PENGANALISA JARINGAN HOTSPOT BERBASIS SUPPORT VECTOR MACHINE

Rahmadya Trias Handayanto
Teknik Komputer Universitas Islam 45 Bekasi
Email: rahmadya_handayanto@unismabekasi.ac.id

ABSTRAK

Penelitian ini membahas penganalisa jaringan menggunakan Support Vector Machine (SVM) setelah pada penelitian sebelumnya menggunakan Jaringan Syaraf Tiruan (JST). Ada banyak aplikasi yang tersedia di pasaran saat ini untuk memberikan grafik jaringan area hotspot kita. Grafik ini akan dianalisis oleh administrator jaringan. Karena area hotspot sering berjalan 24 jam, administrator mengalami kesulitan untuk memonitor lalu lintas setiap saat. Oleh karena itu kami merancang sistem otomatis untuk membantu administrator jaringan dalam memantau jaringan. Sistem ini akan menggantikan keterampilan manusia dalam menafsirkan grafik dengan SVM. Untuk meminimalkan jumlah vektor masukan kita menggunakan nilai rata-rata dari sumbu, sehingga misalnya mikro-komputer notebook, laptop, PDA, dan gadget lainnya dapat menangani sistem ini. Hasil pengujian menunjukkan sistem ini dapat mengklasifikasikan antara normal, lalu lintas tinggi dan un-normal grafik jaringan secara berkala.

Kata Kunci: Penganalisa Jaringan, Support Vector Machine, Hotspot

ABSTRACT

We present a Support Vector Machine-based network analyzer system for hotspot area as the extension of previous research that using Neural Networks (NNs). There are many applications that available in the market today for providing us the network graph of our hotspot areas. These graphs will be analyzed by a network administrator. Because a hotspot area often runs 24 hours, an administrator has a difficulty to monitor the traffic all the time. Therefore we proposed the automatic system to help a network administrator in monitoring the network. This system will replace human skill in interpreting the graph with a Support Vector Machine System. To minimize the number of input vector we use mean value of axis, so the micro-computer e.g. notebook, laptop, PDA, and other gadgets can handle this system. Testing result showed this system could classify between normal, high and un-normal traffic of network graph periodically.h

Keywords: Network Analyzer, Support Vector Machine, Hotspot

1. Pendahuluan

Penelitian ini berusaha membuat sebuah aplikasi yang sanggup mendeteksi kondisi suatu jaringan hotspot dengan algoritma *Support Vector Machine (SVM)*. Proses identifikasi berlangsung secara langsung saat kondisi jaringan dipantau dengan menggunakan aplikasi pemantau jaringan (*sniffer*) dengan bantuan program penangkap gambar yang bisa disetting secara periodik. Sistem yang dirancang diharapkan dapat membantu administrator jaringan karena sistem dapat bekerja mandiri selama 24 jam.

Area *hotspot* menurut ensiklopedia (www.pcmag.com) adalah suatu wilayah dengan region tertentu yang menawarkan akses internet bebas atau untuk kalangan tertentu. Akses internet ini diperuntukkan tidak hanya komputer atau laptop melainkan *gadget* seperti PDA, *handphone*, *tablet*, dan sejenisnya dapat mengakses internet lewat *hotspot*.

Berdasarkan proses pengiriman sinyal, *hotspot* termasuk dalam kategori *broadcast*. Artinya sinyal dari satu titik dalam proses menuju titik/alamat yang dituju sebelumnya di sebar ke seluruh alamat yang berada dalam satu area *local hotspot* tersebut. Akibatnya *broadcast* sangat beresiko turunnya kapasitas jaringan jika jumlah anggotanya bertambah. Apalagi jika salah satu anggotanya memiliki aplikasi yang dapat mengganggu kondisi jaringan seperti memata-matai komputer orang lain, atau mencuri seluruh sumber daya *bandwidth hotspot*. Akan tetapi aplikasi tersebut dapat dimanfaatkan oleh administrator jaringan untuk keperluan pemantau kondisi jaringan (Clincy *et al*. 2005)

Virus dan aplikasi *Denial of Service* (DoS) adalah masalah utama seorang administrator jaringan. Aplikasi tersebut dengan mudah menyebar lewat *hotspot* karena sifatnya yang *broadcast*. DoS dapat dengan mudah mengirimkan paket secara terus menerus ke *server* sehingga *hotspot* menjadi lumpuh. Sehingga tugas utama administrator jaringan adalah memantau pengguna yang baik secara sengaja atau tidak yang berpotensi merusah jaringan. Aplikasi yang dapat menjaga kondisi jaringan banyak tersedia di pasaran misalnya SNORT, TCPDUMP, NRF, dan sebagainya (Roose, 1999). Beberapa riset telah dilakukan dalam membandingkan kinerja *sniffer* (Clncy *et al*, 2005). Sementara itu, sistem yang kami buat menggunakan basis gambar, tidak berbasis teks. Algoritma yang digunakan adalah SVM yang merupakan kelanjuta penelitian sebelumnya yang menggunakan Jaringan Syaraf Tiruan (Handayanto *et al*, 2011).

