BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

- Cahyanto, Adi, dkk. 2014. Investigasi Forensika Pada Log Web Server untuk Menemukan Bukti Digital Terkait dengan Serangan Menggunakan Metode Hidden Markov Models. Yogyakarta. Penelitian ini menggunakan data logweb server untuk melakukan identifikasi dan mencari bukti apabila webserver teridentifikasi terserang oleh peretas (hacker).
- Khairulanam. 2011. Sistem Pendeteksi Serangan pada Jaringan Komputer menggunakan Snort berbasis SMS Gateway (studi kasus : Taman Pintar Yogyakarta). Yogyakarta. Pada penelitian ini mendeteksi serangan terhadap web server oleh peretas (hacker) dengan menggunakan snort yang diimplementasikan ke dalam SMS Gatewai, apabila terjadi penyerangan maka administrator dapat notifikasi dari sistem bahwa telah terjadi penyerangan.
- Abdillah, R, 2014. QUICK REPORT FOR HOST BASED INTRUSION DETECTION SYSTEM (HIDS). Vol. 11. Penelitian ini bertujuan memberikan pemberitahuan cepat dengan menggunakan SMS Gatewai. Pemberitahuan ini terjadi apabila adanya identifikasi penyerangan pada web server yang di analisa melalui konsep Host Based IDS.
- M. Halvey, M.T. Keane, and B. Smyth. Time-Based Segmentation of Log Data for User Navigation Prediction in Personalization. 2005, Chiba, Japan. Pada penelitian ini menggunakan file log untuk mengetahui pola perilaku pengguna mobile web yang diukur dari segmentasi waktu. Untuk mendapatkan pola perilaku penelitian ini menggunakan kriteria time, hits, dan ip addres di mana hasilnya akan diperoleh dengan menggunakan markov model.

2.2 Hidden Markov Model (HMM)

2.2.1 Teori Umum

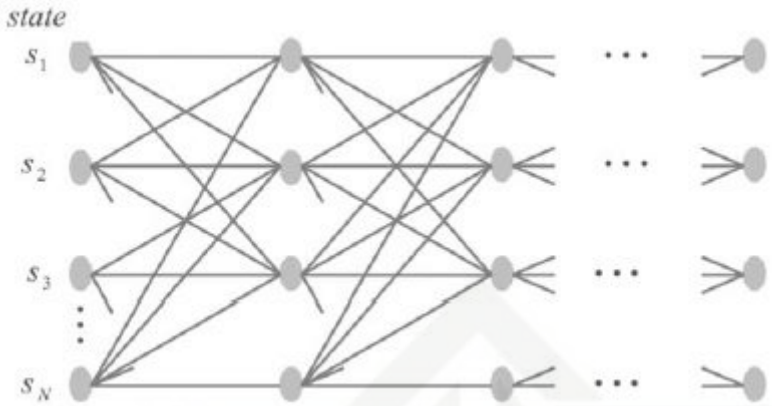
Hidden Markov Model (HMM) pertama kali dikemukakan oleh Leonard. E. Baum pada tahun 1960-an dan telah banyak digunakan dalam menganalisis dan memprediksi suatu permasalahan yang berkaitan dengan waktu. HMM merupakan pengembangan dari Markov chain, dimana setiap state tidak langsung teramati tetapi variables yang terpengaruhi oleh state teramati, yang disebut dengan 'emission'.

Dapat dikatakan HMM adalah model statistika yang memodelkan sistem, sistem yang berbentuk proses Markov dengan parameter yang tidak diketahui, dan HMM digunakan untuk mengatur parameter yang tersembunyi tersebut dari parameter observasi. Model HMM dapat digunakan untuk analisis lebih jauh, seperti untuk analisis pattern recognition applications.

Dalam HMM setiap state memiliki distribusi probabilitas atas simbol-simbol output yang mungkin muncul. Dari rangkaian simbol yang dihasilkan oleh HMM dapat memberikan informasi tentang sekuens atau urutan state.

