KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

PALCOMTECH

SKRIPSI

IMPLEMENTASI SURICATA UNTUK MENINGKATKAN KEAMANAN PADA *CLOUD COMPUTING*

Diajukan Oleh :

SATRIA BAGUS PRIBADI
01150048

Untuk memenuhi Sebagian Dari Syarat-Syarat

Guna Mencapai Gelar Sarjana Komputer

PALEMBANG

2019

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

PALCOMTECH

SKRIPSI

IMPLEMENTASI SURICATA UNTUK MENINGKATKAN KEAMANAN PADA *CLOUD COMPUTING*

Diajukan Oleh :

SATRIA BAGUS PRIBADI
01150048

Untuk Memenuhi Sebagian Dari Syarat-Syarat

Guna Mencapai Gelar Sarjana Komputer

PALEMBANG

2019

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER PALCOMTECH

HALAMAN PENGESAHAN PEMBIMBING SKRIPSI

NAMA	**:**	**SATRIA BAGUS PRIBADI**
NOMOR POKOK	**:**	**011150048**
PROGRAM STUDI	**:**	**TEKNIK INFORMATIKA**
JENJANG PENDIDIKAN	**:**	**STRATA SATU (S1)**
KONSENTRASI	**:**	**JARINGAN**
JUDUL SKRIPSI	**:**	**IMPLEMENTASI SURICATA UNTUK MENINGKATKAN KEAMANAN PADA CLOUD COMPUTING**

Tanggal : 23 Januari 2019

Pembimbing,

Surahmat, S.Kom., M.Kom.

NIDN : 0217058703

Mengetahui,

Ketua,

Benedictus Effendi, S.T.,M.T.

NIP : 09.PCT.13

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER PALCOMTECH

HALAMAN PENGESAHAN PENGUJI SKRIPSI

NAMA	**:**	**SATRIA BAGUS PRIBADI**
NOMOR POKOK	**:**	**011150048**
PROGRAM STUDI	**:**	**TEKNIK INFORMATIKA**
JENJANG PENDIDIKAN	**:**	**STRATA SATU (S1)**
KONSENTRASI	**:**	**JARINGAN**
JUDUL SKRIPSI	**:**	**IMPLEMENTASI SURICATA UNTUK MENINGKATKAN KEAMANAN PADA CLOUD COMPUTING**

Tanggal : 06 Februari 2019	**Tanggal : 07 Februari 2019**
Penguji 1,	**Penguji 2,**
Guntoro Barovih, S.Kom., M.Kom.	**Mahmud, S.Kom., M.Kom.**
NIDN : 0201048601	**NIDN : 0229128602**

Menyetujui,

Ketua,

Benedictus Effendi, S.T., M.T.

NIP : 09.PCT.13

MOTTO *:*

"Cobalah berbeda dari orang lain"
"Terkadang berbeda itu sulit"
"Terkadang juga berbeda itu istimewa"
"Satria Bagus Pribadi**"**

Kupersembahkan Kepada:

- ***Tuhan Yang Maha Esa***
- ***Ayah dan Ibu Tercinta***
- ***Saudara-Saudaraku Tersayang***
- ***Seluruh Keluarga Ku Tercinta***
- ***Orang yang aku sayangi***
- ***Gangster Open Lab***
- ***Teman-Teman Seperjuangan***
- **Bapak Surahmat, S.Kom., M.Kom.**

KATA PENGANTAR

Puji dan Syukur Penulis panjatkan kehadirat Tuhan Yang Maha Esa, atas segala berkat dan karunia-Nya sehingga Penulis dapat menyelesaikan laporan Skripsi ini dengan baik. Laporan ini diberi judul **"Implementasi Suricata untuk Meningkatkan Keamanan Pada Cloud Computing".** Adapun tujuan penulisan laporan Skipsi ini adalah sebaai bentuk pelaporan terhadap apa yang telah Penulis kerjakan, dan dapat diusulkan selama melakukan Skripsi, sehingga apabila laporan Skipsi ini dinilai layak, dapat memenuhi sebagai syarat guna penyusunan Skipsi.

