ANALISIS SELF-SIMILAR UNTUK SISTEM DETEKSI ANOMALI DENGAN ESTIMASI HURST EXPONENT MENGGUNAKAN METODE R/S

SELF-SIMILAR ANALYSIS FOR ANOMALY DETECTION SYSTEM WITH HURST EXPONENT ESTIMATION USING R/S METHOD

Hanif Nurohman[1], Yudha Purwanto[2], Hafidudin[3]
[1,2]Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom
[3]Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom
Nurohman.hanif@gmail.com[1], om_yudha@yahoo.co.id[2], hafidudin@gmail.com[3]

Abstrak

Serangan *Denial of Service (DoS)* merupakan sebuah fenomena yang sedang menjadi topik hangat belakangan ini. Intensitas serangan *DoS* semakin meningkat setiap harinya dengan ditemukannya jenis serangan baru dengan tipe yang sama yaitu *Distributed Denial of Service (DDoS)*. Kedua serangan tersebut menyerang korban dengan cara membanjiri kanal trafik korban dengan banyak kiriman paket pada satu waktu. Hal ini membuat aliran paket yang menuju komputer korban menjadi tersendat sehingga memungkinkan koran tidak mendapatkan paket yang diinginkan karena padatnya trafik pada jaringannya.

Metode *LRD* dan *Self-Similarity* merupakan metode yang cocok dengan sifat trafik jaringan yaitu *variability* dan *burstiness*. Pada metode *LRD* dinyatakan bahwa trafik jaringan menunjukan sebagai memori jangka panjang dimana tingkah laku tersebut berkorelasi melalui waktu yang terpisah jauh. Hal ini menunjukan bahwa setiap paket yang dikirim dan diterima memiliki korelasi dan hubungan tertentu meskipun waktu antar kedatangan paket terpisah cukup jauh. Dalam *DDoS* kemungkinan korelasi dan hubungan tersebut tidak terjadi dalam waktu yang dekat sekalipun. Ini membuat pendeteksian menggunakan *DDoS* menggunakan *LRD* menjadi salah satu metode terbaik. *Self-Similarity* adalah sebuah skala dari invarian yang selalu memiliki kesamaan, jadi ketika *self-similarity* digunakan kedalam pemodelan trafik, maka terlihat plot dari trafik tersebut memiliki kesamaan, walaupun secara waktu memiliki perbedaan.

Kata Kunci : *DDoS, LRD, Self-Similarity, burstiness*

Abstract

Denial of Service (DoS) is a phenomeno which is becoming a hot topic lately. The intensity of DoS attacks is increasing every day with the discovery of a new type of attack with the same type which is Distributed Denial of Service (DDoS). Both attack the victims by flooding with a lot of traffic channels packet at a time. This makes the flow of packets to the victims computer becomes choked and victim don't get the desired package because the density of traffic on its network.

LRD and self-similarity methods is suitable to the network traffic behavior which is variability and burstiness. In LRD method stated that network traffic shows a long-term memory in which behavior is correlated through time apart. This shows that every packet sent and received has a particular relationship although correlation and inter-arival time is quite far apart. In DDoS possibility of correlation and the relationship is not going to happen in the near future though. It makes use of DDoS detection using LRD be one of the best method. Self-similarity is a scale of invariant which is always have the same, so when self-similarity used into traffic modeling, it will show a plot of the traffic will have in common, even though it has a different time.

Keywords : DDoS, LRD, Self-similarity, burstiness

1. Pendahuluan

Pada saat ini jaringan internet merupakan jaringan yang paling banyak dipakai oleh semua orang di seluruh dunia. Hal ini terjadi karena internet dapat menyediakan berbagai macam layanan yang dibutuhkan oleh semua orang. Di internet kita dapat bertukar data dengan mudah dan cepat. Selain karena kemudahan dan kecepatannya internet juga dikenal sebagai jaringan yang fleksibel. Sehingga banyak sekali ditemukan hal-hal baru di jaringan internet. Seperti misalnya, aplikasi-aplikasi baru yang dapat menunjang pekerjaan, foto-foto unik, video menarik dan lainnya.

Namun di balik kelebihan tersebut, internet memiliki kekurangan yang cukup besar. Karena keterbukaannya internet sangat mudah untuk diserang oleh berbagai macam serangan. Biasanya serangan ini dilakukan oleh *hacker*. Korban dari serangan ini biasanya seorang pengguna atau sebuah alamat web sehingga pengguna tersebut tidak dapat mengakses alamat web yang diinginkan. Jika seorang pengguna dijadikan sebagai target serangan maka serangan yang paling terkenal dan sering dipakai adalah *Denial of Service (DOS)* dan *Distributed Denial of Service (DDoS)*.

