Sistem Deteksi Serangan Ddos pada *Software Defined Network* Menggunakan Metode *Entropy*

M. Hafiz Hawarizmi*[1], M Teguh Kurniawan[2], Muhammad Fathinuddin[3]
[1,2,3]Program Studi S1 Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom
E-mail: *[1]**hafizhawarizmi@student.telkomuniversity.ac.id**,
[2]teguhkurniawan@telkomuniversity.ac.id, [3]muhammadfathinuddin@telkomuniversity.ac.id

Abstrak

Teknologi semakin berkembang seiring dengan berkembangnya jaringan komputer. Dalam merancang infrastruktur jaringan yang baik, dibutuhkan arsitektur jaringan yang bersifat dinamis, serta mudah untuk beradaptasi dan dikelola untuk penyesuaian hardware dan software. Untuk mencapai hal tersebut, diperlukan suatu konsep yang dikenal dengan nama Software Defined Network (SDN). Pada jaringan SDN sangat mungkin untuk administrator melakukan penyediaan jaringan dengan cepat tanpa perlu mengkonfigurasi jaringan secara manual karena terpusat di satu controller. Namun SDN juga memiliki beberapa kekurangan yaitu single of failure dan rentan terhadap serangan, salahsatu serangan yang rentan terhadap SDN yakni rentan terhadap serangan DDoS. Untuk mencegah serangan DDoS terjadi, salahsatu hal yang harus dilakukan adalah dengan mendeteksi serangan DDoS yang muncul. Untuk mendeteksi DDoS terdapat beberapa cara atau metode yang dapat diterapkan seperti menggunakan metode machine learning atau statistik. Pada penelitian ini dilakukan penelitian menggunakan metode statistik untuk melakukan deteksi terhadap serangan DDoS yang muncul. Salah satu metode deteksi serangan DDoS statistik yang akan diteliti yaitu melakukan deteksi serangan menggunakan entropy. Untuk mendeteksi serangan menggunakan entropy diperlukan sebuah nilai threshold, windows size, serta count, sebagai parameter untuk mendeteksi kemunculan serangan DDoS. Pada penelitian ini didapatkan sebuah accuracy tertinggi 72.85714285714285% dari 30 kali percobaan menggunakan entropy sebagai metode untuk melakukan deteksi serangan.

***Kata Kunci*:** *SDN, Entropy, DDoS, windows size. Threshold, count.*

1. PENDAHULUAN

Teknologi pada saat ini semakin terus berkembang. Seiring berkembang pesatnyateknologi, diiringi dengan pesatnya jumlah *user* dan *hardware* yang membuat jaringan konvensional bersifat tidak fleksibel dan efisien untuk dilakukan integrasi terhadap jaringan terbaru. Dalam pengoperasiannya, jaringan tradisionalbukan merupakan program yang mudah untuk di lakukan program ulang. Hal tersebutlah yang membuat para peneliti, mengembangkan teknologi terbaru pada jaringan yang terintegrasi dan berbasis *software* [1]

Software Defined Network (SDN) merupakan paradigma jaringan baru yang bertujuan untuk mengubah arsitektur jaringan. Pada jaringan SDN sangat mungkin untuk *administrator* melakukan penyediaan jaringan dengan cepat tanpaperlu mengkonfigurasi jaringan secara manual. Dengan cara yaitu memisahkan antara *control plane* dengan *data plane.* pada SDN *administrator,* tidak perlu mengkonfigurasi perangkat jaringan satu persatu yang sudah ada. Dikarenakan dengan adanya SDN, *administrator* memperoleh kontrol secara penuh pada keseluruhan jaringan di satu titik yang berguna untuk melakukan penyederhanaanterhadap pengoperasian jaringan. [2]

SDN memiliki 3 layer utama yaitu *Application Layer, Control Layer, danInfrastructure layer.* Pada bagian *Control layer.* Merupakan bagian inti pada SDNsebab, *control layer* bertugas untuk mengelola arus lalu lintas pada paket jaringan, kemudian meneruskan paket, dan melakukan pengambilan keputusan terkait perutean, berdasarkan *programming.* Penjelasan singkatnya yaitu *controller* memberikan kebijakan tinggi serta memisahkannya menjadi kebijakan rendah yang akan di install pada *switch,* lalu *switch* bertugas untuk meneruskan paket sesuai kebijakan yang telah ditetapkan oleh *controller.* Sehingga SDN dapat melakukan perancangan, kemudian membangun, hingga mengelola jaringan yangbesar. Untuk mengelola jaringan yang besar terdapat beberapa tantangan yang harus dipertimbangkan Ketika melakukan penerapan SDN yaitu mengenai *availability, performance, scalability, dan security.* [3]

Konsep pada jaringan SDN memiliki kelebihan dan kekurangan, pada segi kelebihannya yaitu memudahkan pengembangan serta eksperimen pada protokol yang baru, mudah dikelola dan memudahkan jaringan Ketika beradaptasi saat terjadi perubahan infrastruktur. (Ramadhan, Primananda, Yahya, 2018) Sedangkan dari segi kekurangan yang terdapat pada SDN yaitu dari *control plane* nya itu sendiri, dikarenakan SDN merupakan jaringan yang bersifat tersentralisasi. (Saputra, Negara, Sanjoyo, 2019) Selain itu, SDN memiliki kekurangan pada sistem algoritma nya, pada algoritma SDN memerlukan memori yang cukup besaruntuk melakukan proses arahan agar berfungsi dengan baik. (Khairi, Ariffin, Muazzah, Latiff, 2021) pada sistemnya, SDN rentan terkena serangan seperti *Denial Of service (DoS)* dan *Defined Denial Of service (DDoS)* yang biasa melakukan serangan terhadap ketersediaan dari jaringan, sehingga pada jaringan tersebut tidak dapat menyediakan layanan atau tidak dapat melayani permintaan. [4]