HMM adalah cara mencari state yang terbatas yang memiliki beberapa jumlah state. Itu memberikan sebuah kerangka peluang untuk pengamatan multivarian pemodelan time series (Hasan & Nath, 2005). HMM terdiri atas sebuah sinyal yang dimodelkan sebagai sebuah rantai Markov keadaanterhingga dan sebuah observasi yang dimodelkan sesuai proses observasi pada rantai Markov, hanya saja proses stokastik pada HMM merupakan proses stokastik ganda dimana salah satu prosesnya tidak dapat diobservasi (hidden).

Jika X ={x1, x2,…..,xT} adalah sebuah proses markov dan O ={O1,O2,…,OT}, adalah sebuah fungsi dari X, maka adalah sebuah HMM yang dapat diobservasi melalui , atau dapat ditulis X =*f*(X) untuk suatu fungsi *f*. Parameter X menyatakan proses parameter-parameter yang tersembunyi, sementara parameter O menyatakan proses parameter-parameter yang diamati. Untuk ilustrasi HMM dapat dilihat dari gambar berikut.



Gambar 2.1 Ilustrasi HMM

Sebuah HMM dikarakteristikkan dengan parameter beriktu (Rabiner, 1989 : 260-261).

1. *N*, jumlah *state* dalam model, dengan ruang *state S = {S1,S2,...,Sn}* dan *state* pada waktu t dinyatakan dengan Xt.
2. *M*, jumlah simbol pengamatan yang dimiliki setiap *state*, dengan ruang observasi O = {O1,O2,…,OT}.

2.2.2 Tipe HMM

Hidden Markov Model dibagi menjadi dua tipe dasar yaitu HMM ergodic dan HMM Kiri Kanan.

- HMM ergodic.

  Pada HMM model ergodic perpindahan keaadaan satu ke keadaan yang lain semuanya memungkinkan.

- HMM kiri-kanan

  Pada HMM kiri-kanan perpindahan keadaan hanya dapat berpindah dari kiri kekanan, perpindahan keadaan tidak dapat mundur ke belakang.

2.2.3 Parameter HMM

$$\bar{a}_{ij} = \frac{\Sigma_k w_k \Sigma_{t+1}^{Tk} \alpha_i^k a_{ij} b_j (O_{t+1}^{(k)} \beta_{t+1}^{(k)}(j)}{\Sigma_k w_k \Sigma_{t=1}^{Tk} \alpha_t^k(i)\, \beta_t^k(i)}$$

$$\bar{b}_{ij} = \frac{\Sigma_k w_k \Sigma_{ot(k)=V_j} \alpha_t^k(i)\beta_t^k(i)}{\Sigma_k w_k \Sigma_{t=1}^{Tk} \alpha_t^k(i)\,\beta_t^k(i)}$$

Dimana A merupakan matrik transisi, a $\in A$.

B adalah matrik observasi, b $\in B$.

π adalah 1 yang merupakan inisial distribusi awal HMM (dipilih secara acak).

$w_k = \frac{1}{P_k}$, $k \in$ [1K] merupakan inverse probabilitas dari perkiraan model urutan training.

Variabel α, β merupakan hasil dari prosedur forward-backward.

2.2.4 Tiga Masalah Utama dalam Hidden Markov Model

Secara umum ada tiga permasalahan yang akan muncul saat menggunakan HMM untuk menyelesaikan permasalahan. Menggunakan akan ada tiga permasalahan, yaitu :

1. Bagaimana menghitung nilai P(O |), yaitu λ: (A,B,π) probabilitas yang dihasilkan dari serangkaian pengamatan O = O1, O2,… , OT.
2. Bagaimana memilih rangkaian state Q = q1, q2, …, qT sehingga dapat mendapatkan rangkaian observasi O = O1, O2, … , OT yang merepresentasikan model λ: (A,B,π).
3. Bagaimana mendapatkan parameter HMM, sehingga nilai P(O |) maksimal.