Serangan DOS adalah jenis serangan terhadap sebuah computer atau server. DOS bekerja di dalam jaringan internet dengan cara menghabiskan sumber (resource) yang dimiliki oleh komputer tersebut sampai komputer tersebut tidak dapat menjalankan fungsinya dengan benar sehingga secara tidak langsung mencegah pengguna lain untuk memperoleh akses layanan dari komputer yang diserang tersebut [1]. Dalam sebuah serangan Denial of Service, *hacker* mencoba untuk mencegah akses seorang pengguna terhadap sistem atau jaringan dengan menggunakan beberapa cara yaitu *traffic flooding, request flooding*, dan mengubah informasi konfigurasi sistem atau bahkan perusakan fisik terhadap komponen dan server. Sasaran serangan ini adalah *link* / *bandwidth* untuk membuat sumber daya *bandwidth* penuh dan sumber daya komputasi (proses, *memory*, *buffer*) pada *server* maupun *node* jaringan untuk membuat sistem pengolah kehabisan sumber daya dan akhirnya *crash/down* sehingga tidak dapat melayani servis yang diminta *user* [2].

Pada penelitian ini digunakan basis *self-similarity traffic anomaly* untuk mendeteksi anomaly. Basis tersebut memiliki empat langkah. Pertama *self-similarity* digunakan untuk mencari jumlah *hurst exponent.* Setelah itu dilakukan perhitungan statistik yang merupakan karakteristik dari *self-similarity.* Karakteristik *self-similar* tersebut terdiri dari tiga hal yaitu peningkatan stasioner, perubahan skala spasial, dan proses agregat. Kelebihan dari metode ini adalah kemampuannya yang dinamis, sehingga dapat beradaptasi dengan berbagai jenis trafik. Hal ini didasarkan pada asumsi bahwa trafik normal memiliki sifat-sifat *self-similarity.* Sehingga walaupun tidak memiliki basis data, metode ini tetap dapat mendeteksi serangan DDoS pada suatu trafik.

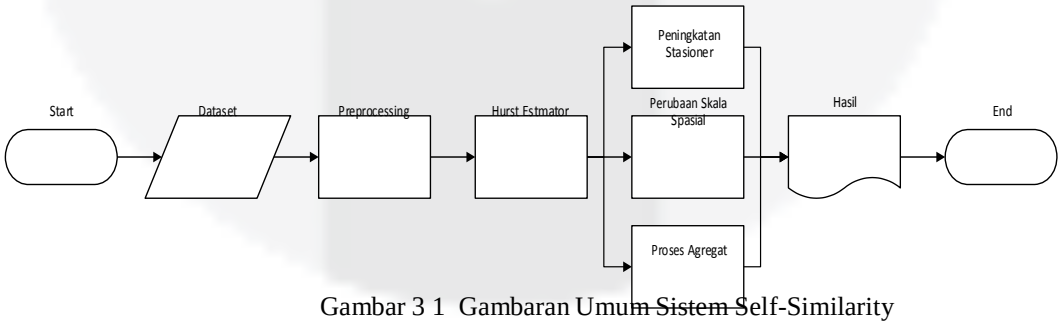
Tujuan dari penelitian ini adalah agar sifat-sifat *self-similarity* dapat mengenali tiga jenis trafik yang diujikan yaitu trafik normal, trafik DDoS dan trafik *flashcrowd*. Selain itu hasil estimasi *hurst exponent* memberikan nilai antara 0,5 dan 1 untuk pengujian dataset normal dan nilai diluar range tersebut untuk pengujian DDoS. Sehingga dapat menjadi dasar yang baik untuk langkah-langkah selanjutnya dalam karakteristik *self-similar.* Dan juga pengukuran jarak untuk pengujian dataset normal memiliki nilai yang lebih kecil dibandingkan dengan pengukuran jarak untuk pengujian dataset DDoS.

Metodologi penelitian yang digunakan adalah pertama dengan melakukan studi literatur yaitu mempelajari literatur-literatur yang ada sesuai dengan permasalahan yang akan dibahas meliputi, konsep deteksi anomali trafik, teori serangan *flooding traffic (DDoS)* dan *flash-crowd*, konsep *preprocessing*, teori estimasi *hurst exponent*, teori dasar dari karakteristik *self-similarity.* Selanjutnya analisis terhadap kebutuhan dan pemodelan sistem untuk proses deteksi anomali trafik lalu dilanjutkan dengan perancangan dan analisis menggunakan tools untuk sistem deteksi anomali trafik dan yang terakhir adalah uji performansi dan analisis hasil penelitian

2. Perancangan

2.1. Gambaran Umum Sistem

Secara umum gambaran sistem dapat digambarkan dalam gambar 1 proses perancangan sistem yang dilakukan diantaranya adalah dengan melakukan persiapan data, melakukan *preprocessing* untuk mendapatkan fitur yang digunakan dalam proses selanjutnya. Dilanjutkan dengan estimasi hurst eksponen. Kemudian dilakukan pengecekan pada tiga ciri khas dari *self-similarity* yaitu peningkatan stasioner, perubahan skala spasial dan proses agregat.



