ISBN 978-623-93343-2-1
Seminar Nasional Informatika, Sistem Informasi dan Keamanan Siber
(SEINASI-KESI)
Jakarta-Indonesia, 19-20 November 2020

Memprediksi Serangan Pada SIM (Security Information Management) Dengan Menggunakan Algoritma *Hidden Markov Model*

Rico Andreas[1], Henki Bayu Seta[2*], Nurul Chamidah[3]

Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
email: rico@upnvj.ac.id, henkiseta@upnvj.ac.id, nurul.chamidah@upnvj.ac.id
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia

Abstrak

Website merupakan suatu aplikasi yang mudah diakses, dalam kemudahan tersebut terdapat serangan yang dilakukan pada *website*. Serangan-serangan tersebut memiliki ancaman yang terdata di OWASP (*Open Web Application Security Project*) pada tahun 2017 sehingga menciptakan informasi yang ada pada OWASP *Top* 10 *Security* – 2017 yang khusus pada aplikasi *web*. Dengan ancaman tersebut penelitian ini dilakukan untuk membuat sistem yang dapat mendeteksi suatu serangan yang terjadi pada *website* dan dapat menampilkan informasi kegiatan yang ada pada *website* dengan *client*. *Security Information Management* (SIM) akan membaca data *access log* dan *error log* yang telah dicatat oleh *web server* lalu data tersebut akan dilakukan *training* dan *testing* dengan menggunakan algoritma *Hidden Markov Model* sehingga mendapatkan model bagi sistem untuk mendeteksi sebuah serangan, serta *log* akan diterjemahkan menjadi suatu informasi yang mudah dibaca oleh *sysadmin* kedalam suatu *dashboard*. Penelitian ini diharapkan dapat menghasilkan suatu model yang dapat mendeteksi sekaligus memantau kegiatan *website* dalam sebuah serangan.

Kata kunci: *Access log*, *Error log*, *Security Information Management*, SIM, *Hidden Markov Model*

1 PENDAHULUAN

Website merupakan salah satu aplikasi populer bagi pengguna *internet* yang bersifat publik. Namun dikarenakan sifatnya yang publik, *website* sering mengalami serangan. Sehingga, menyebabkan suatu kerusakan pada *web server* yang mengelolah *website* tersebut. *Web server* yang memiliki tingkat keamanan yang lemah selalu menjadi sasaran yang tepat bagi para *attacker* saat menyerang *web server*.

Ancaman-ancaman pada *website* yang terjadi pada tahun 2017 sudah didata oleh OWASP (*Open Web Application Security Project*) dan sudah tercatat pada OWASP Top 10 *Security* – 2017. Pada OWASP *Top* 10 *Security*, terdapat beberapa ancaman dan tingkat resiko dari dampak serangan yang telah diklasifikasikan oleh OWASP. Tingkat ancaman yang diberi nilai sudah dihitung dengan kalkulator khusus dari NIST (*National Institute of Standards and Technology*) yang disebut CVSS (*Common Vunlnerability Scoring System*) dengan rentang *score* serangan yaitu 0.0 (*low*) sampai 10.0 (*critical*) (OWASP, 2017).

Dengan adanya ancaman tersebut, maka seorang *sysadmin* harus melakukan penjagaan yang ketat dalam mengamankan *server* khususnya *web server*. Namun, segala kegiatan pada *web server* khususnya kegiatan penyerangan terhadap *website* telah dicatat pada *log* dari *web server*, macam-macam *log* yang tercatat ialah *access log* dan *error log*.

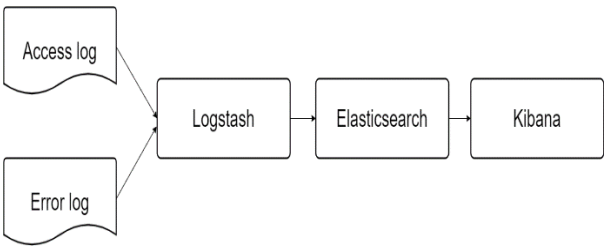
Access log berisi informasi-informasi akses terhadap *website* seperti *IP address client*, waktu *client* mengakses *website*, *request* yang dikirimkan oleh *client*, *web browser* yang digunakan, merekam aktivitas pengguna, melacak upaya otentikasi, serta *file-file* yang ada pada *HTTP Service* (Joshila Grace et al., 2011). *Error log* berisi informasi kegagalan suatu *request* dari *user* / *service* yang ada pada *website*, khususnya yang terjadi pada *web server*. Informasi yang ada pada *error log* ialah waktu *client* mengakses *website*, *IP Address client*, dan kegiatan *error* yang dilakukan *client* (Kabir, 2010). *Log* yang merupakan *record* dari seluruh kegiatan *service* akan berisi sebuah data. Jika *website* tersebut terdapat banyak user yang mengakses, maka data *log* semakin banyak dan *sysadmin* akan sulit membaca *log* tersebut. Sehingga, *sysadmin* tidak dapat mengidentifikasi serangan yang terjadi pada *website*. Oleh karena itu, diperlukan suatu sistem yang memanajemen dan memantau serangan melalui *log* yang tercatat oleh *web server* (Suharjo, 2015). Salah satunya dengan menggunakan metode SIM (*Security Information Management*).