Permasalahan pertama dapat diselesaikan dengan menggunakan forward algorithm, dan untuk permasalahan ketiga dapat diselesaikan menggunakan algoritma Baum-Welch.

2.3 Jaringan Komputer

Jaringan komputer adalah sekumpulan peralatan atau komputer yang saling dihubungkan untuk berbagi sumber daya. Agar terjadi jaringan antar komputer maka setiap bagian dari jaringan komputer meminta dan memberikan layanan (servis). Pihak yang meminta layanan disebut client dan yang memberi layanan disebut server.

2.3.1 Host Based IDS

Host-based IDS (HIDS) memonitor serangan pada sistem operasi, aplikasi maupun tingkat level kernel. HIDS memiliki akses untuk mengaudit *logs*, pesan error, servis dan hak aplikasi dan sumber yang tersedia dari *host* yang diawasi . Sebagai tambahan sebuah HIDS dapat bekerja sebagai tingkat aplikasi . HIDS memiliki pengetahuan tentang bagaimana data aplikasi bekerja, dan bagaimana pula sebuah data aplikasi yang tidak normal. HIDS dapat mengawasi data aplikasi dalam proses pengkodean dan dimanipulasi oleh aplikasi bersangkutan. Manfaat dari HIDS yakni menikmati hak bebas dari akses terhadap *host*.

HIDS lebih baik dalam menentukan dalam proses tingkat keberhasilan serangan. Trafik yang mencurigakan terlihat mirip dengan trafik normal, karena alasan ini NIDS dibuat karena adanya kesalahan peringatan. Disisi lain, HIDS lebih akurat mendeteksi intrusi yang asli karena HIDS tidak membuat volume yang sama dalam *false positive* sebagaimana NIDS.

HIDS yang memiliki pengaruh istimewa terhadap akses sistem dapat mengawasi spesifik komponen tertentu dari *host* yang tidak dapat dibaca aksesnya dari sistem lain. Komponen tertentu dari Sistem Operasi, seperti berkas kunci di *Unix* dan *registry* di Windows, dapat dilihat dari penggunaan yang mencurigakan. Hal ini tentu saja dapat menjadi resiko yang besar jika tipe komponen ini tersedia oleh NIDS untuk diawasi. HIDS dapat diatur dengan *host.*

HIDS memiliki pengetahuan yang lebih dimana hanya tersedia saja kepada IDS dimana komputer yang sedang diawasi itu saja. Selain itu, HIDS dapat memiliki informasi yang spesifik tentang *host* dan bagaimana tipe aktifitas yang normal untuk itu. Trafik yang dikirim kepada *host* dapat muncul sebagai kondisi yang sangat

normal bagi NIDS, tetapi bagi HIDS mungkin saja dapat dianggap sebagai hal yang tidak normal dan berbahaya. Untuk alasan inilah, HIDS dapat menemukan serangan dimana bagi sebuah NIDS hal ini tidak memungkinkan.

Host based IDS juga memiliki beberapa hal yang merugikan. Karena HIDS berada di tempat dimana komputer yang akan diawasi, maka HIDS akan berdampak pada topologi jaringan secara keseluruhan. HIDS tidak dapat mendeteksi serangan yang dimana HIDS tidak terinstalasi pada komputer yang diserang tersebut. Seorang penyerang dapat melakukan aksi kepada mesin yang tidak memiliki HIDS dan kemudian menggunakan aksesnya kepada mesin yang diproteksi sehingga sebuah HIDS menjadi hal yang kurang bijaksana dalam penggunaanya. Sehingga untuk mengawasi serangan yang terjadi HIDS harus dipasang disetiap *host* yang dianggap kritis. Hal ini tentu saja membutuhkan biaya yang sangat besar tergantung kepada *host* yang kritis yang terdapat didalam suatu organisasi yang terus berkembang. Menjalankan IDS pada tingkat *host* juga memiliki arti bahwa kita harus menyiapkan HIDS yang tersedia untuk berbagai versi sistem operasi yang berbeda dari *host* yang harus kita jaga.

