Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 2, No. 9, September 2018, hlm. 3112-3123 http://j-ptiik.ub.ac.id

Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode *Naïve Bayes* Dan *Support Vector Machine* (SVM)

Mercury Fluorida Fibrianda[1], Adhitya Bhawiyuga[2]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]mercuryfluorida@gmail.com, [2]bhawiyuga@ub.ac.id

Abstrak

Serangan *Denial of Service* (DoS) merupakan suatu tindakan untuk melumpuhkan server komputer pada jaringan internet sehingga komputer tidak dapat menjalankan fungsinya dengan benar. Untuk melakukan pendeteksian atau pencegahan berbagai potensi serangan telah dikembangkan *Intrusion Detection System (IDS)*. IDS memiliki dua metode dalam melakukan pendeteksian yaitu *Rule Based (Signature Based)* dan *Behavior Based.* Dalam penelitian ini digunakan metode *behavior based* dimana dalam proses kerjanya membutuhkan sebuah dataset dan metode. Tetapi tidak semua algoritma data mining memiliki kinerja yang baik dalam mengklasifikasi jenis serangan. Oleh karena itu, penelitian ini melakukan perbandingan beberapa algoritma yaitu *Naïve Bayes*, SVM *Linear*, SVM *Polynomial*, dan SVM *Sigmoid*. Dataset yang digunakan dalam penelitian ini adalah dataset dari ISCX2012 *testbed* tanggal 14 Juni 2012. Penelitian ini menganalisis perbandingan metode yang dihasilkan dari proses klasifikasi berdasarkan nilai akurasi *confusion matrix, precision*, *recall*, dan *f1 score. Naive Bayes*, SVM *Linear*, SVM *Polynomial* dan SVM *Sigmoid* menghasilkan persentase akurasi berturut-turut sebesar 85,055%, 99,995%, 99,999%, dan 99,995%. Persentase akurasi tertinggi diperoleh SVM *Polynomial*, sedangkan *Naive Bayes* menghasilkan persentase akurasi terendah.

Kata kunci: Serangan, DoS, IDS, Klasifikasi, *Naive Bayes*, SVM

Abstract

Denial of Service (DoS) attacks is an action to cripple the server computers on the internet so that the computer cannot perform its function properly. To perform the detection or prevention of a wide range of potential attacks, the solution which has been developed is Intrusion Detection System (IDS). IDS has two methods in doing detection that is Rule Based (Signature Based) and Behavior Based. In this study we use the methods of behavior based where in the process of its works require a dataset and methods. The methods that can be used, one of which is the classification of data mining techniques. But not all data mining algorithm has good performance in classifying the type of attack. Therefore, this research did a comparison of several methods i.e. Naïve Bayes, SVM Linear, SVM Polynomial, and SVM Sigmoid. The dataset used in this research is the dataset of a testbed ISCX June 14th, 2012. This research analyzes the comparative method which are resulted from a process of classification such as confusion matrix which resulting into the value of accuracy, precision, recall, and the f1 score. Naive Bayes, SVM Linear, SVM Polynomial, and SVM Sigmoid produce consecutive accuracy with the percentage of 99.995%, 85.055%, 99.999%, and 99.995%. The highest percentage of accuracy obtained by SVM Polynomial, while the Naive Bayes generates the lowest accuracy percentage.

Keywords: *Attacks, DoS, IDS, Classification, Naive Bayes, SVM*

1. PENDAHULUAN

Keamanan jaringan merupakan aspek penting dalam bidang teknologi informasi saat ini. Semakin banyak pengguna dan semakin luas jangkauan komunikasi, maka semakin banyak pula peluang serangan. Sebagai gambaran, pada survey yang dilakukan oleh Lab Kaspersky 2017, 33% organisasi mengalami serangan *DDoS* pada tahun 2017, dibandingkan dengan 17% di tahun 2016. Dari organisasi yang terkena serangan *DDoS*, 20% adalah bisnis yang sangat kecil, 33% adalah UKM, dan 41% adalah perusahaan. Serangan atau *Intrusion* dapat diartikan sebagai sebuah tindakan untuk

memasuki wilayah atau akses yang tidak sah yang dapat membahayakan perangkat jaringan lainnya. (William Stallings, 2005).

Untuk melakukan pendeteksian atau pencegahan berbagai potensi serangan telah dikembangkan suatu sistem atau metode yang dikenal dengan *Intrusion Detection System (IDS)*. IDS adalah sebuah perangkat lunak atau perangkat keras yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan (Monika Kusumawati, 2010). IDS memiliki dua metode dalam melakukan pendeteksian yaitu *Rule Based (Signature Based) dan Behavior Based.* Pendeteksian berbasis *Signature Based* dilakukan dengan mencocokkan lalu lintas jaringan dengan sebuah *rules* yang dibuat oleh administrator dan disimpan dalam sebuah *database*. Pendeteksian jenis ini membutuhkan pembaruan terhadap *database rule* pada *IDS* yang bersangkutan. Berbeda halnya dengan *Behavior Based* yang mendeteksi serangan dengan membandingkan pola dari sebuah dataset menggunakan sebuah metode untuk proses klasifikasi.

Secara umum *Behavior based* dalam proses kerjanya melakukan perbandingan pola atau aktivitas yang ada pada sebuah data, kemudian dilakukan klasifikasi dengan sebuah metode dan menghasilkan sebuah model. Dari model yang sudah dibangun tersebut diuji dengan data *testing* menghasilkan sebuah output untuk melihat akurasi apakah sebuah *traffic* yang ada dapat dikategorikan sebagai intrusi atau bukan. Maka dalam hal ini dibutuhkan sebuah metode yang digunakan untuk proses klasifikasi untuk menghasilkan akurasi yang akurat.

Terdapat beberapa penelitian mengenai perbandingan metode klasifikasi, yang pertama penelitian oleh Srinivas Mukkamala dan Andrew H. Sung (2003). Penelitian ini membahas tentang seleksi fitur yang digunakan untuk IDS dengan menggunakan metode ANN dan SVM agar IDS mencapai performa maksimal dengan menggunakan dataset DARPA 1998. Menurutnya bahwa kinerja algoritma SVM lebih baik jika dibandingkan dengan ANN dalam hal solusi yang dicapai untuk kasus pengklasifikasian IDS. Kemudian penelitian kedua yang dilakukan oleh Mrutyunjaya Panda dan Mana R. Patra (2007) yang menerapkan metode *Naive Bayes* pada deteksi intrusi berbasis anomali menggunakan dataset KDDCup'99. Penelitian ini menyatakan bahwa kinerja algoritma *Bayesian* lebih efisien dalam mengklasifikasikan *Network* IDS (NIDS) dibandingkan ANN. Selanjutnya penelitian yang dilakukan Dwi Widiastuti (2012) yang melakukan perbandingan antara algoritma SVM, *Naive Bayes* dan *Decision Tree* dalam melakukan klasifikasi serangan pada sistem deteksi intrusi dimana data yang digunakan yaitu KDD Cup'99. Penelitian ini menyatakan kinerja algoritma *decision tree* lebih baik dibandingkan dengan algoritma SVM dan NBC. Dari ketiga penelitian tersebut, dataset yang digunakan merupakan dataset lama sehingga dibutuhkan sebuah dataset yang lebih baru untuk deteksi serangan saat ini. Metode klasifikasi yang digunakan juga belum ada yang membandingkan secara spesifik nilai akurasi dari metode *Naive Bayes* dan *Support Vector Machine (SVM)* dengan menggunakan kernel *Linear*, *Polynomial* maupun *Sigmoid*.