Metode klasifikasi banyak dijumpai dalam bentuk *fuzzy, neuro-fuzzy*, ANFIS, dan lain-lain menggunakan pengklasifikasi tertentu (Abe, 2010). Penelitian ini menggunakan SVM karena memiliki kelebihan-kelebihan tertentu seperti kestabilan, kecepatan, dan kemudahan dalam pembelajaran. Metode lain seperti FIS dan JST sangat mengandalkan prosesor dalam pembelajarannya (penyusunan bobot pada neuron atau aturan/rule), sementara SVM mengandalkan prinsip statistik yang lebih mudah (Abe, 2010 dan Miyamoto *et al.* 2009).

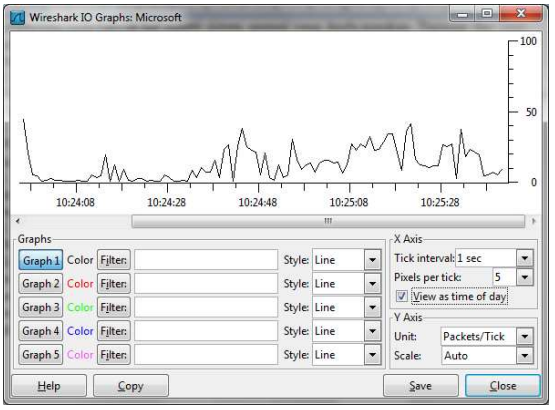
Sistem yang dibuat tidak hanya mampu

mendeteksi kondisi jaringan secara otomatis, melainkan aktivitas lain. Aktivitas-aktivitas tersebut berguna untuk mengevaluasi kondisi jaringan seperti mengetahui sumber virus, DoS, hingga pelanggaran lain yang dilakukan pengguna dalam beraktifitas internet (akses situs porno, memata-matai komputer orang, mencuri data/*password* dan sebagainya).

2. Perancangan Sistem

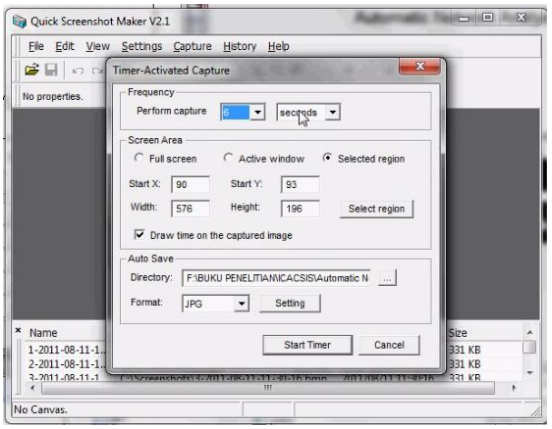
Seperti halnya sistem pembelajaran yang banyak dibuat orang, di sini kami membagi sistem menjadi tiga tahap yaitu: Tahap pembelajaran, Tahap identifikasi dan Tahap forensik. Baik tahap pembelajaran maupun identifikasi menggunakan sistem pemroses citra berukuran 20 x 60. Tahap forensik merupakan tahap pilihan jika terjadi kondisi yang memerlukan penyelidikan lebih lanjut.

*Network Sniffe*r digunakan untuk menangkap grafik kondisi jaringan. Banyak aplikasi yang tersedia di pasaran dan kebanyak bersifat *open source*, salah satunya adalah *Wireshark Sniffer* (sebelumnya bernama *Etherial*). Untuk bisa diolah, disepakati terlebih dahulu ukuran grafik yang akan diproses dan *wireshark* harus menangkap dengan koordinat yang tetap. Misalnya di setting 100 atau 10 paket perdetik dan hindari penggunaan otomatis *scale*, karena mengakibatkan *image* selalu berubah ukurannya (lihat gambar 1).



Gambar 1. Menangkap Grafik Kondisi Jaringan dengan *Wireshark Software*

Ketika *wireshark* menangkap grafik secara terus menerus, aplikasi lain yang bertugas menangkap citra secara periodik disiapkan (lihat gambar 2). Di sini digunakan salah satu aplikasi yang bisa diunduh versi trialnya bernama *Quick Screenshot Maker*.

