Seleksi Notifikasi Serangan Berbasis *IDS Snort* Menggunakan Metode *K-Means*

Ahmadi Yuli Ananta

Abstrak–Berkembangnya teknologi saat ini diikuti juga dengan berkembangnya serangan untuk merusak teknologi tersebut khususnya didalam jaringan komputer seperti DOS attack, port scanning, sniffer dll. Oleh karena itu dibutuhkan sebuah sistem yang bisa mendeteksi serangan secara realtime. Penyeleksian notifikasi serangan pada jaringan komputer berbasis IDS snort menggunakan metode k-means adalah sistem yang mendeteksi serangan jaringan berdasarkan data log pada snort dengan cara mengelompokkan data log tersebut menjadi 2 jenis serangan, bahaya dan tidak bahaya. Kemudian sistem ini akan mengirimkan notifikasi berupa SMS lewat aplikasi SMS gateway untuk serangan yang berjenis bahaya ke ponsel administrator jaringan. Sistem ini mendeteksi jenis serangan dengan proses pengelompokkan dari 100 data training yang diambil secara acak dari rules yang ada di snort. Proses pengelompokkan jenis serangan ini menggunakan metode k-means karena metode ini dapat mengelompokkan data dengan ukuran besar dengan cepat. Tentunya metode ini sangat ditentukan oleh pembangkitan centroid awal yang di ambil secara acak.

Kata Kunci—K-Means, cluster, Snort, IDS, SMS Gateway.

I. PENDAHULUAN

Perkembangan teknologi di bidang pendidikan atau pun bisnis dewasa ini mengalami pertumbuhan yang sangat signifikan, seiring laju perkembangan Teknologi Informasi dan Komunikasi global, lembaga yang telah memutuskan untuk memasang perangkat Teknologi Informasi dan Komunikasi (TIK) harus benar mampu untuk mengimplementasikan secara tepat agar bisa meningkatkan laju organisasi agar lebih baik dan mempunyai daya saing tinggi.

Adanya perangkat teknologi yang serba modern atau canggih akan tidak ada artinya tanpa diimbangi oleh pengaturan dan penggunaan secara tepat efektif dan efisien. Semakin berkembangnya teknologi pasti juga diikuti dengan berkembangnya serangan untuk merusak teknologi tersebut. Oleh karena itu dibutuhkan strategi dan pengaturan yang tepat untuk mendapatkan kehandalan jaringan dan bisa menjadi apa yang diinginkan oleh perusahaan. Seorang administrator jaringan bertanggung jawab penuh atas segala sesuatu ketersediaan dan kerahasiaan informasi. Tidak hanya itu, pemeliharaan perangkat keras maupun lunak, menganalisa masalah, memantau kerja jaringan agar bisa selalu tersedia bagi pengguna menjadi aktivitas keseharian dari seorang administrator jaringan. Maka dari itu tugas seorang administrator cukup berat, sehingga dibutuhkan sistem keamanan yang bisa diandalkan agar membatu kerja seorang administrator.

Seorang administrator jaringan juga tidak mungkin akan berada di depan layar komputer 24 jam selama seminggu. Maka dari itu dibutuhkannya notifikasi langsung tentang pendeteksi serangan pada jaringan yang sering dikenal *Intrusion Detection System* (IDS) melalui *Short Message Service* (SMS). Serangan pada jaringan sendiri bermacam-macam, ada serangan yang berbahaya dan serangan yang tidak berbahaya. Maka dari itu penulis mencoba membuat sistem pendeteksi serangan (IDS) yang bisa memberi notifikasi hanya untuk serangan yang berbahaya melalui SMS.

Ahmadi Yuli Ananta adalah pengajar di Program Studi Teknik Informatika Politeknik Negeri Malang; email : ahmadi@polinema.ac.id

II. TINJAUAN PUSTAKA

Intrusion Detection System (IDS)

Intrusion Detection System (IDS) adalah sebuah sistem yang melakukan pengawasan terhadap *traffic* jaringan dan pengawasan terhadap kegiatan-kegiatan yang mencurigakan didalam sebuah sistem jaringan. Jika ditemukan kegiatan-kegiatan yang mencurigakan berhubungan dengan *traffic* jaringan maka IDS akan memberikan peringatan kepada sistem atau administrator jaringan. Dalam banyak kasus IDS juga merespon terhadap *traffic* yang tidak normal/ anomali melalui aksi pemblokiran seorang user atau alamat IP (*Internet Protocol*).

