PERANCANGAN HOST-BASED INTRUSION DETECTION SYSTEM BERBASIS ARTIFICIAL NEURAL NETWORK

Bondan Himawan, Taufiq Hidayat
Laboratorium Pemrograman dan Informatika Teori
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia
Jln. Kaliurang KM 14,5 Yogyakarta 55501
e-mail: neojavanese@gmail.com, taufiqhid@fti.uii.ac.id

ABSTRAKSI

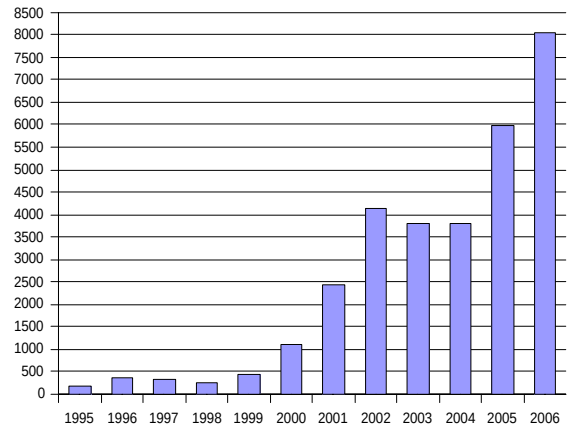
Host-Based Intrusion Detection System (HIDS) adalah sistem yang mampu mendeteksi dan mengidentifikasikan ancaman yang terjadi dalam sistem komputer, dengan cara mengenali setiap pola serangan yang dilakukan oleh intruder. Untuk mendeteksi setiap gejala ancaman tersebut, sistem menggunakan pendekatan anomaly-based intrusion detection, yaitu pengenalan pola berdasarkan pada urutan system call yang menyusun sebuah proses dalam sistem komputer. Artificial Neural Network digunakan untuk mengenali pola-pola dari system call yang dimasukkan ke HIDS, sehingga pola-pola serangan baru dapat dideteksi, walaupun tidak dideklarasikan sebelumnya. Metode yang digunakan untuk mengenali pola adalah metode Multi Layer Perceptron (MLP) dan Self Organizing Map (SOM).

Kata kunci: *intrusion detection, neural netwrok, monitoring system*

1. PENDAHULUAN

1.1 Latar Belakang Permasalahan

Menurut data yang dirangkum oleh CERT, jumlah serangan yang disebabkan oleh vulnerabilities sebuah sistem sangat meningkat. Dari tahun 1995 hingga 2006, telah terjadi sebanyak 30.780 serangan, yang sebagian besar serangan tersebut, yaitu sebanyak 29.274 terjadi dalam tahun 2000 hingga 2006.



Gambar 1. Jumlah serangan dari tahun per tahun.

Peningkatan jumlah serangan yang sangat berarti pada *vulnerability host* ini, dipengaruhi oleh perubahan budaya masyarakat, yang pada saat ini telah menggantungkan seluruh kebutuhannya pada sebuah sistem komputer. Ini menjadi sebuah masalah serius mengingat pada saat ini, komputer telah menjadi sebuah kebutuhan primer manusia dalam menyelesaikan pekerjaannya.

Untuk mengurangi resiko dari setiap serangan, diperlukan usaha-usaha terkait dengan masalah keamanan sistem komputer. Salah satu mekanisme yang dilakukan untuk pengamanan sebuah sistem adalah dengan *intrusion detection*. *Intrusion detection* adalah sebuah mekanisme untuk mendeteksi manipulasi-manipulasi yang tidak diinginkan oleh seorang *intruder*. Tujuan utama dari intrusion detection adalah sebagai alarm, yang akan memberikan peringatan apabila terdapat penetrasi dalam parameter keamanan, dan memberikan solusi terhadap masalah keamanan tersebut.

*Artificial Neural Network (*ANN) dapat dimanfaatkan untuk meyelesaikan permasalahan di bidang jaringan komputer, khususnya *Intrusion Detection System* (IDS). IDS bukanlah sebuah bidang baru dalam dunia komputer, akan tetapi hal ini belumlah di gunakan secara menyeluruh dalam sebuah jaringan komputer komersial. Hal ini dikarenakan sebuah sistem komputer belumlah sempurna untuk melakukan penyelesaian permasalahan/problem solving. Bagaimanapun juga kelak sistem ini akan mempermudah manusia dalam mengelola sebuah sistem komputer.