Oleh karena itu, pada penelitian ini dilakukan prediksi sebuah serangan yang ada pada *log* dengan menggunakan algoritma *Hidden Markov Model*. Dengan menggunakan *Hidden Markov Model*, dapat dilakukan *filtering* untung mengurangi data yang bukan serangan pada *log*. Algoritma ini menggunakan *state* yang tidak dapat diamati secara langsung (tersembunyi), tetapi hanya dapat diobservasi melalui suatu himpunan pengamatan lain dengan menggunakan perhitungan statistik (Cahyanto et al., 2014).

2 METODOLOGI PENELITIAN

2.1 SIM (*Security Information System*)

SIM *(Security Information Management*) merupakan sistem yang digunakan sebagai pemantauan suatu sistem lainnya, yang dimana pemantauan tersebut difungsikan untuk melihat sebuah kegiatan yang bersifat keamanan. Sebuah *server* tidak dapat memantau dirinya sendiri sehingga diperlukan suatu sistem yang dapat memantau kegiatan yang ada pada *server*. *Server management and monitoring* diperlukan agar *sysadmin* dapat memantau dengan mudah apa saja yang terjadi pada *server*. SIM mencakup manajemen ancaman, pemantauan *real-time* insiden keamanan dan memicu reaksi yang tepat jika terjadi insiden. Dengan demikian, data yang dikumpulkan disatukan untuk mengurangi jumlah data dan memfasilitasi penggunaan untuk bereaksi dengan tepat terhadap peristiwa keamanan (Vielberth & Pernul, 2018). Pada sistem SIM ini dibuat dengan menggunakan ELK *stack* yaitu terdiri atas *Elasticsearch*, *Logstash*, dan *Kibana*. Alur dari tiga perangkat tersebut digambarkan pada gambar berikut.



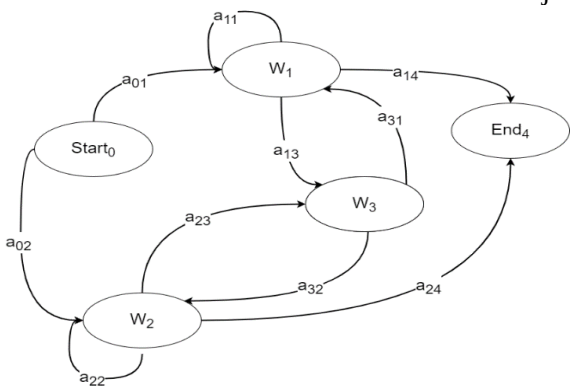
Gambar 1. ELK *Stack*

Penjelasan dari alur pada gambar 3 ialah.

a. *Elasticsearch* adalah mesin pencari yang berkemampuan dalam pencarian dan analisis data secara *real-time*. *Elasticsearch* mempunyai beberapa fitur seperti pencarian multibahasa, *geolocation*, *autocomplete*, JSON dan RESTful API yang memudahkan *Elasticsearch* dalam mengelola data (Advani et al., 2016).
b. *Logstash* membantu dalam membangun jaringan *pipeline* yang dapat memusatkan pengolahan data.. *Logstash* berfungsi untuk memproses *log*, peristiwa, dan data tidak terstruktur (Advani et al., 2016).
c. *Kibana* memvisualisasikan data yang tersimpan pada *cluster Elasticsearch*. *Kibana* menyediakan antarmuka berbasis *browser* yang memudahkan dalam membuat *dashboard* dengan cepat. *Kibana* menyajikan data dalam bentuk histogram, *geomaps*, diagram lingkaran, grafik, tabel, dan lain-lain (Advani et al., 2016).