Dari penjabaran diatas maka dilakukan sebuah penelitian yang berjudul "Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode *Naive Bayes* dan *Support Vector Machine (SVM)*". Tujuan dari penelitian ini untuk melihat tingkat akurasi dari metode *Naive Bayes*, SVM *Linear*, SVM *Polynomial*, dan SVM *Sigmoid* dengan menggunakan dataset ISCX 2012. Dataset ini dipilih karena merupakan dataset baru yang dikembangkan dari tahun 2009 sampai 2012 oleh Fakultas Ilmu Komputer, *Universitas New Brunswick*. Dimana pada penelitian sebelumnya dataset yang digunakan yaitu KDD Cup 1999 dan DARPA 1998 yang merupakan dataset lama sehingga kurang akurat jika dilakukan pengujian deteksi serangan saat ini. Terdapat beberapa tahapan dalam penelitian ini, tahap pertama dataset ISCX dilakukan *preprocess*, selanjutnya menghilangkan beberapa fitur untuk proses klasifikasi, setelah selesai data siap dimasukkan dalam *classifier*. Proses pengujian memanfaatkan pengambilan sampel dengan teknik *random sampling* dimana persentase data *training* 60% data *testing* 40% sehingga menghasilkan persentase model akurasi serta *output* berupa *confusion matrix* dan kurva *ROC (Receiver Operating Characteristic).*

2. DASAR TEORI

2.1. Keamanan Jaringan

Keamanan jaringan dapat dikatakan jika sebuah komputer yang terhubung dengan beberapa jaringan lain yang lebih banyak dapat

memberikan ancaman keamanan daripada sebuah komputer yang sama sekali tidak terhubung pada sebuah jaringan (Ri2M, 2010). Dengan adanya sebuah pengendalian dan pendeteksian maka resiko keamanan jaringan yang tidak diinginkan dapat dilakukan pencegahan.

2.2. *Intrusion Detection System*

Intrusion Detection System (IDS) merupakan sebuah perangkat keras maupun perangkat lunak yang mampu melakukan suatu deteksi pada sebuah aktivitas mencurigakan yang terjadi pada suatu jaringan komputer maupun sistem komputer yang mencurigakan dalam sebuah sistem atau jaringan (Monika Kusumawati, 2010). Menurut (Wu, 2009) *Intrusion Detection System* (IDS) merupakan tindakan untuk melakukan sebuah deteksi dari beberapa *traffic* paket yang dalam prosesnya tidak sesuai atau tidak diharapkan terjadi pada sebuah jaringan.

IDS dalam kemampuannya melakukan deteksi intrusi ataupun serangan pada sebuah jaringan dapat dikategorikan menjadi 2 kategori yaitu sebagai berikut:

1. *Network-based Intrusion Detection System* (NIDS) yang dalam proses melakukan deteksi jika terdapat intrusi atau serangan, maka akan dilakukan analisis untuk seluruh lalu lintas yang terjadi pada jaringan tersebut. Pada dasarnya sebuah NIDS terdapat pada sebuah segmen penting yang ada pada jaringan, yaitu dapat dikatakan sebagai pintu masuk pada sebuah jaringan. Meskipun demikian kelemahan pada NIDS yaitu sedikit lebih rumit saat dilakukan implementasi pada sebuah jaringan yang menggunakan *switch Ethernet*. Namun saat ini beberapa dari vendor *switch Ethernet* sudah melakukan penerapan fungsi IDS pada switch yang telah dibuatnya supaya dapat memonitor koneksi *port*.
2. *Host-based Intrusion Detection System* (HIDS) merupakan IDS yang dalam proses kerjanya hanya dapat mendeteksi sebuah intrusi hanya pada *host* tempat dimana dilakukan implementasi IDS. HIDS melakukan pengamatan maupun pemantauan segala aktivitas hanya pada sebuah *host* jaringan individual apakah didalamnya terdapat aktivitas mencurigakan atau sebuah percobaan penyusupan. HIDS lebih sering diletakkan pada beberapa *server* penting, diantaranya pada *firewall*, *web server*, atau *server* yang terkoneksi ke internet.

2.3. *Naïve Bayes*

Naive Bayes menurut Xhemali, et al (2009) adalah sebuah metode atau algoritma klasifikasi sederhana (*simple*), yang mampu berkontribusi pada keputusan akhir dan pada setiap atributnya memiliki sifat *independent*. Menurut Hang dkk (2006) *Naive Bayes* merupakan proses klasifikasi statistik yang bisa digunakan dalam melakukan prediksi suatu probabilitas pada keanggotaan sebuah *class*. Sedangkan menurut Bustami (2013) *Naive Bayes* merupakan metode pengklasifikasian suatu probabilitas dan statistik yang diperoleh Thomas Bayes seorang ilmuwan Inggris dengan cara melakukan prediksi peluang di masa depan berdasarkan pengalaman pada masa sebelumnya.

2.4. *Support Vector Machine* (SVM)

Support Vector Machine (SVM) dikembangkan oleh Boser, Guyon, dan Vapnik yang dikemukakan pada tahun 1992 di Annual Workshop on Computational Learning Theory. Konsep dasar yang ada pada SVM merupakan beberapa teori komputasi yang sudah ada puluhan tahun sebelumnya, misalnya saja *margin hyperplane*. Pada tahun 1950 seorang Aronszajn memperkenalkan konsep kernel SVM serta kosep-konsep pendukung yang lain. Beberapa komponen yang ada tersebut belum terdapat upaya untuk dirangkai hingga tahun 1992. Terdapat beberapa pilihan fungsi Kernel dari SVM diantranya (Suykens, Gestel, Brabanter, Moor, & Vandewalle, 2002):

- SVM *Linear*
- SVM *Polynomial*
- Kernel RBF *(Radial Basis Function)*
- Kernel MLP *(Multi Layer Perceptron)*
- *Tangent Hyperbolic (sigmoid)*

2.5 Evaluasi Kinerja *Classifier*

Percobaan dari penelitian dapat dilakukan sebuah evaluasi dengan pengukuran nilai *accuracy, precision, recall dan f-score*. Menurut Xhemali, et al (2009) *Confusion Matrix* dapat dilakukan pengukuran dengan cara menggunakan tabel klasifikasi yang bersifat prediktif.

2.5.1 *Confusion Matrix*

Confusion matrix menurut Han dan Kamber (2011) dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali tuple dari kelas yang berbeda. Nilai dari *True-Positive* dan *True-Negative* memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar, sedangkan *False-Positive* dan *False-Negative* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data.

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	*P'*	*N'*	*P + N*

Gambar 1 *Confusion Matrix* menampilkan total *positive* dan *negative tuple* (Han dan Kamber, 2011)

TP *(True Positive)* → Jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif
FP *(False Positive)* → Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif
FN *(False Negative)* → Jumlah data dengan nilai sebenarnya positif dan nilai prediksi negatif
TN *(True Negative)* → Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif

2.5.2 Kurva ROC *(Receiver Operating Characteristic)*

Kurva ROC menunjukan visualisasi dari akurasi model dan membandingkan perbedaan antar model klasifikasi. *Receiver Operating Characteristic* (ROC) mengekspresikan *confusion matrix* (Vercellis, 2009). ROC merupakan grafik dua dimensi dimana *false positives* sebagai garis horizontal sedangkan *true positives* untuk mengukur perbedaaan performasi metode yang digunakan. Kurva ROC merupakan teknik untuk memvisualisasi dan menguji kinerja pengklasifikasian berdasarkan performanya (Gorunescu, 2011). Model klasifikasi yang lebih baik adalah yang mempunyai kurva ROC lebih besar (Vercellis, 2009).