Gambar 3 1 Gambaran Umum Sistem Self-Similarity

2.2. Dataset

Terdapat tiga dataset yang digunakan pada simulasi ini, yaitu dataset normal menggunakan dataset CAIDA 2014, dataset DDoS menggunakan dataset CAIDA 2007 dan dataset *flashcrowd* menggunakan dataset *world cup 1998.* Dataset normal dan dataset DDoS merupakan *raw* trafik dengan format data

pcap. Untuk dapat diolah maka data tersebut harus diubah terlebih dahulu ke dalam format excel ataupun csv dengan menggunakan menu import pada aplikasi *wireshark* dengan terlebih dahulu melakukan *filter* pada *IP destination* untuk analisis 5 menit dan proses agregat dengan *windowing* 15 menit. Dan juga melakukan *filter* pada *IP source* dan *IP destination* untuk analisis *perflow.*

Sedangkan pada dataset *flashcrowd* terdapat perbedaan dalam mengolah dataset. Perbedaan data terdapat pada format yang didapatkan untuk dataset tersebut yaitu berupa *rar binary*. Dimana format tersebut tidak bisa langsung dibaca oleh program di dalam komputer. Sehingga dibutuhkan *tools* untuk dapat merubah data tersebut menjadi format excel ataupun csv. *Tools* tersebut adalah *command of format (COF).* Karena masalah privasi, alamat IP yang tertera dari COF bukanlah IP yang sebenarnya tapi bisa dipastikan semua alamat IP yang muncul terwakili oleh alamat IP yang disediakan oleh data tersebut.

2.3. *Preproccessing*

Preprocessing merupakan tahap selanjutnya setelah memperoleh dataset trafik yang digunakan. Setiap dataset diproses untuk mendapatkan masukan langkah-langkah selanjutnya. Masukan yang diperlukan adalah jumlah paket yang datang pada suatu *IP destination* setiap detik, jumlah *size* dari paket yang datang pada suatu *IP Destination* setiap detik, dan jarak antara waktu kedatangan paket. Jarak antara waktu kedatangan paket dan jumlah *size* per paket yang datang pada trafik tersebut digunakan pada proses estimasi hurst eksponen. Sedangkan jumlah paket pada setiap detik dan jumlah *size* pada setiap detik digunakan dalam pengecekan sifat-sifat *self-similarity.*

Pada pengecekan karakteristik *self-similarity* dilakukan tiga analisis yaitu analisis 5 menit, analisis proses agregat dengan *windowing* 15 menit dan analisis *perflow.* Sehingga *preprocessing* untuk setiap dataset dilakukan sebanyak tiga kali untuk tiga analisis yang berbeda. Proses yang dilakukan pada *preprocessing* ini bertujuan sama, yaitu untuk mencari masukan pada proses selanjutnya. Perbedaan terdapat pada dataset mentah yang dimasukan ke dalam proses ini. Dimana terdapat perbedaan *filtering* yang dilakukan pada *IP Source* dan *IP destination.* Dibawah ini merupakan algoritma dari tahap *preprocessing.*

2.4. *Hurst Estimator (Rescaled Adjusted Range)*

Salah satu teknik estimasi nilai *hurst exponent* adalah *Rescaled Adjusted Range (R/S).* Hurst mengemukakan sebuah normalisasi pengukuran nondimensional yang dapat menggambarkan dan mengantisipasi adanya perubahan-perubahan dalam mengestimasi *hurst exponent* [3]. Metode yang dikembangkan oleh Hurst ini sangat sering dipakai untuk mengestimasi nilai dari *hurst exponent*. Hal ini disebabkan metode *R/S* ini sangat mudah untuk diterapkan serta variable-variabel yang dibutuhkan sangat mudah untuk didapatkan. Kelebihan dari metode ini adalah tidak bergantung pada distribusi data marginal. Dan dari penelitian sebelumnya, metode *R/S* merupakan metode yang paling baik dalam domain waktu.

Metode *R/S* sendiri diawali dengan mencari nilai *R(n)* yang pada dasarnya merupakan range dari sebuah dataset. Dimana rumusnya adalah :

$$R(n) = \max \Delta_k - \min \Delta_k \qquad (1)$$

dimana

$$\Delta_k = \Sigma X_i - kX \qquad (2)$$

rumus diatas membedakan range pada proses R(n) dengan range pada umumnya dimana range pada umumnya merupakan pengurangan antara nilai maksimum dengan nilai minimum. Setelah didapatkan nilai R(n), maka langkah selanjutnya adalah membandingkanya dengan S(n). Dimana rumus S(n) adalah :

$$S(n) = \left[\frac{1}{(n-1)\Sigma(X_i - n)^2}\right] \qquad (3)$$

Setelah didapat nilai perbandingan antara R(n) dengan S(n) maka langkah selanjutnya adalah membuat sebuah diagram *log-log scale*. Dimana sumbu y merupakan log{M[R(n)/S(n)]} dan sumbu x merupakan log (n). pada tahap ini didapatkan dua garis yang memiliki kenaikan secara konstan. Kemiringan kedua garis ini yang digunakan dalam menestimasi *hurst exponent* dengan menggunakan metode *least-square* [3].

.

2.5. Pengecekan sifat-sifat *self-similarity*

Setelah mengumpulkan data-data yang dibutuhkan dalam penelitian ini, maka pada tahap selanjutnya adalah membuat mekanisme dalam mendeteksi serangan DDoS. Dimana masukan pada tahap ini berupa permintaan paket dari *IP source* per satuan waktu kepada target yang direpresentasikan dengan pengenalan karakteristik sifat-sifat *self-similarity* untuk setiap fitur yang didapatkan. Pada *self-similarity* terdapat beberapa parameter yang digunakan sehingga sebuah trafik dikatakan memiliki karakteristik *self-similar*. Parameter-parameter tersebut diantaranya adalah peningkatan stasioner, perubahan spasial terhadap waktu, dan proses agregat [1].