2.2 *Hidden Markov Model* (HMM)

Hidden Markov Model atau biasa disingkat HMM merupakan pengembangan model statistik dari model *Markov*. Model ini dipandang sebagai proses *bivariate parametric* dalam waktu diskrit. Proses yang terjadi dalam HMM merupakan *finite-state* yang homogen dari *Markov Model* dan tidak dapat diamati. Proses kedua merupakan aliran variabel acak kondisional yang diberikan oleh *Markov Model*. Pada saat apapun, distribusi untuk setiap variabel acak dipengaruhi oleh nilai *Markov Model* pada waktu tersebut saja. Oleh karena itu, HMM merupakan bagian dari statistika parametrik (Eko & Prasetyo, 2011). Pada *Markov Chain*, setiap busur antar *state* berisi probabilitas yang mengindikasikan kemungkinan jalur tersebut akan diambil. Jumlah probabilitas semua busur yang keluar dari sebuah simpul adalah satu. Contoh dari *Markov Chain* ialah ditunjukan pada gambar 2.



Gambar 2. *Markov Chain*

Pada gambar di atas, aij adalah probabilitas transisi dari *state i* ke *state j*. Misalkan, dari simpul *start0* keluar dua kemungkinan, *a02* dan *a01*. Maka jumlah probabilistik *a01* + *a02* adalah satu. Hal ini juga berlaku bagi simpul-simpul yang lain. *Markov Chain* bermanfaat untuk menghitung probabilitas suatu kejadian dapat dirumuskan yaitu.

$$P(\sigma_1)=P(\sigma_t|\sigma_{t-1}, \sigma_{t-2}, \sigma_{t-3}, \ldots) \quad (1)$$

Pada Persamaan 1, σ_1 adalah kondisi saat ini, dan σ_t adalah kondisi pada waktu tertentu yang berhubungan dengan σ_1. Sedangkan σ_{t-1} adalah kondisi sebelum σ_t. Kemudian kita dapat

berasumsi bahwa sebelah kanan persamaan bersifat invariant, yaitu, dihipotesiskan dalam keseluruhan sistem, transisi di antara keadaan tertentu tetap sama dalam hubungan probabilistiknya.
Berdasarkan asumsi tersebut, kita dapat membuat suatu set keadaan probabilistik a_{ij} diantara dua keadaan S_i dan S_j:

$$a_{ij}=P(\sigma_1 =S_i \mid \sigma_{t-1}=S_j), \quad 1 \le i, j \le N \qquad (2)$$

Pada Persamaan 2, karena i dan j dapat sama, maka berlaku batasan berikut:

$$a_{ij} \ge 0 \ \textit{and} \ \sum_{i=1}^{n} a_{ij} = 1 \qquad (3)$$

Suatu HMM dapat dianggap sebagai jaringan *Bayesian* dinamis yang sederhana (*simplest dynamic Bayesian network*).
HMM adalah variasi dari *finite state machine* yang memiliki kondisi tersembunyi Q, suatu nilai output O (observasi), kemungkinan transisi A, kemungkinan output B, sebuah kondisi awal . Kondisi saat ini tidak terobservasi. Tetapi, setiap keadaan menghasilkan output kemungkinan B. biasanya, Q dan O dimengerti, jadi HMM disebut triple A,B.

a. Himpunan *observed state*: $O = o_1, o_2, ..., o_n$.
b. Himpunan *hidden state*: $S = S_1, S_2, ..., S_{n-1}, S_n$.
c. Probabilitas transisi: $A = a_{01}, a_{02}, a_{n1}, ..., a_{nm}.$; a_{ij} adalah probabilitas dari *state* i ke *state* j.
d. Probabilitas emisi atau *observation likelihood*: $B = b_i(O_t)$, merupakan probabilitas observasi O_t dibandingkan oleh *state* i.
e. State awal dan akhir: q_0, q_{end}, yang tidak terkait dengan observasi.

2.3 Algoritma Viterbi

Algoritma *Viterbi* adalah algoritma *dynamic* programming untuk menemukan kemungkinan rangkaian status yang tersembunyi (biasa disebut *Viterbi path*) yang dihasilkan pada rangkaian pengamatan kejadian, terutama dalam lingkup HMM (Irfani et al., 2014).
Untuk menemukan sebuah rangkaian status terbaik, $q = (q_1, q_2, \ldots\ldots, q_r)$, untuk rangkaian observasi $O = (o_1, o_2, \ldots\ldots, o_r)$, perlu didefinisikan kuantitas:

$$\delta_t(i) = \max_{q1, q2, \ldots qt-1} P[q_1, q_2, \ldots, q_{t-1}, q_{t=i}, o_1, o_2, \ldots, o_t \mid \lambda] \qquad (4)$$

Pada Persamaan 4, $\delta_t(i)$ adalah rangkaian terbaik, yaitu dengan kemungkinan terbesar, pada waktu t dimana perhitungan untuk pengamatan t pertama dan berakhir pada status i. Dengan menginduksi, didapat:

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i)_{ij} \right] \cdot b_j(o_{1+1}) \qquad (5)$$