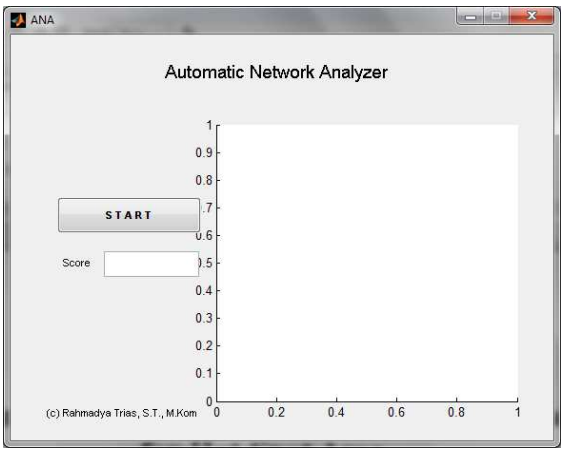


Gambar 2. Timer-Activated Capture of Network's Graph

Setelah menentukan bidang yang akan direkam, *timer* dinyalakan. *Timer* berfungsi menentukan berapa lama tiap kali merekam gambar dari kondisi jaringan. Makin cepat proses perekaman, makin banyak sumber daya yang dibutuhkan (prosesor dan RAM).

Citra yang ditangkap oleh aplikasi perekam grafik akan dianalisa oleh sistem apakah masuk kategori normal, beban puncak ataukah tidak normal. Beberapa jenis bahasa pemrograman tersedia, salah satunya adalah Matlab. Penggunaan *Graphical User Interface* (GUI) sepertinya sudah menjadi keharusan setiap program yang dibuat agar mempermudah pengguna dalam pengoperasiannya. Ketika tombol "*Start*" ditekan, sistem akan mulai melakukan analisa terhadap jaringan (gambar 3).

Sistem menganalisa tiap gambar yang berhasil direkam oleh aplikasi perekam. Citra diletakan dalam lokasi folder khusus yang dapat diakses oleh program dan akan dihapus ketika proses analisa selesai agar menghemat memori, kecuali jika terjadi kondisi tidak normal, agar bisa dianalisa lebih lanjut oleh administrator jaringan. Tentu saja proses terjadi jika ada gambar yang direkam, oleh karena itu pengguna harus memastikan alat perekam periodik berjalan dengan baik.



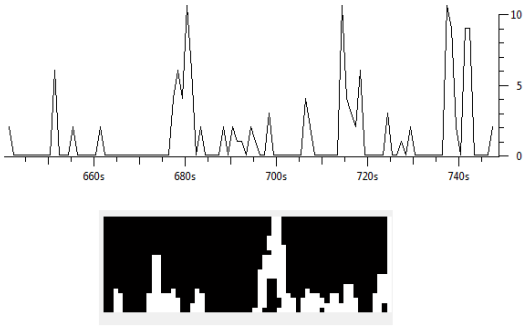
Gambar. 3 GUI dari sistem yang diusulkan

Setelah proses analisa, program akan mengirimkan pesan ke administrator jaringan baik berupa suara, gambar, dan lain-lain. Kita bisa saja menambahkan aplikasi lain seperti SMS *Gateway* yang akan mengirimkan SMS ke administrator.

Tahap Pembelajaran

Sistem yang kami usulkan menggunakan algoritma *Support Vector Machine* yang merupakan jenis *Supervised Learning*. Pelatihan perlu dilakukan untuk mengajari sistem bahwa suatu jenis grafik masuk dalam kategori normal atau tidak. Karena hanya dua kategori yaitu normal atau tidak normal, maka kami hanya menggunakan satu tahap evaluasi.

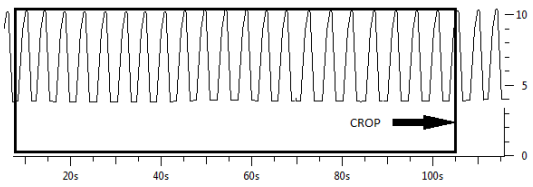
Karena citra yang direkam merupakan gambar real, agar dapat diproses harus dikonversi terlebih dahulu menjadi ukuran 100 x 300 dengan tahapan konversi hitam putih, kontras, pembersihan, dan *cropping*. Hasil akhirnya adalah matriks berukuran 20 x 60 seperti tampak pada gambar 4.



Gambar 4. Kondisi grafik sebelum proses (atas) and setelah proses pengolahan citra(bawah).

Resolusi dikurangi agar proses pembelajaran lebih mudah, dan walaupun resolusi rendah, ciri yang direkam masih tetap terjaga. Dengan ukuran yang kecil harapannya tiap laptop atau bahkan *netbook* berkapasitas prosesor rendah sanggup mengolahnya. Fungsi *cropping* perlu ditambahkan agar hanya grafik

saja yang diproses lebih lanjut.