2.4.1. Peningkatan Stasioner

Untuk sebuah permodelan trafik, sebuah proses X(t) harus memilki sifat-sifat stasioner dimana struktur dari sebuah data harus invarian atau tidak terjadi perubahan ketika dilakukan pergeseran waktu [4]. Tanpa adanya sifat stasioner, maka sebuah model trafik kehilangan kegunaannya sebagai pemberi gambaran dari sebuah fenomena dalam trafik. Proses stasioner merupakan model yang paling baik untuk menggambaran *long memory* [6]. Diasumsikan bahwa X(t) merupakan jumlah *size* atau jumlah paket yang dibawa oleh beberapa paket pada saat waktu t. Atau dengan kata lain merupakan nilai kumulatif dari sebuah trafik selama waktu pengamatan. Dalam hal ini waktu pengamatan sebesar satu detik secara terus menerus.

Pada pengecekan peningkatan stasioner masukan berupa jumlah *size* atau jumlah paket yang masuk pada detik ke t dan jumlah *size* atau jumlah paket dari seluruh paket pada *window* yang dilakukan pengujian. *Size* tersebut didapatkan dari hasil *preprocessing* dataset. Untuk nantinya diproses dengan rumus sebagai berikut :

$$\{X(t+\Delta t)-X(\Delta t)\}=\{X(t)-X(0)\} \quad (4)$$

Atau

$$X(t)=X(t+t) \quad (5)$$

Tanda sama dengan pada persamaan (4) dan persamaan (5) menyatakan bahwa kedua persamaan tersebut harus memiliki kesamaan dalam tingkat *finite dimensional distribution* [1]. Dalam penghitungan *finite dimensional distribution*, keluaran yang didapatkan dari persamaan (5) dijumlahkan secara terus menerus dengan data keluaran dari proses *finite dimensional distribution* itu sendiri. Sehingga ketika terjadi pergeseran waktu pada trafik normal tidak berpengaruh besar pada keluaran dari proses peningkatan stasioner. Sedangkan pada trafik anomali sangat berpengaruh karena adanya lonjakan pada jumlah paket dan jumlah *size* yang datang pada waktu tertentu.

2.4.2. Perubahan Skala Spasial

Perubahan Skala Spasial merupakan perubahan skala dari data trafik yang dijadikan masukan. Hal ini menandakan bahwa pada trafik normal terdapat kesamaan dalam jumlah paket dan jumlah *size* yang datang pada satu waktu pengamatan. Kesamaan itu tetap ada meskipun telah terjadi perubahan skala dalam data trafik tersebut. Sedangkan pada trafik anomali tidak terdapat kesamaan tersebut karena adanya perbedaan jumlah paket dan *size* yang signifikan pada awal dan pertengahan terjadinya anomali tersebut. Pada langkah ini dilakukan perubahan skala 1:2 pada data trafik yang dimasukan ke dalam proses perubahan skala spasial. Untuk nantinya dimasukan ke dalam rumus :

$$\{X(t)\}=\{\sqrt{2}X(t)\} \quad (6)$$

Sama dengan proses sebelumnya tanda sama dengan pada persamaan (6) menyatakan bahwa kedua persamaan tersebut harus memiliki kesamaan dalam tingkat *finite dimensional distribution*. Sehingga ketika terjadi ketika dilakukan perubahan skala pada trafik normal tidak berpengaruh besar pada keluaran dari proses perubahan skala spasial [3]. Sedangkan pada trafik anomali sangat berpengaruh karena adanya lonjakan pada jumlah paket dan jumlah *size* yang datang pada waktu tertentu.

2.4.3. Proses Agregat

Proses Agregat merupakan keseluruhan dari data trafik yang masuk ke dalam *IP destination*. Pada proses ini dilakukan perbandingan nilai dari proses agregat dengan nilai dari *autocovariance* dari sebuah data trafik. Sebelum langkah ini, pemenuhan dari setiap persamaan menandakan bahwa data tersebut memiliki sifat stasioner yang lemah. Sebuah proses harus memiliki sifat *self-similarity wide sense* atau *self-similarity* pada orde kedua. *Self-similarity* pada orde kedua dapat menangkap atau cocok dengan karakteristik dari *autocovariance* dalam proses agregat [7]. Pada pengecekan *Agregated Process*, dicek apakah sebuah dataset memiliki karakteristik *self-similar wide-sense*. Fitur yang dibutuhkan merupakan variansi dari kumpulan dataset pada suatu *window* yang sedang diteliti. Untuk nantinya dimasukan ke dalam rumus :

$$\gamma(k) = \frac{\sigma^2}{2}[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}] \quad (7)$$