2.6 Dataset ISCX 2012

Dataset *Information Security Center of eXcellence* (ISCX) merupakan data yang dikembangkan oleh Fakultas Ilmu Komputer, *Universitas New Brunswick* dari tahun 2009 sampai tahun 2012. Penelitian yang dilakukan oleh (Shiravi, Tavallaee dan Ghorbani, 2011) menjelaskan tentang ISCX dan jenis pendekatan yang digunakan dalam mengembangkan dataset. Seluruh dataset berlabel ISCX terdiri dari 157867 paket dengan 19 *features* dan mengumpulkan lebih dari tujuh hari aktivitas jaringan (yaitu normal dan intrusi). Berikut kumpulan data simulasi ISCX berdasarkan skenario serangan:

1. Infiltrasi jaringan dari dalam
2. HTTP *Denial of Service*
3. *Distributed DoS* menggunakan IRC *Botnet*
4. *Brute-force SSH*

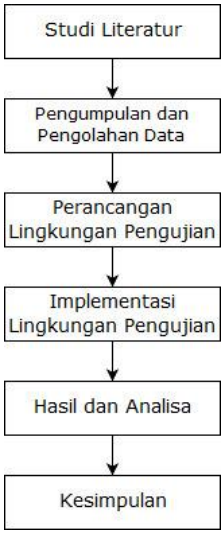
Berikut *features* pada *Intrusion Detection System* dataset ISCX 2012:

Tabel 1 Daftar *Features* pada Dataset ISCX

No.	Nama	No.	Nama
1	*appName*	11	*Src TCPFlagsDescription*
2	*totalSourceBytes*	12	*Dst TCPFlagsDescription*
3	*totalDestinationBytes*	13	*Source*
4	*totalDestinationPacket*	14	*protocolName*
5	*totalSourcePacket*	15	*sourcePort*
6	*Src PayloadAsBase64*	16	*Destination*
7	*Src PayloadAsBaseUTF*	17	*destinationPort*
8	*Dst PayloadAsBase64*	18	*startDateTime*
9	*Dst PayloadAsBaseUTF*	19	*stopDateTime*
10	*direction*		

3. METODOLOGI

Metodologi penelitian akan dibahas secara sistematik melalui langkah-langkah yang spesifik untuk digunakan dalam menyelesaikan masalah penelitian. Tahap-tahap penelitian ini disajikan pada Gambar 2

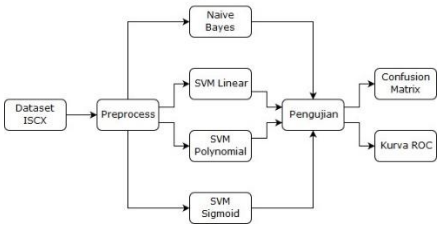


Gambar 2 Diagram Alir Metodologi Penelitian

4. SIMULASI

4.1 Perancangan Lingkungan Pengujian

Rancangan lingkungan pengujian dilakukan untuk memberikan suatu gambaran bagaimana penelitian ini akan dilakukan.



Gambar 3 Diagram Alur Perancangan

Lingkungan Pengujian

Dari Gambar 3 menggambarkan alur bagaimana dataset ISCX dilakukan klasifikasi hingga menghasilkan *output confusion matrix* dan kurva ROC. Dataset ISCX memiliki format xml. Sebelum dimasukkan kedalam *classifier*, dataset perlu dilakukan sebuah ekstrasi file dari **xml* ke dalam bentuk **csv* untuk memudahkan pengolahan data. Hasil dari ekstrasi data tersebut menghasilkan beberapa kolom fitur dengan beberapa data *string* dan *integer*. Untuk dapat dilakukan pengolahan, beberapa data string harus diubah ke dalam bentuk *integer*. Setelah dilakukan pemilihan fitur, selanjutnya dataset siap dimasukkan kedalam *classifier*. Metode klasifikasi yang digunakan pada penelitian ini yaitu *Naive Bayes*, SVM *Linear*, SVM *Polynomial*, dan SVM *Sigmoid*. Dari keempat metode klasifikasi tersebut selanjutnya dilakukan perbandingan nilai *accuracy, precision, recall, dan f1 score* yang dihasilkan untuk mencari metode terbaik. Hasil pengujian dari proses klasifikasi dataset tersebut menghasilkan *output confusion matrix* dan kurva ROC.

4.1.2 Implementasi Lingkungan Pengujian

Implementasi lingkungan pengujian memuat proses implementasi data dari perancangan yang sebelumnya sudah dirancang, mulai dari *preprocess* dataset untuk dapat diolah dalam *classifier Naive Bayes*, SVM *Linear*, SVM *Polynomial* dan SVM *Sigmoid* yang selanjutnya dilakukan pengujian hingga menghasilkan nilai dari *confusion matrix, accuracy, precision, recall, dan f1 score* dan kurva ROC.

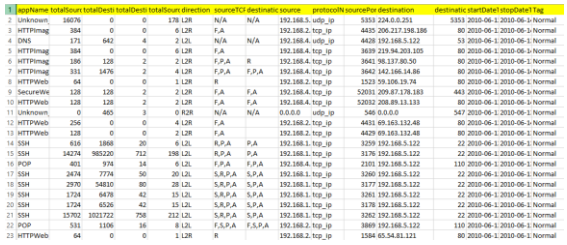
4.1.3 Preprocess Dataset

Dataset ISCX yang digunakan dalam penelitian ini didapatkan dari jaringan *testbed* dengan format *Pcap* yang di arsip dalam 7 zip pengarsipan dan 1 arsip dengan format *xml* yang mencakup keseluruhan data *testbed* tanggal 11 juni hingga 17 juni. Dari 7 zip data tersebut kemudian dipecah menjadi beberapa folder data serta satu folder data yang memiliki format *xml*. Pada penelitian ini data yang digunakan yaitu data pada folder hasil *testbed* pada tanggal 14 juni yang merupakan serangan *HTTPDoS*. Berikut file dataset ISCX *testbed* 14 juni:

```
<?xml version="1.0" encoding="UTF-8"?>
<dataroot xmlns:od="urn:schemas-microsoft-com:officedata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="TestbedMonJun14Flows.xsd" generated="2014-03-11T18:21:14">
<TestbedMonJun14Flows>
<appName>Unknown_UDP</appName>
<totalSourceBytes>16076</totalSourceBytes>
<totalDestinationBytes>0</totalDestinationBytes>
<totalDestinationPackets>0</totalDestinationPackets>
<totalSourcePackets>178</totalSourcePackets>
<sourcePayloadAsBase64></sourcePayloadAsBase64>
<destinationPayloadAsBase64></destinationPayloadAsBase64>
<destinationPayloadAsUTF></destinationPayloadAsUTF>
<direction>L2R</direction>
<sourceTCPFlagsDescription>N/A</sourceTCPFlagsDescription>
<destinationTCPFlagsDescription>N/A</destinationTCPFlagsDescription>
<source>192.168.5.122</source>
<protocolName>udp_ip</protocolName>
<sourcePort>5353</sourcePort>
<destination>224.0.0.251</destination>
<destinationPort>5353</destinationPort>
<startDateTime>2010-06-13T23:57:19</startDateTime>
<stopDateTime>2010-06-14T00:11:23</stopDateTime>
<Tag>Normal</Tag>
```