2.4 IDENTIFIKASI DAN KLASIFIKASI WEB LOG

Pengembangan website telah menjadi tuntutan pemiliknya seiring dengan dinamika dan kemajuan teknologi internet. Website yang tidak mempunyai informasi dan tampilan yang menarik dapat menyebabkan kinerja yang tidak bagus. Hal ini berefek pada kerugian bisnis bagi perusahaan yang mengandalkan marketing melalui internet. Web log merupakan data yang dipakai untuk mengetahui kinerja suatu website. Identifikasi dan klasifikasi data web log yang ada pada webserver ini digunakan untuk menganalisis pola akses pengunjung untuk perbaikan keamanan dan kinerja website.

2.4.1 Struktur Web log

Web log adalah file webserver tentang informasi pengunjung suatu website setiap kali pengunjung menggunakan sumber daya dari situs tersebut. Sebagian besar web log menggunakan struktur Combined Log Format. Berikut ini adalah contoh web log dari webserver, yaitu :

110.137.231.82 - - [22/Mar/2010:15:08:07 +0700] "GET /prodi/mice/image/banner-h.gif HTTP/1.1" 200 47000 "http://www.pnj.ac.id/prodi/mice.php" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6"

Keterangan :

• **110.137.231.82** : Remote IP address atau domain name. Sebuah IP address adalah host dengan 32-bit yang sudah ditetapkan oleh Internet Protocol; domain name yang digunakan untuk menentukan Internet Protocol bersifat unik untuk setiap host di internet. IP address biasanya didefinisikan untuk satu nama domain.

• **- -** : Authuser (Username dan password) digunakan jika server memerlukan otentikasi pengguna.

• **[22/Mar/2010:15:08:07 +0700]** : Tanggal dan waktu akses pengunjung.

• **"GET /prodi/mice/image/banner-h.gif HTTP/1.1"** : Mode permintaan: GET, POST atau metoda HEAD dari CGI (Common Gateway Interface).

• **200** : Kode status saat client mengunjungi suatu halaman website, misalnya, 200 adalah "OK" atau 404 adalah "tidak ditemukan".

• **47000** : Kapasitas dokumen yang ditransfer (Bytes).

• **"http://www.pnj.ac.id/prodi/mice.php"** : Baris permintaan yang berasal dari client.

• **"Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6"** : Remote log dan log agent.

Format web log tersebut dapat dimodifikasi agar menampilkan informasi yang benar-benar dibutuhkan oleh administrator server.

2.4.2 Identifikasi Log Web

Aktivitas apa saja yang dilakukan oleh pengguna sistem akan selalu dicatat oleh sistem ke dalam bentuk file log. Banyaknya aktivitas yang dilakukan pengguna sistem akan menyulitkan proses pencarian data tertentu yang terdapat pada log terutama data yang terkait dengan serangan ke log web server.

Identifikasi log web server yakni melakukan pengecekan terhadap integritas log web server. Seorang peretas kemungkinan akan mencoba untuk mengubah data log agar susah dilacak saat proses investigasi forensik. Pada web server apache (httpd), sering kali tidak memanfaatkan mekanisme perlindungan terhadap log file dalam modekonfigurasi default. Oleh karena itu, proses identifikasi logdibuat untuk memeriksa log agar dapat mendeteksi tamper data. Deteksi tamper data terhadap log dibuat dengan menerapkan algoritma Grubbs outlier test.

Untuk pengindentifikasi data log di peroleh dari aplikasi ossec di hids. Deteksi tamper data dibuat dengan menerapkan algoritma Grubbs outlier test. (http://webspace.ship.edu/pgmarr/Geo441/Lectures/OPT%201%20%20Outlier%20Detection.pdf)

$$G = \frac{X_{max} - \bar{X}}{\sigma_n}$$

G merupakan maksimum delay yang ditemukan pada log

X_{max} merupakan nilai maksimum

$\bar{X}$ merupakan rata-rata aritmatika

σ_n merupakan standart deviasi delay waktu request yang tersimpan pada log.