Pada kekurangan SDN disebutkan salah satu kekurangannya bahwa SDN rentan terhadap serangan DDoS, DDoS merupakan serangan yang terdistribusi yang mempunyai tujuan untuk menghabiskan bandwith atau sumber daya yang tersediapada tujuan yang akan diserang dengan cara membanjiri server, jaringan yang bertautan, serta perangkat jaringan yang traffic nya tidak sah. [5]

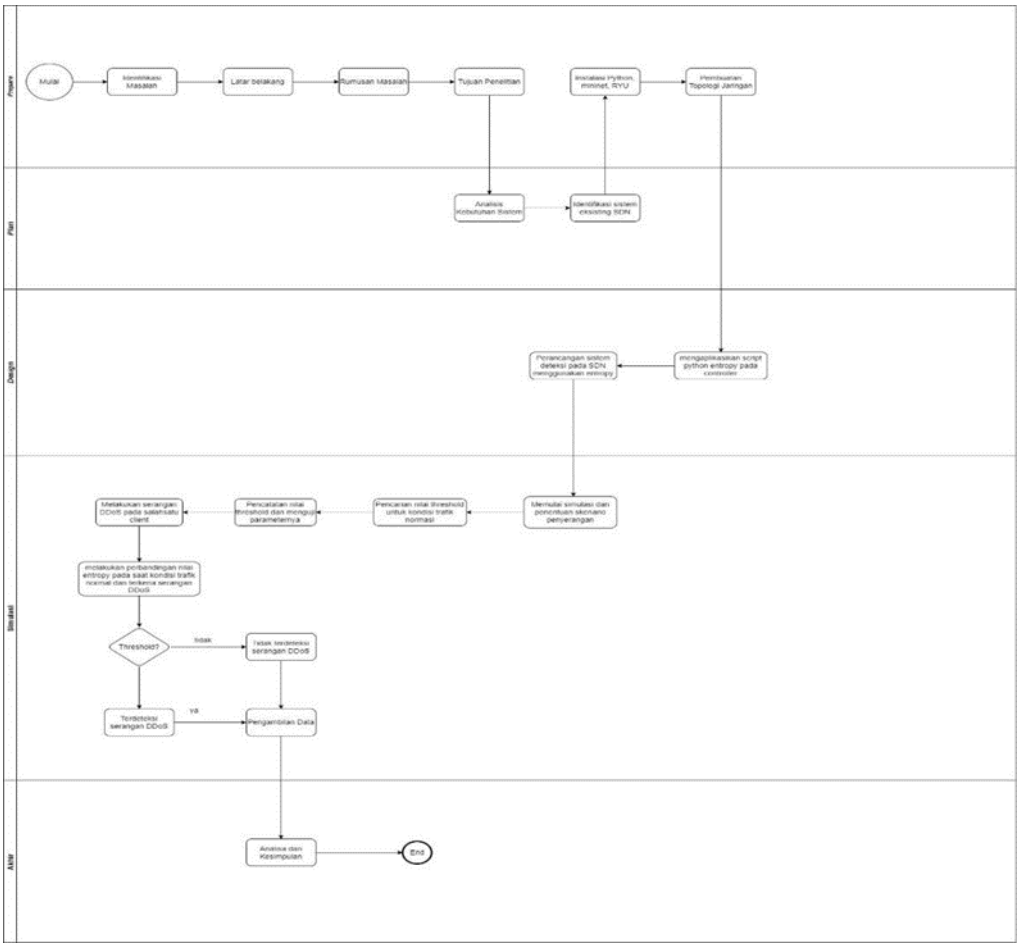
Serangan DDoS biasanya memanfaatkan *protocol*UDP yang bersifat *connectionless* dengan tujuan untuk menyerang target. Kemudian paket data serangan lainnya juga banyak dikirimkan pada target yang akan diserang supaya paket tersebut bisa membanjiri komputer target. padaSebagian kasus tertentu komputer berubah menjadi *hang* ketika paket data serangan tersebut dikirimkan. [6]

Untuk mengatasi permasalahan terkait serangan DDoS, pada tugas akhir ini akanmerancang sistem deteksi DDoS menggunakan *Entropy* yang akan digunakan untuk mengatasi serangan yang terjadi pada SDN. Pada penelitian (Marilanda, Widiastuti, Sumadi, Nastiti, Faiqurrahman, 2019) [7] menyebutkan bahwa *Entropy* merupakan metode yang digunakan dalam proses mendeteksi serangan DDoS dengan cara mengecek tingkat keacakan dari 250 paket pertama, dengan menyeleksi probabilitas kemunculan pada paket berdasarkan pada variabel yang terdapat pada IP sumber serta IP tujuan.

Berdasarkan penelitian tersebut, penulis berinisiatif untuk meneliti terkait deteksidan mitigasi serangan DDoS pada SDN menggunakan metode *Entropy* dengan mencoba menambahkan apa yang telah peneliti (Marilanda, Widiastuti, Sumadi, Nastiti, Faiqurrahman, 2019)[7] tambahkan.

2. METODE PENELITIAN

Metode Berikut merupakan langkah - langkah sistematika penelitian yang dapat di ilustrasikan pada gambar III.1:



Gambar 1. Sistematika Penelitian

Penjelasan sistematika penelitian yang digunakan adalah sebagai berikut:

1. Tahap *prepare*

Tahap penelitian ini merupakan tahapan untuk melakukan identifikasi terhadap permasalahan yang nantinya akan menjadi latar belakang terhadap adanya penelitian ini. Setelah latar belakang tersusun, maka dapat ditentukan untukperumusan masalah serta tujuan penelitian yang dituju mengenai sistem deteksi dan mitigasi serangan DDoS pada SDN menggunakan *Entropy*. Kemudian pada tahapan ini terdapat dilakukannya persiapan terhadap praktik penelitian yaitu melakukan instalasi *Phyton*, *mininet*, *RYU* serta melakukan pembuatan topologi jaringan.

2. Tahap *plan*

Pada tahapan *plan* yaitu dilakukannya analisa mengenai kebutuhan sistem yang dibutuhkan serta melakukan identifikasi terhadap kondisi eksisting sistem pada SDN. Hasil yang telah didapat dari identifikasi tersebut, dapat memberikan gambaran mengenai apa saja yang diperlukan untuk merancang apa saja yang dibutuhkan pada penelitian ini.

