Analisis Sistem Deteksi Anomali Trafik Menggunakan Algoritma *Clustering* Denstream Dengan Modifikasi pada Proses Update *Micro-Cluster*

Analysis Traffic Anomaly Detection System Using Clustering Algorithm With Modified Denstream on Update Process Micro-Cluster

Paundra Dwi Laksono[1], Yudha Purwanto[2], Fiky Yosef Suratman[3]

[1,2,3]Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

[1]phizound@student.telkomuniversity.ac.id, [2]omyudha@telkomuniversity.ac.id, fysuratman@telkomuniversity.ac.id

Abstrak

Sistem deteksi berdasarkan anomali trafik adalah suatu sistem sistem keamanan jaringan yang berfungsi untuk mengetahui adanya keanehan atau gangguan dalam sebuah jaringan internet. Berkembangnya teknologi internet telah meningkatkan jumlah aktivitas masyarakat terhadap penggunaan internet sehingga membuat jumlah user dalam mengakses internet meningkat dan memicu terjadinya kemunculan anomali trafik. Anomali trafik dapat berupa serangan *Distributed Denial of Service* (DDoS). Untuk menangani anomali trafik maka dirasa penting untuk membuat sebuah sistem deteksi anomali trafik yang dapat membedakan antara trafik normal dan trafik serangan. Pada penelitian ini dibangun sebuah metode *Intrusion Detection System* (IDS) dengan teknik *unsupervised learning* yang menggunakan algoritma *clustering*. Pada penelitian ini menggunakan algoritma Denstream dengan modifikasi pada proses *update micro-cluster* untuk menghasilkan deteksi paling baik dengan parameter *purity*. Hasil dari penelitian ini, sistem yang dapat menghasilkan sistem deteksi dengan tingkat rata - rata purity mencapai 97.07%.

Kata Kunci : ***Anomaly Traffic*, DDoS , Denstream, *Clustering*, *Update micro-cluster***

Abstract

Traffic anomaly detection system is based on a system of network security system that serves to detect an oddity or a disruption in the Internet network. Development of internet technology has increased the number of people for the use of internet, so as to make the number of users accessing Internet increases and trigger the emergence of anomalous traffic. Anomaly traffic can be a Distributed Denial of Service (DDoS). To handle the traffic anomaly it is deemed necessary to make a traffic anomaly detection system that can distinguish between normal traffic and attack traffic. In this research constructed a method Intrusion Detection System (IDS) with unsupervised learning techniques that use *clustering* algorithm. In this research, using algorithms Denstream with modifications on *micro-cluster* update process to produce the best detection with purity parameters. Results from this study, a system that can produce a detection system at the level of the average - average purity reached 97.07%.

Keywords: traffic anomalies, DDoS, Denstream, *Clustering*, Update *micro-cluster*

1. Pendahuluan

Distributed Denial of Service (DDoS) adalah suatu jenis serangan terhadap sebuah komputer atau server dengan salah satu cara membanjiri lalulintas jaringan dengan banyak permintaan (*request flooding*) sehingga tidak dapat diakses oleh *user* yang berhak.

Dalam penelitian ini akan menggunakan algoritma pengembangan dari *density-based* DBSCAN yang disebut algoritma Denstream Algoritma Denstream sendiri menggunakan teknik *micro-cluster* yang digunakan untuk mengindentifikasi *cluster* beserta atribut didalamnya dan memproses banyak *outlier* dalam sebuah *data stream*.

Dalam survey [1], algoritma Denstream dapat mengatasi kelemahan yang terdapat pada DBSCAN terhadap outlier. Algoritma Denstream ini merupakan perkembangan dari algoritma DBSCAN dengan penerapan proses *micro-cluster*. Dengan metode ini maka algoritma dapat mengatasi banyaknya data inputan pada data stream.

Fokus penelitian tugas akhir ini menerapkan algoritma Denstream ke dalam sistem deteksi anomali trafik dan penerapan modifikasi pada proses update *micro-cluster* untuk mengeefektifkan perhitungan terhadap *micro-cluster* proses dalam sistem deteksi anomali trafik.