IDS sendiri muncul dengan beberapa jenis dan pendekatan yang berbeda yang intinya berfungsi untuk mendeteksi *traffic* yang mencurigakan didalam sebuah jaringan. IDS ada dua jenis yaitu, yang berbasis jaringan (NIDS) dan berbasis host (HIDS).

Ada IDS yang bekerja dengan cara mendeteksi berdasarkan pada pencarian ciri-ciri khusus dari percobaan yang sering dilakukan. Cara ini hampir sama dengan cara kerja perangkat lunak anti virus dalam mendeteksi dan melindungi sistem terhadap ancaman. Kemudian ada juga IDS yang bekerja dengan cara mendeteksi berdasarkan pada pembandingan pola *traffic* normal yang ada dan kemudian mencari ketidaknormalan traffic yang ada. Ada IDS yang fungsinya hanya sebagai pengawas dan pemberi peringatan ketika terjadi serangan dan ada juga IDS yang bekerja tidak hanya sebagai pengawas dan pemberi peringatan melainkan juga dapat melakukan sebuah kegiatan yang merespon adanya percobaan serangan terhadap sistem jaringan dan komputer[1].

Snort

Salah satu aplikasi Linux yang dapat dipakai untuk meningkatkan keamanan komputer adalah Snort. Secara garis besar, Snort adalah sebuah program yang memiliki tiga fungsi atau tiga modus operasi. Snort dapat dipakai dalam *packet sniffer mode* sehingga bekerja sebagai *sniffer* sama seperti Wireshark. Sama seperti Wireshark, Snort juga dapat menyimpan setiap *packet* yang di-capture ke dalam media penyimpan di modus *packet logger mode*. Akan tetapi berbeda dengan Wireshark, Snort dapat dipakai sebagai komponen NIDS dengan menjalankannya pada *Network Intrusion Detection System* (NIDS) *mode*. Pada modus yang terakhir ini, Snort akan menganalisa *packet* berdasarkan rule yang ada untuk mengenali adanya upaya serangan *hacker* [2].

K-Means

K-means merupakan salah satu metode *clustering* non hirarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih *cluster*. Metode ini mempartisi data ke dalam *cluster* sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam cluster yang lain. Secara umum algoritma dasar dari K-means clustering adalah sebagai berikut:

1. Tentukan jumlah cluster.
2. Alokasikan data ke dalam cluster secara random.
3. Hitung centroid / rata-rata dari data yang ada di masing-masing cluster.
4. Alokasikan masing-masing data ke centroid / rata-rata terdekat.
5. Kembali ke tahap 3, apabila masih ada data yang berpindah cluster atau bila ada perubahan nilai centroid, jika tidak ada data yang berpindah cluster atau ada perubahan nilai centroid maka proses clustering sudah selesai.

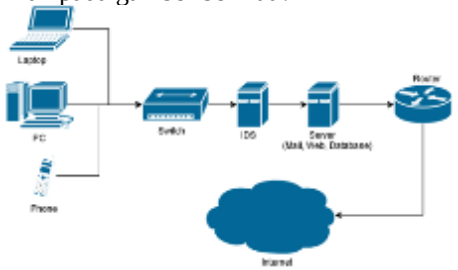
Distance space digunakan untuk menghitung jarak antara data dan centroid. Adapun persamaan yang dapat digunakan salah satunya yaitu *Euclidean Distance Space*. *Euclidean distance space* sering digunakan dalam perhitungan jarak, hal ini dikarenakan hasil yang diperoleh merupakan jarak terpendek antara dua titik yang diperhitungkan[3].