Ada beberapa kelompok yang telah mencoba menggunakan sistem yang mengkombinasi dua disiplin ilmu ini, berdasarkan dari pendekatan masing-masing. Di *MIT Lincoln Laboratory*, sistem ini dimplementasikan dengan pendekatan *misuse detection*. Data yang diisikan untuk NN berisikan parameter-parameter serangan dalam sebuah lalu-lintas jaringan dari sebuah sistem UNIX, yang mengijinkan penyerang untuk mengakses hak-hak *root* dalam sebuah server(Lippman, 1999). Di UBILAB Laboratory, pendeteksian penyerangan dilakukan dengan menggunakan pendekatan NN *Self Organizing Map* (SOM). SOM akan mengklasifikasikan parameter-parameter dari lalu lintas jaringan dalam dua-dimensional array untuk fisualisasi (Giradin, 1998). Hal yang sama juga dilakukan di *Departement of Computer Science*

Rensselaer Polytechnic Institute, New York. Pada penelitian tersebut, pendekatan yang digunakan adalah pada pendeteksian intrusinya, yaitu dengan *Network-Based Intrusion Detection System*, dengan menggunakan methode *Multi Layer Perceptron* (MLP) dan SOM.

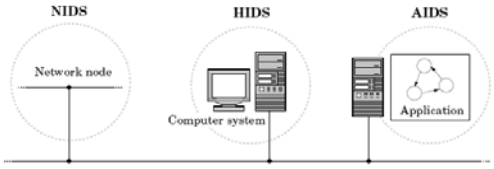
1.2 Tujuan

Tujuan dari penelitian tentang *Host-Based Intrusion Detection System* adalah merancang sebuah sistem cerdas berbasiskan Artificial Neural Network yang digunakan untuk *Intrusion Detection System*.

2. DASAR TEORI

2.1 Host-Based Intrusion Detection System

Intrusion Detection adalah proses mengidentifikasi dan menanggapi setiap manipulasi sistem mencurigakan dalam sistem komputer dan *communication resource* (Vigna, 2005; Crothers, 2002). Manipulasi-manipulasi tersebut dapat berupa sebuah intrusi yang dilakukan oleh seorang *hacker* dengan memanfaatkan celah-celah keamanan, ataupun juga oleh seorang *script kiddies* dengan sebuah *software* keamanan. *Intrusion Detection System* (IDS) adalah sebuah sistem yang digunakan untuk *monitoring system* dan mencoba untuk memberikan respon terhadap setiap intrusi dalam sebuah sistem atau *network resource*. Berdasarkan sumber kejadiannya, IDS dibagi menjadi 4 macam, yaitu: *Network Intrusinons Detection System* (NIDS), *Host-Based Intrusinon Detection System* (HIDS), *Protocol-Based Intrusion Detection System* (PIDS), *Application Protocol-Based Intrusion Detection System* (AIDS), serta *Hybrid Intrusion Detection System* (HIDS).

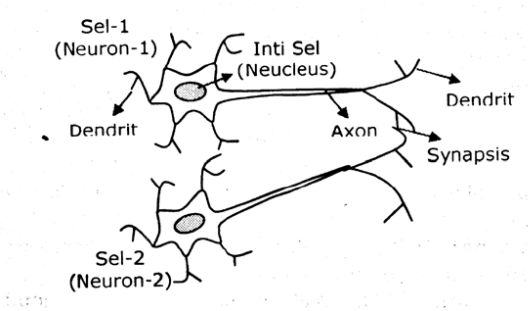


Gambar 2. IDS berdasarkan pada tempat kejadian.