3. Tahap *design*

Tahap *design* merupakan tahapan setelah dilakukannya tahapan *prepare* dantahapan *plan*, pada tahapan ini yaitu tahapan dilakukannya pengaplikasian *script phyton* pada *Entropy* dan perancangan sistem deteksi menggunakan *Entropy* pada SDN.

4. Tahap simulasi

Setelah melakukan pengaplikasian script *phyton* pada *Entropy* dan perancangan sistem, pada tahap ini dilakukanlah simulasi dan pengujian mengenai pencarian nilai *threshold* pada *Entropy*, melakukan uji parameter, mecoba melakukan ngumpulan data serta melakukan perbandingan nilai *Entropy* pada saat kondisi normal dan terkena serangan DDoS

5. Tahap akhir

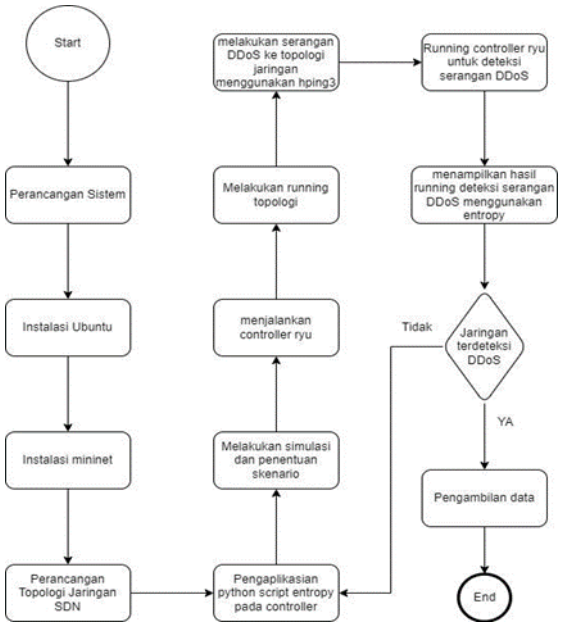
Setelah tahap *prepare* sampai tahap simulasi dilakukan , kemudian menuju pada tahap akhir pada penelitian ini yaitu tahap akhir. Tahap akhir merupakan tahap dilakukannya analisa pada penelitian yang telah dilakukan serta kesimpulan dari keseluruhan penelitian yang telah diteliti.

3. HASIL DAN PEMBAHASAN

3.1. Pembahasan Perancangan

Bab ini menjelaskan tentang perancangan dari sistem yang dibangun pada tugas akhir. Penjelasan lebih detail mengenai konsep untuk deteksi serangan DDoS menggunakan *Entropy* pada jaringan SDN. mulai dari pembuatan topologi sampai melakukan deteksi serangan DDoS.

3.1.1. Perancangan Sistem



Gambar 2. Perancangan Sistem

Berdasarkan Gambar IV.1 Perancangan Sistem. dibawah ini peneliti jabarkan mengenai detail yang dituliskan dengan 7 poin utama yaitu:

1. Menginstalasi perangkat yang dibutuhkan seperti *ubuntu*, *mininet* serta *controller* yaitu

menggunakan *RYU controller*, *wireshark* serta, *hping3*untuk pengiriman paket serangan.

2. Membuat topologi jaringan pada direktori *ubuntu mininet*/custom. Pada tahap ini akan membuat koneksi untuk seluruh *switch* dan *controller*. Topologi ini dibangun menggunakan 2 *switch* dan 2 *host*. Dengan ketentuanyaitu 15 *host* untuk *switch* 1 serta 15 *host* untuk *switch* 2.

3. Tahap selanjutnya yaitu menjalankan *controller RYU* dan melakukan *run* topologi pada *controller RYU*.

4. Selanjutnya yaitu menjalankan *controller RYU*, untuk mendeteksi serangan DDoS menggunakan *Entropy*

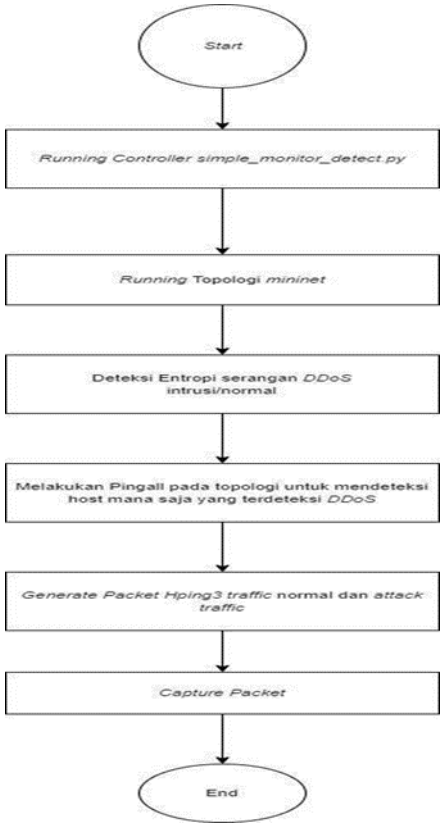
5. Kemudian hasil deteksi DDoS ditampilkan pada terminal *controller RYU* melakukan *run*

6. Selanjutnya yaitu paket di *generate* menggunakan *Hping3* serta di *Capture* menggunakan *wireshark*

7. Tahap terakhir yaitu, analisis pengujian deteksi serangannya

3.1.2. Menjalankan Controller Dan Generate Packet Hping3

Controller dan *hping3* merupakan *tools* yang penting bagi SDN jika hendak melakukan deteksi serangan DDoS. Hal ini didasarkan pada fungsi *controller* yaitu sebagai otak dari suatu sistem SDN yang terhubung, sedangkan *hping3* merupakan *tools* yang berfungsi untuk meng-*generate packet* adapun *packet* yangdi*generate* merupakan *packet traffic* normal ataupun *attack traffic*.



