Pengujian Dan Analisa Keamanan Website Terhadap Serangan *SQL Injection* (Studi Kasus : *Website* UMK)

Moh Dahlan, Anastasya Latubessy, Mukhamad Nurkamid, Lelly Hidayah Anggraini

ABSTRACT

Security is an important factor to develop a website. It has been a challenge for website developer since there is no guarantee for the definition of 'secure'. 'There is no totally secure system', is not only a statement but has been proof in reality. UMK website's is a website used for information gateway in campus. Since this website has been accessed widely, it was needed to pay attention on its security. There are some ways to test the website security like using SQL Injection. SQL injection is susceptibility when attacker has a chance to inject the Structured Query Language (SQL) through application back end. This research was aimed to found the weakness of UMK website. Those weakness would be analyzed so the solution can be used to develop secure website in the future.

Keyword : *Analysis, Security, Website, SQL Injection*

ABSTRAK

Keamanan merupakan salah satu faktor penting yang harus diperhatikan dalam membangun sebuah *website*. Hal tersebut menjadi sebuah tantangan tersendiri bagi para pengembang *website,* karena tidak ada jaminan yang pasti akan defenisi 'aman' itu sendiri. "Tidak ada sistem yang benar-benar aman", bukanlah sebuah pernyataan semata, namun telah dirasakan dalam realitas. *Website* UMK merupakan *website* yang digunakan sebagai media dan sarana informasi kampus. Mengingat *website* ini dapat diakses secara luas, maka dinilai perlu memperhatikan keamanan *website*. Terdapat beberapa cara yang dapat digunakan untuk melakukan pengujian terhadap kemanan *website*. Salah satunya adalah dengan melakukan *SQL Injection.* SQL *injection* adalah kerentanan yang terjadi ketika penyerang memiliki kemampuan untuk mempengaruhi *Structured Query Language* (*SQL*) *query* yang melewati suatu aplikasi ke *database back-end*. Dengan diadakannya penelitian ini, diharapkan dapat diperoleh kelemahan dari *website* UMK. Kelemahan tersebut akan dianalisa sehingga memperoleh solusi kedepan guna pengembangan *website* yang lebih aman.

Kata Kunci : *Analisa, Keamanan, Website, SQL Injection*

A. PENDAHULUAN

1. Latar belakang

Website adalah sebuah cara untuk menampilkan diri di *Internet*. Dibaratkan w*ebsite* adalah sebuah tempat di *Internet* dimana siapa saja di dunia ini dapat mengunjunginya. Keamanan merupakan salah satu indikator penting dalam membangun sebuah *website*, mengingat akses ke *Internet* yang terbuka bebas bagi masyarakat umum. Selain itu, saat ini w*ebsite* tidak hanya dijadikan layanan untuk memberikan informasi statis, tetapi telah berkembang dengan ditambahkannya fitur-fitur untuk melakukan transaksi secara *on-line*. Sampai saat ini tidak ada *website* yang dapat dikatakan benar-benar aman.

Website UMK (Universitas Muria Kudus) dengan domain umk.ac.id merupakan *website* yang digunakan sebagai media dan sarana informasi kampus. Mengingat *website* ini dapat diakses secara luas, maka dinilai perlu memperhatikan keamanan *website*. Terdapat beberapa cara yang dapat digunakan untuk melakukan pengujian terhadap kemanan *website*. Salah satunya adalah dengan melakukan *SQL Injection.*

SQL injection adalah kerentanan yang terjadi ketika penyerang memiliki kemampuan untuk mempengaruhi *Structured Query Language (SQL) query* yang melewati suatu aplikasi ke *database back-end*. Dengan mampu mempengaruhi apa yang akan diteruskan ke *database*, penyerang dapat memanfaatkan sintaks dan kemampuan dari *SQL*, serta kekuatan dan fleksibilitas untuk mendukung fungsi operasi *database* dan fungsionalitas sistem yang tersedia ke *database*. Injeksi *SQL* bukan merupakan kerentanan yang eksklusif mempengaruhi aplikasi *Web*, kode yang menerima masukan dari sumber yang tidak dipercaya dan kemudian menggunakan *input* yang membentuk *SQL* dinamis bisa rentan[1]. Kasus *SQL Injection* terjadi ketika seorang penyerang dapat memasukkan serangkaian pernyataan *SQL* ke *query* dengan memanipulasi data input ke aplikasi[2].