III. METODE PENELITIAN

Perancangan Sistem

1. Desain Arsitektur Jaringan

Desain arsitektur jaringan dari sistem penyeleksian notifikasi serangan pada jaringan komputer ini ditunjukkan pada gamber berikut :



Gambar 1 Desain Arsitektur Jaringan

2. Pendeteksian Serangan Pada Jaringan Komputer Menggunakan Snort

Perangkat lunak Snort merupakan suatu perangkat lunak yang melakukan pendeteksian serangan. Pendeteksian serangan pada snort berdasarkan rule yang ada. Letak rule snort pada sistem ini terdapat di direktori /usr/local/snort/rules. Perintah untuk menjalankan snort sebagai berikut:

```
/usr/local/snort/bin/snort -i wlan0 –c
/usr/local/snort/etc/snort.conf -l
/var/log/snort -g snort -u snort
```

Dengan penjelasan berikut:

/usr/local/snort/bin/snort = direktori letak aplikasi snort

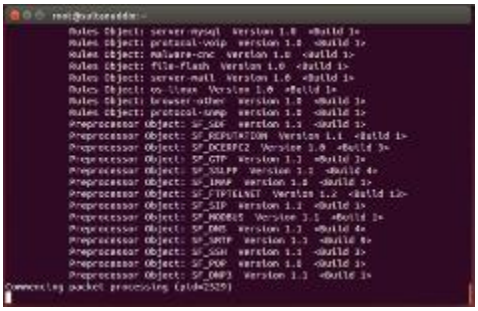
-i wlan0 = interface yang digunakan

-c /usr/local/snort/etc/snort.conf = konfigurasi snort yang digunakan

-l /var/log/snort = letak log snort disimpan

-g snort –u snort = group dan user yang dipakai

Hasil dari running snort ditunjukkan pada Gambar 2.



Gambar 2 *Running Snort*

3. Penulisan Alert Snort ke Database Menggunakan Barnyard2

Barnyard2 berfungsi untuk menuliskan hasil dari log snort ke dalam database. Log yang dibaca disini adalah log yang bertipe unified. Snort akan menulis log bertipe unified dengan snort.u2.xxxxx. perintah untuk menjalankan barnyard2 sebagai berikut:

```
barnyard2 -c
/usr/local/snort/etc/barnyard2.conf -
d /var/log/snort -f snort.u2 -w
/var/log/snort/barnyard2.waldo -g
snort -u snort
```

Dengan penjelasan berikut:

barnyard2 = perintah untuk menjalankan barnyard2

-c /usr/local/snort/etc/barnyard2.conf = letak file konfigurasi barnyard2

-d /var/log/snort = letak direktori log snort

-f snort.u2 = untuk mencari file snort.u2 pada direktori log snort

-w /var/log/snort/barnyard2.waldo = letak file barnyard2.waldo

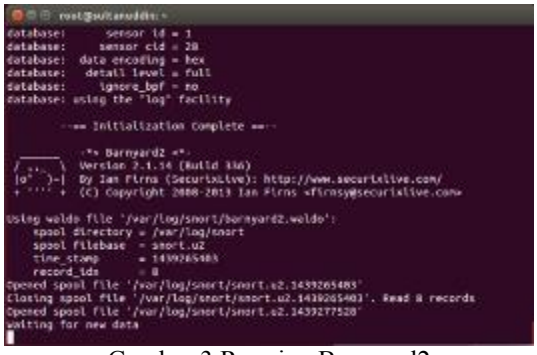
-g snort –u snort = group dan user yang dipakai

Hasil dari running barnyard2 ditunjukkan pada Gambar 3

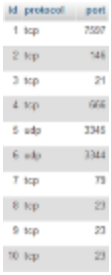
4. Proses Normalisasi Data Training

Untuk proses clustering data dilakukan dengan menggunakan algoritma K-Means. Pada proses clustering ini hanya menggunakan 2 parameter untuk menentukan jenis serangan yaitu data protocol dan

destination port. Parameter tersebut ditunjukkan pada Gambar 4.



Gambar 3 Running Barnyard2



Gambar 4. Data serangan

Agar data serangan tersebut bisa diproses pada algoritma K-Means maka data serangan harus dinormalisasikan terlebih dahulu karena masih mengandung data non-numeric dan terdapat juga nilai yang memiliki range sangat jauh antara nilai terendah dan nilai tertingginya. Proses normalisasi dilakukan dengan menggunakan PHP sesuai aturan sebagai berikut:

Data protocol: TCP = 1, UDP = 2, ICMP = 3

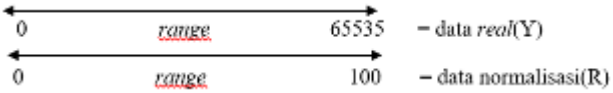
Kode untuk normalisasi protocol sebagai berikut:

```
If ($data['protocol'] ==
'tcp'){$data['protocol'] = 1;}

If ($data['protocol'] ==
'udp'){$data['protocol'] = 2;}

If ($data['protocol'] ==
'icmp'){$data['protocol'] = 3;}
```

Data *destination port*:



d = (X – Ymin) * (Rmax – Rmin) / (Ymax – Ymin) + Rmin

dimana:

d = data hasil normalisasi
X = data yang akan dinormalisasi
Ymin = nilai minimal data real
Ymax = nilai maksimal data real
Rmin = nilai minimal data normalisasi
Rmax = nilai maksimal data normalisasi

Kode untuk normalisasi destination port sebagai berikut:

```
$port1 = ($data['port'] - 0) * (100 -
0) / (65535 - 0) + 0;
```

5. Proses Clustering menggunakan algoritma K-Means

Proses clustering data serangan menggunakan nilai dari normalisasi kedua parameter. Tahap pertama adalah menentukan centroid yang mana ditentukan centroid awal adalah:

```
Centroid Protocol 1 ($centroid1a) = 1
Centroid Port 1 ($centroid1b) =
11.59227893
Centroid Protocol 2 ($centroid2a) = 1
Centroid Port 2 ($centroid2b) =
0.03204395
```

Kode untuk menghitung jarak dengan metode Euclidean Distance Space sebagai berikut:

1. Jarak data dengan centroid 1 (Protocol 1 dan Port 1)

```
$jarak1 = sqrt(pow(($protocol-
$centroid1a),2)+pow(($port-
$centroid1b),2));
```

2. Jarak data dengan centroid 2 (Protocol 2 dan Port 2)

```
$jarak2 = sqrt(pow(($protocol-
$centroid2a),2)+pow(($port-
$centroid2b),2));
```

Setelah menghitung jarak maka akan diperoleh jarak serangan terhadap masing-masing centroid. Tahap selanjutnya adalah mengelompokkan dan pelabelan data serangan sesuai dengan cluster-nya, yaitu dengan cara memilih jarak centroid terpendek. Jika jarak data serangan dengan centroid 1 yang terpendek, maka data tersebut diberi label "bahaya". Apabila jarak dengan centroid 2 yang terpendek, maka data tersebut diberi label "tidak bahaya".

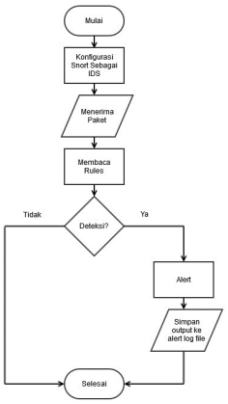
Tahap selanjutnya setelah pengelompokkan dan pelabelan data adalah menghitung centroid baru.

***Flowchart* Sistem**

Flowchart merupakan serangkaian bagian-bagian yang berfungsi untuk menerangkan alur dari jalannya program. Dalam sistem penyeleksian notifikasi serangan ini empat flowchart, yaitu proses pendeteksian serangan, proses penulisan log serangan ke database, proses clustering k-means dan proses pengiriman notifikasi melalui SMS Gateway.

1. *Flowchart* Pendeteksian Serangan

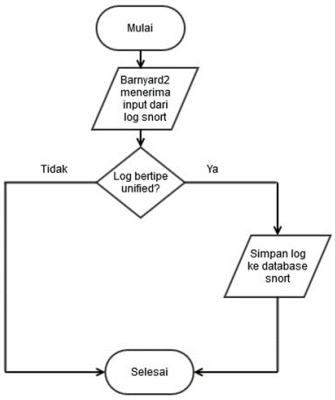
Flowchart ini menerangkan tentang proses pendeteksian serangan menggunakan IDS snort.



Gambar 5 Flowchart Deteksi Serangan

2. *Flowchart* Penulisan *Log* Serangan ke *Database*

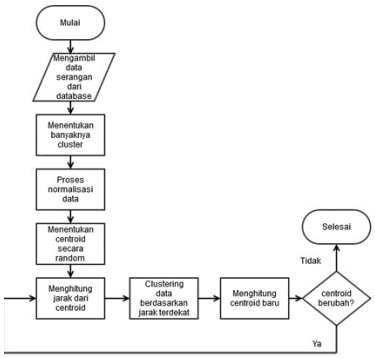
Flowchart ini menerangkan tentang proses penulisan log serangan ke database menggunakan Barnyard2.