Gambar 3. Alur *Controller* RYU dan *hping3*

Penjelasan mengenai alur *Controller RYU* dan *hping3* yang berjalan pada sistemSDN, yaitu sebagai berikut:

1. Tahap pertama, *host* melakukan *running controller* digabungkan dengan filedeteksi *Entropy* DDoS yang telah dibuat yaitu *simple_monitor_detect*.py.

2. Tahap kedua yaitu menjalankan topologi *mininet* yang akan terkoneksidengan *controller*

3. Tahap ketiga yaitu melakukan deteksi *Entropy* serangan DDoS.

4. Tahap keempat yaitu melakukan *ping* semua *host* topologi dengan perintah"*ping*all' untuk mendeteksi *host* yang terdeteksi serangan DDoS.

5. Tahap kelima yaitu meng-*generate packet* serangan menggunakan *hping3* untuk *traffic* normal dan *attack traffic*.

6. Tahap terakhir yaitu melakukan *Capture packet* pada *wireshark* untukmenganalisis *packet* yang sudah di *generate*

3.2.2. Akurasi Entropy

Akurasi *Entropy* merupakan tingkat pendekatan terkait pengukuran kuantitas*Entropy* dengan nilai sebenarnya. Berikut merupakan rumus untuk mendeteksi akurasi :

Detection *Accuracy* : $\frac{TP+TN}{TP+TN+FP+FN}$

Keterangan:

- *True Positive* (TP) merupakan jumlah catatan serangan yang telah diklasifikasikan sebagai bentuk serangan

- *True Negative* (TN) merupakan jumlah catatan trafik normal yang telahdiklasifikasikan tidak adanya serangan.

- *False Positive* (FP) merupakan jumlah catatan *traffic* normal yang diklasifikasikan sebagai bentuk serangan

- *False Negative* (FN) merupakan jumlah serangan yang sebenarnya namundiklasifikasikan sebagai *traffic* normal.

3.2. Implementasi dan Pengujian

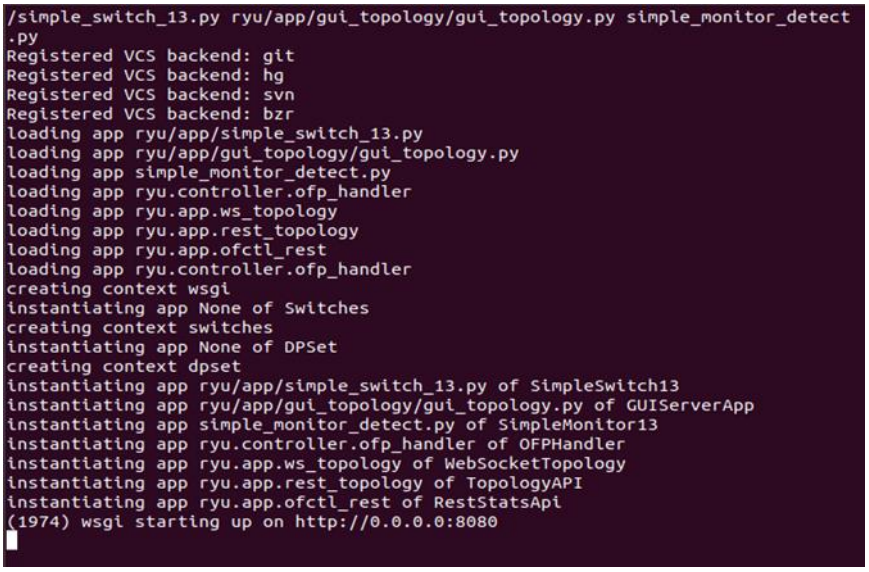
3.2.1. Implementasi Ryu Controller

Pada subbab ini, dilakukan implementasi *RYU controller*. *RYU controller* digunakan sebagai sistem yang pada penelitian ini berfungsi sebagai sistem yangdigunakan untuk mendeteksi serangan DDoS menggunakan *Entropy* serta menjalankan topologi yang *host* nya akan dideteksi *host* mana saja yang terserangoleh DDoS.

Berikut merupakan perintah yang digunakan untuk melakukan implementasi *RYUcontroller* :

```
root@hafiz-virtualbox:/home/hafiz/RYU# sudo RYU-manager –observe-links
RYU/app/simple_switch_13.pyRYU/app/gui_topology/gui_topology.py
simple_monitor_detect.py
```

1. *Running controller RYU* untuk deteksi serangan DDoS

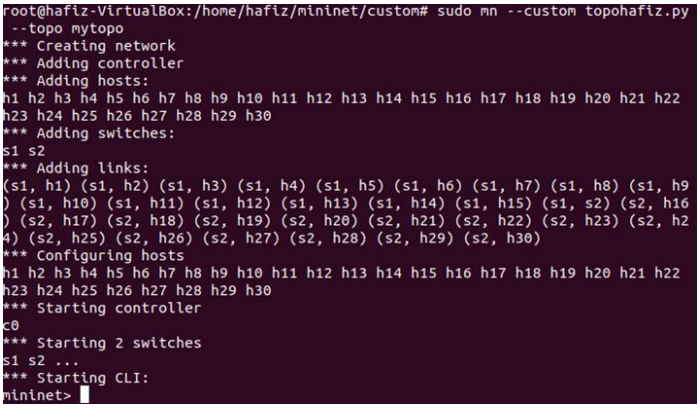


Gambar 4. *Running controller RYU*

Pengujian ini dilakukan untuk menjalankan *controller RYU* yang bertugas untuk mendeteksi serangan DDoS. Untuk mendeteksi serangan DDoS dilakukan *running Source code* pada *controller* yaitu *simple_monitor_detect*.py.