Dalam identifikasi log web server dapat dilakukan dengan menguji keaslian data log web server menggunakan algoritma grubbs outlier test untuk memastikan bahwa data log tersebut tidak mengalami modifikasi dengan melihat timestamp log saat tersimpan di web server dengan melihat timestamp log terakhir diakses.

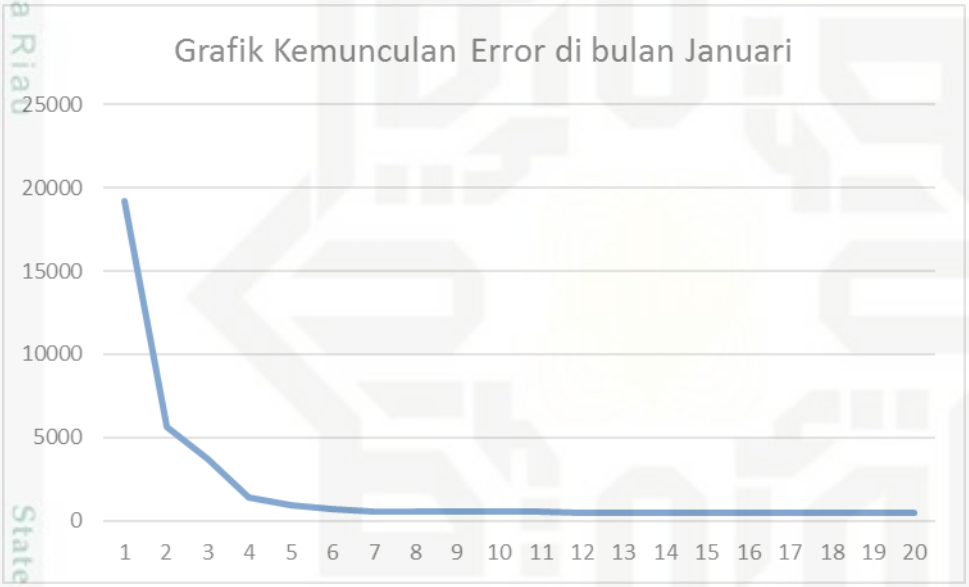
Data		Mean (x)	1935.3
19179	Max	q(n)	4263.381214
5635			

standarisasi	hasil standarisasi	Outlier
4.044606647	4.044606647	outlier
0.867785406	0.867785406	

Ip Address	Date	Time
100.8.135.37	26-Jan-16	06:03:54 +0700
103.10.169.198	26-Jan-	08:47:28

							16	+0700
3705	-	-	0.415093071	0.415093071		103.10.169.198	26-Jan-16	08:43:54 +0700
1386	-	-	-0.128841399	0.128841399		103.10.169.198	26-Jan-16	08:47:28 +0700
930	-	-	-0.235798759	0.235798759		103.10.169.198	27-Jan-16	09:54:51 +0700
672	-	-	-0.296314108	0.296314108		103.10.66.78	29-Jan-16	14:10:55 +0700
580	-	-	-0.317893224	0.317893224		103.10.66.78	29-Jan-16	14:10:59 +0700
535	-	-	-0.328448227	0.328448227		103.19.37.2	27-Jan-16	12:20:01 +0700
521	-	-	-0.331732005	0.331732005		103.22.134.66	25-Jan-16	12:28:15 +0700
518	-	-	-0.332435672	0.332435672		103.22.134.66	25-Jan-16	12:28:27 +0700
515	-	-	-0.333139339	0.333139339		103.224.152.53	25-Jan-16	12:24:43 +0700
510			-0.334312117	0.334312117		103.233.144.107	25-Jan-16	09:51:03 +0700
510			-0.334312117	0.334312117		103.233.144.107	27-Jan-16	10:31:06 +0700
505			-0.335484895	0.335484895		103.233.144.107	28-Jan-16	08:23:33 +0700
505			-0.335484895	0.335484895		103.241.5.196	25-Jan-16	14:45:59 +0700
504			-0.335719451	0.335719451		103.241.5.196	25-Jan-16	14:46:04 +0700
503			-0.335954006	0.335954006		103.55.33.7	26-Jan-16	18:57:51 +0700
50			-	0.33665		103.55.	26-	18:57:5