Gambar 6 Flowchart Penyimpanan Log ke Database

3. *Flowchart* Proses Clustering *K-means*

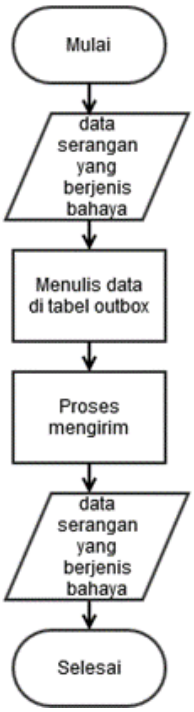
Flowchart ini menerangkan tentang proses clustering k-means menggunakan kode PHP.



Gambar 7 Flowchart Proses Clustering K-Means

4. *Flowchart* Proses Mengirim SMS

Flowchart ini menerangkan tentang proses mengirim SMS dengan *SMS Gateway*.



Gambar 8 Flowchart Proses SMS

IV. HASIL DAN PEMBAHASAN

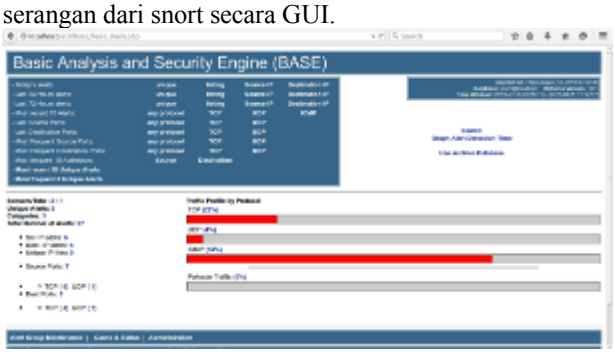
Bab ini merupakan hasil dan pembahasan sistem seleksi notifikasi serangan berbasis ids snort menggunakan metode k-means. Proses melakukan identifikasi dilakukan dengan melaksanakan penelitian dan observasi pada sistem yang sedang berjalan. Hasil penelitian-penelitian tersebut dipergunakan untuk masukan ke dalam sistem yang akan dikembangkan.

Tampilan Antarmuka

Antar muka pengguna digunakan untuk memudahkan *user* dalam melakukan operasi perhitungan. Pada antar muka ini akan dijabarkan di beberapa bagian yang akan digunakan oleh *user*.

1. Halaman ACIDBASE

ACIDBASE adalah halaman untuk melihat isi log serangan dari snort secara GUI.



Gambar 9 Halaman ACID

2. Halaman Pengiriman SMS

Halaman ini berfungsi untuk proses cluster dan pengiriman sms yang berlabel bahaya.

Gambar 10 Halaman Pengiriman SMS

3. Notifikasi SMS

Notifikasi SMS yang di terima oleh administrator jaringan.

Gambar 11 Notifikasi SMS

Implementasi Sistem

Pada tahap ini akan dilakukan implementasi sistem penyeleksian notifikasi serangan pada jaringan komputer berbasis IDS snort dengan Metode K-means. Di bawah ini merupakan langkah-langkah penggunaan aplikasi:

Proses Deteksi Serangan

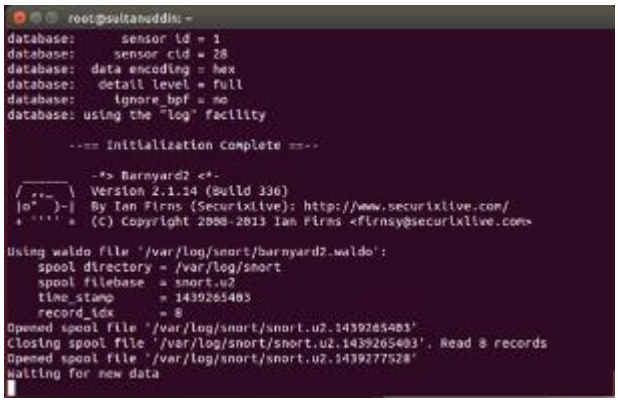
1. Jalankan Snort dengan perintah `/usr/local/snort/bin/snort -i wlan0 -c /usr/local/snort/etc/snort.conf -l /var/log/snort -g snort -u snort`. Jika snort berhasil dijalankan akan muncul "Commencing packet processing (pid=xxxx)" seperti Gambar 12

Gambar 12 Snort Berhasil Dijalankan

2. Dalam mendeteksi serangan snort membutuhkan rules. Direktori rules pada sistem ini ada di /usr/local/snort/rules/community.rules.
3. Log serangan yang dideteksi oleh snort akan disimpan dalam direktori /var/log/snort.