Untuk mendapatkan kembali rangkaian status, perlu adanya penyimpanan hasil yang memaksimalkan Persamaan 5, untuk tiap i dan j, dengan menggunakan tabel $A_r(j)$. Prosedur lengkap untuk menemukan kumpulan status-status terbaik bisa dirumuskan sebagai:

a) Inisialisasi

$$\delta_1(i) = \prod_i b_i(o_1), 1 \le i \ge N \qquad (6)$$

$$A_r(1) = 0 \qquad (7)$$

b) Rekursif

$$\delta_t(i) = \max_{1 \le i \le N} \left[\delta_{t-1}(i)\, a_{ij} \right] b_j(o_t) \qquad (8)$$

$2 \leq t \leq T,\ 1 \leq j \leq N$ (9)

$A_t(j) = \arg\max_{1 \leq i \leq N} [\delta_{t-1}(i)\ a_{ij}]$ (10)

$2 \leq t \leq T,\ 1 \leq j \leq N$ (11)

c) Terminasi

$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$ (12)

$q_T^* = \arg\max_{1 \leq i \leq N} [\delta_T(i)]$ (13)

d) Lintasan status

$q_t^* = A_t(t+1)\ (q_{t+1}^*)$ (14)

$t = T-1, T-2, \ldots, 1.$ (15)

3 HASIL DAN PEMBAHASAN

3.1 *Dataset Access Log* dan *Error Log*

Pada *dataset* ini, dilakukan pengumpulan data yang ada pada UPT TIK Universitas Pembangunan Nasional Veteran Jakarta yang terdapat pada *website* SIAKAD (Sistem Informasi Akademik), data berupa sebuah *log web application* yang dicatat oleh *Apache web server*. Data tersebut ialah *Access log* dan *Error log*, data yang dikumpulkan berawal dari tanggal 08 September 2019 03:42:07 sampai tanggal 22 September 2019 03:03:19 di mana *access log* memiliki 6 variabel dan 1,308,704 baris data, untuk *error log* memiliki 4 variabel dan 250 data yang di mana dataset ini melalui proses integrasi yang di mana data *access log* dan *error log* disatukan sehingga menjadi satu *dataset* utuh lalu dilakukan *labeling* secara manual pada klasifikasi *Attack* dan *Non-attack* berdasarkan pakar yaitu IT *Security Engineer* sebagai *red team* dan IT *Security Auditor* sebagai *blue team* yang telah melalui wawancara.

3.2 *Modeling System*

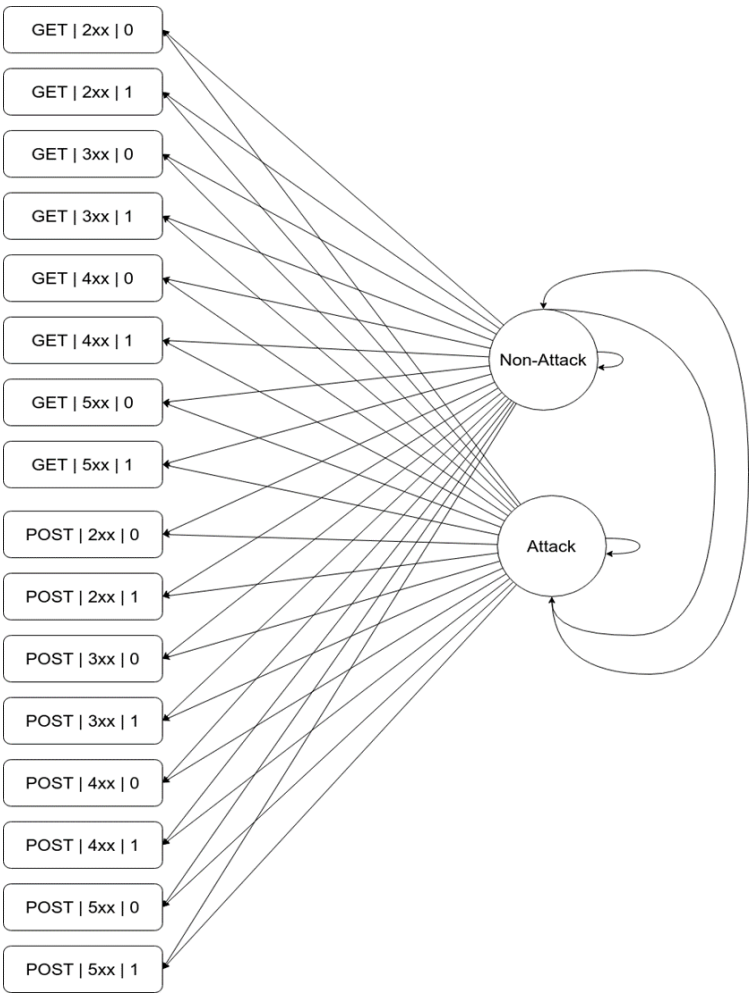
Pemodelan yang ada pada HMM ialah pembentukan *markov chain* yang di mana akan dibentuk *hidden state* untuk dilakukan prediksi dan *observation state* untuk kondisi yang terlihat (yaitu pada datanya). Pada penelitian ini mengambil kombinasi antara variabel tersebut yang di mana value variabel tersebut ialah sebagai berikut.