Gambar 4 Dataset ISCX 14 Juni.*xml*

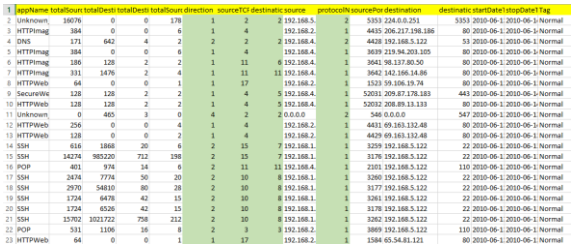
Untuk mempermudah pengolahan data maka dataset tersebut dilakukan ekstraksi data kedalam bentuk *csv*. Data di *convert* dari format *XML ke CSV* menggunakan XML to CSV *converter* yang kemudian menghasilkan *Excel Spreadsheet*. Dilakukan normalisasi data yaitu mengubah format *string* menjadi *integer* untuk mempermudah proses perhitungan dan pengolahan data. Dimana jenis paket dataset ISCX *testbed* 14 juni ini beberapa fiturnya berjenis *"string"* sehingga tidak bisa dilakukan perhitungan.



	appName	totalSour	totalDesti	totalDesti	totalDesti	totalSour	direction	sourceTCI	destinatic	source	protocolN	sourcePor	destination	destinatic	startDate1	stopDate1	Tag
2	Unknown	16076	0	0	178	L2R		N/A	N/A	192.168.5.	udp_ip		5353	224.0.0.251			Normal
3	HTTPImag	384	0	0	6	L2R		F,A		192.168.2.	tcp_ip		4435	206.217.198.186			Normal
4	DNS	171	642	4	2	L2L		N/A	N/A	192.168.4.	udp_ip		4428	192.168.5.122			Normal
5	HTTPImag	384	0	0	6	L2R		F,A		192.168.4.	tcp_ip		3639	219.94.203.105			Normal
6	HTTPImag	186	128	2	2	L2R		F,P,A	R	192.168.4.	tcp_ip		3641	98.137.80.50			Normal
7	HTTPImag	331	1476	2	4	L2R		F,P,A	F,P,A	192.168.4.	tcp_ip		3642	142.166.14.86			Normal
8	HTTPWeb	64	0	0	1	L2R		R		192.168.2.	tcp_ip		1523	59.106.19.74			Normal
9	SecureWe	128	128	2	2	L2R		F,A	F,A	192.168.4.	tcp_ip		52031	209.87.178.183			Normal
10	HTTPWeb	128	128	2	2	L2R		F,A	F,A	192.168.4.	tcp_ip		52032	208.89.13.133			Normal
11	Unknown	0	465	3	0	R2R		N/A	N/A	0.0.0.0	udp_ip		546	0.0.0.0			Normal
12	HTTPWeb	256	0	0	4	L2R		F,A		192.168.2.	tcp_ip		4431	69.163.132.48			Normal
13	HTTPWeb	128	0	0	2	L2R		F,A		192.168.2.	tcp_ip		4429	69.163.132.48			Normal
14	SSH	616	1868	20	6	L2L		R,P,A	P,A	192.168.1.	tcp_ip		3259	192.168.5.122			Normal
15	SSH	14274	985220	712	198	L2L		R,P,A	P,A	192.168.1.	tcp_ip		3176	192.168.5.122			Normal
16	POP	401	974	14	6	L2L		F,P,A	F,P,A	192.168.4.	tcp_ip		2101	192.168.5.122			Normal
17	SSH	2474	7774	50	20	L2L		S,R,P,A	S,P,A	192.168.1.	tcp_ip		3260	192.168.5.122			Normal
18	SSH	2970	54810	80	28	L2L		S,R,P,A	S,P,A	192.168.1.	tcp_ip		3177	192.168.5.122			Normal
19	SSH	1724	6478	42	15	L2L		S,R,P,A	S,P,A	192.168.1.	tcp_ip		3261	192.168.5.122			Normal
20	SSH	1724	6526	42	15	L2L		S,R,P,A	S,P,A	192.168.1.	tcp_ip		3178	192.168.5.122			Normal
21	SSH	15702	1021722	758	212	L2L		S,R,P,A	S,P,A	192.168.1.	tcp_ip		3262	192.168.5.122			Normal
22	POP	531	1106	16	8	L2L		F,S,P,A	F,S,P,A	192.168.2.	tcp_ip		3869	192.168.5.122			Normal
23	HTTPWeb	64	0	0	1	L2R		R		192.168.2.	tcp_ip		1584	65.54.81.121			Normal

Gambar 5 Dataset ISCX 14 Juni.csv sebelum dilakukan normalisasi

Setelah dilakukan normalisasi terhadap dataset ISCX *testbed* 14 Juni, berikut hasil normalisasinya



	appName	totalSour	totalDesti	totalDesti	totalDesti	totalSour	direction	sourceTCI	destinatic	source	protocolN	sourcePor	destination	destinatic	startDate1	stopDate1	Tag
2	Unknown	16076	0	0	178	1	2	2	192.168.5.	2	5353	224.0.0.251					Normal
3	HTTPImag	384	0	0	6	1	4		192.168.2.	1	4435	206.217.198.186					Normal
4	DNS	171	642	4	2	2	2	2	192.168.4.	2	4428	192.168.5.122					Normal
5	HTTPImag	384	0	0	6	1	4		192.168.4.	1	3639	219.94.203.105					Normal
6	HTTPImag	186	128	2	2	1	11	6	192.168.4.	1	3641	98.137.80.50					Normal
7	HTTPImag	331	1476	2	4	1	11	11	192.168.4.	1	3642	142.166.14.86					Normal
8	HTTPWeb	64	0	0	1	1	17		192.168.2.	1	1523	59.106.19.74					Normal
9	SecureWe	128	128	2	2	1	4	5	192.168.4.	1	52031	209.87.178.183					Normal
10	HTTPWeb	128	128	2	2	1	4	5	192.168.4.	1	52032	208.89.13.133					Normal
11	Unknown	0	465	3	0	4	2	2	0.0.0.0	2	546	0.0.0.0					Normal
12	HTTPWeb	256	0	0	4	1	4		192.168.2.	1	4431	69.163.132.48					Normal
13	HTTPWeb	128	0	0	2	1	4		192.168.2.	1	4429	69.163.132.48					Normal
14	SSH	616	1868	20	6	2	15	7	192.168.1.	1	3259	192.168.5.122					Normal
15	SSH	14274	985220	712	198	2	15	7	192.168.1.	1	3176	192.168.5.122					Normal
16	POP	401	974	14	6	2	11	11	192.168.4.	1	2101	192.168.5.122					Normal
17	SSH	2474	7774	50	20	2	10	8	192.168.1.	1	3260	192.168.5.122					Normal
18	SSH	2970	54810	80	28	2	10	8	192.168.1.	1	3177	192.168.5.122					Normal
19	SSH	1724	6478	42	15	2	10	8	192.168.1.	1	3261	192.168.5.122					Normal
20	SSH	1724	6526	42	15	2	10	8	192.168.1.	1	3178	192.168.5.122					Normal
21	SSH	15702	1021722	758	212	2	10	8	192.168.1.	1	3262	192.168.5.122					Normal
22	POP	531	1106	16	8	2	3	3	192.168.2.	1	3869	192.168.5.122					Normal
23	HTTPWeb	64	0	0	1	1	17		192.168.2.	1	1584	65.54.81.121					Normal

Gambar 6 Dataset ISCX 14 Juni.csv setelah dilakukan normalisasi

Supaya data layak digunakan dalam pengolahan, maka dilakukan pengurangan jumlah atribut atau *features* serta dilakukan dengan mengubah beberapa data *string* menjadi sebuah angka (*integer*) dengan memanfaatkan rumus *excel*.

4.1.4 Klasifikasi Dataset

Setelah dataset dilakukan normalisasi maka siap dimasukkan pada *classifier*. Pada sub bab ini memuat proses *input* dataset ISCX yang telah dilakukan normalisasi yaitu mengubah paket data berjenis *string* menjadi *integer* untuk mempermudah proses pengolahan data.

Dalam proses klasifikasi dataset ISCX *Testbed* 14 Juni pada penelitian ini, bahasa pemrograman yang digunakan adalah bahasa pemrograman *python* dengan menggunakan *library* bernama *scikit-learn*.