Proses Penulisan Log serangan ke Database

1. Jalankan Barnyard2 dengan perintah `barnyard2 -c /usr/local/snort/etc/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo -g snort -u snort`. Jika barnyard2 berhasil dijalankan akan muncul "Waiting for new data".



Gambar 13 Barnyard2 Berhasil Dijalankan

Proses Pengaktifan Daemon SMS Gateway

1. Cara mengaktifkan daemon untuk gammu SMS Gateway dengan memberikan perintah `/etc/init.d/gammu-smsd start`.
2. Cek ulang untuk memastikan daemon gammu bekerja dengan perintah /etc/init.d/gammu-smsd status. Apabila terdapat tulisan [OK] maka daemon gammu sudah berjalan.

Proses Cluster Data Serangan dan Mengirim Notifikasi SMS

1. Sebelum membuka website untuk proses cluster dan mengirim notifikasi SMS cek dulu daemon dari apache2 dengan perintah /etc/init.d/apache2 status. Apabila terdapat tulisan [OK] berarti daemon apache2 sudah berjalan
2. Untuk membuka website proses cluster dan mengirim notifikasi SMS, buka link localhost/sms/index.php pada browser.

Ujicoba Sistem

Setelah melakukan proses pendeteksian serangan, penulisan log ke database, cluster data training, clustering data serangan dan pengiriman notifikasi SMS, maka untuk melihat apakah hasil dari proses-proses diatas telah berhasil seperti yang diinginkan akan dilakukan pengujian.

Untuk tahap pengujian sistem menggunakan metode blackbox. Metode ini memungkinkan adanya pengembangan untuk melatih seluruh fungsi pada sistem. Metode ini digunakan untuk mendemonstrasikan jalannya aplikasi dan menemukan kesalahan saat aplikasi dijalankan. Dengan menggunakan metode ini dapat dinilai apakah input yang diterima dan output yang dihasilkan sudah tepat atau belum. Berikut hasil dari pengujian sistem:

Tabel 1 Pengujian Sistem

Pengujian Sistem				
No	Skenario Pengujian	Hasil yang diharapkan	Hasil pengujian	Keterangan
1	Pendeteksian serangan dengan IDS Snort	Terdeteksi dan menulis *log*	Berhasil	Hasil pendeteksian tergantung pada penulisan *rules* snort
2	Penulisan *log* snort ke *database* snort menggunakan Barnyard2	Tabel acid_event berisi data yang ada di *log* snort	Berhasil	*Log* yang ditulis hanya yang bertipe *unified* (snort.u2)
3	Membuka ACID BASE lewat browser	ACID menampilkan data serangan dari database	Berhasil	Data serangan yang ditampilkan adalah hasil dari penulisan dari Barnyard2
4	Implementasi metode K-Means dengan 100 data *training*	Tidak ada perubahan pada nilai *centroid*	Berhasil	Proses perhitungan nilai *centroid* dilakukan 2 kali
5	Implementasi pada data serangan *real*	Pengelompokkan jenis serangan bahaya dan tidak	Berhasil	*Centroid* yang digunakan adalah *centroid* dari hasil 100 data *training*
6	Pengiriman notifikasi serangan dengan SMS Gateway	Mengirimkan serangan yang berjenis bahaya ke tabel outbox di *database*	Berhasil	Serangan yang berlabel tidak bahaya tidak akan diproses

Pada Tabel 1 dapat dilihat hasil pengujian sistem menggunakan blackbox. Berdasarkan dari hasil pengujian sistem menggunakan blackbox, maka dapat ditarik kesimpulan bahwa sistem penyeleksian notifikasi pada jaringan komputer berbasis IDS snort dengan Metode K-Means sudah berjalan sesuai dengan harapan.