Berdasarkan defenisi tersebut, dapat dikatakan bahwa serangan *SQL Injection* sangat berbahaya karena penyerang yang telah berhasil memasuki *database* sistem dapat melakukan manipulasi data yang ada pada *database* sistem. Proses manipulasi data yang tidak semestinya oleh penyerang dapat menimbulkan kerugian bagi pemilik *website* yang terinjeksi. Kebocoran data dan informasi merupakan hal yang fatal. Data-data tersebut dapat disalahgunakan oleh pihak yang tidak bertanggung jawab.

Keamanan data dan informasi sangat penting dalam menjaga ketahanan sebuah *website.* Berdasarkan uraian-uraian tersebut, maka dinilai perlu untuk menguji kemanan *website* UMK terhadap serangan *SQL Injection,* serta melakukan analisa terhadap kelemahan sistem yang ada, sehingga dapat diperoleh tindakan selanjutnya untuk perbaikan sistem.

2. Rumusan masalah

Berdasarkan pada latar belakang yang dijelaskan sebelumnya, maka rumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana cara melakukan pengujian terhadap keamanan *website* UMK?
2. Bagaimana melakukan analisa terhadap keamanan *website* UMK?

3. Tujuan penelitian

Tujuan dari penelitian ini adalah :

1. Melakukan pengujian terhadap keamanan *website* UMK.
2. Melakukan analisa terhadap hasil pengujian keamanan *website* UMK.

4. Manfaat penelitian

Manfaat dari penelitian ini adalah :

1. Dapat mengetahui kelemahan *website* UMK, apakah sistem rentan terhadap serangan.
2. Dapat mengetahui langkah atau tindakan pencegahan, berdasarkan hasil analisa terhadap pengujian keamanan *website* UMK.

B. LANDASAN TEORI

Beberapa penelitian terkait *forensic* jaringan antara lain Ruchandani, dkk (2006) telah melakukan eksperimen dasar forensik jaringan dengan menangkap lalu lintas paket pada jaringan, menganalisis karakteristiknya, dan mencoba untuk mengetahui aktivitas yang berbahaya dalam membantu mengidentifikasi sumber aktivitas sebagai kerusakan yang dilakukan pada jaringan menggunakan tools *tcp-dump, ethereal*, dan *n-map*. Disimpulkan bahwa *tcp-dump*, *ethereal*, dan *n-map* sangat ampuh untuk membantu menangkap dan menganalisis paket jaringan diantaranya paket *sniffing* dan *port scanning*[3].

Kaushik dan Joshi (2010) melakukan forensik jaringan untuk serangan ICMP. forensik jaringan adalah teknologi investigasi khusus yang menangkap, merekam dan menganalisis paket jaringan dan menentukan anomali dalam lalu lintas apakah sebuah serangan atau bukan. Tantangan forensik jaringan adalah pengumpulan, pemeriksaan, analisis, identifikasi fitur untuk menentukan serangan dan capture paket dengan volume yang besar. Proses analisis forensik melibatkan persiapan, pengumpulan, pelestarian, pemeriksaan, analisis, investigasi dan tahap presentasi. Kaushik dan Joshi mengusulkan model sistem forensik jaringan untuk ICMP untuk mengumpulkan data jaringan, mengidentifikasi paket yang mencurigakan, memeriksa protokol dan validasi serangan. Untuk mengatasi masalah jumlah data yang besar yang akan diperiksa maka hanya digunakan informasi header paket ICMP saja dengan format paket capture: libcap dan ekstensi file pcap. Eksperimen dilakukan menggunakan *nmap, sing* dan *traceroute*[4].