Host-Based Intrusion Detection System (HIDS) adalah sistem pengawasan dan pendeteksian setiap aktivitas dalam sebuah komputer. HIDS menggunakan *host log information*, *system activity* dan *scanner*, seperti virus scanner untuk mendeteksi setiap serangan yang ditujukan terhadap single komputer tersebut. Seperti NIDS, yang secara dinamis memeriksa setiap paket yang dilewatkan dari sebuah jaringan komputer, HIDS mendeteksi program yang mengakses *resource* dalam sistem komputer tersebut. HIDS akan memeriksa setiap perubahan dalam sistem, dan akan menyimpannya dalam sebuah basis-data mengenai RAM, *file-system* dan semua parameter-parameter yang dicurigai sebagai sebuah intrusi.

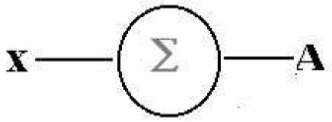
2.2 Artificial Neural Network

Artificial Neural Networks (ANN) atau lebih dikenal dengan *Neural Networks* (NN) merupakan sebuah metode bagaimana komputer dapat mempelajari serta mengenali sesuatu tugas yang dihadapi komputer tersebut. Hal ini merupakan representasi dari jaringan jaringan biologis yang dimiliki oleh manusia. Dalam jaringan syaraf biologis, terdapat sebuah jaringan yang sangat luas, yang terdiri dari neuron-neuron yang saling berhubungan. Ada tiga komponen penting yang dimiliki oleh setiap neuron, yaitu *dendrit*, *soma,* dan *akson*. *Dendrit* dan *akson* bertugas untuk menyampaikan informasi yang berupa *impuls*[1] elektrik dari satu neuron ke neuron yang lain. *Soma* atau badan sel, akan menjumlahkan informasi-informasi yang disampaikan dendrit melalui *sypnatic gap*. Semakin banyak informasi yang dijumlahkan oleh *soma*, maka akan semakin besar pula informasi dari sebuah neuron, yang berarti akan menjadikan kepintaran dari orang bertambah.



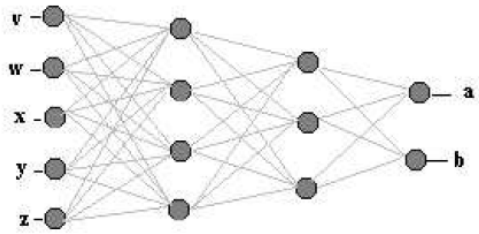
Gambar 3. Jaringan syaraf biologis.

Dalam ANN, proses komunikasi antar node[2] juga berlangsung setiap kali terdapat impuls dari satu node ke node yang lain.Setiap node terhubung dengan node yang lain melalui sebuah layer. Untuk mengimplementasikan penambahan informasi yang dilakukan oleh *soma* pada jaringan syaraf biologis, di ANN, setiap layer telah memiliki bobot tertentu, yang nantinya juga akan selalu dijumlahkan. Bobot-bobot inilah yang nantinya akan digunakan oleh node-node untuk menyelesaikan suatu permasalahan. Pada jaringan biologis, bobot-bobot tersebut dapat dianalogikan dengan kasi pada proses kimia yang terjadi pada syptatic gap.



Gambar 4. Sebuah NN dengan neuron tunggal.

1 Impuls adalah rangsangan-rangsangan yang berisi informasi yang disampaikan dari satu neuron ke neuron yang lainnya,
2 Node merupakan istilah neuron dalam ANN.



Gambar 5. Interkoneksi dari neuron-neuron yang menhasilkan dua input.

Berdasarkan pembelajarannya, ANN dibagi menjadi beberapa jenis, yang di makalah ini hanya digunakan dua jenis arsitektur ANN, yaitu *Multi Layer Perceptron* (MLP) dan *Kohonen Self Organizing Map* (SOM).

Kohonen Self Organizing Map (SOM) merupakan salah satu bentuk dari *Unsupervised Learning Methode*, yang diperkenalkan pertama kali oleh Professor Teuvo Kohonen pada tahun 1982. Dalam hal ini, SOM-Kohonen digunakan untuk mengelompokkan (clustering) data berdasarkan karakteristik/fitur-fitur data. Sebuah cluster data terdiri atas sekelompok vektor yang nilainya tidak jauh berbeda. Data yang mirip memiliki topologi yang tidak jauh berbeda. (Erwin, 2005).