Pembahasan

Pada bagian ini dibahas mengenai hasil percobaan metode pengklasteran dengan algoritma k-means yang diimplementasikan pada pengelompokan data log sebanyak dua kelompok yang sudah ditentukan sebelumnya. Hasil percobaan metode pengklasteran ini dilakukan pada 100 data serangan random yang ada di rules snort. Hasil percobaan dapat disajikan seperti berikut:

1. Pembahasan Cluster Centroid Awal

Percobaan ini digunakan untuk mengetahui hasil klaster pertama dari 100 data training yang telah diambil secara

random dari rules yang ada di snort. Pada percobaan ini centroid dibangkitkan dengan random. Hasil dari pengklasteran data training ditunjukkan pada Gambar 14

K-Means

No	Protocol	Port	Jarak Cluster 1	Jarak CLuster 2	Label
1	1	11.59227893	0	11.56023498	bahaya
2	1	0.22278172	11.36949721	0.19073777	tidak bahaya
3	1	0.03204395	11.56023498	0	tidak bahaya
4	1	1.01625086	10.57602807	0.98420691	tidak bahaya
5	2	5.10414282	6.5647475337507	5.1697376091109	tidak bahaya
6	2	5.10261692	6.5662556304211	5.1682405365941	tidak bahaya
7	1	0.12054627	11.47173266	0.08850232	tidak bahaya
8	1	0.03509575	11.55718318	0.0030518	tidak bahaya
9	1	0.03509575	11.55718318	0.0030518	tidak bahaya
10	1	0.03509575	11.55718318	0.0030518	tidak bahaya
11	1	0.03509575	11.55718318	0.0030518	tidak bahaya
12	1	0.03509575	11.55718318	0.0030518	tidak bahaya
13	1	0.03509575	11.55718318	0.0030518	tidak bahaya
14	1	0.03509575	11.55718318	0.0030518	tidak bahaya
15	1	0.03509575	11.55718318	0.0030518	tidak bahaya
16	1	0.03509575	11.55718318	0.0030518	tidak bahaya
17	1	0.03509575	11.55718318	0.0030518	tidak bahaya
18	1	0.03509575	11.55718318	0.0030518	tidak bahaya
19	1	0.03509575	11.55718318	0.0030518	tidak bahaya
20	2	47.81414511	36.235667367413	47.792564183819	bahaya
21	2	47.81414511	36.235667367413	47.792564183819	bahaya
22	1	42.21408408	30.62180515	42.18204013	bahaya
23	1	42.21408408	30.62180515	42.18204013	bahaya
24	1	42.21408408	30.62180515	42.18204013	bahaya
25	2	41.87685969	30.301086314008	41.856762945963	bahaya

Gambar 14 Running Cluster Awal

Hasil centroid yang di dapat pada percobaan ini berbeda dari centroid sebelumnya, maka dari itu percobaan akan dilanjutkan ke percobaan kedua dengan nilai centroid yang baru. Hasil dari centroid baru yang didapat ditunjukkan pada Gambar 15.

Centroid 1	Centroid 2
1	11.59227893
1	0.03204395

Continue

New Centroid 1	New Centroid 2
1.42857143	23.49940235
1.13924051	0.33071516

Gambar 15 Centroid Awal dan Centroid Baru

2. Pembahasan Hasil Centroid Baru

Percobaan kedua ini digunakan untuk mengetahui hasil klaster kedua dari 100 data training sebelumnya dimana centroid di ambil dari nilai rata-rata dari tiap kelompok. Hasil dari pengklasteran data training ditunjukkan pada Gambar 16.

Hasil centroid yang di dapat pada percobaan kedua ini sama dari centroid sebelumnya. Jadi untuk menentukan hasil klaster untuk serangan realtime pada log snort akan menggunakan nilai centroid ini. Hasil dari centroid baru yang didapat ditunjukkan pada Gambar 17.