2. *Running* Topologi menggunakan *controller*

```
root@hafiz-virtualbox:/home/hafiz/mininet/custom# sudo mn – custom
topohafiz.py –topo mytopo –controller remote,ip=127.0.0.1
```



Gambar 5. *Running* Topologi

3. Tampilan ketika Deteksi serangan DDoS terjadi

Gambar 6. Tampilan Deteksi Serangan DdoS

3.2.2. Skenario Pengujian

Skenario yang akan dicoba pada penelitian ini yaitu terdapat 4 skenario. Skenario pertama yaitu skenario dengan trafik normalyaitu dengan tanpa serangan, sedangkan untuk skenario 2 yaitu dengan adanya serangan dengan pengiriman paket sebanyak 250, 500, dan 1000 paket. Untuk pengujiannya pada skenario pertama yaitu dengan *host* 1 mencoba mengirimkan paket tanpa adanya serangan kepada seluruh *host*, sedangkan untuk *host* 2 yaitu dengan mengirimkan 250, 500, dan 1000 paket ke *host* terakhir yaitu *host* 30. Kemudian untuk skenario yang ke 3 yaitu pengujian dari penyerangan yang berada pada *switch* yang sama dengan masing – masing penyerang mengirimkan 500 paket serangan kepada *host* 30. Yang terakhir skenario 4 yaitu pengujian dari penyerangan yang berada pada *switch* yang berbeda dengan masing – masing penyerang mengirimkan 500 paket serangan kepada *host* 30.

Tabel 1. Skenario Pengujian

No.	Skenario	Terdeteksi/ Normal	Host	Paket yang dikirimkan	Keterangan
1	Skenario 1	Tidak ada serangan	*Host* 1 ke seluruh *host*	1000 paket	Pada Skenario ini dilakukan pencarian nilaipada kondisi trafik normal dengan tanpa serangan. Paket data disebar 1000 paket dengan *window size 26* dikirim ke seluruh *host* yang dianalisa pada *host* 30.
2	Skenario a	Ada serangan	*host* 2 ke *host* 30	250 paket	Pada skenario ini dilakukan pengamatan pada trafik normal dengan adanya serangan sebesar 25% yaitu 250 paket serangan yang dikirim dari *host* 2 ke *host* 30 yang menyebabkan *switch* tidak mengenalinya.

	Skenario b	Ada serangan	*host* 2 ke *host* 30	500 paket	Pada skenario ini dilakukan pengamatan pada trafik normal dengan adanya serangan sebesar 50% yaitu 500 paket serangan yang dikirim dari *host* 2 ke *host* 20 yang menyebabkan *switch* tidak mengenalinya.
3	Skenario 3	Ada serangan	*Host* 2dan *host* 5 ke *Host* 30 Secara bersamaan	1000 paket	Pada skenario ini dilakukan pengamatan pada trafif normal dengan adanya serangan sebesar 100% yaitu 1000 paket yangdikirim dari *host* 2 dan

3.2.3. Analisis

1. Mencari nilai threshold *entropy*

Pada penelitian ini, nilai threshold diambil berdasarkan penelitian dari beberapa jurnal yang telah peneliti teliti, setelah melalui beberapa tahapan untuk menentukan nilai threshold yang akan peneliti ambil, peneliti memutuskan untuk mengembangkan penelitian [1] dengan judul penelitian yaitu Analisis Kinerja Jaringan Pada Software Defined Network Terhadap Serangan DDoS dan Deteksi Serangan Menggunakan *Entropy*. Alasan peneliti mengembangkan nilai *threshold* dari penelitian (Sanubari, 2019) tersebut adalah terkait dari nilai *threshold* yang akan berpengaruh terhadap *windows size* serta *count* untuk menghasilkan keakuratan *Accuracy* deteksi serangan *Entropy* yang peneliti lakukan.

Berdasarkan beberapa jurnal yang telah peneliti teliti, peneliti memutuskan untukmengembangkan penelitian terkait nilai *threshold* yang telah peneliti [1] dapatkan. Nantinya peneliti akan menemukan *output* dari penelitian ini terkait windows size, *count*, dan *Accuracy* tertinggi yang telah penelitian (Sanubari, 2019) lakukan.

Tabel 2. Analisis nilai threshold *entropy*

No.	**Parameter**	**Penjelasan**	**Rumus**
1	*Mean*	Mean merupakan nilai rata-rata dari banyaknya data. Nilai *mean* dapat ditentukan dengan cara membagi jumlah data dibagi banyak data.	$\bar{x} = \frac{\sum_{i=1}^{n} f_i x_i}{\sum_{i=1}^{n} f_i}$
2	Standar	banyaknya data. Nilai *mean* dapat ditentukan dengan cara membagi	
	Deviasi	jumlah data dibagi banyak data.	
4	*Confidence interval min*	*Confidence interval min* merupakan batasbawah nilai darisuatu sampel rata-rata	-
5	*Confidence interval max*	*Confidence interval max* merupakan batas nilai dari suatu sampel rata rata	-

Berikut merupakan tabel perbedaan nilai *traffic* normal dan maksimun pada *traffic* serangan untuk mendapatkan nilai *threshold* yang didapatkan :

Tabel 3. nilai *threshold*

	Normal *Traffic*	**25% *Attack***
Mean	1,059368	0,787320
Standard deviasi	0,061600	0,357098

Confidence interval	0,005195	0,081918
Convidence Intervalmax	1,064563	0,869239
Convidence Intervalmin	1,054172	0,705402
Perbedaan trafik normal minimum dan *attack traffic maximum*	0,184932	
Threshold	0,87	

Menurut peneliti [1] untuk menentukan *threshold* pada *Entropy* harus terdapat pebandingan nilai entropi untuk kondisi trafik normal dan trafik dengan kondisi serangan sebanyak 25%. Pada tabel tersebut menunjukan bahwa perbandingan antara kondisi *traffic* normal dengan *Entropy* saat adanya serangan 25%. Dari perhitungan tersebut didapatkan rata-ratauntuk kondisi *traffic* normal sebanyak 1.05936807614 serta rata-rata untuk adanya serangan 25% sebanyak 0,787320. Hal ini dapat disimpulkan bahwa dapat terlihat entropi mengalami penurunan sebesar 0.27204729662 yang disebabkan terdepat penambahan *traffic* data pada *controller*. Untuk mencari *threshold* pada entropi, maka dilakukan untuk perhitungan terkait standar deviasi dan *confidence interval*.Pada perhitungan tersebut didapatkan standar deviasi untuk *traffic* normal sebanyak 0,061600 serta standar deviasi untuk *traffic* adanya serangan 25%sebanyak 0,357098. *Confidence interval* untuk *traffic* normal sebanyak 0,05195, *interval max* sebanyak 1,064563, serta interval min sebanyak 1,054172. *Confidence interval* dengan trafik serangan 25% sebanyak 0,081918, *confidence interval max* sebanyak 0,869239, serta *confidence interval min* sebanyak 0,705402. Untuk nilai *threshold* yang diperoleh yaitu sebesar 0,87 yang berasal dari *confidence interval max* dan kondisi serangan 25%.