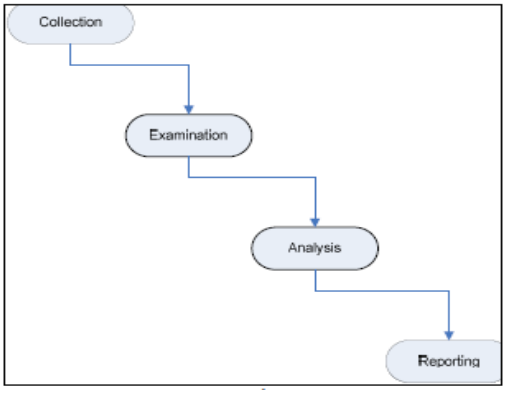
Beberapa penelitian terkait yang membahas tentang *SQL Injection* antara lain, Halfond dan Orso (2005) menyajikan dan mengevaluasi teknik baru untuk mendeteksi dan mencegah serangan *SQL Injection*. Teknik menggunakan pendekatan berbasis model untuk mendeteksi *query* illegal sebelum dieksekusi pada basis data. Halfond dan Orso (2005) mengembangkan alat AMNESIA (*Analysis and Monitoring for NEutralizing SQLInjection Attacks*) yang menerapkan teknik secara statis dan dinamis untuk mengevaluasi teknik pada aplikasi *web*. Teknis statis menggunakan analisis program untuk membangun model *query* yang sah yang dapat dihasilkan aplikasi, sedangkan teknik dinamis menggunakan pemantauan *runtime* untuk memeriksa yang dihasilkan *query* dan memeriksa model statis yang dibangun. Halfond dan Orso (2005) mengusulkan model baru untuk melawan SQLIA yaitu kerentanan yang disebabkan oleh input pengguna yang tidak divalidasi dengan menggunakan kombinasi teknik analisis statis dan dinamis dan menerapkannya pada alat *prototype* AMNESIA[5].

Pomeroy dan Tan (2011) membahas mengenai tantangan rekaman jaringan dan manfaat penggunaannya di masa depan. Solusi perekaman adalah untuk mendeteksi dan mengungkap serangan. *SQL Injection* merupakan salah satu serangan teratas selain XSS (*Cross Site Scripting*) dari tahun 2002 hingga tahun 2008. Cara untuk meningkatkan rekonstruksi serangan *web* adalah memahami dan memperbaiki kelemahan aplikasi *web* serta dukungan hukum pidana dan perdata. *Firewall* tidak efektif untuk memblokir lalu lintas sedangkan IDS (*Intrusion Detection System*) untuk memicu perekaman aplikasi jaringan yang dapat meningkatkan efektifitas rekonstruksi serangan *SQL Injection*[6].

Penelitian yang akan dikerjakan saat ini, akan membahas tentang serangan *SQL Injection*, dengan melakukan pengujian terhadap sistem keamanan *website* Universitas Muria Kudus. *Tools* yang digunakan adalah *IDS Snort*, dimana *IDS Snort* adalah *tools* otomatis yang membantu memantau aktivitas jaringan serta pengujian penetrasi untuk menemukan dan mengeksploitasi kerentanan *SQL Injection* pada halaman *web*.

C. METODE PENELITIAN

Pada penelitian ini metodologi yang digunakan secara garis besar menggunakan dua pendekatan, yaitu pendekatan proses forensik dan pustaka. Diagram alir penelitian diadopsi dari model proses *forensic* dari Baryamureeba dan Tushabe, 2004[7]. Model proses *forensic* ini meliputi beberapa tahapan yaitu *collection, examination, analysis, reporting.* Ditunjukan pada Gambar 1.



Gambar1. Diagram Alir Penelitian[7]

Tahapan-tahapan yang digunakan dalam proses forensik antara lain :

1. Identifikasi (*Collection*)
   Pada tahap ini dilakukan identifikasi terhadap kebutuhan-kebutuhan, baik kebutuhan fungsional maupun identifikasi

kondisi jaringan *website* Universitas Muria Kudus. Meliputi identifikasi kebutuhan alat dan bahan, identifikasi variabel yang diteliti, jangka waktu dan tempat penelitian.

a. Alat dan Bahan :
   i. Satu buah komputer sebagai IDS Snort Server.
   ii. Tools IDS Snort
   iii. Tools wireshark
   iv. OS Linux (Ubuntu)
b. Waktu Penelitian : Juni 2013 - Januari 2014
c. Tempat Penelitian : Pusat Sistem Informasi UMK
d. Variabel yang diteliti : Website UMK

2. Pengujian (*Examintaion*)
   Pada tahap ini mulai dilakukan pengujian terhadap keamanan *website* Universitas Muria Kudus (UMK). Peneliti mulai melakukan SQL *Injection* terhadap *website* UMK. Serangan disini hanya dilakukan untuk melihat apakah penyerang dapat memasuki *database website* UMK tanpa melakukan manipulasi terhadap *database* yang ada, sehingga tidak akan mengganggu kondisi *website* yang sedang berjalan.