5 HASIL DAN PEMBAHASAN

5.1 Hasil

Hasil dari lingkungan pengujian pada proses klasifikasi dengan menggunakan metode *Naive Bayes*, SVM *Linear*, SVM *Polynomial*, serta SVM *Sigmoid* menampilkan persentase nilai *accuracy*, *precision*, *recall*, *dan f1 score* dari masing-masing *classifier*.

5.1.2 Confusion Matrix

Dari hasil perhitungan *confusion matrix* yang dilakukan pada proses klasifikasi pada modul *classify* dengan metode *Naive Bayes*, maka dihasilkan ringkasan nilai sebagai berikut:

Tabel 2 Perhitungan *Confusion Matrix (Naive Bayes)*

Jenis Paket	Jumlah Paket	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
TN	0	85,055	100	85,055	91,924
FP	0				
FN	5978				
TP	34022				

Dari hasil perhitungan *confusion matrix* yang dilakukan pada proses klasifikasi pada modul *classify* dengan metode SVM *Linear*, maka dihasilkan ringkasan nilai sebagai berikut:

Tabel 3 Perhitungan *Confusion Matrix* (SVM *Linear*)

Jenis Paket	Jumlah Paket	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
TN	0	99,995	99,995	100	99,997
FP	1				
FN	0				
TP	19999				

Dari hasil perhitungan *confusion matrix* yang dilakukan pada proses klasifikasi pada modul *classify* dengan metode SVM *Polynomial*, maka dihasilkan ringkasan nilai sebagai berikut:

Tabel 4 Perhitungan *Confusion Matrix* (SVM *Polynomial*)

Jenis Paket	Jumlah Paket	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
TN	0	99,999	99,994	99,994	99,994
FP	1				
FN	1				
TP	19998				

Dari hasil perhitungan *confusion matrix* yang dilakukan pada proses klasifikasi pada modul *classify* dengan metode SVM *Sigmoid*, maka dihasilkan ringkasan nilai sebagai berikut:

Tabel 5 Perhitungan *Confusion Matrix* (SVM *Sigmoid*)

Jenis Paket	Jumlah Paket	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
TN	0	99,995	99,995	100	99,997
FP	1				
FN	0				
TP	19999				

5.1.3 Kurva ROC *(Receiver Operating Characteristic)*

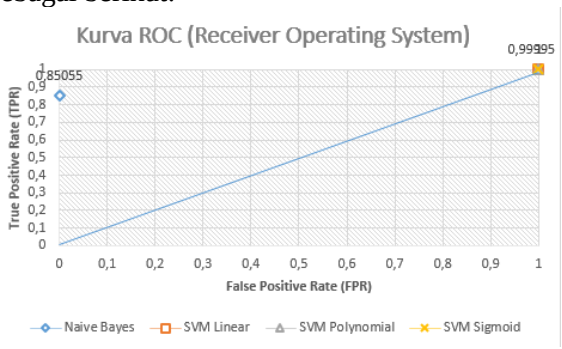
Jika pada *confusion matrix* hanya menyajikan informasi dalam bentuk angka, maka jika ingin menampilkan informasi kinerja algoritma klasifikasi dalam bentuk grafik dapat digunakan Kurva *Receiver Operating Characteristic* (ROC). Kurva ROC digunakan untuk membandingkan kinerja diagnostik dari dua atau lebih tes laboratorium atau diagnostik (Griner et al., 1981). Kurva ROC digunakan untuk melakukan sebuah analisa terhadap model *classifier* yang telah dibuat. Kurva ROC dibuat berdasarkan nilai yang telah didapatkan pada perhitungan *confusion matrix*, yaitu antara *False Positive Rate* dengan *True Positive Rate*. Dimana:

- *False Positive Rate* (FPR) = *False Positive* / (*False Positive* + *True Negative*)
- *True Positive Rate* (TPR) = *True Positive* / (*True Positive* + *False Negative*)

Tabel 6 Perhitungan FPR dan TPR

Jenis Classifier	**False Positive Rate**	**True Positive Rate**
Naive Bayes	0	0,85055
SVM Linear	1	1
SVM Polynomial	1	0,9995
SVM Sigmoid	1	1

Pada tabel 6 merupakan hasil perhitungan *False Positive Rate* dan *True Positive Rate* yang didapatkan dari nilai *false-positive, true-negative, true-positive, dan false-negative*. Dari tabel tersebut maka dapat dibuat sebuah kurva ROC dengan memanfaatkan *False Positive Rate* sebagai sumbu X dan *True Positive Rate* sebagai sumbu Y, maka dapat ditampilkan kurva ROC sebagai berikut:



Gambar 7 Kurva ROC

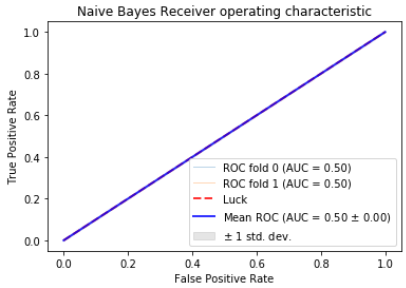
Pada Gambar 7 merupakan kurva ROC, dimana *Naive Bayes* berada pada titik {0, 0,85055}, SVM *Linear* berada pada titik {1,1} yaitu sejajar dengan garis *threshold*, SVM *Polynomial* berada pada titik {1, 0,99995} sedikit dibawah garis *threshold*, sementara itu SVM *Sigmoid* juga sejajar dengan garis *threshold* (1,1).

5.1.4 Kurva ROC dengan *Cross-Validation*

Cross-Validation merupakan metode statistik validasi silang yang dilakukan dengan melakukan evaluasi serta perbandingan. Metode ini dilakukan dengan cara membagi data menjadi dua segmen. Segmen pertama untuk melatih model yaitu data *training* sedangkan segmen kedua untuk memvalidasi model yaitu data uji atau data *testing*. *Cross-Validation* yang populer yaitu *k-fold cross-validation*. Dalam proses kerjanya, dataset dibagi menjadi sejumlah K-buah partisi secara random. Setelah terbagi dalam k-buah partisi maka dilakukan sebanyak K-kali eksperimen. Pada masing-masing eksperimen, digunakan data partisi ke-K sebagai data *testing* dan partisi yang lain sebagai data *training*. Dalam proses *Cross-Validation* dilakukan proses validasi silang, yaitu data training dijadikan data testing, dan sebaliknya data testing dijadikan data training. Dalam penelitian ini digunakan metode evaluasi *standard* yaitu *stratified* 2 *fold cross validation.* Maksudnya adalah dilakukan proses pengujian sebanyak 2 kali, dan menghasilkan pengukuran dengan nilai (*mean*) rata-rata dari 2 pengujian tersebut. Dari hasil pengukuran tersebut dihasilkan nilai *Area Under Curve* (AUC) atau disebut juga luas area dibawah kurva. Nilai AUC berguna dalam menentukan model klasifikasi terbaik. Dalam penelitian ini digunakan data *training* dengan persentase 60% dan data *testing*

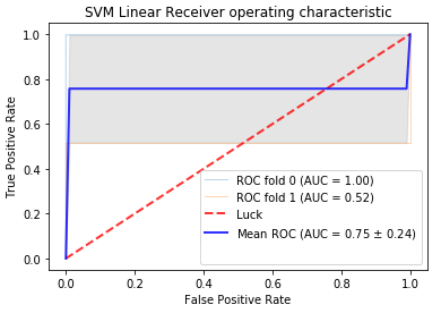
dengan persentase 40%.

