rata pengaruh			5898,70	3226,30

Dari perhitungan yang dilakukan menunjukkan setiap serangan pada sensor Kippo akan mempengaruhi traffic masuk sebesar 5 KB dan 3 KB pada *traffic* keluar berdasarkan rata-rata yang telah dihitung. Nilai tersebut lebih kecil dibanding dengan peangaruh dari serangan Glastopf.

Tabel 7. Pengaruh Dionaea pada Traffic

Serangan	*Traffic In*	*Traffic Out*	In (Byte / Serangan)	Out (Byte / Serangan)
56	47178	84106	360,46	175,17
163	226335	411510	3556,55	1163,14
35	79439	1677929	279,32	6253,5
Rata-rata pengaruh			1398,78	2530,60

Hasil perhitungan serangan pada sensor Dionaea pada tabel diatas tidak mempelihatkan pengaruh serangan yang cukup besar terhadap *traffic* antarmuka jaringan. Namun pada Tabel 7 terlihat peningkatan yang signifikan pada *traffic* antarmuka jaringan yang mengikuti kenaikan jumlah serangan sensor Dioanea. Hal dapat terjadi dikarenakan sernsor Dionaea yang berfungsi sebagai pendeteksi malware, biasanya Dionaea akan mendownload berkas malware yang dikirim penyerang untuk dianalisis.

4.3. Analisis Pengaruh Jumlah Serangan pada CPU

Serangan yang masuk ke *Honeypot* dilihat pada grafik kemudian dibandingkan dengan grafik penggunaan CPU. Kenaikan jumlah serangan yang masuk ke sensor *Honeypot* terlihat mempengaruhi penggunaan CPU perangkat. Terlihat pada Gambar 5 bahwa jumlah serangan mempengaruhi penggunaan CPU *user*. Dari grafik yang tertampil pada gambar tersebut merupakan grafik serangan pada sensor *Honeypot* "madu-DSSDI-1-UGM" pada tanggal 25 Juli 2018, sebelum pukul 08.00 terdapat kenaikan jumlah serangan pada sensor Dionaea. Kenaikan tersebut diikuti dengan kenaikan penggunaan CPU pada waktu tersebut.

Gambar 5. Perbandingan grafik serangan dan *CPU* 25 Juli 2018 08.00

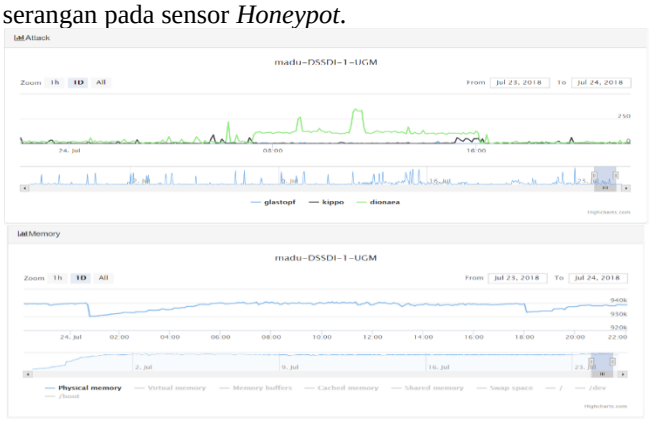
Pengkatan jumlah serangan yang mencapai 331 serangan pada sensor Dionaea diikuti dengan kenaikan penggunaan CPU user menjadi 8 %. Penggunan CPU sistem tidak mngalami perubahan yang tetap berada pada nilai 0. Jumlah serangan pada masing-masing sensor *Honeypot* terlihat memiliki pengaruh berbeda beda terhadap penggunaan CPU, seperti terlihat pada Gambar 6 yang merupakan grafik serangan dan penggunaan CPU dari sensor *Honeypot* "maduDSSDI-2-UGM".

Gambar 6. Perbandingan serangan dan *CPU* "madu- DSSDI - 2 - UGM"

Pada grafik yang diambil dari sensor "madu-DSSDI-2UGM" terlihat pada bagian yang ditandai terdapat peningkatan dari serangan dan penggunaan CPU. Pada sensor *Honeypot* tersebut terpasang dua buah sensor yaitu Kippo dan Dionaea. Dari grafik tersebut terlihat bahwa kenaikan penggunaan CPU pada tiap sensor berbeda.