3. PEMBAHASAN

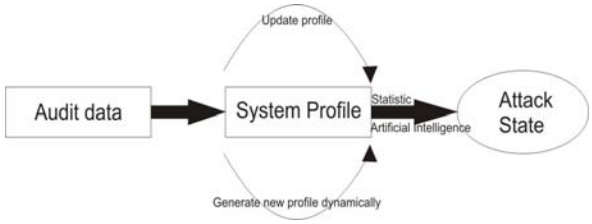
3.1 Metode dan Analisis Desain

Host-based IDS menggunakan informasi yang diberikan oleh sistem operasi untuk mengidentifikasi sebuah serangan. Untuk mendapatkan informasi tersebut, sistem membutuhkan operasional-operasional dasar sebuah sistem operasi, yang disebut sebagai *system call*, sehingga dapat dikatakan bahwa sistem akan berjalan pada level *kernel*. Salah satu metode yang digunakan adalah dengan *OS-level auditing*. Dalam *OS-level auditing*, terdapat banyak teknik, tergantung sistem operasi apa yang digunakan oleh sistem.

Operating System Audit (OS Audit) adalah sebuah mekanisme untuk memperoleh informasi dari aktifitas-aktifitas user dan *aplikasi* (Vigna, 2005; Khuegel, 2005). OS mengawasi setiap aktivitas user dan aplikasinya, berdasarkan pada *system call* yang dijalankan pada *kernel*. *System call* adalah sebuah perantara atau alat-alat khusus yang menghubungkan *user program application*, dengan *kernel*. *System call* merupakan sebuah kepercayaan dari *kernel,* untuk mengakses seluruh *resource* yang terhubung dengan sistem komputer tersebut. Oleh karena itu, banyak *malicious program* yang memanfaatkannya untuk mengakses sistem, seperti *trojan horse*, *virus*, *backdoor* dan sebagainya. Sistem operasi bertugas untuk mengontrol setiap *resource* yang terhubung ke dalamnya. Untuk itulah, mekanisme auditing diimplementasikan pada sistem operasi tersebut, untuk menjaga kestabilan dan inkonsistensi sistem. Pada setiap pengaksesan *resource* tersebut, proses akan diaudit, untuk didapat informasi-informasi resource dan prosesnya.

Anomaly-Based Approach

Teknik IDS, *anomaly-based detection* merupakan teknik pendeteksian intrusi yang didasarkan pada sebuah keanehan dalam sebuah aktivitas-aktivitas rutin dalam sebuah sistem. Teknik pendeteksiannya didasarkan pada model tingkah laku normal yang dilakukan oleh user, yang disebut dengan *profile*. Banyak penyimpangan dari *profile* inilah kemudian diaanggap sebagai sebuah serangan (Vigna & Kruegel, 2005). Penganalisisan profile dapat menggunakan teknik statistika, atau dapat juga dengan menggunakan *artificial intelligence* (Konrad, 2004). Dengan mengidentifikasi sebuah perilaku ubnormal, yang didasarkan atas sebuah tingkah laku normal, maka dapat diketahui bahwa aksi tersebut merupakan sebuah ancaman atau tidak. Dengan cara di atas, maka sebuah ancaman baru, maka dapat diidentifikasi oleh sistem.



Gambar 6. Skema *anomaly-based detection system*

Untuk beberapa sistem, pendekatan yang dilakukan untuk membuat *profile intrusion detection system* adalah dengan *system call*. Untuk membangun *profile*, sistem mempercayakan sebuah model yang berisi ururtan dari *system call* dalam operasi normal. Setiap operasi komputer, melibatkan sejumlah *system call* dalam pengaksesan seperangkat *resource*. Rangkaian *system call* inilah yang dijadikan sebagai model *profile*. Selama proses pengawasan sistem, *profile* inilah yang dijadikan dasar dalam penentuan apakah operasi yang berlangsung merupakan sebuah ancaman atau tidak.