2. Dasar Teori dan Perancangan

2.1. Sistem Deteksi Anomaly

Dalam mendeteksi dan mengatasi anomali yang terdapat dalam jaringan komputer dikenal istilah *Intrusion Detection System (IDS)* dan *Intrusion Prevention System (IPS)* [2]. *Intrusion Detection System (IDS)* bekerja dengan cara dalam sistem tersebut harus memonitor dan mendeteksi adanya serangan (intrusi) dan memberikan peringatan (*alarm*) bila terjadi serangan kepada administrator sehingga serangan tersebut dapat diatasi. Dengan adanya metode IDS maka muncul sebuah pengembangan yaitu *Intrusion Prevention System (IPS)* dimana dalam metode ini memiliki kemampuan memonitor dan mendeteksi yang sama dengan IDS dan dapat secara otomatis menangani serangan yang terdapat dalam jaringan tersebut [2] [3].

2.2. Distributed-Denial of Service (DDoS)

Denial of Service (Dos) merupakan suatu bentuk serangan *flooding* yang bertujuan membuat suatu sumber (*resource)* yang dimiliki suatu komputer target habis dan tidak dapat memberikan layanan kepada pengguna yang sah. *Distributed Denial of Service* (DDoS) adalah salah satu jenis serangan DoS, contoh dari serangan DDoS adalah *TCP SYN flooding*. Target serangan dalam DoS dan DDoS [2] adalah *bandwidth,* dimana *bandwidth* dalam sebuah jaringan tersebut akan dibuat penuh dan sumber daya komputasi pada server maupun node jaringan habis.

Kondisi tersebut disebabkan oleh sumber daya komputasi baik dari proses, *memory*, *buffer* pada *server* maupun *node* jaringan akhirnya menjadi *crash/down* sehingga tidak dapat melayani servis yang diminta *user*.

2.3. Density Based *Clustering*

Clustering merupakan suatu metode dalam data mining yang dapat digunakan untuk mendeteksi suatu anomali trafik. *Clustering* merupakan teknik untuk memisahkan data yang tidak diketahui informasinya kedalam struktur kelompok yang memiliki kemiripan maksimum dengan data lain dalam satu *cluster*. *Density based Clustering* [4] merupakan suatu teknik dalam *clustering* dimana *cluster* terbentuk pada bagian yang memiliki kerapatan tertentu

2.4. Denstream *Clustering*

Salah satu cara untuk mendeteksi anomali trafik adalah dengan menggunakan metode *clustering*. Dalam penelitian ini menggunakan Algoritma *Density-Based Clustering* dengan menggunakan teknik *micro-cluster* [4] yang disebut dengan Algoritma Denstream. Dalam proses untuk melakukan *clustering* pada *Data Stream* maka akan membutuhkan proses *damped window* [6]. Algoritma Denstream menggunakan teknik *micro-cluster* yang digunakan untuk memproses outlier dalam sebuah Data Stream, *micro-cluster* pada algoritma Denstream dibedakan menjadi tiga yaitu Core *micro-cluster*(c-*micro-cluster*) , Potential C-*micro-cluster*(p-*micro-cluster*) , dan Outlier C-*micro-cluster*(o-*micro-cluster*). c-*micro-cluster* merupakan *micro-cluster* utama pada algoritma ini dan c-*micro*-cluter mempunyai tiga atribut utama yaitu weight, center dan radius. Setelah melakukan proses DBSCAN untuk membentuk sebuah *cluster* awal maka akan dilakukan perhitungan weight total sebuah *cluster* dengan perhitungan berikut :

$$w_{total} = \sum_{j=1}^{n} d_{ij} 2^{-\lambda \Delta t} \quad (1)$$

Setelah dilakukan perhitungan weight total maka akan dilakukan perhitungan untuk mencari CF_1, CF_2, Radius dan *Center* untuk *cluster* dengan perhitungan sebagai berikut :