Dari nilai *True-Postive* dan *False-Positive* yang dihasilkan dari perhitungan *confusion matrix* yang dilakukan pada proses klasifikasi pada modul *classify* dengan metode *Naive Bayes*, maka dihasilkan kurva ROC sebagai berikut:



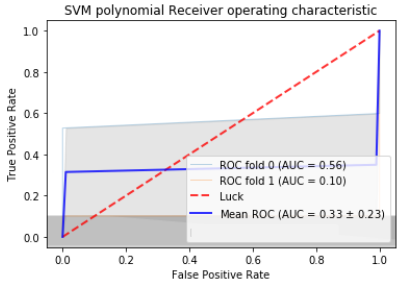
Gambar 8 Kurva ROC *(Naive Bayes)*

Dari nilai *True-Postive* dan *False-Positive* yang dihasilkan dari perhitungan *confusion matrix* yang dilakukan pada proses klasifikasi pada modul *classify* dengan metode SVM *Linear*, maka dihasilkan kurva ROC sebagai berikut:



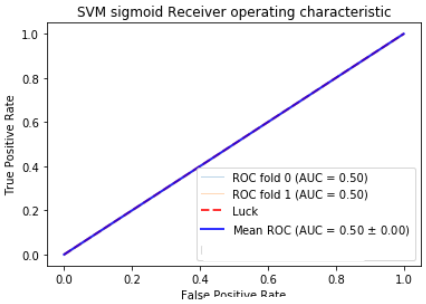
Gambar 9 Kurva ROC (SVM *Linear*)

Dari nilai *True-Postive* dan *False-Positive* yang dihasilkan dari perhitungan *confusion matrix* yang dilakukan pada proses klasifikasi pada modul *classify* dengan metode SVM *Polynomial*, maka dihasilkan kurva ROC sebagai berikut:



Gambar 10 Kurva ROC (SVM *Polynomial*)

Dari nilai *True-Postive* dan *False-Positive* yang dihasilkan dari perhitungan *confusion matrix* yang dilakukan pada proses klasifikasi pada modul *classify* dengan metode SVM *Sigmoid*, maka dihasilkan kurva ROC sebagai berikut:



Gambar 11 Kurva ROC (SVM *Sigmoid*)

5.1.5 Running Time

Running time atau yang biasa disebut sebagai waktu komputasi yang dibutuhkan oleh sebuah sistem dalam melakukan penyelesaian masalah dan juga membangun sebuah model pada sebuah komputer. Waktu komputasi dilakukan perhitungan mulai dari algoritma berjalan hingga saat algoritma berhenti. Berdasarkan hasil penelitian, dalam melakukan proses klasifikasi hingga memberikan hasil output, masing-masing metode *Naive Bayes*, SVM *Linear*, SVM *Polynomial*, dan SVM *Sigmoid* membutuhkan waktu yang berbeda-beda, dapat dilihat pada tabel berikut:

Tabel 7 Hasil Perbandingan *Running Time Classifier*

Classifier	***Running Time (seconds)***
Naive Bayes	16.189985990524292
SVM *Linear*	1201.6530289649963
SVM *Polynomial*	652.7967011928558
SVM *Sigmoid*	75.86262702941895

5.2 Pembahasan

5.2.2 Confusion Matrix

Hasil dari perhitungan *confusion matrix* dengan metode *Naive Bayes* pada Tabel 5.1 dengan jumlah paket yang terdeteksi sebagai *True-Negative* sebesar 0 data, *False-Positive* sebesar 0 data, *False-Negative* sebesar 5978 data dan *True-Positive* sebesar 34022 data, menghasilkan nilai *accuracy* 85,055%, *precision* 100%, *recall* 85,055% dan F1 *score* 91,924%. Hal ini berarti nilai yang dihasilkan dari

perbandingan data, atau yang diidentifikasikan apakah benar merupakan *attack* atau data normal dari total keseluruhan data hanya 85,055% yang diidentifikasikan benar. Jika dilihat dari tingkat ketepatan antara informasi yang diminta dengan jawaban yang diberikan sistem menghasilkan nilai 100% yang berarti sangat tepat. Tetapi jika dilihat dari tingkat perolehan keberhasilan sistem dalam menemukan kembali sebuah informasi hanya sebesar 85,055%. Pada pengujian ini dapat dikatakan bahwa kualitas klasifikasi cukup berhasil karena memperoleh nilai *precision* dan *recall* yang tinggi.

Hasil dari perhitungan *confusion matrix* dengan metode SVM *Linear* pada Tabel 5.2 dengan jumlah paket yang terdeteksi sebagai *True-Negative* sebesar 0 data, *False-Postive* sebesar 1 data, *False-Negative* sebesar 0 data dan *True-Positive* sebesar 19999 data, menghasilkan nilai *accuracy* 99,995%, *precision* 99,995%, *recall* 100% dan F1 *score* 99,997%. Hal ini berarti nilai yang dihasilkan dari perbandingan data, atau yang diidentifikasikan apakah benar merupakan *attack* atau data normal dari total keseluruhan data memiliki persentase yang cukup tinggi sebesar 99,995%. Jika dilihat dari tingkat ketepatan antara informasi yang diminta dengan jawaban yang diberikan sistem menghasilkan nilai 99,995% yang berarti tingkat ketepatannya tinggi. Tetapi jika dilihat dari tingkat perolehan keberhasilan sistem dalam menemukan kembali sebuah informasi memiliki persentase yang sangat tinggi sebesar 100%. Pada pengujian ini dapat dikatakan bahwa kualitas klasifikasi berhasil karena memperoleh nilai *precision* dan *recall* yang sangat tinggi.

Hasil dari perhitungan *confusion matrix* dengan metode SVM *Polynomial* pada Tabel 5.3 dengan jumlah paket yang terdeteksi sebagai *True-Negative* sebesar 0 data, *False-Postive* sebesar 1 data, *False-Negative* 1 data, dan *True-Positive* sebesar 19998 data, menghasilkan nilai *accuracy* 99,999%, *precision* 99,994%, *recall* 99,994% dan F1 *score* 99,994%. Hal ini berarti nilai yang dihasilkan dari perbandingan data, atau yang diidentifikasikan apakah benar merupakan *attack* atau data normal dari total keseluruhan data memiliki persentase yang hampir sempurna yaitu 99,999%. Jika dilihat dari tingkat ketepatan antara informasi yang diminta dengan jawaban yang diberikan sistem dengan tingkat perolehan keberhasilan sistem dalam menemukan kembali sebuah informasi memiliki persentase yang sama yaitu sebesar 99,994%. Hal ini berarti, pada pengujian yang dilakukan dapat dikatakan bahwa kualitas klasifikasi cukup berhasil karena memperoleh nilai *precision* dan *recall* yang tinggi.

Hasil dari perhitungan *confusion matrix* dengan metode SVM *Sigmoid* pada Tabel 5.4 dengan jumlah paket yang terdeteksi sebagai *True-Negative* sebesar 0 data, *False-Postive* sebesar 1 data, *False-Negative* sebesar 0 data, dan *True-Positive* sebesar 19999 data, menghasilkan nilai *accuracy* 99,995%, *precision* 99,995%, *recall* 100% dan F1 *score* 99,997%. Hal ini berarti nilai yang dihasilkan dari perbandingan data, atau yang diidentifikasikan apakah benar merupakan *attack* atau data normal dari total keseluruhan data memiliki persentase yang cukup tinggi sebesar 99,995%. Jika dilihat dari tingkat ketepatan antara informasi yang diminta dengan jawaban yang diberikan sistem menghasilkan nilai 99,995% yang berarti tingkat ketepatannya tinggi. Tetapi jika dilihat dari tingkat perolehan keberhasilan sistem dalam menemukan kembali sebuah informasi memiliki persentase yang sangat tinggi sebesar 100%. Pada pengujian ini dapat dikatakan bahwa kualitas klasifikasi berhasil karena memperoleh nilai *precision* dan *recall* yang sangat tinggi.