Sebagai contoh, diketahui terdapat sebuah kode program, yaitu:

```
#include <stdio.h>
#include <signal.h>
void sighup();
void sigint();
void sigquit();
main() {
        int pid;
        if ((pid = fork()) < 0) {
                perror("fork");
                exit(1);
        }
        if (pid == 0) {
                signal(SIGHUP, sighup);
                signal(SIGINT, sigint);
                signal(SIGQUIT, sigquit);
                for (;;);
        } else {
                printf("PARENT  : sending
                SIGHUP\n\n");
                kill(pid, SIGHUP);
```

```
                sleep(3);
                printf("PARENT : sending
                SIGINT\n\n");
                kill(pid, SIGINT);
                sleep(3);
                printf("PARENT : sending
                SIGQUIT\n\n");
                kill(pid, SIGQUIT);
                sleep(3);
        }
}
void sighup() {
        signal(SIGHUP, sighup); /
}
void sigint() {
        signal(SIGINT, sigint);
}
void sigquit() {
        exit(0);
}
```

Dari baris program tersebut, dapat dibuat sebuah rangkaian *system call*. Apabila s_n adalah variabel yang akan menyimpan nilai dari setiap *system call*, maka urutan *system call* yang dihasilkan dari baris program di atas adalah sebagai berikut:

s_1	execve()	s_{22}	fstat64()
s_2	brk()	s_{23}	mmap2()
s_3	access()	s_{24}	write()
s_4	mmap2()	s_{25}	kill()
s_5	open()	s_{26}	rt_sigprocmask()
s_6	fstat64()	s_{28}	rt_sigaction()
s_7	mmap2()	s_{29}	rt_sigprocmask()
s_8	close()	s_{30}	nanosleep()
s_9	access()	s_{31}	write()
s_{10}	open()	s_{32}	write()
s_{11}	read()	s_{33}	kill()
s_{12}	fstat64()	s_{34}	rt_sigprocmask()
s_{13}	mmap2()	s_{35}	rt_sigaction()
s_{14}	mmap2()	s_{36}	rt_sigprocmask()
s_{15}	mmap2()	s_{37}	nanosleep()
s_{16}	close()	s_{38}	write()
s_{17}	mmap2()	s_{39}	write()
s_{18}	set_thread_area()	s_{40}	kill()
s_{19}	mprotect()	s_{41}	rt_sigprocmask()
s_{20}	munmap()	s_{42}	rt_sigaction()
s_{21}	close()	s_{43}	rt_sigprocmask()
		s_{44}	nanosleep()

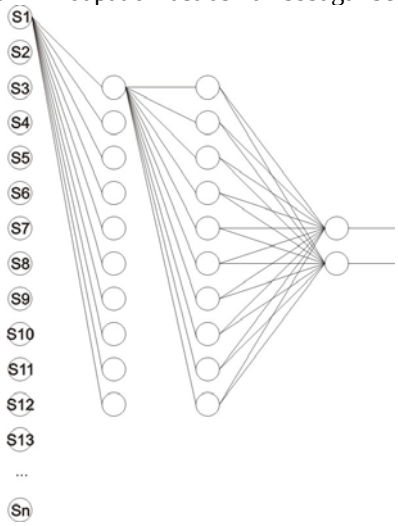
Gambar 7. Urutan *system call* dalam sebuah program yang dijalankan.

Dalam *anomaly-based detection system*, dikenal dua mode, yakni mode pembelajaran dan mode deteksi. Mode pembelajaran merupakan proses pembuatan model yang berisikan urutan *system call* yang syah, yang dijalankan selama eksekusi operasi normal. Mekanisme ini akan memberikan sebuah pola dalam *profile*. Mode deteksi adalah proses pengujian sejumlah input, sehingga akan dihasilkan output, yaitu apakah input tersebut dianggap sebagai sebuah ancaman terhadap sistem atau tidak. Misal *S* adalah proses yang tersusun dari beberapa *system call*, dan s_n adalah *system call*, maka $S_{in} = \{s_1, s_2, s_3, \ldots s_n\}$ merupakan *system call* yang akan disimpan dalam OS. Setiap $s \in S$, maka $s = n$, dimana *n* adalah sebuah nilai *integer*, *length.s*. Apabila $S_{out} = \{s_1, s_2, s_3, \ldots\}$ merupakan pengujiannya maka $(S_{in} \rightarrow S_{out}) \in I$, dimana *I* merupakan bilangan *integer* 0 dan 1.