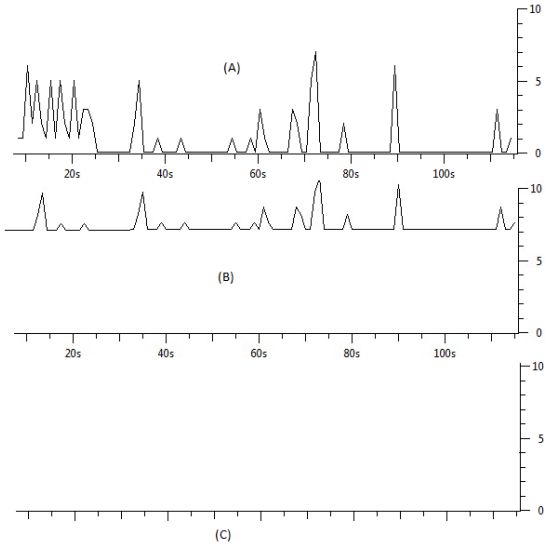


Gambar 5. Mekanisme Croping untuk membuang elemen yang tidak diperlukan.

Banyak riset-riset yang menerapkan pencocokan pola (Czajka *et al*., 2002, Kalera *et al*., 2004 dan Sabourin, 1997). Karena sistem yang kami buat melakukan proses pencocokan pola, maka sistem tidak jauh berbeda pada penelitian di atas. Hanya saja karena hanya membandingkan kondisi beban jaringan maka agar mempermudah proses analisa, sisi ordinat saja yang diperhatikan kecenderungannya.

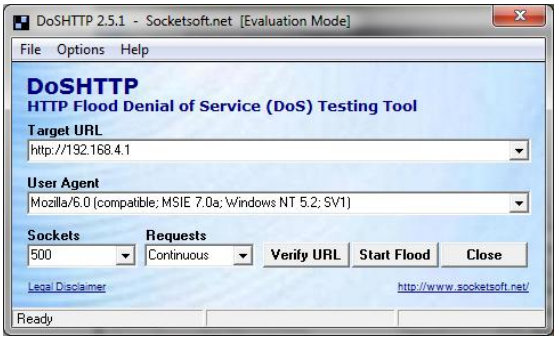
Banyak Literatur yang membahas pengolahan citra baik dengan aplikasi matlab (Gonzales, 2009) maupun teori dasarnya yang lengkap (Leondes, 1998). Untuk memastikan hasil yang baik, terkadang dilakukan pelatihan beberapa kali. Tahap pembelajaran adalah jantung dari sistem dimana kemampuan menganalisa grafik oleh manusia akan digantikan oleh sistem. Karena pelatihan yang dilakukan terpandu, maka diperlukan suatu label/target pembelajaran. Gambar 6 memperlihatkan grafik-grafik yang akan menjadi dasar pembelajaran.

Kondisi normal terjadi jika trafik menunjukkan kondisi rata-rata dan minimal, sementara kondisi tidak normal terjadi karena dua kemungkinan, sistem melewati batas grafik (terserang virus atau DoS), atau tidak ada jaringan yang terdeteksi (*hotspot* mati).

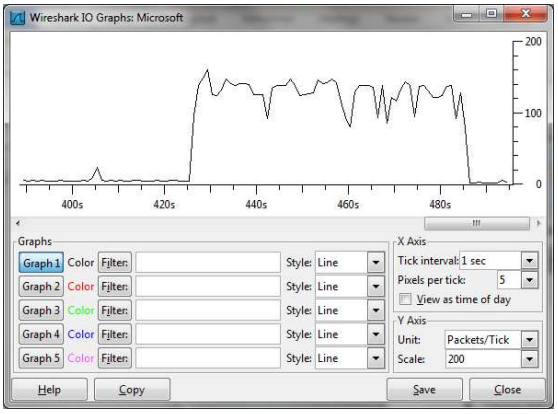


Gambar 6. jenis kondisi trafik, a) Normal, b) high traffic, and c) Un-normal

Untuk menguji perilaku *hotspot* yang terkena serangan Dos, digunakan program yang didownload di internet dan bersifat *open source*. Untuk negara tertentu seperti Amerika Serikat, aplikasi tersebut *illegal*. Gambar 7 memperlihatkan aplikasi yang digunakan untukmenguji sistem yaitu DosHTTP, yang prinsipnya membanjiri *server* dengan data *broadcast* (http://www.socketsoft.net).



Gambar 7. DosHTTP Application for Traffic Generation

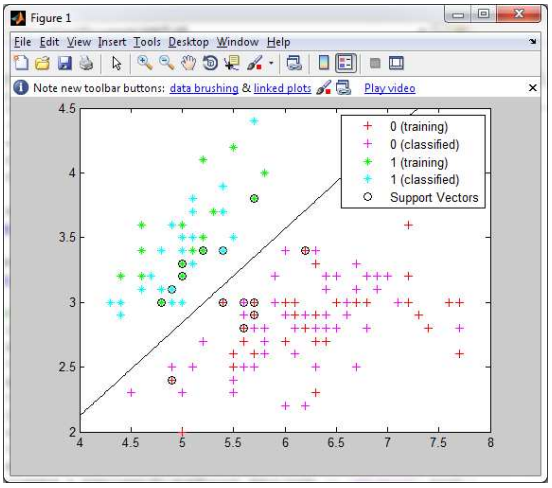


Gambar 8. Hasil serangan DosHTTP selama 30 detik (atas) Grafik yang diperlihatkan Wireshark (bawah).