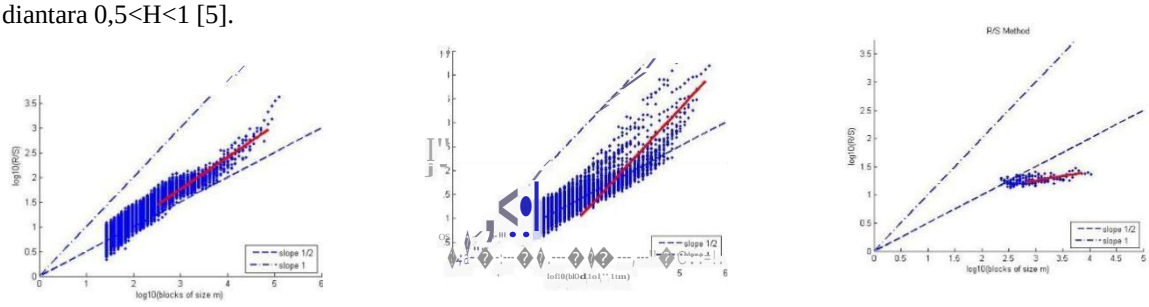
Pada dua langkah sebelumnya, persamaan yang ada dibandingkan dengan terlebih dahulu mengubah nilai keluaran menjadi nilai *finite dimensional distribution*. Sedangkan pada proses agregat perbandingan yang dilakukan adalah dengan membandingkan nilai keluaran langsung dengan nilai *autocovariance* [7]. Hal ini dikarenakan *self-similarity* orde kedua dapat menangkap karakteristik dari nilai *autocovariance. Self-similarity* orde kedua telah menjadi *framework* dominan untuk permodelan trafik jaringan [1].

3. Pembahasan

Penggunaan deteksi trafik anomali dengan analisis tingkah laku *self-similarity* dalam internet trafik merupakan sesuatu hal yang baru. Karenanya ketiga ciri khas *self-similarity* memiliki peranan penting dalam menentukan trafik normal dan trafik anomali. Input dari simulasi ini adalah dataset CAIDA 2014 untuk pengecekan trafik normal dan trafik CAIDA 2007 DDoS dimana kedua dataset tersebut telah diakui validitasnya. Sementara untuk dataset *flashcrowd* digunakan dataset *world cup 1998*. Analisis terbagi ke dalam empat bagian yaitu estimasi hurst eksponen, peningkatan stasioner, perubahan skala spasial, dan proses agregat.

3.1. Estimasi Hurst eksponen

Gambar 1(A) merupakan hasil dari estimasi hurst eksponen untuk dataset normal. Masukan dari estimasi hurst eksponen untuk dataset normal ini merupakan jarak antar kedatangan paket dan *size* dari dataset normal CAIDA 2014 dalam rentang waktu 0-60 detik yang menuju *IP destination* 189.116.21.85. Estimasi hurst eksponen untuk dataset normal, didapatkan nilai hurst eksponen sebesar 0,6455. Nilai dari hurst eksponen yang didapatkan sesuai dengan penelitian sebelumnya dimana trafik normal memiliki nilai ekponen *hurst* diantara 0,5<H<1 [5].



(A) (B) (C)

Gambar 1 Pengujian estimasi hurst eksponen untuk dataset normal(A), dataset DDoS(B), dan dataset flashcrowd(C)

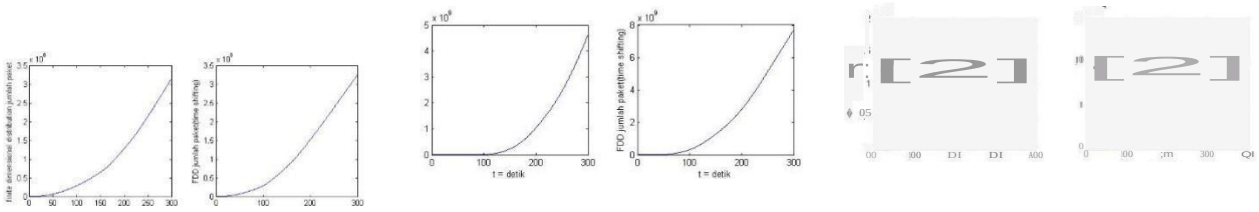
Gambar 1(B) merupakan hasil dari estimasi hurst eksponen untuk dataset DDoS. Sama seperti estimasi dataset normal, masukan untuk estimasi *hurst* ini adalah jarak kedatangan paket dan *size* dari dataset *DDoS* CAIDA 2007 dalam rentang waktu 0-60 detik. Estimasi hurst eksponen untuk dataset *DDoS*, didapatkan nilai hurst eksponen sebesar 1,0326. Nilai dari hurst eksponen yang didapatkan sesuai dengan penelitian sebelumnya dimana trafik anomali memiliki nilai hurst eksponen diluar dari range hurst eksponen untuk dataset normal [6].

Gambar 1(C) merupakan hasil dari estimasi hurst eksponen untuk dataset *flashcrowd*. Seperti pada dataset sebelumnya, masukan untuk estimasi *hurst* ini adalah jarak kedatangan paket dan *size* dari dataset *World Cup*

1998 dengan rentang waktu 0-60 detik. Estimasi hurst eksponen untuk dataset *flashcrowd*, didapatkan nilai hurst eksponen sebesar 0,1586. *Flashcrowd* merupakan fenomena yang termasuk anomali namun tidak dikategorikan sebagai serangan pada suatu trafik internet. Sehingga nilai yang didapatkan sesuai dengan penelitian sebelumnya dimana trafik anomaly memiliki nilai hurst eksponen diluar range hurst eksponen untuk dataset normal.