3.2 Membangun Neural Network

Mode pembelajaran merupakan sebuah mekanisme dalam pengbaharuan *profile*. Dalam mode ini, sistem akan diberikan beberapa input *system call* secara normal, yaitu suatu kondisi sistem tidak mengalami ancaman. Input tersebut dimasukkan ke dalam sebuah mekanisme pembelajaran. *Neural Network* (NN), merupakan sebuah mekanisme pembelajaran yang sangat sesuai dengan metode ini, karena NN mampu mengenali setiap pola baru dari setiap input. Keunggulan lain NN adalah dapat memberikan keputusan berdasarkan *knowledge* yang dimiliki untuk mendeteksi setiap penyimpangan yang terjadi pada input sistem. Dengan digunakan metode ini, diharapkan sebuah ancaman baru akan dapat dideteksi oleh sistem. Model NN yang digunakan untuk pembelajaran adalah dengan *Multi Layar Perceptron* (MLP). MLP dipilih karena mekanisme pembelajaran yang sederhana, dan *error rate* yang dihasilkan tidak besar.

Apabila ditentukan bahwa *S* adalah sebuah proses yang dideteksi dalam sistem, sedangkan *s*, merupakan *system call* penyusun proses tersebut, maka $S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, \ldots s_n\}$ dan NN mempunyai 4 layer, yaitu *input layer*, 2 *hidden layer*, dan 1 *output layer*, maka arsitektur NN dapat diilustrasikan sebagai berikut:

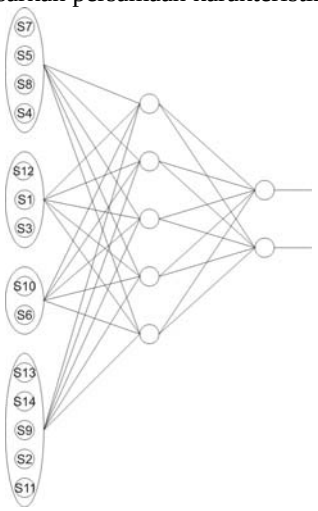


Gambar 8. *Neural Network* dengan 14 input *system call.*

Dari ilustrasi di atas, maka apabila diketahui sebuah proses *S*, maka akan tersusun sebuah NN sebanyak *n(S)* input. Sedangkan node *hidden layer* akan menjadi *n(S)/2* node, dengan 2 *hidden layer* dan 2 node untuk *output layer*. Apabila terdapat bobot W dari node input ke masing-masing node dalam hidden layer, maka akan tersedia $n(W) = (n(S) * n(S)/2)$, atau sebuah himpunan dimana $W = \{w_1, w_2, w_3, \ldots, w_{(n(S) * n(S)/2)}\}$ dan *w float*. *W* akan selalu berubah setiap kali iterasi dilakukan, dan atau setiap kali mode pembelajaran diaktifkan. Dari sebanyak

n(S) node tersebut akan diidentifikasi sebuah pola baru, yang disimpan dan di-*update* dalam *profile*.

Dalam kasus yang lain, sistem ini akan menjadi sangat bermasalah apabila terdapat sebuah operasi yang melibatkan banyak node. Seperti halnya *buffer overflow*, maka operasi ini akan menjadi ancaman sendiri bagi sistem, karena sistem tidak disiapkan untuk menangani operasi-operasi yang membutuhkan *resource* yang cukup banyak. Untuk menangani permasalahan ini, maka digunakan teknik *Self Organizing Map Kohonen (SOM)*. SOM merupakan salah satu teknik dalam NN, yang berguna untuk mengklasifikasikan setiap input berdasarkan persamaan karakteristiknya.