$$CF_1 = \sum_{j=1}^{n} f(t - t_{ij}) P_{ij} \quad (2)$$

$$CF_2 = \sum_{j=1}^{n} f(t - t_{ij}) P_{ij}{}^2 \quad (3)$$

$$C = \frac{\sum_j^n f(t-t_{ij}) x_{ij}}{w_{total}} \quad (4)$$

$$R = \sqrt{\frac{|CF_2|}{w} - \frac{(|CF_1|)^2}{w}} \quad (5)$$

p-micro-cluster merupakan *micro-cluster* yang berpotensi menjadi *cluster* utama maupun menjadi sebuah outlier yang akan dihapus tergantung pada berat dari *p-micro-cluster* sendiri sedangkan *o-micro-cluster* merupakan *micro-cluster* yang berpotensi menjadi p-*micro-cluster* maupun menjadi sebuah *outlier* tergantung

pada berat dari *o-micro-cluster* tersebut. Algoritma Denstream selalu melakukan pertimbangan berat terhadap setiap *micro-cluster*nya apakah *micro-cluster* tersebut layak menjadi *micro-cluster* utama atau hanya akan dihapus karena dianggap hanya sebagai outlier.

Sebelum datangnya set point pada Denstream akan dilakukan perhitungan Tp dimana Tp adalah waktu minimum yang dibutuhkan untuk *p-micro-cluster* berubah menjadi *outlier.* Formula dari Tp adalah sebagai berikut

$$Tp = \left[\frac{1}{\lambda}\log\left(\frac{\beta\mu}{\beta\mu-1}\right)\right] \quad (6)$$

Denstream akan menerima set point dimana digunakan DBSCAN untuk membuat initial set dari *p-micro-cluster* . Saat initial *p-micro-cluster* terbentuk , Denstream menunggu sampai sebuah point datang dari *stream*. Point yang baru datang ini awalnya mencoba bergabung ke *p-micro-cluster* terdekat ketika point tersebut kurang dari radius *p-micro-cluster* terdekat maka dia akan bergabung dengan *p-micro-cluster* tersebut lalu radius,center dan berat dari *p-micro-cluster* akan dilakukan *update*

Apabila point gagal bergabung dengan *p-micro-cluster* , maka point akan mencoba bergabung dengan *o-micro-cluster*. Ketika *o-micro-cluster* yang baru mempunyai berat yang cukup maka *o-micro-cluster* akan menjadi *p-micro-cluster* dan dihapus dari *outlier buffer*. Ketika point tidak bisa bergabung dengan *micro-cluster* lainnya maka Denstream membuat *o-micro-cluster* pada point tersebut dan menempatkannya di *outlier buffer*. Setiap ada point yang datang Denstream secara periodik mengecek dan menghapus semua *o-micro-cluster* yang memiliki berat kurang dari *lower limit of weight. Lower limit of weight* didefinisikan sebagai berikut

$$\xi(t_c, t_o) = \frac{2^{-\lambda(t_c - t_o + T_p)} - 1}{2^{-\lambda T_p} - 1} \quad (7)$$

Dimana t_c adalah waktu sekarang dan t_o adalah waktu yang dibutuhkan untuk membuat *o-micro-cluster*. Setelah semua proses selesai maka akan dilakukan proses Generating *Cluster* dimana proses ini adalah proses terakhir dari algoritma Denstream. Dalam proses ini akan dilakukan pembentukan *cluster* terkahir dengan menggunakan DBSCAN

2.5. Euclidean Distance

Pada pnelitian sebelumnya [5] mendeteksi anomal mendeteksi *anomaly traffic* menggunakan euclidean distance sebagai rumus pengukuran jaraknya, mendapatkan hasil yang cukup baik dalam perhitungannnya. *Euclidean Distance* [8] adalah sebuah formula yang digunakan untuk menghitung jarak antara dua titik (point) dalam *Euclidean space*. Formula ini biasa digunakan untuk menghitung jarak antara titik (point) pada 1-dimensi, 2-dimensi, 3-dimensi sampai n-dimensi. Secara umum formula untuk perhitungan n-dimensi pada *Euclidean Distance* adalah sebagai berikut :