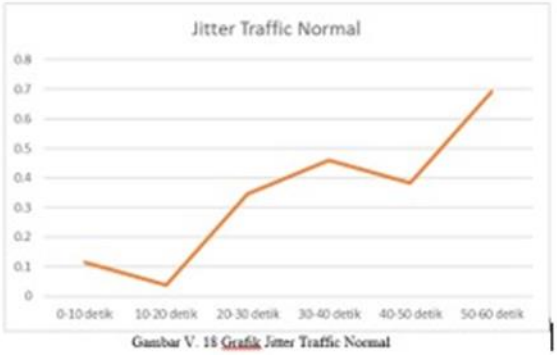
Tabel V. 5 *Delay Traffic* Normal

No.	Waktu per Detik	*Delay* rata-rata paket masuk per detik
1	0-10	0.651648
2	10-20	0.217216
3	20-30	0.1954943
4	30-40	0.2606591
5	40-50	0.3258239
6	50-60	0.3909886

Gambar V. 17 Grafik Delay Traffic Normal

Tabel V. 6 Jitter Traffic Normal

No.	Waktu per Detik	*Jitter* rata-rata paket masuk per detik
1	0-10	0.115093
2	10-20	0.38364
3	20-30	0.345278
4	30-40	0.460371
5	40-50	0.383643
6	50-60	0.690557



Gambar V. 18 Grafik Jitter Traffic Normal

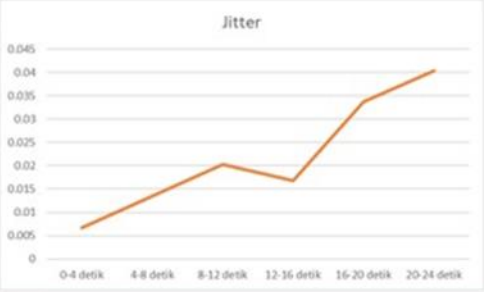
P-ISSN: 2089-676X
E-ISSN: 2549-0796

Tabel V. 7 *Delay 250* paket serangan

No.	Waktu per Detik	*Delay* rata-rata paket masuk per detik
1	0 - 4 detik	1.52543
2	4 - 8 detik	0.30503
3	8 - 12 detik	0.76272
4	12 - 16 detik	0.61017
5	16 - 20 detik	1.27119
6	20 - 24 detik	0.91526

Tabel V. 8 Jitter 250 paket serangan

No.	Waktu per Detik	*Jitter* rata-rata paket masuk per detik
1	0 - 4 detik	0.006734
2	4 - 8 detik	0.013468
3	8 - 12 detik	0.020202
4	12 -16 detik	0.016835
5	16 - 20 detik	0.033671
6	20 - 24 detik	0.040405

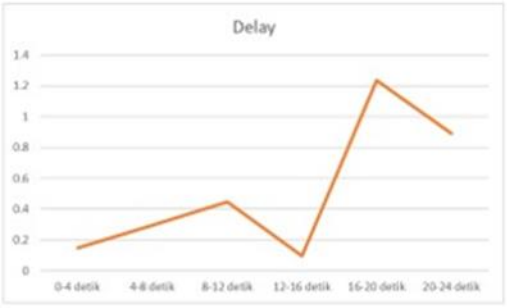


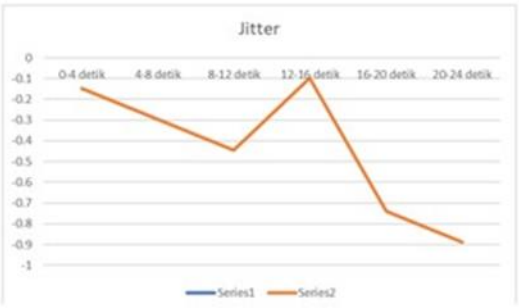
Tabel V. 9 Delay 500 paket serangan

No.	Waktu per Detik	*Delay* rata-rata paket masuk per detik
1	0 - 4 detik	0.14849
2	4 - 8 detik	0.296979
3	8 - 12 detik	0.445469
4	12 - 16 detik	0.098993
5	16 - 20 detik	1.23741
6	20 - 24 detik	0.890937

Tabel V. 10 Jitter 500 paket serangan

No.	Waktu per Detik	*Jitter* rata-rata paket masuk per detik
1	0 - 4 detik	-0.14816
2	4 - 8 detik	-0.29631
3	8 - 12 detik	-0.44447
4	12 - 16 detik	-0.09877
5	16 - 20 detik	-0.74078
6	20 -24 detik	-0.88894



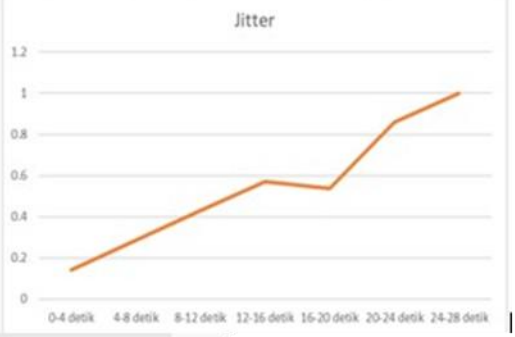


Tabel V. 11 Delay 1000 paket serangan

No.	Waktu per Detik	*Delay* rata-rata paket masuk per detik
1	0 - 4 detik	0.134573
2	4 - 8 detik	0.269146
3	8 - 12 detik	0.403719
4	12 - 16 detik	0.370076
5	16 - 20 detik	0.672865
6	20 - 24 detik	0.504648
7	24 - 28 detik	0.94201