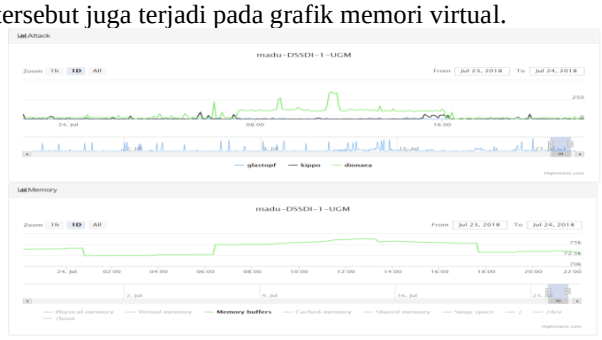
4.4. Analisis Pengaruh Jumlah Serangan pada Memori

Penggunaan memori pada sistem sensor *Honeypot* dilihat pada grafik memori. Pada informasi memori terdapat beberapa jenis memori, yasitu diantaranya memori fisik, memori virtual, memori *swap*, memori *buffer*, memori *cache* dan lainya. Jumlah serangan yang masuk ke *Honeypot* dibandingkan dengan penggunaan memori dan dilihat memori mana yang berpengaruh terhadap jumlah serangan pada sensor *Honeypot*.



Gambar 7. Perbandingan serangan dan memori fisik

Dari jumlah serangan yang masuk ke *Honeypot* pada tanggal 24 terlihat adanya serangan yang cukup signifikan dan terjadi cukup lama. Pengamatan grafik dilakukan pada sensor "madu-DSSDI-1-UGM". Penggunaan memori fisik perangakat sensor tersebut mengalami kenaikan secara perlahan ketika ada peningkatan jumlah serangan kemudian turun setelah jumlah serangan turun. Pada grafik serangan terjadi peningkatan jumlah serangan pada pukul 6.00, pada saat yang sama tidak terjadi peningkatan penggunaan memori fisik pada perangkat. Tidak adanya peningkatan memori fisik pada perangakat dapat disebabkan karena memori fisik yang terpakai sudah mencapai batas maksimal. Hal ini terlihat dari grafik memori fisik yang sudah menigkat sebelumnya sampai batas maksimal, kanaikan grafik mulai berhanti pada pukul 5.30 dan setalah itu grafik terlihat stabil stabil. Hal tersebut menunjukkan bahwa penggunaan memori sudah mencapai batas maksimal. Kemudian grafik memori fisik turun pada saat beberapa jam setelah jumlah serangan turun. Hal tersebut juga terjadi pada grafik memori virtual.



Gambar 8. Perbandingan serangan dan memori *buffer*

Dari grafik memori *buffer* terjadi peningkatan secara drastis pada penggunaan memori *buffer* pada saat ada peningkatan jumlah serangan, penggunaan memori tersebut juga menurun secara drastis setelah jumlah serangan menurun. Hal tersebut terlihat pada grafik yang ditampilkan oleh Gambar 8. Dari grafik memori *buffer* tersebut menunjukkan pengkatan penggunaan memori terjadi bersamaan dengan peningkatan jumlah serangan pada perangkat sensor. Setiap terjadi peningkatan jumlah serangan yang signifikan, penggunaan memori *buffer* juga ikut meningkat. Penggunaan memori *buffer* turun pada saat satu jam setelah jumlah serangan menurun. Penggunaan memori *buffer* yang terpengaruh jumlah serangan pada *Honeypot* dikarenakan fungsi dari memori *buffer* yang berperan sebagai tempat penampungan sementara untuk data yang sedang dikirim atau diterima dari perangkat luar.

V. KESIMPULAN

Berdasarkan hasil pemantauan dan analisi performa dari sistem *Honeypot* dengan SNMP, didapat kesimpulan bahwa sistem pemantauan performa pada sistem *Honeypot* dapat dibangun dengan menggunakan SNMP untuk mengambil informasi sensor *Honeypot*. Dari sistem tersebut terlihat penggunaan memori pada perangkat sensor *Honeypot* telah melebihi kapasitas. Selain ituj umlah serangan yang masuk perangkat sensor *Honeypot* mempengaruhi performa dari perangkat sensor *Honeypot* yang terlihat dari peningkatan *traffic*, beban CPU dan penggunaan memori pada saat terjadi peningkatan jumlah serangan. Pengaruh jumlah serangan terhadap *traffic* antarmuka jaringan berbeda-beda tergantung dari jenis serangan yang masuk, serangan yang terdeksi pada sensor Glast*o*fp dapat menambah *traffic* data rata-rata 67 KB/serangan pada *traffic* keluar dan Kippo sebesar 6 KB/serangan, sedangakan pada sensor Dionaea tidak tentu karena sensor tersebut berfungsi deteksi *malware*.