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2} \quad (8)$$

2.6. Dataset DARPA 1998

Pada penelitian ini menggunakan *dataset* yang sudah sering digunakan pada peneletian serupa sebelumnya. Untuk pengujian *traffic* DDoS digunakan *dataset* DARPA 1998 [9], *dataset* tersebut sudah menjalani proses *preprocessing* agar mudah dianalisia. Untuk pengujian metode IDS yang dirancang, dilakukan simulasi menggunakan bahasa pemrograman Java

2.7. Parameter Uji

Beberapa parameter yang digunakan digunakan untuk mengetahui performansi algoritma Denstream dengan modifikasi pada proses *update micro-cluster* dengan menggunakan *euclidean distance* dalam melakukan pembedaan antara *traffic* normal dan *traffic anomaly*. Beberapa parameter awal yang digunakan dalam mengukur keakuratan algoritma sebagai berikut :

Tabel 1. Parameter Uji

Prediksi	**Aktual**	
	Serangan	**Normal**
Serangan	Serangan = Serangan (TP)	Normal = Serangan (FP)
Normal	Serangan = Normal (FN)	Normal = Normal (TN)

True positive (TP) adalah kondisi dimana algoritma mendeteksi data sebagai serangan dan kelanjutan sebenarnya memang data tersebut merupakan serangan. *True nagative* (TN) adalah dimana algoritma mendeteksi data sebagai kondisi normal dan kenyataannya memang data tersebut merupakan data normal. *False positive* (FP) adalah dimana kondisi algoritma mendeteksi data dengan kondisi normal tetapi disebut sebuah serangan. *False negative* (FN) adalah kondisi dimana algoritma melakukan salah deteksi yang menyatakan data dengan kondisi serangan disebut segabagi kondisi normal.

2.8. Purity

Purity adalah ukuran tingkat kemurnian dari sebuah *cluster* dimana *Purity* tersebut menghitung ada atau tidaknya suatu data yang berbeda dalam sebuah *cluster* tersebut dengan perhitungan sebagai berikut :

$$PUR = \frac{\sum_{i=1}^{K} \left| \frac{c_i^d}{c_i} \right|}{K} X100 \quad (9)$$

Dimana $\left|C_i^d\right|$ merupakan jumlah data dengan label tarbanyak, $|C_i|$ merupakan total keseluruhan data yang terdapat dalam *cluster* tersebut dan K merupakan jumlah total *cluster* yang terbentuk

3. Pembahasan

Preprocessing adalah suatu proses untuk normalisasi sebuah data trafik agar mudah/cocok untuk digunakan pada proses pendeteksian. Pada penelitian sebelumnya [10] menggunakan metode *preprocessing* memudahkan menganalisis dan meningkatkan hasil analisis yang dilakukan. Tujuan dalam proses *preprocessing* untuk melakukan mendapatkan fitur yang relevan dari *raw data,* dalam penelitian ini dilakukan pada dataset DARPA 1998. Fitur yang digunakan dapat dilihat pada Tabel 2. Algoritma Denstream modifikasi dapat dilihat pada Algoritma 1, Algoritma 2 dan Algoritma 3. *Dataset* dalam penelitian ini diberi label untuk mempermudah analisa hasil akhir yang didapatkan.

Tabel 2. Ekstraksi Fitur

Nama Fitur	**Jenis Koneksi**	**Penjelasan**
Count	-	Jumlah *traffic* dalam satu window
IP_source	IP Source dan IP Destination sama	Jumlah *traffic* dari IP Source ke IP Destination yang sama
Protocol		Jumlah protocol yang sama
SYN		Jumlah *traffic* "SYN"
ACK		Jumlah *traffic* "ACK"
Port_Out		Jumlah *traffic* menuju ke port out yang sama
Length		Jumlah *traffic* dengan length yang sama
Different_Source	IP Destination sama	Jumlah *traffic* dengan IP Source berbeda
New_IP	-	Jumlah kemunculan IP baru