Serangan DoS sangat berbahaya karena mengakibatkan jaringan hotspot mati. Jenis serangan ini harus diketahui secara cepat oleh administrator jaringan. Seperti tampak pada gambar 7, akibat serangan DoS grafik menunjukkan kenaikan 15 kali dari kondosi normalnya.

Digunakan lima sample untuk tiap kondisi untuk membentuk persamaan pemisah. Terdiri dari lima belas vector karena citra trafik untuk pelatihan sebanyak lima belas (lima untuk tiap kondisi).

Target yang digunakan adalah = [1 1 1 1 1 5 5 5 5 5] , [1 1 1 1 1 10 10 10 10 10], dan [5 5 5 5 5 10 10 10 10 10]. Artinya tiap kondisi jaringan akan dipetakan menjadi satu yang berarti kondisi normal lima yang berarti padat dan 10 yang berarti kondisi tidak normal. Fungsi pada matlab yang digunakan adalah "*svmtrain*" untuk *training*. Gambar di bawah contoh hasil pemisahan/klasifikasi dengan SVM.



Gambar 9. Hasil Pemisahan dengan SVM.

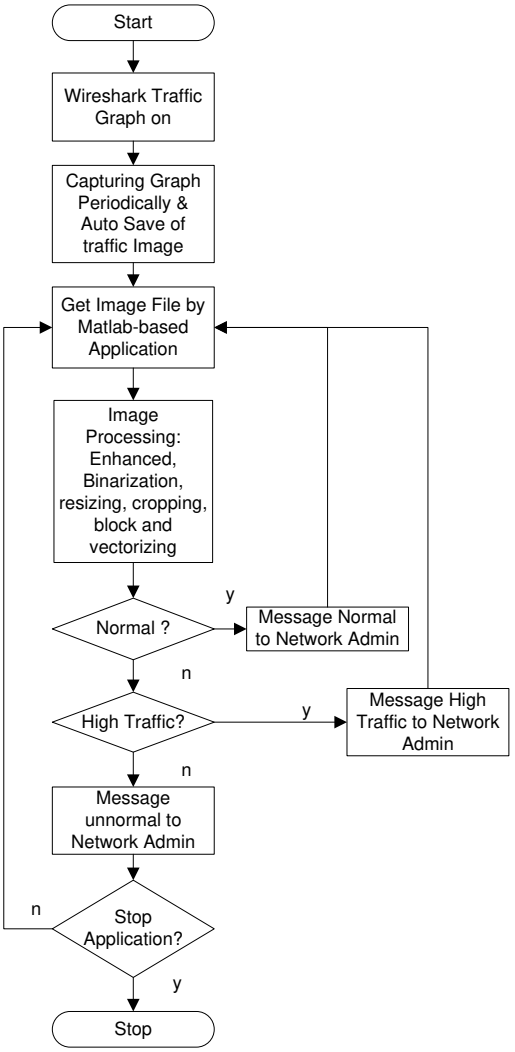
Garis pemisah berhasil dibentuk dan menjadi dasar penentu suatu kondisi yang akan diuji, seperti pada sub bab setelah ini. Istilah *support vector* adalah suatu anggota himpunan yang letaknya paling dekat dengan kelas yang lainnya. Jika kita ingin mengklasifikasikan dengan tiga kondisi, katakanlah kondisi tengah-tengah yaitu beban puncak, maka kita dapat menggunakan metode *tree*, dengan membandingkan satu dengan yang lain (*winner take all*) dimana pemenangnya adalah kelas yang terpilih.

Tahap Identifikasi

Berbeda dengan verifikasi, identifikasi mengharuskan sistem memilih suatu kelas terhadap data yang akan diuji. Diagram alir dapat dilihat pada gambar 9 di bawah ini.

Setelah *wireshark* dijalankan dan memantau kondisi grafik jaringan *hotspot*, aplikasi yang merekam gambar serta aplikasi berbasis SVM dihidupkan maka sesaat setelah tombol rekam dijalankan secara periodik, aplikasi klasifikasi kondisi jaringan siap dijalankan. Karena hasil rekam tersimpan

didalam folder kerja Matlab maka ketika aplikasi SVM dijalankan, maka tiap gambar diklasifikasikan apakah masuk kategori normal atau tidak normal. Ketika kondisi tidak normal ditemukan maka pesan akan dikirim ke administrator jaringan.



Gambar 10. Diagram Alir Sistem

Kode harus dibuat agar Matlab secara periodik menghidupkan *Timer.* Berikut ini kode yang sering digunakan.