3. Analisa (*Analysis*)
   Pada tahapan ini, dilakukan analisa terhadap hasil serangan SQL *Injection*, hal ini berguna untuk menemukan kelemahan-kelemahan pada *website* UMK. Proses analisa menggunakan bantuan *tools wireshark* , sehingga tiap *frame* pada paket data yang mengalir di jaringan dapat terbaca. Berdasarkan hasil analisa, juga diharapkan dapat diperoleh solusi untuk pengembangan keamanan sistem.

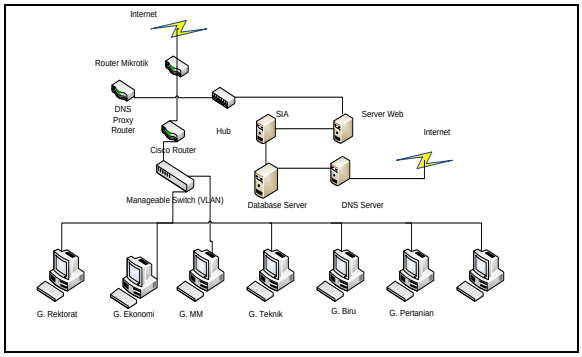
4. Pelaporan (*Reporting*)
   Pada tahap pelaporan, mulai dilakukan dokumentasi terhadap hasil penelitian beserta analisisnya.

D. HASIL DAN PEMBAHASAN

1. Topologi Jaringan UMK

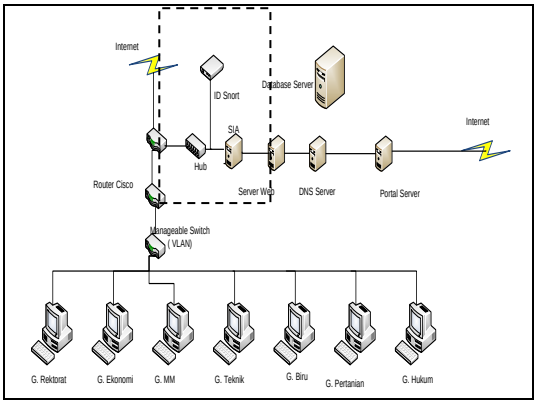
Jaringan *intranet* UMK adalah jaringan yang menghubungkan komputer-komputer yang tersebar dilingkungan Universitas Muria Kudus baik yang terhubung secara *Local Area Network*(LAN) maupun terhubung secara *offline* menggunakan fasilitas *dial-up.* Pusat jaringan(*backbone*) *intranet* terletak di Unit Pelaksana Teknis Perencanaan Sistem Informasi[8].

Topologi Jaringan yang berjalan pada jaringan di Universitas Muria Kudus dapat dilihat seperti yang terdapat pada Gambar 2. Dalam hal ini server DNS, proxy, dan mikrotik router dipisahkan secara *hardware*, sehingga beban kerja pada *proxy server* menjadi terpisah.



Gambar 2. Topologi Jaringan UMK Sebelum Penelitian[8]

Dalam penelitian ini, ditambahkan sebuah *server* untuk IDS Snort. Server IDS Snort ini diletakan didepan jaringan internet, sebelum *web server* sistem. Dengan penambahan *server* IDS Snort seperti yang ditunjukan pada Gambar 3 maka semua paket data yang akan masuk ke *web server* UMK dapat dipantau.



Gambar 3. Topologi Jaringan UMK Setelah Penambahan IDS *Server*

2. Implementasi IDS Snot

Penelitian ini menambahkan sebuah komputer *server* yang bertindak sebagai IDS Snort. Gambar 4 menunjukkan hasil implementasi dari *server* IDS Snort yang

diletakan pada Pusat Sistem Informasi(PSI) Universitas Muria Kudus.