Algoritma 1 : Merging (DS, CS, eps, β ,μ, λ Point)

1. Masukan point p ke p-*microcluster* cp;
2. ***if*** rp (radius terbaru dari cp) ≤ eps ***then***
3. Gabung p ke cp;
4. ***else***
5. Masukan point p ke o-*microcluster* co;
6. ***if*** ro (radius terbaru dari co) ≤ eps ***then***
7. Gabung p ke co;
8. ***if*** w (weight baru dari co) > βμ ***then***
9. hapus co buat baru p-*microcluster* dari co
10. ***end if***
11. ***else***
12. Buat o-*microcluster* dari p
13. ***end if***
14. ***end if***
15. Simpan sebagai *Cluster* (CS1)

Algoritma 2 : Denstream Modifikasi (DS, CS1, eps, β ,μ, λ)

1. ***get*** $Tp = \left[\frac{1}{\lambda}\log(\frac{\beta\mu}{\beta\mu-1})\right]$
2. Dapatkan titik p selanjutnya dari waktu sekarang pada datastream;
3. ***merging*** (p);
4. ***if*** (t mod tp) = 0 ***then***
5. ***for*** setiap p-*micro-cluster* ***do***
6. ***if*** wp (weight dari cp) < βμ ***then***
7. simpan cp ke Temporary;
8. ***end if***
9. ***end for***
10. ***for*** setiap o-*micro-cluster* Co ***do***
11. ***get*** $\xi(t_c, t_o) = \frac{2^{-\lambda(t_c-t_o+Tp)}-1}{2^{-\lambda Tp}-1}$
12. ***if*** wo (weight dari co) < ξ ***then***
13. simpan co ke Temporary;
14. ***end if***
15. ***end for***
16. ***end if***
17. ***do*** *generating cluster*;
18. Simpan sebagai *Cluster*2 (CS2)

Algoritma 3 : Update *Micro-Cluster* (DS, eps, Temp)

1. ***get*** cp Temporary;
2. ***get*** co Temporary;
3. ***get*** neweps (cp,co);
4. ***for*** setiap (cp,co) **do**
5. ***merging*** with neweps
6. **algoritma Denstream**
7. ***end for***
8. ***do*** *generating cluster*;
9. Simpan sebagai *Cluster*2(CS2)

3.1. Pengujian Dataset DDos

Proses *preprocessing* dahulu dilakukan pada dataset DARPA 1998, dikarenakan dataset darpa masih berupa *raw data*. Tujuan dari proses *preprocessing* untuk medapatkan karekteristik dari *traffic* DDoS sehingga hasil performansi deteksi yang dihasilkan lebih baik. Pengujian dilakukan pada *dataset* normal dan serangan mendapati hasil yang beragam dengan menggunakan masukan sistem seperti pada tabel 3. Hasil pengujian dari *dataset* DARPA 1999 dapat dilihat pada tabel 4, tabel 5, tabel 6, tabel 7 dan gambar 1 dibawah

Tabel 3. Masukan Sistem

Masukan sistem				
eps	*minpts*	λ	β	μ
0.8	3	0.2	2	3

Tabel 4. Pengujian Week 1 Wednesday

Week 1 Wednesday				
Prediksi	Actual			
	Normal	Smurf	Neptune	Back
Normal	4812	53	-	-
Smurf	-	-	-	-
Neptune	-	-	-	-
Back	-	-	-	-
Purity = 99.63%				

Tabel 5. Pengujian Week 2 Friday

Week 2 Friday				
Prediksi	Actual			
	Normal	Smurf	Neptune	Back
Normal	22	-	-	42
Smurf	-	-	-	-
Neptune	-	-	-	-
Back	484	-	-	913
Purity = 87.32%				

Tabel 6. Pengujian Week 6 Tuesday

Week 6 Tuesday					
Prediksi	Actual				
	Normal	Smurf	Neptune	Back	portsweep
Normal	7131	-	-	-	468
Smurf	-	-	-	-	-
Neptune	-	-	-	-	-
Back	-	-	-	-	-
portsweep	-	-	-	-	-
Purity = 93.84%					