Tabel V. 12 Jitter 1000 paket serangan

No.	Waktu per Detik	*Jitter* rata-rata paket masuk per detik
1	0 - 4 detik	0.143046
2	4 - 8 detik	0.286092
3	8 - 12 detik	0.429138
4	12 - 16 detik	0.572184
5	16 - 20 detik	0.536423
6	20 - 24 detik	0.858276
7	24 - 28 detik	1.001322

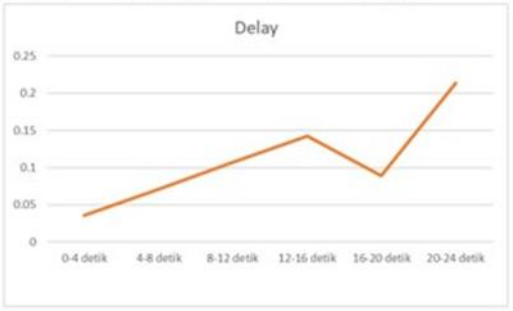


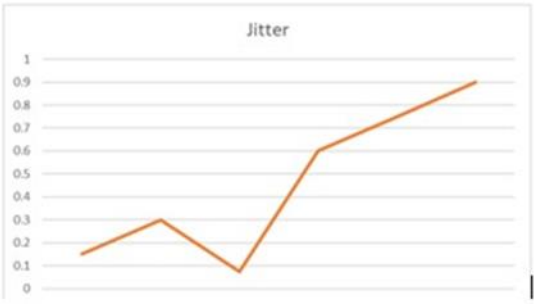
Tabel V. 13 Delay 500 paket serangan dengan *switch* yang sama

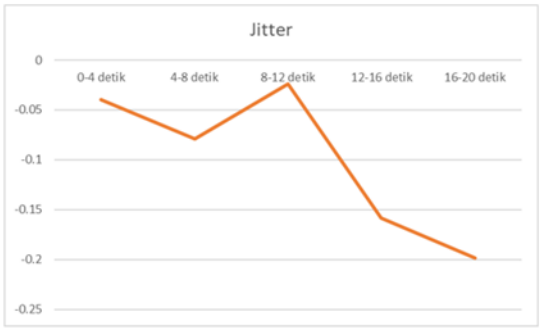
No.	Waktu per Detik	*Delay* rata-rata paket masuk per detik
1	0-4 detik	0.035625
2	4-8 detik	0.07125
3	8-12 detik	0.106875
4	12-16 detik	0.142501
5	16-20 detik	0.089063
6	20-24 detik	0.213751

Tabel V. 14 Jitter 500 Paket Serangan dengan *Switch* yang Sama

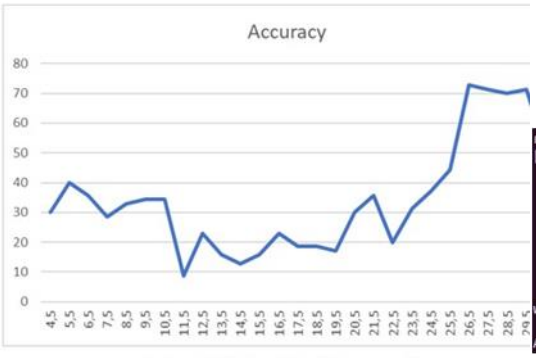
No.	Waktu per Detik	*Jitter* rata-rata paket masuk per detik
1	0-4 detik	0.1501499
2	4-8 detik	0.3002999
3	8-12 detik	0.075075
4	12-16 detik	0.6005998
5	16-20 detik	0.7507497
6	20-24 detik	0.9008997







3.2.4. Hasil Akurasi Entropy



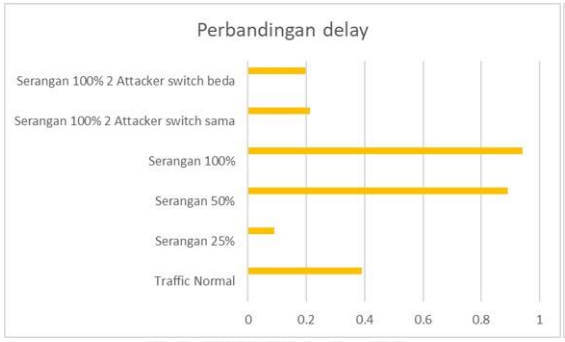
Gambar V. 29 Grafik Hasil *Accuracy Entropy*

```
root@hafiz-VirtualBox:/home/hafiz# python3 accuracyentropy.py
[[24 19]
 [ 0 27]]
              precision    recall  f1-score   support

           0       1.00      0.56      0.72        43
           1       0.59      1.00      0.74        27

    accuracy                           0.73        70
   macro avg       0.79      0.78      0.73        70
weighted avg       0.84      0.73      0.73        70

Accuracy: 72.85714285714285
```

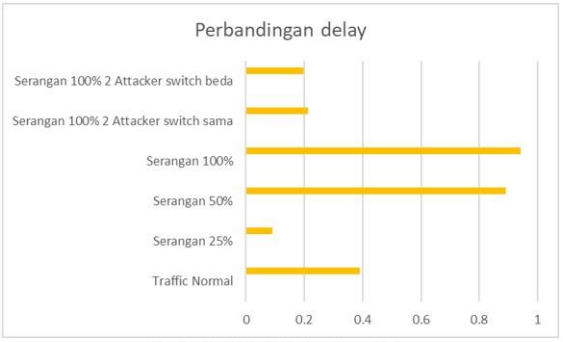
Gambar V. 30 Hasil *Accuracy Entropy* Ubuntu

Berdasarkan grafik *Accuracy Entropy* diatas, peneliti melakukan beberapa percobaan untuk menentukan akurasi deteksi serangan menggunakan *Entropy* yang paling tinggi, hasilnya dari 30 kali percobaan, akurasi yang paling tinggi yang peneliti dapat yaitu terdapat pada *windows size* sebesar 26, dan *count* 5 dari*threshold* yang digunakan dengan nilai sebesar 0,87.