Method: 1. GET
2. POST

HTTP Status Code:
1. 2xx (*client request successful*)
2. 3xx (*request redirection*)
3. 4xx (*client request incomplete*)
4. 5xx (*server errors*)

Error (status): 1. 0 (*Non Error*)
2. 1 (*Error*)

Dari variabel-variabel tersebut akan membentuk kombinasi sehingga terbentuk suatu *behavior user* dalam mengakses. Dengan kombinasi yang terbentuk akan menjadi satu *state* pada *observation state*. Untuk *hidden state* yaitu kelas yang ada pada *dataset*, hal ini ialah *Attack* dan *Non Attack* yang digunakan sebagai *hidden state*.



Gambar 3. Model HMM Yang Dibentuk Untuk *Dataset*

3.3 Hasil

Implementasi HMM dalam memprediksi serangan dilakukan dengan menggunakan metode *hold-out estimation* dalam pembagian *data train* dan *data test*, lalu dibagi menjadi 3 skenario yaitu skenario 1 dari pembagian *data train* 70% *data test* 30%, lalu skenario 2 yaitu *data train* 80% *data test* 20%, pembagian yang terakhir yaitu *data train* 90% *data test* 10%. Hasil prediksi dari 3 skenario tersebut ditampilkan pada tabel berikut.

Tabel 1. Hasil Prediksi Skenario 1

No.	IP Address	Timestamp	Method	HTTP Status Code	Error	Class	Class predict
0	112.215.201.29	17-09-19 9:33	GET	404	0	Non Attack	Non Attack
1	112.215.201.29	17-09-19 9:33	GET	404	0	Non Attack	Non Attack
2	112.215.201.29	17-09-19 9:33	GET	404	0	Non Attack	Non Attack
3	120.188.38.207	17-09-19 9:33	GET	302	0	Non Attack	Non Attack
4	120.188.38.207	17-09-19 9:33	GET	200	0	Non Attack	Non Attack
5	120.188.38.207	17-09-19 9:33	GET	404	0	Non Attack	Non Attack
6	120.188.38.207	17-09-19 9:33	GET	404	0	Non Attack	Non Attack
7	120.188.38.207	17-09-19 9:33	GET	404	0	Non Attack	Non Attack
8	120.188.38.207	17-09-19 9:33	GET	404	0	Non Attack	Non Attack
9	120.188.38.207	17-09-19 9:33	GET	404	0	Non Attack	Non Attack
10	120.188.38.207	17-09-19 9:33	GET	404	0	Non Attack	Non Attack
..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..
385391	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack
385392	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack
385393	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack
385394	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack

Tabel 2. Hasil Prediksi Skenario 2

No.	IP Address	Timestamp	Method	HTTP Status Code	Error	Class	Class predict
0	182.0.149.37	18-09-19 12:35	GET	200	0	Non Attack	Non Attack
1	223.255.230.236	18-09-19 12:35	GET	200	0	Attack	Non Attack
2	114.124.233.75	18-09-19 12:35	GET	404	0	Non Attack	Non Attack
3	182.0.149.37	18-09-19 12:35	GET	404	0	Non Attack	Non Attack
4	182.0.149.37	18-09-19 12:35	GET	404	0	Non Attack	Non Attack
5	182.0.149.37	18-09-19 12:35	GET	404	0	Non Attack	Non Attack
6	182.0.149.37	18-09-19 12:35	GET	404	0	Non Attack	Non Attack
7	114.124.233.75	18-09-19 12:35	GET	404	0	Non Attack	Non Attack
8	114.124.233.75	18-09-19 12:35	GET	404	0	Non Attack	Non Attack
9	114.124.233.75	18-09-19 12:35	GET	404	0	Non Attack	Non Attack
10	114.124.233.75	18-09-19 12:35	GET	404	0	Non Attack	Non Attack
..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..
256926	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack
256927	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack
256928	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack
256929	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack

Tabel 2. Hasil Prediksi Skenario 3

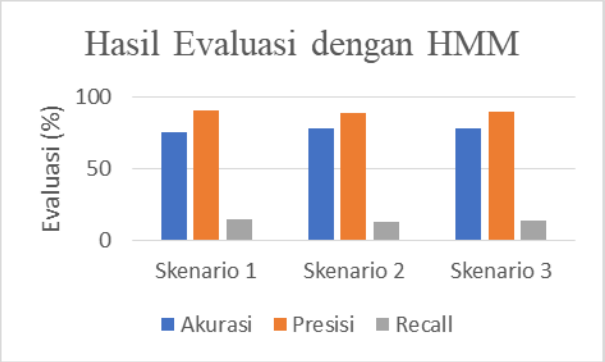
No.	IP Address	Timestamp	Method	HTTP Status Code	Error	Class	Class predict
0	180.243.177.228	19-09-19 22:34	GET	404	0	Non Attack	Non Attack
1	180.243.177.228	19-09-19 22:34	GET	404	0	Non Attack	Non Attack
2	180.243.177.228	19-09-19 22:34	GET	404	0	Non Attack	Non Attack
3	180.243.177.228	19-09-19 22:34	GET	404	0	Non Attack	Non Attack
4	180.243.177.228	19-09-19 22:34	GET	404	0	Non Attack	Non Attack
5	180.243.177.228	19-09-19 22:34	GET	404	0	Non Attack	Non Attack
6	180.243.177.228	19-09-19 22:34	GET	404	0	Non Attack	Non Attack
7	180.243.177.228	19-09-19 22:34	GET	404	0	Non Attack	Non Attack
8	180.243.177.228	19-09-19 22:34	GET	404	0	Non Attack	Non Attack
9	180.243.177.228	19-09-19 22:34	GET	404	0	Non Attack	Non Attack
10	125.161.139.106	19-09-19 22:34	GET	200	0	Non Attack	Non Attack
..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..
128461	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack
128462	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack
128463	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack
128464	36.88.159.47	22-09-19 3:03	GET	404	0	Non Attack	Non Attack

3.4 Evaluasi

Setelah melalui percobaan dengan tiga skenario yaitu dengan pembagian data menggunakan *hold out estimation* dengan skenario 1 yaitu *data train* 70% dan *data test* 30%, lalu skenario 2 yaitu *data train* 80% dan *data test* 20%, terakhir skenario 3 yaitu *data train* 90% dan *data test* 10% akan dilakukan evaluasi terhadap data, sehingga didapat perhitungan akurasi, presisi, dan *recall*.

Tabel 3. Nilai Hasil Akurasi, Presisi, dan *Recall*

Pembagian Data	Akurasi	Presisi	*Recall*
Skenario 1	75.33%	90.02%	14.4%
Skenario 2	77.83%	88.64%	13.13%
Skenario 3	78.28%	89.7%	13.5%