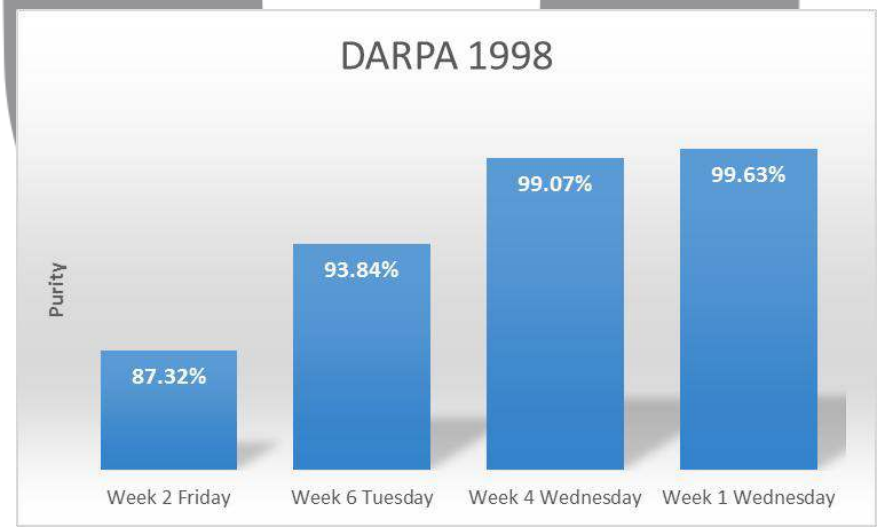
Tabel 7. Pengujian Week 4 Wednesday

Week 4 Wednesday						
	Actual					
Prediksi	Normal	Smurf	Neptune	Back	ipsweep	portsweep
Normal	8451	-	-	-	95	62
Smurf	-	-	-	-	-	-
Neptune	-	-	-	-	-	-
Back	-	-	-	-	-	-
ipsweep	-	-	-	-	-	-
portsweep	-	-	-	-	-	-
Purity = 99.07%						

Pada pengujian dataset *Week 1 Wednesday, Week 2 Friday, Week 6 Tuesday,* dan *Week 4 Wednesday* dapat dilihat dalam tabel 4, tabel 5, tabel 6 dan tabel 7 diatas bahwa setiap dataset memiliki hasil yang berbeda dan beragam, hasil yang diperoleh tersebut berasal dari masing – masing dataset yang memiliki data berbeda.

Pada dataset *Week 1 Wednesday* terdapat trafik normal dan trafik serangan, algoritma Denstream modifikasi dapat mendeteksi *cluster* yang didalamnya terdapat trafik normal dan serangan. Dalam dataset *Week 1 Wednesday* didapatkan hasil berupa banyak *cluster* yang terbentuk adalah sebanyak 3 *cluster* dengan hasil deteksi berupa 4812 trafik normal dan 53 trafik serangan yang dideteksi sebagai trafik normal, dalam hasil deteksi dataset *Week 1 Wednesday* terdapat kesalahan deteksi sebanyak 53 trafik serangan yang dideteksi sebagai trafik normal, kesalah deteksi tersebut bisa dikatakan rendah karena hasil parameter *Purity* yang dihasilkan oleh dataset *Week 1 Wednesday* adalah sebesar 99.63%. Algoritma Denstream dengan modifikasi pada proses *update micro-cluster* dapat mendeteksi mayoritas *cluster* yang berisi data trafik normal dengan baik.