Gambar 4. IDS *Server* di Pusat Sistem Informasi UMK

3. Hasil Analisa Paket Data IDS Snort

Server yang diletakan di PSI dipantau selama kurang lebih satu bulan, banyak sekali paket data yang mengalir. Terdapat beberapa paket data yang mengalir merupakan ancaman, sehingga IDS akan memberikan *alert.* Beberapa *alert* yang muncul pada IDS Snort seperti yang ditunjukan pada Gambar 5. Baris pertama sampai baris kelima pada Gambar 4.4 menunjukan adanya *false postif*, merupakan gejala yang sebenarnya tidak ada. Sehingga tidak terlalu berbahaya. Baris 7 sampai baris 8 menunjukan adanya pengintaian yang melihat apakah ada celah yang terbuka.

```
1  [**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**]
2  [Classification: Attempted Denial of Service] [Priority: 2]
3  01/06-06:44:50.070725 192.168.1.3 -> 198.52.235.98
4  UDP TTL:64 TOS:0x0 ID:16758 IpLen:20 DgmLen:820
5  Frag Offset: 0x039D   Frag Size: 0x0320
6
7  [**] [123:8:1] (spp_frag3) Fragmentation overlap [**]
8  [Priority: 3]
9  01/06-06:44:50.843844 192.168.1.3 -> 198.52.235.98
10 UDP TTL:64 TOS:0x0 ID:53652 IpLen:20 DgmLen:820
11 Frag Offset: 0x039D   Frag Size: 0x0320
```

Gambar 5. Alert pada IDS Snort

Gambar 6 menunjukkan salah satu frame paket data yang mengalir di jaringan. Paket data ini di-*capture* pada tanggal 6 Januari 2014, dengan panjang frame 81 bytes (648 bits). Protokol jaringan uang digunakan adalah UDP.

```
Frame 1: 81 bytes on wire (648 bits), 81 bytes captured (648 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Jan  6, 2014 09:49:15.689472000 SE Asia Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1388976555.689472000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 81 bytes (648 bits)
    Capture Length: 81 bytes (648 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:dns]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
```

Gambar 6. Frame paket data di jaringan

Ethernet merupakan lapisan fisik dan data link pada *Local Area Network.* Gambar 7 menunjukkan ethernet yang digunakan, baik dari sumber dan tujuan paket data di jaringan.

```
Ethernet II, Src: Hewlett-_c3:c8:20 (d8:d3:85:c3:c8:20), Dst: Tp-LinkT_22:21:80 (00:1d:0f:22:21:80)
  Destination: Tp-LinkT_22:21:80 (00:1d:0f:22:21:80)
    Address: Tp-LinkT_22:21:80 (00:1d:0f:22:21:80)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: Hewlett-_c3:c8:20 (d8:d3:85:c3:c8:20)
    Address: Hewlett-_c3:c8:20 (d8:d3:85:c3:c8:20)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Type: IP (0x0800)
```

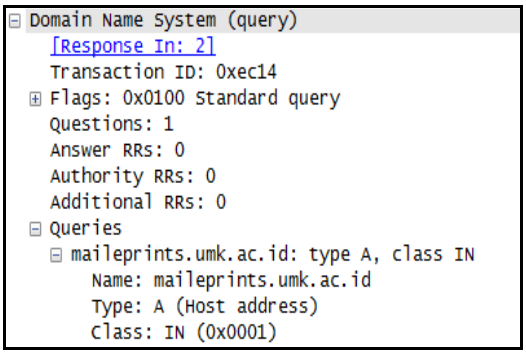
Gambar 7. Ethernet paket data di jaringan

Paket data yang mengalir pada jaringan menggunakan IP Versi 4, dengan alamat sumber adalah 192.168.1.4 dengan tujuan 223.165.7.209. Informasi detailnya seperti yang ditunjukkan pada Gambar 8.

```
Internet Protocol Version 4, Src: 192.168.1.4 (192.168.1.4), Dst: 223.165.7.209 (223.165.7.209)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
    .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
  Total Length: 67
  Identification: 0xf8a0 (63648)
  Flags: 0x02 (Don't Fragment)
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
  Fragment offset: 0
  Time to live: 64
  Protocol: UDP (17)
  Header checksum: 0x98e6 [correct]
  Source: 192.168.1.4 (192.168.1.4)
  Destination: 223.165.7.209 (223.165.7.209)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
```

Gambar 8. IP paket data di jaringan

Informasi Domain Name System(DNS) ditunjukkan pada Gambar 9. Terlihat bahwa DNS yang coba diakses adalah maileprints.umk.ac.id.