0	current date	0.336657673	7673		33.7	Jan-16	1 +0700
498		-0.337126785	0.337126785		103.55.33.7	26-Jan-16	18:57:51 +0700
495		-0.337830451	0.337830451		103.55.33.7	26-Jan-16	18:57:51 +0700
					103.55.33.7	28-Jan-16	13:09:19 +0700



2.4.3 Tahapan Persiapan Analisis Data Akses Web Server

Tahapan ini dilakukan dengan menganalisis data akses pada web log untuk mengetahui tingkah laku pengunjung. Proses analisis ini dilakukan dalam beberapa tahap, yaitu raw web log data, data cleaning, user identification, session identification, dan database of clean log. Ada beberapa tahapan untuk menunjukkan proses persiapan data web log,yaitu :

1. Raw Web Log Data

Raw web log merupakan proses penyiapan data web log yang terdapat pada web server. Data yang terdapat pada web log tidak dapat langsung digunakan untuk proses analisis karena banyak terdapat informasi yang tidak relevan untuk mendapatkan pola akses pengunjung web server.

Data yang terdapat pada web log, seperti IP address, date, time, request mode, kode status, Byte, refferer dan user agent digunakan untuk mengidentifikasi user dan session.

2. Data Cleaning

Proses ini dilakukan untuk menyaring informasi yang tidak relevan pada web log asli, sehingga web log hanya menyimpan data yang dibutuhkan saja untuk proses selanjutnya. Informasi yang dapat dibersihkan pada proses penyaringan ini antara lain, request terhadap file multimedia (gambar, icon, animasi, suara dan video), client-side script file, dan cascading style sheet file. Informasi tersebut diabaikan karena merupakan bagian dari suatu request terhadap sebuah halaman web.

3. Identifikasi User

Identifikasi User adalah proses identifikasi setiap user yang mengakses website. Setiap user memiliki IP address yang unik dan masing-masing IP address tersebut merupakan salah satu user.

Identifikasi user sangat rumit dengan adanya cache lokal dan proxy server didalam suatu jaringan. Aturan yang digunakan untuk identifikasi user dalam menanggulangi masalah adanya cache lokal dan proxy server adalah sebagai berikut:

• Jika ada alamat IP yang baru, maka dianggap ada user baru.
• Jika ada alamat IP yang sama, tapi sistem operasi atau browser berbeda, maka dianggap sebagai user baru.

4. Identifikasi Session

dentifikasi suatu session pada web log dilakukan dengan mengelompokkan request-request dari alamat IP yang sama dalam suatu rentang waktu tertentu. Tujuan dari identifikasi session adalah untuk menemukan pola akses setiap pengguna dan halaman yang sering diakses. Metode yang paling sederhana adalah

menggunakan timeout, di mana jika waktu antara permintaan halaman melebihi batas tertentu, dianggap pengguna memulai session baru.

5. Database of clean log

Setelah tahapan raw web log data, data cleaning, identifikasi user, dan identifikasi session, data web log siap digunakan untuk menentukan pola akses pengunjung .

6. Penentuan Pola

Berdasarkan penelitian korzmaz, Turgay dan Yu, Xinran (2015) pola suatu web log diprediksikan berdasarkan banyaknya user (IP Address) mengakses web pada tiap waktu. Dari data tersebut kemudian digambarkan dalam bentuk grafik line untuk menggambarkan signifikan atau tidaknya suatu IP Address mengakses suatu web. Berikut contoh penggambaran tabel dan grafik..

Number of total visitings	Number of paths found
>900	169
>1000	18
>1100	5
>1200	3
>1300	1
>1400	0