Gambar 4. Diagram Akurasi, Presisi dan Recall HMM Semua Skenario

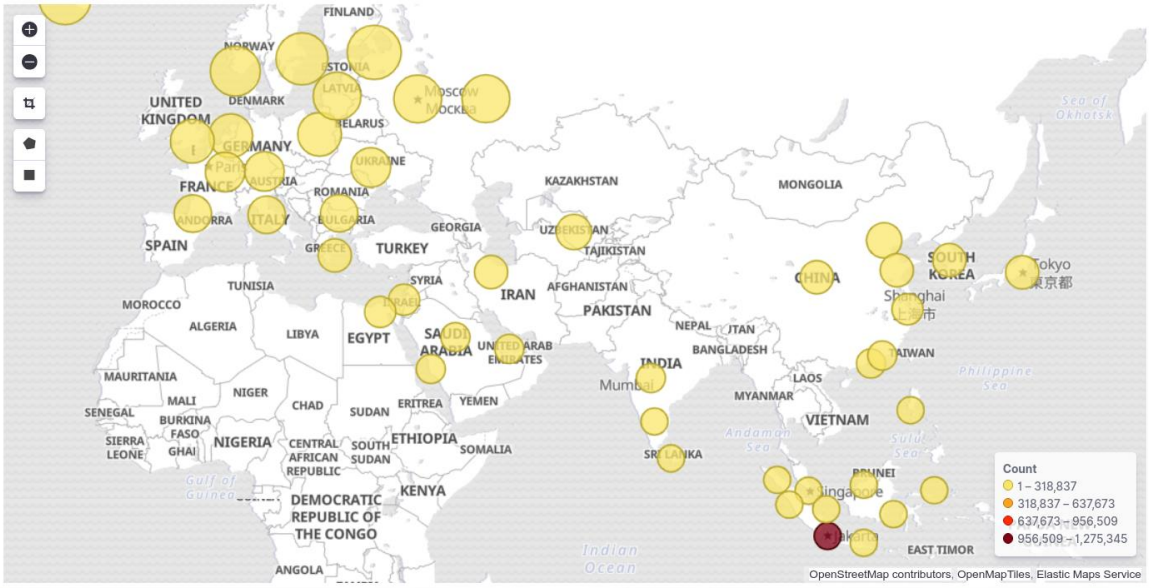
3.5. *Dashboard*

Pada tahap ini, dilakukan pembentukan SIM (*Security Information Management*) dengan menggunakan ELK (*Elasticsearch*, *Logstash*, *Kibana*). *Dataset* dimasukan kedalam ELK sehingga terbentuk suatu *data structural* yang bisa dilakukan pembuatan *dashboard* yang dibutuhkan untuk membuat SIM yang di mana visualisasi yang ditampilkan berdasarkan saran dan wawancara pakar.

Gambar 5. Tampilan *Discover* pada SIM

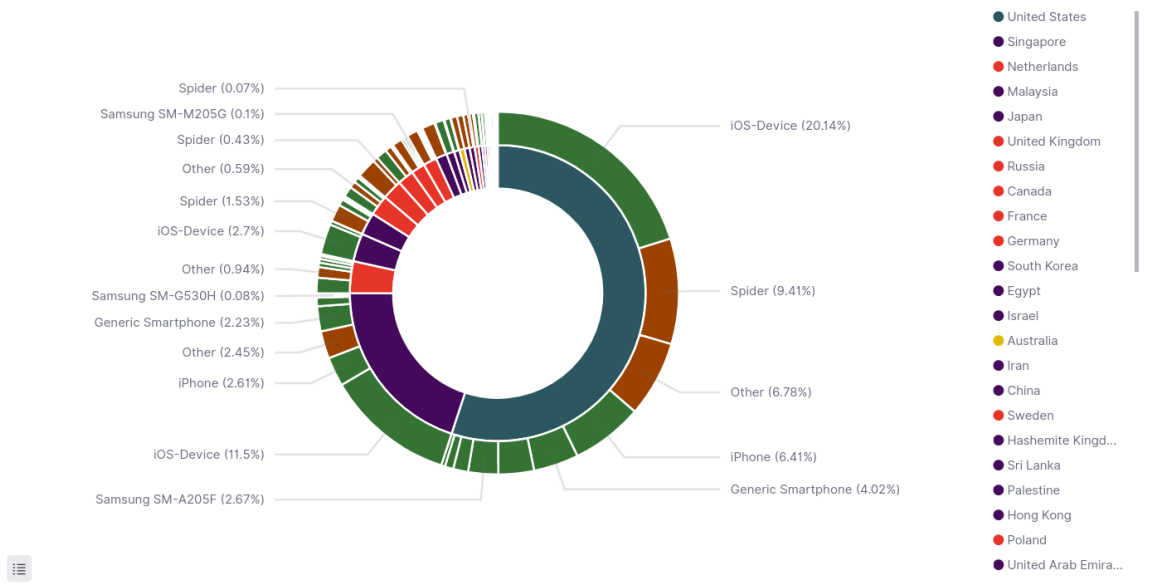
Discover diperlukan untuk memunculkan data yang ingin diperlihatkan atau dibutuhkan dengan cara dilakukan *filtering* dengan memilih variabel yang diperlukan untuk ditampilkan. *Discover* dapat menunjukan atribut clientip untuk *IP Address*, *geoip.city_name* untuk nama kota, *geoip.country_name* untuk nama negara, *request* untuk data yang diakses *user*, *response* yaitu *HTTP Status Code*, *useragent.os* yaitu sistem operasi yang digunakan *client* dalam mengakses SIAKAD UPNVJ, dan *useragent.device* yaitu perangkat yang digunakan client dalam mengakses SIAKAD UPNVJ. Setelah itu dapat dibentuk visualisasi lainnya yang menggambarkan data *log server Apache* pada *website* SIAKAD.

ISBN 978-623-93343-2-1
Seminar Nasional Informatika, Sistem Informasi dan Keamanan Siber
(SEINASI-KESI)
Jakarta-Indonesia, 19-20 November 2020



Gambar 6. Titik Koordinat user dalam mengakses SIAKAD UPNVJ

Geopoint map diperlukan untuk memerlihatkan titik koordinat *user* mana saja yang mengakses SIAKAD dengan warna yang menginterpretasikan banyaknya akses pada suatu titik di mana warna tersebut jika diurutkan berdasarkan terbanyak ialah ungu, merah, jingga, dan terakhir kuning.