K-Means

No	Protocol	Port	Jarak Cluster 1	Jarak CLuster 2	Label
1	1	11.59227893	11.914833679482	11.262424537612	tidak bahaya
2	1	0.22278172	23.28056574535	0.17617476293526	tidak bahaya
3	1	0.03204395	23.471271455647	0.32953362697595	tidak bahaya
4	1	1.01625086	22.487235810408	0.69953349855425	tidak bahaya
5	2	5.10414282	18.404132790943	4.8504142632231	tidak bahaya
6	2	5.10261692	18.405657955316	4.8489125901295	tidak bahaya
7	1	0.12054627	23.382783946313	0.25210886923885	tidak bahaya
8	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
9	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
10	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
11	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
12	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
13	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
14	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
15	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
16	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
17	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
18	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
19	1	0.03509575	23.468220164499	0.32677018712515	tidak bahaya
20	2	47.81414511	24.32145649618	47.491231050755	bahaya
21	2	47.81414511	24.32145649618	47.491231050755	bahaya
22	1	42.21408408	18.719588289431	41.883600370652	bahaya
23	1	42.21408408	18.719588289431	41.883600370652	bahaya
24	1	42.21408408	18.719588289431	41.883600370652	bahaya
25	2	41.87685969	18.386339197285	41.555060347808	bahaya

Gambar 16 Running Cluster Kedua

Centroid 1	Centroid 2
1.42857143	23.49940235
1.13924051	0.33071516

Continue

New Centroid 1	New Centroid 2
1.42857143	23.49940235
1.13924051	0.33071516

Gambar 17 Centroid Awal dan Baru Tidak Berubah

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Sistem telah dikembangkan dan telah diujicoba dengan hasil berjalan dengan baik, secara garis besar dapat ditarik beberapa kesimpulan sebagai berikut:

a. Hasil pengujian menunjukkan bahwa sistem keamanan jaringan bisa
b. Serangan dapat terdeteksi atau tidak tergantung dari pola data serangan tersebut ada di dalam rule snort atau tidak.
c. Hasil pengujian menunjukkan bahwa sistem bisa mengelompokkan jenis serangan menjadi dua kelompok, bahaya dan tidak bahaya.
d. Notifikasi SMS berhasil dikirim secara realtime dengan menggunakan PHP.
e. Hasil pengujian menunjukkan bahwa SMS gateway berhasil mengirimkan data serangan yang berlabel “bahaya” saja.

B. Saran

Pada penelitian ini sistem menghitung normalisasi data, menentukan centroid, menghitung jarak, menghitung centroid baru dan mengirim SMS dengan Bahasa PHP, agar perhitungan menjadi lebih efisien dapat dilakukan penelitian lanjutan dengan menggunakan platform Java dalam penentuan centroid, normalisasi data, menghitung jarak dan mengirimi SMS.

DAFTAR PUSTAKA

[1] A. R. Baker *et al.*, *Snort 2.1 Intrusion Detection*, Second Edi. Massachusetts: Syngress Publishing, Inc, 2004.

[2] C. Scott, P. Wolfe, and B. Hayes, *Snort? For Dummies*. Indiana: Willey Publishing, Inc, 2004.

[3] Y. Agusta, "K-Means–Penerapan, Permasalahan dan Metode Terkait," *J. Sist. dan Inform.*, vol. 3, no. 1, pp. 47–60, 2007.

[4] R.U. Rehman, *Intrusion detection systems with Snort: advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID*. Prentice Hall Professional, 2003.

[5] A. Ramadhika, *SMS Gateway Menggunakan Gammu Dan MySQL*, 2012. [Online] Tersedia: http://www.ubaya.ac.id/ubaya/articles_detail/33/SMS-Gateway-menggunakan-Gammu-dan-MySQL.html [25 Januari 2015]

[6] J.E. Kendall, & K.E. Kendall, *Analisis dan Perancangan Sistem*. Jakarta: Indeks, 2010.

[7] A. Kadir, *Dasar Pemrograman Web Dinamis Menggunakan PHP*. Yogyakarta: Andi Offset, 2008.

[8] A. Kadir, *Mudah Mempelajari Database MySQL*. Yogyakarta: Andi Offset, 2010.

[9] A. S. Putra, *Apache Web Server*. Yogyakarta: Andi Offset, 2003.

[10] R. Danyliw, *Analysis Console for Intrusion Databases*. 2003. [Online] Tersedia: http://www.andrew.cmu.edu/user/rdanyliw/snort/snortacid.html [14 April 2015]

[11] R. S. Pressman, *Software Engineering: A Practition's Approach*. New York: McGraw-Hill, 2005.