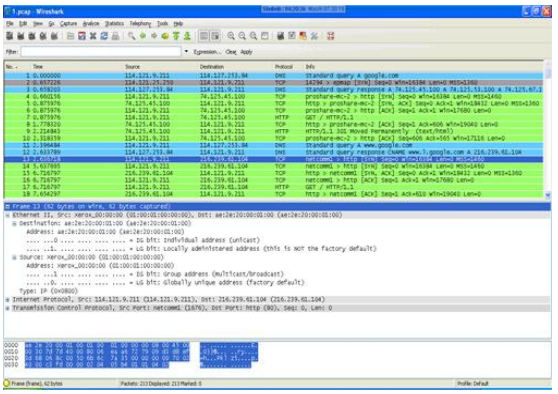
```
t = timer('TimerFcn', 'stat=false; disp("Timer!")',...
                'StartDelay',4);
start(t)
stat=true;
while(stat==true)
<script of image processing and neural network simulation is here>
End
```

Diperlukan kode tambahan agar sistem mampu mengambil data citra tiap beberapa periode tertentu. Selain itu diperlukan juga kode untuk mengganti nama dan menghapus gambar setelah proses klasifikasi berjalan dan kondisi normal diperolah.

Tahap Forensik

Tahap forensik dilaksanakan ketika terjadi suatu masalah (jaringan tidak normal). Masalah yang terjadi biasanya ketika penyerang dapat mengakses root (Albertson *et al*., 2004). Dalam sistem ini, setelah dilakukan forensik terhadap suatu masalah, dapat diketahui sumber di mana masalah tersebut berasal dengan memanfaatkan *software wireshark*.

Gambar 10 memperlihatkan kemampuan *wireshark* mengendus *hotspot*. Jika pengguna yang masuk ke jaringan *hotspot* sedikit tetapi kondisi jaringan sibuk, dapat dipastikan ada klien yang menyebar virus, DoS dan sejenisnya.



Gambar 11. Pengecekan Paket dengan Wireshark

Forensik bukan pekerjaan yang dilaksanakan tiap hari, melainkan hanya jika terjadi masalah pada jaringan dan kita ingin mengetahui penyebabnya. Sistem yang kami

rancang, mampu memberi peringatan kepada administrator jaringan sebelum proses forensik dilaksanakan dan bermanfaat karena bekerja selama 24 jam.

Yang diteliti pada tahap forensik adalah file *log* yang berisi aktivitas jaringan dengan ekstensi "pcap". Tugas mengecek *log file* ini adalah administrator jaringan. Terkadang administrator jaringan mampu mengetahui sumber virus dari klien mana saja.

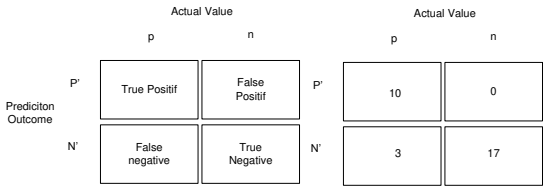
3. Hasil Eksperimen

Digunakan sepuluh sampel untuk tiap kondsisi trafik yang tidak digunakan ketika proses pembelajaran untuk menguji sistem. Intruksi yang digunakan untuk mengecek klasifikasi pada SVM dengan Matlab adalah "*svmclassify*". Dengan memasukkan nama persamaan garis pemisahnya, maka diperoleh jawaban apakah sistem berharga satu yang berarti normal, ataupun sepuluh yang berarti tidak normal. Jika kita menambahkan satu kelas lagi, katakanlah jaringan sibuk/padat maka bisa kita tambah kategori baru, katakanlah lima. Tabel 1 memperlihatkan hasil pengujian sistem.

Tabel 2. Hasil percobaan

No	Traffic	Result	Sim Result	Similarity
1	Normal	1	"Normal"	0,06
2	Normal	1	"Normal"	0,35
3	Normal	1	"Normal"	0,27
4	Normal	1	"Normal"	0,39
5	Normal	1	"Normal"	0,47
6	Normal	1	"Normal"	0,55
7	Normal	1	"Normal"	0,51
8	Normal	1	"Normal"	0,51
9	Normal	1	"Normal"	0,04
10	Normal	1	"Normal"	0,03
11	High	10	"Un-normal"	3,24
12	High	5	"High"	2,98
13	High	5	"High"	2,56
14	High	5	"High"	0,64
15	High	5	"High"	3,76
16	High	5	"High"	4,64
17	High	10	"Un-normal"	4,07
18	High	10	"Un-normal"	4,56
19	High	1	"Normal"	3,67
20	High	5	"High"	2,00
21	Un-normal	10	"Un-normal"	0,03
22	Un-normal	10	"Un-normal"	0,03
23	Un-normal	10	"Un-normal"	0,03
24	Un-normal	10	"Un-normal"	0,03
25	Un-normal	10	"Un-normal"	0,03
26	Un-normal	10	"Un-normal"	0,03
27	Un-normal	10	"Un-normal"	0,03
28	Un-normal	10	"Un-normal"	0,03
29	Un-normal	10	"Un-normal"	0,03
30	Un-normal	10	"Un-normal"	0,03

Pembelajaran terpandu mampu mengklasifikasikan target menjadi kategori normal, tinggi, dan tidak normal dengan indikator 1, 5, dan 10 atau angka lainnya. Kondisi tidak normal merupakan kondisi yang tidak diharapkan dan harus memberi warning kepada administrator jaringan. Gambar 10 memperlihatkan *Receiver Operating Characteristic* (ROC) yang menggambarkan kesalahan positif atau negatif.