Gambar 7. Negara luar Indonesia dengan perangkat mengakses SIAKAD

Pada visualisasi ini dapat memperlihatkan terlihat bahwa terdapat beberapa negara yang menggunakan perangkat yang tidak diketahui seperti menggunakan perangkat dengan label *Other* lalu juga dengan menggunakan perangkat yang bernama *Spyder*, sehingga dapat dianalisis bahwa perangkat-perangkat tersebut memiliki kecurigaan dalam mengakses SIAKAD UPNVJ.

Gambar 8. Dashboard SIAKAD UPNVJ SIM

Pada Gambar 8 terlihat berbagai macam visualisasi yang telah dibuat sebelumnya sehingga terbentuk *dashboard* yang diperlukan dalam pembuatan SIM (*Security Information Management*) untuk keperluan *monitoring* sebuah *website* yang di mana bahwa terdapat elemen *log aggregation*, *dashboard*, *reporting*, lalu untuk *alert* terdapat pada sistem yang terbentuk oleh HMM untuk mendeteksi sebuah serangan.

4 KESIMPULAN

Setelah dilakukan penelitian terhadap prediksi serangan dengan menggunakan HMM dengan data yang digunakan ialah *access log* dan *error log* dari *website* SIAKAD UPNVJ, sehingga dapat disimpulkan ialah hasil evaluasi dari 3 Skenario menunjukan bahwa model dapat memprediksi sebuah serangan (kelas *attack*) di atas 85%, keakuratan dalam memprediksi serangan dan tidak serangan ialah 70%, namun untuk ketepatan model dalam memprediksi kelas sebenarnya pada data. Dari banyaknya kelas serangan pada data sebenarnya, hanya dibawah 15% model dapat memprediksi benar. Sehingga model kurang baik dalam memprediksi serangan dikarenakan adanya *imbalance* pada data. Pada SIM terlihat berbagai macam akses yang mencurigakan seperti akses yang berasal dari luar Indonesia. Lalu terdapat perangkat mencurigakan / perangkat yang tidak umum digunakan oleh *user* dalam mengakses *website* SIAKAD UPNVJ, sehingga dapat dianalisis bahwa terdapat akses yang *anomaly* pada *website*.

Referensi

Aditya Nugroho, P., Saptono, R., & Eko Sulistyo, M. (2016). Perbandingan Metode Probabilistik Naive Bayesian Classifier dan Jaringan Syaraf Tiruan Learning Vector Quantization dalam Kasus Klasifikasi Penyakit Kandungan. *Jurnal Teknologi & Informasi ITSmart*, *2*(2), 21. https://doi.org/10.20961/its.v2i2.628

Advani, S., Mridul, M., Vij, S. R., Agarwal, M., Palak, L., & Sanketa, K. (2016). Log analytics using ELK stack on Cloud platform. *International Journal of Advanced Research in Computer and Communication Engineering*, *5*(4), 50–52. https://doi.org/10.17148/IJARCCE.2016.5413

Cahyanto, T. A., Jember, U. M., Prayudi, Y., & Indonesia, U. I. (2014). *INVESTIGASI FORENSIKA PADA LOG WEB SERVER UNTUK MENEMUKAN BUKTI DIGITAL TERKAIT DENGAN SERANGAN MENGGUNAKAN METODE BUKTI DIGITAL TERKAIT DENGAN SERANGAN MENGGUNAKAN*. *June*.

Eko, M., & Prasetyo, B. (2011). Teori Dasar Hidden Markov Model. *Makalah II2092 Probabilitas Dan Statistik, Sem. I*.

Irfani, A., Amelia, R., & P, D. S. (2014). *Algoritma Viterbi dalam Metode Hidden Markov Models pada Teknologi Speech Recognition*. 1–5.

Joshila Grace, L. K., Maheswari, V., & Nagamalai, D. (2011). Analysis of Web Logs And Web User In Web Mining. *International Journal of Network Security & Its Applications*, *3*(1), 99–110. https://doi.org/10.5121/ijnsa.2011.3107

Kabir, M. J. (2010). Apache Server 2 bible. In *Hungry Minds, Inc.* (Vol. 34, Issue 4). https://doi.org/10.1007/s11013-010-9189-4

OWASP. (2017). The Ten Most Critical Web Application Security Risks Important Notice Request for Comments GM Golden Master. *Top 10 Security - 2017*. https://www.owasp.org/images/0/0a/OWASP_Top_10_2017_GM_%28en%29.pdf

Suharjo, I. (2015). Log Analysis in the User Access on the Web Services Server. *Jurnal AgriSains*, *6*(1), 19–35.

Vielberth, M., & Pernul, G. (2018). A Security Information and Event Management Pattern. *Federal Ministry of Education and Research*, *1*, 1–12.