Gambar 9. *Cluster* dari *Neural Network.*

Apabila diinputkan $S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}\}$, maka S akan di klasifikasikan berdasarkan karakteristik yang sama antar satu dengan yang lainnya. Dari proses tersebut maka dihasilkan $S' = \{\{s_7, s_5, s_8, s_4\}, \{s_{12}, s_1, s_3\}, \{s_{10}, s_6\}, \{s_{13}, s_{14}, s_9, s_2, s_{11}\}\}$. Sehingga dapat diketahui bahwa layer input telah berkurang menjadi *n(S')*, dan *hidden layer* dengan masing-masing layer terdapat *n(S')/2* node. Secara proses, mekanisme ini akan mengurangi beban memori yang sangat berarti, sehingga akan menentukan waktu (*t*) proses.

4. KESIMPULAN

Banyak metode telah dikembangkan sebagai sebuah *Intrusion Detection System (IDS)*, khususnya *Host-Based Detection System (HIDS)*. Salah satu metode yang dikembangkan adalah dengan *Artificial Neural Network (ANN)*. ANN dipilih sebagai sebuah metode IDS dikarenakan kemampuannya dalam mengenali setiap pola-pola yang ada, yang dalam hal ini adalah pola dari *system call*. Karena itulah, dengan menggunakan ANN, maka kemungkinan dideteksi sebuah serangan baru dalam sebuah intrusi sangat besar. Tentunya, pengenalan pola serangan baru ini akan cukup membantu *administrator* dalam mengelola sistem komputernya.

Penggunaan ANN juga terdapat kekurangan. Dalam kasus ini, sistem tidak dapat menemukan titik optimal dari setiap pembelajaran yang dilakukan, karena jumlah neuron selalu berubah-ubah dari satu state ke state yang lainnya. Hal ini dapat terselesaikan apabila sistem dapat mencari jumlah hidden layer maksimum, sehingga akan dapat ditentukan bobot yang optimum.Untuk mendapatkan hasil tersebut, maka sistem juga harus dilengkapi dengan *Aritifial Intelligence*, sehingga dapat menentukan berapa jumlah hidden layer yang optimum.

PUSTAKA

[1] Haryono, M.E.A. *Diktat Kuliah Jaringan Syaraf Tiruan.* Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia. Yogyakarta. 2005

[2] De Boer, Pieter; Pels, Martin. "*Host-Based Intrusion Detection System".* http://staff.science.uva.nl/~delaat/snb-2004-2005/p19/report.pdf terakhir diakses 12 Mei 2007

[3] Mitchel, Mark; Oldham, Jeffrey; Samuel, Alex. *Advanced Linux Programming.* New Riders Publishing. 2001.

[4] Palagrini, Chandrika. "*Network-Based Intrusion Detection System".* Departement of Mathematics and Computer Science Rensselaer Polytechnic Institute. 1998. http://www.cs.rpi.edu/~szymansj/theses/pelagrini.ms.02.pdf terakhir diakses tanggal 12 Mei 2007

[5] Rieck, Konrad. "*An Intelligence Host-Based Intrusion Detection System".* Departement of Mathematics and Computer Science Freie Universitat Berlin. 2004. http://ida.first.fhg.de/~rieck/docs/talks/ml-ids-en.pdf terakhir diakses tanggal 7 Mei 2007

[6] Sundaran, Aurobindo. "*An Introduction to Intrusion Detection ".* http://www.acm.org/crossroads/xrds2-4/intrus.html terakhir diakses tanggal 3 Mei 2007

[7] Vigna, Geovannni; Kruegel, Christopher. "*Host-Based Intrusion Detection System".* Technical University Vienna. 2005. http://www.auto.tuwien.ac.at/~chris/research/doc/infsec05_hids.pdf terakhir diakses 3 Mei 2007

[8] _____. "*Self-organizing map*". http://en.wikipedia.org/wiki/Self-organizing_map#Preliminary_definitions terakhir diakses 11 Mei 2007

[9] _____. "*Kohonen's Self-organizing Map (SOM)".* http://www.willamette.edu/~gorr/classes/cs449/Unsupervised/SOM.html terakhir diakses 12 Mei 2007

[10] _____. "*Multi-Layer Perceptron Networks".* http://www.cis.hut.fi/harri/thesis/valpola_thesis/node43.html. terkahir diakses 12 Mei 2007