3.2. Peningkatan Stasioner

Pada pengujian peningkatan stasioner dilakukan pengujian sampel pada 0-5 menit awal pada dataset normal yang menuju *IP destination* 189.116.21.85.. Sedangkan pada dataset DDoS dan *flashcrowd* dilakukan pengujian saat 5 menit awal terjadi fenomena tersebut yang menuju pada satu *IP Destination*, Dengan masukan berupa jumlah paket yang masuk ke dalam trafik selama 5 menit.



(A) (B) (C)

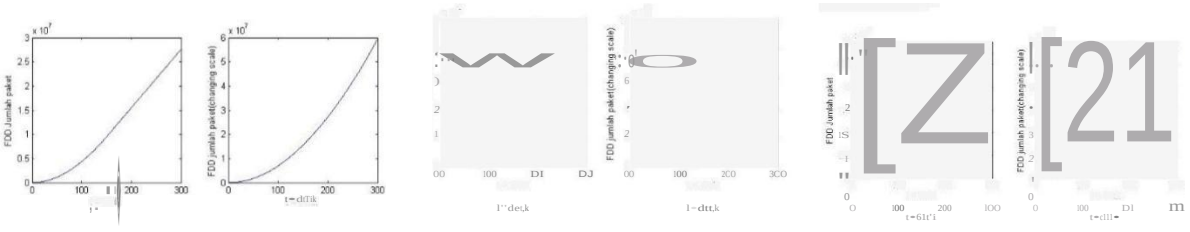
Gambar 2 Pengujian proses peningkatan stasioner untuk dataset normal(A), dataset DDoS(B), dataset flashcrowd(C)

Gambar 2(A) merupakan hasil dari simulasi trafik normal untuk proses peningkatan stasioner. Gambar tersebut merupakan hasil distribusi *finite dimensional* dari persamaan (5). Dimana masukan dari proses ini adalah jumlah paket yang masuk setiak satu detik. kita dapat melihat pada gambar di atas bahwa kedua grafik memiliki garis yang serupa. Hal ini membuktikan bahwa trafik normal telah memenuhi salah satu elemen dalam tingkah-laku *self-similar*. Hal ini menandakan bahwa pada setiap detiknya trafik normal memiliki rata-rata jumlah paket yang konstan. Hal ini membuat tidak adanya pengaruh berarti ketika dilakukan pergeseran waktu terhadap data trafik normal tersebut. Untuk identifikasi dilakukan penghitungan jarak dengan menggunakan metode *mahalanobis distance.* Untuk identifikasi dataset normal, nilai dari *mahalanobis distance* adalah 18,7669.

Bila kita bandingkan dengan hasil simulasi trafik DDoS untuk proses peningkatan stasioner memiliki garis yang berbeda. Dimana masukan dari proses ini adalah jumlah paket yang masuk setiap satu detik dari datset DDoS CAIDA 2007 Hal ini menunjukan bahwa ketika sebuah proses DDoS mengalami pergeseran waktu mengakibatkan perbedaan secara kumulatif yang sangat besar. Ini mengindikasikan bahwa serangan DDoS mengirim banyak paket dalam waktu bersamaan. Hasil simulasi dari dataset DDoS dapat dilihat pada gambar 2(B). nilai dari *mahalanobis distance* untuk pengujian dataset DDoS adalah 35,0057.

Pada gambar 2(C) terdapat perbedaan dari grafik pengujian tersebut. Hal ini menandakan fenomena *flashcrowd* mempengaruhi trafik normal dengan menigirimkan banyak paket meskipun paet tersebut tidak masuk ke dalam trafik secara tiba-tiba namun dengan peningkatan jumlah paket secara bertahap. nilai dari *mahalanobis distance* untuk pengujian dataset *flashcrowd* adalah 17,5053.

3.3. Perubahan Skala Spasial



(A) (B) (C)

Gambar 3 Pengujian proses perubahan skala spasial untuk dataset normal (A), dataset DDoS(B), dataset flashcrowd(C)

Pada pengujian perubahan skala spasial ini sama seperti pengujian peningkatan stasioner. Yaitu dilakukan pengujian yaitu pengujian sampel pada 0-5 menit awal untuk dataset normal yang menuju pada *IP destination* 189.116.21.85. Sedangkan untuk dataset DDoS dan *flashcrowd* pengujian sampel

dilakukan pada 5 menit awal terjadinya DDoS dan *flashcrowd*. Masukan untuk pengujian dataset normal adalah dataset normal CAIDA 2014 sedangkan untuk masukan pengujian dataset DDOS adalah dataset DDoS CAIDA 2007. Untuk pengujian *flashcrowd* digunakan dataset *world cup 1998.*