Gambar 9. DNS paket data di jaringan

4. Solusi *Rule* yang Ditawarkan

Aturan snort yang ditawarkan peneliti pada *sysadmin* dalam pengembangan selanjutnya adalah aturan snort untuk melihat adanya kemungkinan adanya serangan SQL *Injection*(varchar) seperti yang ditunjukkan pada Gambar 10.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"< IN >
WEB_SERVER Possible SQL Injection (varchar)";
flow:established,to_server; content:"varchar("; nocase; http_uri;
classtype:attempted-admin;
reference:url,doc.emergingthreats.net/2008175;
reference:url,www.emergingthreats.net/cgi-
bin/cvsweb.cgi/sigs/WEB_SERVER/WEB_SQL_Injection_Monster_List;
sid:2008175; rev:5;)
```

Gambar 10. Aturan Snort tentang Possibility SQL Injection(varchar)

Menurut *rule*, apabila terdapat paket TCP dari luar melalui port apapun menuju ke dalam melalui port HTTP (80), yang sesuai dengan pola (baca keterangan *rule* 1) maka akan mengirimkan pesan "WEB_SERVER *Possible SQL Injection* (varchar)".

Keterangan *Rule* :

- alert adalah tanda peringatan.
- tcp adalah jenis protokol transport.
- $EXTERNAL_NET any adalah host asal yang melewati port manapun.
- -> adalah aliran dari host asal ke host tujuan.
- $HTTP_SERVERS $HTTP_PORTS adalah server HTTP melewati HTTP port (80).
- msg:"< IN > WEB_SERVER Possible SQL Injection (varchar)"; adalah pesan yang akan diterima apabila terjadi sebuah event.
- flow:established,to_server; adalah koneksi TCP yang terbentuk dalam host sumber ke host tujuan.
- content:"varchar("; artinya konten spresifik yang dicari
- nocase; adalah pengabaian case untuk menetapkan pola yang dicari.
- http_uri; adalah pencarian pola yang sesuai dengan konten pada normalized URI.
- classtype:attempted-admin; artinya mencoba mendapatkan hak user, ini memiliki prioritas yang tinggi.
- reference:url,doc.emergingthreats.net/2008175; reference:url,www.emergingthreats.net/cgibin/cvsweb.cgi/sigs/WEB_SERVER/WEB_SQL_Injection_Monster_List; merupakan referensi ke sistem pengidentifikasi serangan eksternal.
- sid:2008175; merupakan id dari aturan snort.
- rev:5;) artinya revisi aturan snort ke 5.

E. KESIMPULAN

Berdasarkan penelitian yang dilakukan dapat disimpulkan bahwa :

1. Tidak ada sistem yang dikatakan benar-benar aman, sehingga aktifitas jaringan perlu dipantau setiap saat.
2. Dengan adanya IDS Snort, aktifitas jaringan yang berjalan di Pusat Sistem Informasi (PSI) Universitas Muria Kudus dapat dipantau setiap saat.
3. Pemberian aturan snort / *rule* yang tepat dapat memberikan peringatan/*alert* sehingga serangan dari *intruder* terhadap jaringan dapat diketahui oleh *sysadmin*.

DAFTAR PUSTAKA

[1] Clarke, J., 2009, *SQL Injection Attacks and Defense*. Burlington: Syngress Publishing and Elseiver.

[2] Anley, C., 2002, Advanced SQL Injection in SQL Server Applications. *An NGSSoftware Insight Security Research (NISR) Publications*: Next Generation Security Software Ltd.

[3] Ruchandani, B., Kumar, M., Kumar, A., Kumari, K., Sinha., A.,K., 2006, Ekperimentation In Network Forensics Analysis. *Proceedings of the Term Paper*

Series under CDACCNIE Bangalore, India, December 2006.

[4] Kaushik,A.,K., Joshi,R.,C., 2010, Network Forensic System for ICMP Attacks. *International Journal of Computer Applications*, Vol 2, No.3, May 2010.

[5] Halfond, W.G.J., Orso, A., 2005., AMNESIA: Analysis and Monitoring for NEutralizing SQLInjection Attacks. *IEEE and ACM Intern. Conf. On Automated Software Engineering (ASE 2005)*. Hal. 174–183, Nov. 2005.

[6] Pomeroy, A., Tan, Q., 2011, Effective SQL Injection Attack Reconstruction Using Network Recording. *IEEE International Conference on Computer and Information Technology*.Canada.

[7] Baryamureeba,V., Tushabe, F., 2004, The Enhanced Digital Investigation Process Model. *Proceedings of the Fourth Digital Forensic Research Workshop*, May 27.

[8] Nurkamid, M., 2011, Analisa Keefektifan Jaringan Local Area Network (Intranet) Universitas Muria Kudus, Jurnal Sains dan Teknologi, vol. 4 no. 2, Universitas Muria Kudus, Kudus.