5.2.3 Kurva ROC *(Receiver Operating Characteristic)*

Kurva ROC yang dihasilkan pada penelitian ini merupakan kurva untuk mengevaluasi kualitas keberhasilan dari *output* klasifikasi menggunakan *cross-validation*. Kurva ROC menampilkan *True Positive Rate* pada sumbu Y dan *False Positive Rate* pada sumbu X. Hal ini berarti bahwa sudut kiri atas plot merupakan titik ideal yaitu *False Positive Rate* 0 dan *True Positive Rate* 1. Kurva ROC ini diperlukan dalam evaluasi kinerja *classifier* untuk mengevaluasi tingkat keberhasilan rata-rata dari sekian pengujian yang telah dilakukan pada proses klasifikasi. Dari kurva yang dihasilkan, jika garis kurva berada diatas garis *threshold* maka pengujian semakin baik, tetapi jika garis kurva berada dibawah garis *threshold* maka dapat dikatakan bahwa proses klasifikasi tersebut buruk.

Untuk klasifikasi data mining menurut

Gorunescu (2011), bahwa nilai AUC dapat dibagi menjadi beberapa kelompok:

1. 0,90 – 1.00 = *Excellent Classification*
2. 0,80 – 0,90 = *Good Classification*
3. 0,70 – 0,80 = *Fair Classification*
4. 0,60 – 0,70 = *Poor Classification*
5. 0,50 – 0,60 = *Failure*

Hasil pada Gambar 7 merupakan kurva ROC, dimana *Naive Bayes* berada pada titik {0, 0,85055}. Hal ini terlihat bahwa titik *naive bayes* berada diatas garis *threshold*, yang berarti semakin berada diatas garis *threshold* dan mendekati nilai 1 maka dapat dikatakan *naive bayes* merupakan *classifier* yang baik dalam proses pengklasifikasian. Berbeda halnya dengan SVM *Linear* yang berada pada titik {1,1} yaitu sejajar dengan garis *threshold*. Hal ini berarti memiliki nilai AUC sebesar 0,5, dimana berdasarkan nilai AUC menurut Gorunescu (2011) dapat dikatakan *classifier* SVM *Linear* bernilai failure yaitu memiliki tingkat akurasi yang buruk dalam pengklasifikasian. Sementara itu, SVM *Polynomial* berada pada titik {1, 0,99995} sedikit dibawah garis *threshold*. Jika dilihat dari kurva ROC, semakin dibawah garis *threshold*, maka *classifier* tersebut dapat dikatakan memiliki klasifikasi yang buruk. Sedangkan SVM *Sigmoid* juga sejajar dengan garis threshold (1,1). Sama halnya dengan SVM Linear yang memiliki nilai AUC sebesar 0,5, dimana berdasarkan nilai AUC menurut Gorunescu (2011) dapat dikatakan *classifier* SVM *Sigmoid* bernilai failure yaitu memiliki tingkat akurasi yang buruk dalam proses pengklasifikasian.

Hasil pada Gambar 8 hingga Gambar 11 merupakan kurva ROC dengan meto*de K-Fold Cross Validation*, yaitu menggunakan teknik validasi silang. Teknik *Cross Validation* dipakai sebagai sebuah teknik untuk melakukan validasi suatu metode, dimana untuk lebih menguatkan keakuratan hasil dari metode tersebut. Metode ini dilakukan dengan dua kali proses validasi dengan pengambilan data secara acak dari data *testing*. *Fold* 0 merupakan percobaan pertama dan *fold* 1 merupakan percobaan kedua. Dari hasil pada Gambar 8 menunjukkan nilai AUC yang dihasilkan pada kurva ROC *Naive Bayes* yaitu sebesar 0,50 yang berarti dalam proses klasifikasi memliki kekuatan nilai diagnostik yang sangat lemah. Dapat dilihat garis pada kurva yang berada persis pada garis *threshold* atau garis batas nilai, yang berarti diagnostik klasifikasinya cenderung *failure* atau tidak sesuai. Selanjutnya pada Gambar 9 menunjukkan nilai AUC yang dihasilkan pada kurva ROC SVM *Linear* yaitu sebesar 0,75 yang berarti dalam proses klasifikasi memliki kekuatan nilai diagnostik sedang *(fair classification).* Terlihat dari garis kurva yang berada diatas garis *threshold* atau garis batas nilai mendekati angka 1, sehingga menghasilkan nilai AUC atau area dibawah kurva 1, dan kemudian turun dibawah garis *threshold* yang menyebabkan nilai AUC turun menjadi 0,52. Sehingga dihasilkan nilai rata-rata nilai AUC pada kurva ROC SVM Linear sebesar 0,75. Hal ini berarti dalam proses diagnostic klasifikasinya cenderung sesuai (*Fair Classification).* Pada Gambar 10 menunjukkan nilai AUC yang dihasilkan pada kurva ROC SVM *Polynomial* memiliki nilai yang lebih rendah dari tipe klasifikasi yaitu sebesar 0,33 yang berarti dalam proses klasifikasi memliki kekuatan nilai diagnostik sangat lemah atau sangat buruk. Terlihat pada garis kurva yang lebih banyak berada dibawah garis *threshold*. Semakin berada dibawah garis *threshold* maka proses klasifikasi tersebut dikatakan buruk. Pada kurva didapatkan nilai AUC atau luas area dibawah kurva sebesar 0,56 dan 0,1 sehingga didapatkan nilai rata-rata AUC hanya sebesar 0,33. Sedangkan nilai AUC yang dihasilkan pada Gambar 11 pada kurva ROC SVM *Sigmoid* memiliki nilai yang sama dengan *Naive Bayes* yaitu sebesar 0,50 dengan garis pada kurva yang berada persis pada garis *threshold* atau garis batas nilai. Hal ini berarti dalam proses klasifikasi memliki kekuatan nilai diagnostik sangat lemah (*Failure*) atau cenderung tidak sesuai.

Dari keseluruhan hasil, pada kurva ROC yang tanpa menggunakan teknik *Cross Validation*, *classifier* terbaik yaitu *Naive Bayes.* Sedangkan dari hasil kurva ROC yang menggunakan teknik *Cross Validation, classifier* terbaik yaitu SVM Linear. Akan tetapi keakuratan suatu model yang dibangun belum dapat dianggap valid jika perhitungan atau pengukuran yang dilakukan tidak menggunakan teknik untuk memvalidasi. Untuk itu digunakan sebuah teknik *Cross Validation* yang berguna untuk menilai dan melakukan validasi atas suatu keakuratan sebuah model yang dibangun.

5.2.4 Running Time

Pada tabel 7 memuat tentang *running time* atau waktu komputasi yang dibutuhkan oleh masing-masing *classifier Naive Bayes*, SVM

Linear, SVM *Polynomial*, dan SVM *Sigmoid* untuk membangun sebuah model. Waktu komputasi yang dibutuhkan oleh *Naive Bayes* yaitu selama 16 *seconds*. Berbeda halnya dengan waktu komputasi yang dibutuhkan oleh SVM *Linear* yaitu 1201 *seconds*. Sementara itu SVM *Polynomial* membutuhkan waktu komputasi selama 652 *seconds*. Sedangkan SVM *Sigmoid* membutuhkan waktu komputasi selama 75 *seconds*. Dari keempat waktu komputasi yang dibutuhkan oleh masing-masing *classifier*, dapat dilihat bahwa hasil pengujian waktu komputasi untuk metode Naive Bayes menghasilkan waktu komputasi yang lebih cepat yaitu selama 16 seconds. Dibandingkan dengan waktu komputasi untuk metode SVM Linear menghasilkan waktu komputasi lebih lama yaitu 1201 *seconds*. Hal ini sesuai dengan kelebihan yang ada pada algoritma *naïve bayes* dibandingkan beberapa algoritma lain seperti *support vector machine* maupun *neural network* yang membutuhkan waktu cukup lama dalam melakukan komputasi data. Metode *naive bayes* selain simpel juga tidak adanya proses *training* dalam membuat algoritma, maka relatif lebih cepat dari sisi waktu komputasi.