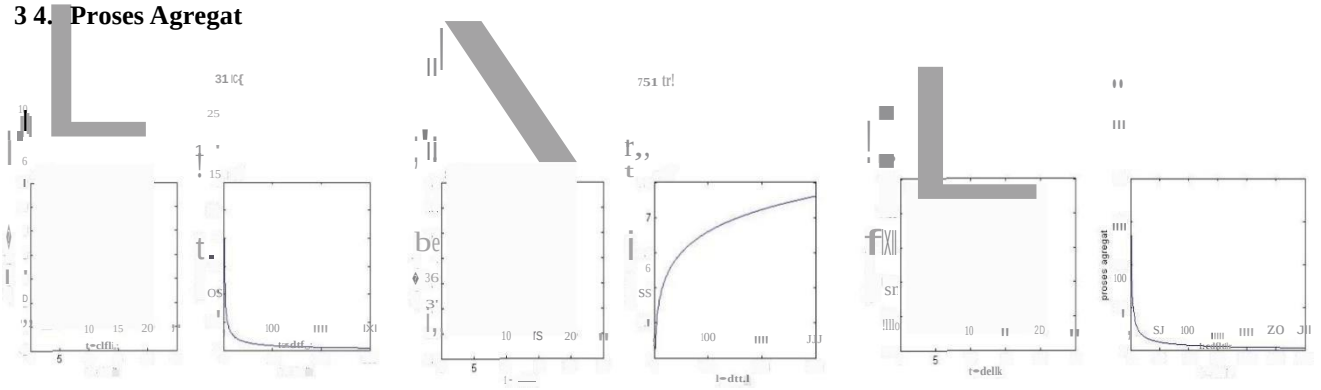
Untuk proses perubahan skala spasial memiliki kesamaan langkah-langkah pengujian dengan proses peningkatan stasioner. Kesamaan tersebut pada distribusi finite dimensional dari setiap persamaan. Untuk pengujian ini masukan masih sama yaitu berupa jumlah paket per detik dari dataset normal CAIDA 2014. Perbedaanya terdapat pada persamaan yang diuji. Untuk perubahan skala spasial persmaan yang dibandingkan adalah persamaan (6) dimana dicari terlebih dahulu distribusi finite dimensional dari kedua persamaan tersebut.

Gambar 3(A) merupakan hasil dari simulasi trafik normal untuk proses perubahan skala spasial. Gambar 3(A) menunjukan bahwa meskipun terjadi perubahan skala, namun tetap terdapat kesamaan dengan nilai data sesungguhnya. Hal ini berarti *self-similarity* mengabaikan perubahan skala waktu dan terdapat kesamaan meskipun terjadi perubahan waktu yang teratur. Dan untuk pengujian dataset normal, seperti terlihat di gambar telah memenuhi persamaan dari proses perubahan skala spasial. Untuk identifikasi dilakukan penghitungan jarak dari grafik tersebut dengan menggunakan metode *mahalanobis distance*. Untuk pengujian dataset normal didapatkan nilai dari *mahalanobis distance* adalah sebesar 27,7116.

Bila kita bandingkan dengan pengujian dataset DDoS untuk proses perubahan skala spasial memiliki garis grafik yang berbeda. Masukan masih berupa jumlah paket setiap detiknya dari dataset DDoS CAIDA 2007. Hasilnya dapat dilihat pada gambar 3(B). Pada gambar di bawah ini, dataset DDoS telah mengurangi karakteristik dari self-smilarity. Hal ini berarti dataset DDoS tidak memiliki kesamaan ketika dilakukan perubahan waktu secara teratur. Adanya lonjakan jumlah paket secara tiba-tiba mempengaruhi terjadinya perbedaan grafik tersebut. Nilai *mahalanobis distance* dari pengujian DDoS adalah sebesar 28,9515.

Gambar 3(C) merupakan hasil dari simulasi trafik *flashcrowd* untuk proses perubahan skala spasial. Sama seperti pengujian *DDoS* terdapat perbedaan antara kedua grafik. Hal ini dikarenakan adanya peningkatan jumlah paket pada fenomena *flashcrowd*. Hal ini membuat penskalaan yang dilakukan memiliki data yang timpang antara data awal sebelum dan sesudah terjadi *flashcrowd*. Untuk pengujian *flashcrowd* didapatkan nilai *mahalanobis distance* sebesar 36,8535.

3 4. Proses Agregat



(A) (B) (C)

Gambar 4 Pengujian proses agregat untuk dataset normal(A), dataset DDoS(B), dataset flashcrowd(C)

Pada pengujian perubahan skala spasial ini sama seperti pengujian peningkatan stasioner yaitu dilakukan pengujian sampel pada 0-5 menit awal dari setiap masing-masing dataset yang menuju pada satu *IP Destination* 189.116.21.85. Untuk dataset normal digunakan dataset CAIDA 2014, untuk dataset DDoS digunakan dataset CAIDA 2007, sedangkan untuk pengujian *flashcrowd* digunakan dataset *world cup 1998.*

Untuk proses agregat ada sedikit perbedaan dengan kedua langkah sebelumnya dalam membedakan trafik normal dan trafik anomali. Dalam proses agregat, masih membandingkan dua persamaan seperti pada proses sebelumnya. Namun persamaan yang dibandingkan adalah nilai pasti yang didapatkan tanpa harus diubah ke distribusi *finite dimensional.* Untuk proses agregat persamaan (7) harus mengikuti bentuk dan fungsi *autocovariance.*

Gambar 4(A) menunjukan hasil dari pengujian trafik normal untuk proses agregat. Pada gambar tersebut dapat dilihat bahwa *second order self-similarity* mengikuti bentuk dari *autocovariance.* Untuk identifikasi dilakukan penghitungan jarak dengan menggunakan *mahalanobis distance*. Untuk pengujian dataset normal didapatkan nilai *mahalanobis distance* sebesar 5,0191.