DAFTAR PUSTAKA

[1] M. Rouse and M. Cobb, "*Honeypot*," Juni 2018. [Online]. Available: https://searchsecurity.techtarget.com/definition/ *Honeypot*. [Accessed 2 Juli 2018].

[2] Cisco, "Cisco," 2016. [Online]. Available: https://www.cisco.com/c/en/us/products/securit y/what -is-*Network*-security.html. [Accessed 13 Juli 2018].

[3] H. Bhagaskara, D. Adhipta and L. E. Nugroho, "Uji Penetrasi Sistem Keamanan jaringan Universitas Gadjah Mada dengan Information

System Security Assessment Framework (ISSAF)," Universitas Gadjah Mada,
Yogyakarta, 2015.

[4] B. Fredianto and Soedjatmiko, "Manajemen Keamanan Jaringan Informasi Menggunakan
OSSIM,"
Universitas Gadjah Mada, Yogyakarta, 2005.

[5] A. N. Singh and R. C. Joshi, "A *Honeypot System* for Efficient Capture and Analysis *of Network* Attack Traffic," IEEE, 2011.

[6] I. L. Pribadi, "Implementasi Sistem Keamanan Jaringan Menggunakan *Honeypot* Dionaea, IDS, Dan Cuckoo Sandbox," Universitas Telkom, Bandung, 2013.

[7] R. A. Habsoro and N. R. Rosyid, "Implementasi *Honeypot* Untuk Mengungkap Port Scanning Attacks dalam Jaringan," Universitas Gadjah Mada, Yogyakarta, 2015.

[8] J. V. Ho*of*, "Can a SSH *Honeypot* Be Used to
Attract
Attackers and Improve Security?," 29 September 2014. [Online].Available: https://securityintelligence.com/can-a-ssh*Honeypot*be-used-to-attract-attackers-andimprove-security/. [Accessed 20 Juli 2018].

[9] L. Rist, "Know Your Tools: Glastopf - A dynamic, lowinteraction web application *Honeypot*," 15 November 2010. [Online].
Available:
https://www.honeynet.org/papers/KYT_glastop f. [Accessed 20 Juli 2018].

[10] I. Laksana and N. R. Rosyid, "Implementasi *Honeypot* Sebagai Pemantauan Parameter Pada Http *Request* Untuk Mengetahui Tujuan Serangan," Universitas Gadjah Mada,
Yogyakarta, 2017.

[11] Anomali, Inc., "Modern Honey *Network*,"
[Online]. Available: https://github.com/threatstream/mhn.
[Accessed 30 Juni 2018].

[12] W. Septian, W. Najib and S. Sumaryono, "Implementasi *Honeypot* Menggunakan Platform Modern Honey *Network* (studi Kasus Di Direktorat Sistem Dan Sumber Daya
Informasi, Universitas Gadjah Mada)," Universitas Gadjah Mada, Yogyakarta, 2017.

[13] RaspberryPi, "Raspberry Pi 3 Model B," 2016. [Online].

[14] J. Prakoso and A. K. Sari, "Perbandingan Performa Snort dan Suricata Sebagai Sistem Deteksi Intrusi pada Raspberry Pi," Universitas Gadjah Mada, Yogyakarta, 2018.

[15] Techopedia, "*Network Management System* (NMS)," 2018. [Online]. Available: https://www.techopedia.com/definition/11988/ networ k-*Management-System*-nms. [Accessed 30 Juni 2018].

[16] Cisco, "*Network Management System*: Best Practices White Paper," 11 Juli 2007. [Online].
Available:
https://www.cisco.com/c/en/us/support/docs/av ailabili ty/high-availability/15114-NMSbestpractice.html. [Accessed 30 Juni 2018].

[17] R. Lukitawati and A. Subardono, "Implementasi Dan Analisis Sistem Monitoring Performance Jaringan Dengan Parameter *Quality Of Service* (qos)," Universitas Gadjah Mada, Yogyakarta, 2017.

[18] Y. Sholikatin and N. R. Rosyid, "Implementasi *Fault Management* (manajemen Kesalahan)
Pada *Network Management System* (NMS) Berbasis SNMP," Universitas Gadjah Mada, Yogyakarta, 2017.

[19] M. Rouse, J. Scarpati, A. Ranjan, C. Karbinski and J. Mathew, "*Simple Network Management Protocol* (SNMP)," Januari 2018. [Online].
Available: https://search*Network*ing.techtarget.com/definit ion/SN MP. [Accessed 1 Juli 2018].

[20] S. Wilkins, "SNMP Concepts and Configuration," 20 Juli 2011. [Online]. Available:http://www.ciscopress.com/articles/ar ticle.asp?p=1730 888. [Accessed 2 Juli 2018].