3.2.4. Analisis Perbandingan



Gambar V. 31 Grafik Perbandingan *Delay*



Gambar V. 31 Grafik Perbandingan *Delay*

2. KESIMPULAN

Dari penelitian yang telah peneliti teliti, peneliti menyimpulkan beberapa hal yakni sebagai berikut :

1. *Software Defined Network* (SDN) merupakan paradigma jaringan baru yang bertujuan untuk mengubah arsitektur jaringan. Pada jaringan SDN sangat mungkin untuk administrator melakukan penyediaan jaringan dengan cepat tanpa perlu mengkonfigurasi jaringan secara manual. Dengan cara yaitu memisahkan antara control plane dengan data plan.

2. Untuk melihat nilai *threshold Entropy* diperlukan parameter seperti nilai rata-rata, standar deviasi, interval max, interval min, confidence interval, serta perbedaan dari *traffic* normal dan *traffic* serangan.

3. Nilai *Threshold* pada *Entropy* dibutuhkan untuk menentukan jenis serangan atau *traffic* normal (intrusi/normal)

4. Nilai *Threshold* pada penelitian ini mengembangkan penelitian dari[1] dengan nilai *threshold* 0,87 yang dikembangkan peneliti untuk mencapai akurasi dengan dataset yang telah peneliti miliki **5.**Akurasi penelitian menggunakan *Entropy* untuk deteksi serangan DDoS padapenelitian ini mencapai 72.85714285714.

DAFTAR PUSTAKA

[1] Sanubari, D.T. (2019). Network Performance Analysis In Software Defined Network Against DDOS Attack And Detection Using *Entropy*.
[2] N Afif, S.R., Sukarno, P., & Nugroho, M.A. (2018). E-procedding of engineering: Analisis Perbandingan Algoritma Naive Bayes dan Decision Tree untuk Deteksi Serangan Denial Of Service (DoS) pada Arsitektur SoftwareDefined Network (SDN), 5(3), 7515.
[3] Panjaitan, A.F.A., Sukarno, P., & Nugroho, M.A. (2018). E-procedding of engineering: Pendeteksian DoS Pada *Controller* Software Defined Network Dengan Menggunakan Algoritma Berbasis Entropi, 5(3), 7867.
[4] Putra, Y.A., Negara M.R., Yasirandi R. (2021). E-Procedding of Engineering : Implementasi *Adaptive Blocking Intrusion Detection System* Pada Jaringan *Internet Of Things* Berbasis *Software Defined Network.* 8(2), 1418.
[5] Q Sihombing, J.C.J., Kartikasari, D.P., Bhawiyuga, A. (2019). ImplementasiSistem Deteksi dan Mitigasi Serangan *Distributed Denial of Service* (DDoS) menggunakan SVM *Classifier*

pada Arsitektur *Software Defined Network* (SDN).Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. 3(10), 9608- 9613.

[6] Yasin, A., Mohidin, I. (2018). Dampak Serangan DDOS Pada Software Based OpenFow *Switch* Di Perangkat HG553. JTech. 6(2), 72-74.https://doi.org/10.30869/jtech.v6i2.206, p-issn/e-issn:2252- 4002/2546-558X.

[7] Marilanda, G.A.A., Widiastuti, W., Sumadi, F.D.S., Nastiti, V.R.S., Faiqurrahman, M. (2019). Mitigasi Serangan Mac Flooding Pada SDNMenggunakan *Entropy*. Seminar Nasional Teknologi dan Rekayasa. 59 -63

[8] Cintasari, E.P. (2018). *Analisis Kinerja Jaringan Software DefinedNetwork (SDN) Dengan Protokol OpenFlow pada Mininet.* (Skripsi Sarjana,Universitas Islam Negeri Syarif Hidayatullah Jakarta, 2018) Diakses dari https://repository.uinjkt.ac.id/dspace/bitstream/123456789/53507/1/EDDYTA%20PUTRI%20CINTASARI-FST.pdf .

[9] Putra, M.W., Pramukantoro, E.S., Yahya, W. (2018). Analisis Perbandingan Performansi Kontroler Floodlight, Maestro,RYU, POX dan ONOSdalam Arsitektur Software Defined Network (SDN). Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer. 2(10), 3779-3787

[10] Simarmata, F.R., Tulloh, R., Haryani. Y.S. (2018). E-Procedding of Applied Science : Simulasi Jaringan Software Defined Network Menggunakan Protokol Routing OSPF Dan Ryu Controller. 4(3). 2889

[11] Fajar, A.P., & Purboyo, T.W. (2018). A Survey Paper Of Distributed Denial-Of-Service Attack in Software Defined Networking (SDN). *InternationalJournal of Applied Engineering Research ISSN 0973-4562*, 13, 476

[12] Carvalho R.N., Bordim J.L., & Alchieri E.A.P. (2019). *IEEE International Parallel and Distributed Processing Symposium Workshops* : *Entropy* Based DoSAttack identification in SDN, 627-634

[13] Oliphant, T. E. (2007). Python for Scientific Computing. Computing in Science & Engineering (Volume: 9, Issue: 3, May-June 2007), 10-20

[14] Rahayu, P.T., Ardiyanta, S.A. (2019). Hubungan Minat Belajar dengan Kemampuan Siswa Dalam Mengoperasikan Virtualbox Sebagai Media Pembelajaran Sistem Operasi Jaringan Kelas XI TKJ SMKN 1 KRAS. Jurnal Of Education And Information Communication Technology. 3(1). 37- 46.

[15] Santosa, B., Boedi, D, P., & Putra, Y, I. (2010). Remastring Distro UbuntuUntuk Menunjang Pembelajaran Informatika. Seminar Nasional Informatika 2010(SemnasIF 2010), C 57 - C 65