Seperti yang terlihat pada gambar 4(B) di sebelah kiri merupakan hasil simulasi persamaan untuk trafik DDoS dan gambar sebelah kanan merupakan fungsi *autocovariance*. Grafik tersebut menunjukan bahwa hasil persamaan mengikuti bentuk dari fungsi *autocovariance*. Bila dibandingkan dengan simulasi DDoS untuk

proses agregat, hasil menunjukan perbedaan bentuk antara persamaan dengan fungsi *autocovariance*. Hal ini menunjukan serangan DDoS mengurangi kehadiran elemen *self-similar* dalam sebuah trafik. Seperti pada gambar di atas menunjukan serangan DDoS tidak dapat memnuhi *self-similar* pada orde kedua dimana *self-similar* pada orde kedua telah menjadi *framework* dominan untuk memodelkan trafik jaringan. Nilai *mahalanobis distance* dari pengujian dataset DDoS adalah 70,6250.

Pada gambar 4(C) dapat dilihat hasil dari pengujian untuk dataset *flashcrowd*. Grafik tersebut menunjukan bahwa hasil persamaan masih mengikuti pola dari *autocovariance.* Perbedaannya dengan pengujian DDoS dan normal adalah fungsi *autocovariance* tidak mendekati nilai nol melainkan berada lebih tinggi dari nilai nol. Hal ini yang membedakan pengujian *flashcrowd* dengan pengujian normal dan DDoS. Hal ini menandakan bahwa meskipun fenomena *flashcrowd* merupakan sebuah anomaly namun memiliki karakteristik seperti trafik normal. Sama seperti pengujian dataset normal dan DDoS dilakukan penghitungan jarak dengan menggunakan *mahalanobis distance.* Untuk pengujian dataset *flashcrowd* didapatkan nilai *mahalanobis distance* sebesar 1,0936.

4. Kesimpulan

1. Sifat-sifat dari self-similarity dapat mengenali tiga jenis trafik yang diujikan, hal itu dapat dilihat dari nilai-nilai yang didapatkan dari pengukuran jarak dengan menggunakan metode *mahalanobis distance.*
2. Nilai *mahalanobis distance* dari pengujian dataset DDoS memiliki nilai yang lebih besar dibandingkan dengan nilai dari pengujian dataset normal. Sedangkan untuk *flashcrowd* memiliki nilai yang lebih kecil bila dibandingkan dengan pengujian dataset normal.
3. Sistem deteksi berdasarkan sifat *self-similarity* dapat diapllikasikan dengan baik dengan catatan lebar *windowing* sangat berperan terhadap keberhasilan dan tingkat akurasi sistem.
4. Nilai dari *hurst eksponen* untuk pengujian dataset normal, DDoS dan *flashcrowd* berturut-turut adalah 0.6455, 1.0326 dan 0.1586. hal ini sesuai dengan penelitian sebelumnya yang menyebutkan kondisi trafik normal memberikan nilai antara 0,5 hingga 1 dan nilai diluar range tersebut untuk pengujian dataset anomali.
5. Pengukuran jarak untuk pengujian dataset normal dapat dijadikan patokan untuk mengidentifikasi anomali yang terjadi pada trafik. Hal ini dikaenakan adanya perbedaan nilai *mahalanobis distance* pada pengujian ketiga dataset uji.

Daftar Pustaka

[1] Y. Purwanto, K. H. and B. R. , "Survey : Metode dan Kemampuan Sistem Deteksi Anomali Trafik," *Security and Protection*, 2014.

[2] T. Karagiannis and M. Faloutsos, "SELFIS : A Tool for Self-Similarity and Long-Range Dependence Analysis," *Workshop on Fractals and Self-Similarity in Data Mining*, 2002.

[3] O. I. Sheluhin, S. M. Smolskiy and A. V. Osin, Self-Similar Processes in Telecommunications, Chicester: John Wiley & Sons Ltd., 2007.

[4] D. Brignoli, "DDoS Detection Based On Traffic Self-Similarity," 2008.

[5] T. Karagiannis, M. Molee and M. Faloudtsos, "Long-Range Dependence Ten Years of Internet Modelling," *IEEE Internet Computing*, vol. 4, pp. 1089-1096, 2004.

[6] J. Beran, "Statistical Methods fr Data With Long-Range Dependence," *Statistical Science*, vol. 7, no. 4, pp. 404-416, 1994.

[7] W. E. Leland, W. Wilinger, M. S. Taqqu and D. V. Wilsoon, "On The Self-Similar Nature of Ethernet Traffic," *Computer Communication Review*, pp. 203-213, 1989.