6 KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan pada hasil pengujian, yang diperoleh mengenai proses klasifikasi pada beberapa metode classifier *Naive Bayes*, SVM *Linear*, SVM *Polynomial*, dan SVM *Sigmoid* serta dari pembahasan yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Tahapan klasifikasi serangan menggunakan metode *behavior based* membutuhkan sebuah dataset dan metode. Dengan melakukan perbandingan pola atau aktivitas yang ada pada sebuah data, kemudian dilakukan klasifikasi dengan sebuah metode dan menghasilkan sebuah model. Dari model yang sudah dibangun tersebut diuji dengan data *testing* menghasilkan sebuah *output* untuk melihat akurasi apakah sebuah *traffic* yang ada dapat dikategorikan sebagai intrusi atau bukan.
2. Mekanisme pengolahan dataset ISCX 2012 dilakukan dalam beberapa tahap, yaitu dataset ISCX dilakukan *preprocess* dengan mengubah format *xml* menjadi *csv*, dan mengubah beberapa data dari *string* menjadi *integer*, selanjutnya menghilangkan beberapa fitur untuk proses klasifikasi, hingga data siap dimasukkan dalam *classifier*.
3. Fitur yang digunakan dalam proses klasifikasi yaitu *totalSourceBytes, totalDestinationBytes, totalDestinationPacket, totalSourcePacket, direction, Source TCPFlagsDescription, Destination TCPFlagsDescription, protocolName, sourcePort, Destination, destinationPort, startDateTime,* dan *stopDateTime.*
4. Performa yang dihasilkan dari *confusion matrix* pada masing-masing *classifier Naive Bayes*, SVM *Linear*, SVM *Polynomial*, dan SVM *Sigmoid* menghasilkan persentase akurasi berturut-turut sebesar 85,055%, 99,995%, 99,999% dan 99,995%.
5. Performa kinerja klasifikasi yang dihasilkan dari kurva ROC pada *classifier Naive Bayes* yaitu baik, SVM *Linear* lemah, SVM *Polynomial* sangat lemah, dan SVM *Sigmoid* lemah. Sedangkan jika dilihat dari kurva ROC dengan *cross-validation* menunjukkan bahwa *classifier Naive Bayes* yaitu lemah dengan nilai AUC 0,5, SVM *Linear* baik dengan nilai AUC 0,75, SVM *Polynomial* sangat lemah dengan nilai AUC 0,33 dan SVM *Sigmoid* lemah dengan nilai AUC 0,5.

6.2 Saran

Untuk meningkatkan nilai akurasi dari sebuah metode dapat dilakukan dengan beberapa teknik diantaranya teknik *bagging* dan *boosting*. Dalam penelitian ini menggunakan teknik *random sampling* dan belum menggunakan kedua teknik tersebut, karena penelitian ini hanya terbatas pada perbandingan metode *Naive Bayes*, SVM *Linear*, SVM *Polynomial* dan SVM *Sigmoid*. Penelitian ini juga menngunakan data dari ISCX *testbed* 14 Juni saja. Untuk itu diharapkan pada penelitian selanjutnya dapat digunakan teknik *bagging* maupun *boosting* untuk peningkatan akurasi serta menggunakan dataset ISCX dari *testbed* yang lain yaitu *testbed* ISCX pada tanggal 11-17 Juni.

DAFTAR PUSTAKA

A. Shiravi, H. Shiravi, M. Tavallaee. And A. A. Ghorbani, *"Toward developing a systematic approach to generate*

benchmark datasets forintrusion detection," Comput. Secur., vol.31, no.3, pp. 357-374, 2011.

Stallings, William, 2005. *Intrusion Cryptography and Network Security*.

Gunn, S, R. 1998. *Support Vector Machines For Classification and Regression*. Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science

Maimon, O., & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook.*

Wang, X., Wen, J., Zhang, Y., & Wang, Y. (2014*). Optik Real estate price forecasting based on SVM optimized by PSO*. Optik - International Journal for Light and Electron Optics, 125(3), 1439–1443.

Bustami., 2013, *Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi*, TECHSI : Jurnal Penelitian Teknik Informatika, Vol. 3, No.2, Hal. 127-146

Yaman, S dan Pelecanos, J. 2013. *Using Polynomial Kernel Support Vector Machines for Speaker Verification*. IEEE SIGNAL PROCESSING LETTERS, VOL. 20, NO. 9, SEPTEMBER 2013

Sopharak, A., Uyyanonvara, B dan Barman, S. 2014. *Comparing SVM and Naïve Bayes Classifier for Automatic Microaneurysm Detection*. World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:8, No:5, 2014

Lin, Hsuan-Tien dan Lin, Chih-Jen. 2017. *A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods*. Department of Computer Science and Information Engineering National Taiwan University

Cristianini dan Tayor, S. 2000, *An introduction to Support Vector Machines*, Cambridge University Press.

Napsiah., Stiawan, D dan Heryanto, A.2016. *Visualisasi Serangan Denial of Service dengan Clustering Menggunakan K-Means Algorithm*. Universitas Sriwijaya

Han, J. dan M. Kamber. 2001. *Data Mining: Concepts and Techniques* . *Tutorial*. Morgan Kaufman Publisher. San Francisco

Nugroho, A, Satrio. 2007. *Pengantar Support Vector Machine*. BPP Teknologi.

M. K. Han Jiawei , 2000. *Data Mining: Concepts and Technique*

M ukkamala, Srinivas. dan Andrew H. Sung. 2003. *Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines* . Jurnal. Department of Computer Science, MIT. USA

Panda, Mrutyunjaya. And Mana R. PATRA. 2007. *Network Intrusión Detection Using Naive Bayes* . Jornal. Department of Computer Science, Behampur University. India.

Widiastuti, Dwi, 2012. *Analisa Perbandingan Algoritma SVM, Naive Bayes dan Decision Tree Dalam Mengklasifikasikan Serangan (Attacks) Pada Sistem Pendeteksi Intrusi.* Universitas Gunadarma

Kusumawati, Monika. (2010*). Implementasi IDS (Intrusion Detection System) serta Monitoring Jaringan dengan Interface Web Berbasis B ASE pada Keamanan Jaringan* , Skripsi, Universitas Indonesia Jakarta

Xhemali, D., Hinde, C.J. & Stone, R.G. 2009. *Naive Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages*. International Journal of Computer Science Issues 4 (1): 16 -23. (Online) http://ijcsi.org/papers/4 -1-16-23.pdf (16 Mei 2013)

Gorunescu, F. 2011. *Data Mining Concepts, Models and Tehniques*. Intelligent Systems Reference Library, Volume 12. Springer-Verlag Berlin Heidelberg .

Kusrini & Emha Taufiq Luthfi. (2009). *Algoritma Data Mining.* Yogyakarta: Andi.

Suykens, J., Gestel, T., Brabanter, J. D., Moor, B. D., and Vandewalle, J. (2002). *"Least Squares Support Vector Machines"*, World Scientific, Singapore

Vercellis, Bernadth. (2009), *Sistem Informasi* , Lokomedia, Yogyakarta