Gambar. 12. Receiver Operating Characteristic (ROC) of System

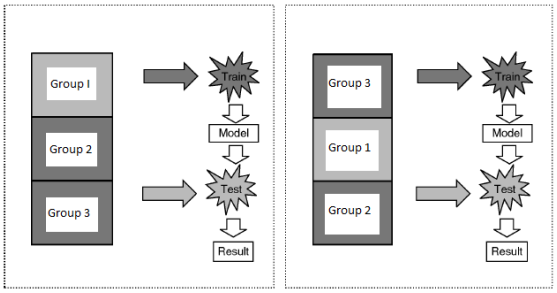
Perlu dibuat kode pada sistem untuk kategorisasi sistem, baik dengan if- else atau switch – case.

x<1.49, Untuk kondisi normal

1.5<x<7.9, Untuk kondisi tinggi

x> 8, Untuk kondisi tidak normal.

Selain itu perlu dilakukan pengujian dengan validasi silang. Beberapa literatur dijumpai pembahasan tentang teori tersebut ((McQuarie *et al*., 1998). Ketika satu grup dilatih sebagai model, grup terakhir digunakan sebagai *tester*. Berikutnya posisi berganti hingga seluruh grup pernah menjadi pembelajar dan penguji (lihat gambar 11).



Gambar 13. Teknik Validasi Silang

Digunakan tiga sampel untuk tiap-tiap grup yang menggambarkan normal, tinggi dan tidak normal, jadi kita memiliki Sembilan sampel. Dengan metode ini tiap sampel pernah berfungsi sebagai pembelajar dan juga sebagai tester terhadap yang lainnya. Hasilnya dapat dilihat pada table 3.

Tabel 3. Hasil Validasi Silang

No.	Learning	Testing	Sim	Kelas	Summary
1	Group 1	Group 3			
		normal	1	normal	ok
		high	10	un-normal	no
		un-normal	10	un-normal	ok
2	Group 3	Group 2			
		normal	5	high	no
		high	5	high	ok
		un-normal	10	un-normal	ok
3	Group 2	Group 1			
		normal	5	high	no
		high	5	high	ok
		un-normal	10	un-normal	ok

Tabel 3 memperlihatkan adanya kesalahan sebesar 30 persen untuk ketiga model yang diuji oleh grup lain. Hal ini terjadi karena jumlah sampel yang hanya tiga. Akan tetapi akurasi sebesar 70 persen dirasa sudah cukup memadai. Tentu saja untuk memperoleh hasil yang lebih akurat kita memerlukan data sampel yang ukurannya besar.

Meningkatkan Efisiensi Pembelajaran

Random Access Memory (RAM) adalah komponen penting dalam komputer mikro seperti laptop dan *gadget* lainnya. Untuk mengatasinya diperlukan penyederhanaan matriksnya, karena kondisi grafik terlihat dari besarnya ordinat dari grafik yang dianalisa. Oleh karena itu kita dapat menggunakan nilai rata-rata dari sisi ordinat. Dengan demikian matriks yang berukuran 20 x 60 dapat disederhanakan menjadi 20 x 1 berupa vektor. Jika kita memiliki matriks berikut:

a =

0.8147 0.6324 0.9575 0.9572 0.4218
0.9058 0.0975 0.9649 0.4854 0.9157
0.1270 0.2785 0.1576 0.8003 0.7922
0.9134 0.5469 0.9706 0.1419 0.9595

Dengan penggunaan nilai rata-rata diperoleh vektor:

Average =

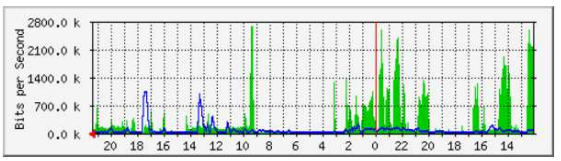
0.7567
0.6739
0.4311
0.7064

Tentu saja metode ini tidak dapat menganalisa kondisi dari dua sisi, seperti pencocokan pola pada citra. Akan tetapi, hal ini dapat meringankan beban komputer mikro yang menjalankan sistem ini, apalagi bekerja secara terus menerus.