Sedangkan pada dataset *Week 2 Friday* dapat diketahui hasil dari deteksi adalah menghasilkan 4 *cluster* dengan hasil deteksi 22 trafik normal dideteksi dengan benar, 484 trafik normal dideteksi sebagai trafik serangan, 42 trafik serangan dideteksi sebagai trafik normal dan 913 trafik serangan dideteksi dengan benar. Kesalahan deteksi yang terdapat dalam *Week 2 Friday* dikarenakan persebaran data pada dataset tidak memiliki perbedaan yang signifikan antara trafik normal dan trafik serangan, sehingga berpengaruh pada tingkat *Purity*. *Purity* yang dihasilkan sebesar 87.32% pada *Week 2 Friday* ini dapat dikatakan memiliki hasil yang bagus karena dapat mendeteksi mayoritas data trafik serangan pada dataset dan memiliki sedikit kesalahan deteksi. Hasil perbandingan *purity* dapat dilihat pada gambar 1 berikut



Gambar 1. *Purity* Algoritma Modifikasi *Denstream* dengan Dataset DARPA 1998

4. Kesimpulan

Kesimpulan yang didapat dari penelitian ini adalah sistem deteksi anomali trafik menggunakan metode *clustering* dengan algoritma Denstream dengan modifikasi pada *proses update micro-cluster* dapat membedakan antara trafik normal dan trafik serangan dengan baik walaupun terdapat sedikit kesalahan dalam deteksi. Modifikasi yang dilakukan yaitu pada proses *update micro-cluster* dapat mengurangi banyaknya data yang hilang selama proses algoritma dengan begitu algoritma ini dapat menghasilkan junlah *cluster* yang lebih baik. Masukan sistem sangat mempengaruhi pembentukan jumlah *cluster* sehingga mempengaruhi *purity* yang dihasilkan oleh algoritma.

Untuk pengembangan sistem deteksi anomali trafik selanjutnya dapat dilakukan dalam kondisi *real time* dimana sistem dapat langsung mendeteksi anomali trafik yang terjadi dalam sebuah jaringan komputer. Dan menentukan *cluster* yang akan dibentuk dalam algoritma karena jumlah *cluster* yang dihasilkan mempengaruhi waktu yang digunakan dalam proses algoritma.

Daftar Pustaka

[1] R. Xu and D. Wunsch, "Survey of *Clustering* Algorithms," *Neural Networks, IEEE Transactions,* vol. 16, no. 3, pp. 645 - 678, 2005.

[2] Y. Purwanto, Kuspriyanto, Hendrawan and B. Rahardjo, "Traffic Anomaly Detection in DDoS Flooding," *International Conference on Telecommunication Systems Services and Applications (TSSA),* vol. 8, pp. 313-318, 2014.

[3] K. Ramadhani, M. Yusuf and H. E. Wahanani, "Pendeteksian Dini Sserangan UDP Flood Berdasarkan Anomali Perubahan Ttraffic Jaringan Berbais Cusum Algoritm," *Computer security,* 2013.

[4] A. Amini and T. Y. Wah, "A Comparative Study of Density-based *Clustering* Algorihtms on Data Streams : *Micro-clustering* Approaches," *Intelligent Control and Innovative Computing,* pp. 275-287, 2012.

[5] M. Matysiak, "Data Stream Mining Basic Methods and Techniques," 2012.

[6] F. Cao, M. Ester, W. Qian and A. Zhou, "Density-Based *Clustering* over an Evolving Data Stream with Noise," *SIAM Conference Data Mining, Bethesda,* 2006.

[7] G. Münz, S. Li and G. Carle, "Traffic Anomaly Detection Using KMeans *Clustering*," *In GI/ITG Workshop MMBnet,* 2007.

[8] M. K. Amit Singla, "Comparative Analysis & Evaluation of Euclidean Distance Function and Manhattan Distance Function Using K-means Algorithm," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. II, no. 7, pp. 298-300, 2012.

[9] "1998 DARPA Intrussion Detection Evaluation Data Set," LINCOLN LABORATORY MASSACHUSSETS INSTITUTE OF TECHNOLOGY, [Online]. Available: http://www.ll.mit.edu/ideval/data/1998data.html. [Accessed 26 5 2015].

[10] Made Indra Wira Pramana, Yudha Purwanto and Fiky Yosef Suratman, "DDoS Detection Using Modified K-Means *Clustering* with Chain Initialization Over Landmark Window," *International Confrence on Control, Electronics, Renewable Energy, and Communication,* 2015.