4. Perbandingan Dengan Sistem Lain

Mudah membandingkan dengan sistem yang banyak dijumpai di pasaran (SNORT, TCPDUMP, *Network Flight Recorder* (NRF), dan lain-lain) karena sistem yang kami rancangan menggunakan pencocokan pola gambar grafik. Beberapa sistem juga memanfaat *soft computing* tetapi menggunakan algoritma genetika (Kandeerban, 2010). Beberapa sistem yang dibuat sanggup mendeteksi kondisi jaringan yang terserang DoS, *Probe*, U2R, dan R2L yang berjalan dalam tipe *switch*, sedangkan sistem yang kami rancang hanya berjalan dalam tipe *broadcast*. Namun karena *hotspot* berkarakter *broadcast*, sistem yang kami rancang cukup memadai.

Karena sistem yang kami rancang menggunakan metode pencocokan pola grafik, maka sistem ini dapat diterapkan lebih luas dan dikombinasikan oleh penghasil grafik lain selain *wireshark*, misalnya *Multi Router Traffic Grapher* (MRTG). Sehingga tidak hanya diterapkan pada hotspot yang bertipe LAN, melainkan terhadap suatu korporasi yang bertipe WAN. Gambar 12 mengilustrasikan grafik yang dapat diterapkan untuk WAN.



Gambar 14. Example of MRTG (source: www.en.wikipedia.org)

5. Kesimpulan dan Riset Yang Akan Datang

Hasil pengujian menunjukan sistem yang kami rancang dapat mengidentifikasikan tipe jaringan apakah normal, beban penuh, atau tidak normal yang dapat menjadi alarm pertama terhadap administrator jaringan dengan akurasi 70 hingga 90 persen. Alarm yang dihasilkan dari bagian pengambil keputusan pada sistem dapat membantu

administrator jaringan mengetahui jika terjadi gangguan pada hotspotnya.

Karena sistem yang dibuat menganalisa citra grafik, maka dapat diterapkan pada sistem yang lebih luas yang memiliki grafik untuk dianalisa. Ke depan perlu ditambahkan analisa terhadap text hasil *sniffing* oleh *wireshark*. Jika diterapkan pada jaringan non *hotspot* yang biasanya menggunakan *Switch*, perlu digunakan aplikasi *sniffer* yang berjalan pada jaringan yang menggunakan *switch*.

Daftar Acuan

Abe, Sigeo,"*Support Vector Machines for Pattern Classification*", London: Springer-Verlag London Limited, 2010.

Albertson, Mikael & Peter Gunnarsson,"Computer Forensic", TDDCO3 Project, Spring, Sweden, 2004, pp. 1-6.

Anoname, "*Encyclopedia - Definition of: Hotspot*", from: http://www.pcmag.com/encyclopedia_term/0,2542,t=Wi-Fi+hotspot&i=61778,00.asp, The Computer Language Company Inc., 21November 2011.

Anoname, "*Hotspot (Wi-Fi)*", from: http://en.wikipedia.org/wiki/Hotspot_%28Wi-Fi%29, 21 November 2011.Clincy, V.A., Nael Abu-Halaweh.2005. A Taxonomy of Free Network Sniffers for Teaching and Research. *Consortium for Computing Sciences in Collegees, Georgia*, pp. 64-75.

Czajka, Adam & Andrzej Pacut.2002. Neural networks for signature classification and identity verification secure, *Materialy Konferencyjne*. pp: 1 – 7.

Fausett, Laurene V.1993. Fundamentals of Neural Network: Architectures, Algorithms and Applications. New Jersey. Prentice Hall.

Gonzalez, Rafael C., Richard E. Woods, Steven L. Eddins.2009. Digital Image Processing Using Matlab, Second Edition. USA. Gatesmark Publishing.

Kalera, Meenakshi K., Srihari Sargur & Aihua Xu. 2004. Offline Signature Verification and Identification Using Distance Statistics. *International Journal of Pattern Recognition and Artificial Intelligence*. 18: 1339-1360.

Leondes, Cornelius T. 1998. Image Processing and Pattern Recognition. USA. Academic Press.

Kandeeban, S. Selvakani and Rengan S. Rejash. 2010. Integrated Intrusion Detecting System Using Soft Computing. *International Journal of Network Security*. 10:2.

McQuarrie, Allan D. R., Chih-Ling Tsai. 1998. Regression and Time Series Model Selection. Singapore. World Scientific Publishing, Co. Pte. Ltd.

Miyamoto, Sadaaki, Hidetomo Ichihashi and Katsuhiro Honda. 2008. *Algorithms for Fuzzy Clustering – Methods in c-Means Clustering with Application*. Germany.Springer-Verlag Berlin Heidelberg.

Roesch, Martin,1999. *SNORT – Lightwight Intrusion Detection For Network"*, in *Proceeding of LISA '99: 13th System Administration Conference*. Seatle, Washington 7-12th of November 1999, pp 229-237.

Sabourin, Robert. 1997. Off-line Signature Verification: Recent Advanced and Perspective. *Journal of Laboratoire d'Imagerie de Vision et d'Intelligence Artificielle (LIVIA)*. 19: 976-988.