Adapun selama penulisan dan penyusunan skripsi ini, Penulis mendapatkan banyak bimbingan, bantuan dan dukungan dari berbagai pihak. Oleh karena itu sudah menjadi kewajiban bagi Penulis untuk mengucapkan terima kasih kepada berbagai pihak tersebut, yaitu :

1. Kepada Ketua STMIK PalComTech, Bapak Benedictus Effendi, S.T., M.T.,
2. Kepada Pembantu Ketua 1, Bapak D.Tri Octafian, S.Kom., M.Kom.,
3. Kepada Ketua Program Studi Teknik Informatika, Bapak Alfred Tenggono, S.Kom., M.Kom.,
4. Kepada Dosen Pembimbing Skripsi Bapak Surahmat,S.kom., M.Kom.
5. Kepada Kedua Orang Tua Penulis tercinta.
6. Kepada Saudara yang selalu memberi dukungan.

7. Kepada Teman dan Sahabat Seperjuangan.

8. Kepada Semua Pihak yang telah banyak membantu dan memberi dukungan.

Demikian kata pengantar dari Penulis, dengan harapan semoga Skripsi ini dapat bermanfaat dan berguna bagi para pembaca, dengan kesadaran Penulis bahwa penulisan Skripsi masih mempunyai banyak kekurangan dan kelemahan sehingga membutuhkan banyak saran dan kritik yang membangun untuk menghasilkan sesuatu yang lebih baik. Terima kasih.

Palembang, 09 Februari 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

ABSTRAK

SATRIA BAGUS PRIBADI. Implementasi Suricata Untuk Meningkatkan Keamanan Pada *Cloud Computing*.

Cloud Computing saat ini banyak dikembangkan dan digunakan oleh perusahaan yang membutuhkan sumber daya komputasi besar dan efisien. Sebagai perkembangan teknologi maka ancaman keamanan dalam layanan komputasi awan yang terus meningkat. Berbagai ancaman di teknologi *cloud computing* dapat dihindari dengan memaksimalkan identifikasi celah keamanan. Ancaman informasi mengenai *cloud computing* tuntutan keamanan jaringan dan layanan dari kemungkinan serangan. Suricata adalah ids yang dapat mendeteksi ancaman serangan jaringan aktivitas yang dibantu dengan aturan yang ada. Ketika serangan terdeteksi maka suricata akan menciptakan sebuah log serangan-serangan yang dilakukan, Suricata juga dapat melakukan deteksi otomatis pada layer 7. Penulis dikumpulkan hasil dari serangan yang di log Suricata dan penulis juga mengevaluasi apakah Suricata dapat mendeteksi serangan *port scanning*, *bruteforce*, *denial of service*, dan *backdoor* ke *Cloud Computing*. Dari hasil ini pengujian mendapatkan hasil optimal dari hasil serangan terdeteksi oleh log sistem deteksi intrusi (IDS) Suricata di dir /var/log/suricata/fast.log, dan juga penulis menambahkan konfigurasi suricata tidak hanya untuk mendeteksi, sehingga bisa juga untuk melakukan eksekusi drop bila ada aktifitas mencurigakan dengan menggunakan *netfilter* yang sudah ada pada suricata dan konfigurasi oleh penulis untuk meningkatkan keamanan pada *cloud computing* secara optimal.

Kata kunci : ***Intrusion Detection System, Suricata, Cloud Computing***

ABSTRACT

SATRIA BAGUS PRIBADI. *The Implementation Of Suricata For Increase Security On Cloud Computing*.

Cloud Computing currently many developed and used by the companies that need huge computing resources and efficient. As the development of technology then security threats in the cloud services are on the rise. Various threats in cloud computing technology can be avoided by maximizing the identification of security gaps. Threat information about Cloud Computing network security demands and the services of a possible attack. Suricata is ids that can detect network attack threat activities that assisted with the existing rules. When an attack is detected then the suricata will create a log of attacks conducted, Suricata can also perform automatic detection at layer 7. The author collected the results from an attack in log Suricata and writers also evaluate whether Suricata can detect bruteforce attacks, port scanning, denial of service, and a backdoor into Cloud Computing. From the results of this testing get optimum results from the results of the attacks detected by the log intrusion detection system (IDS) Suricata in dir/var/log/suricata/fast.log, and is also the author of the add configuration suricata is not only to detect, so that it can also to perform a execution drop when there is suspicious activity using netfilter already on the suricata and configuration by the author to improve the security in cloud computing optimally.

Keyword : ***Intrusion Detection System, Suricata, Cloud Computing***

DAFTAR PUSTAKA

Ariyanto, Yuri. Harijanto, Budi. Watequlis S, Yan. (2017). *"Implementasi Suricata pada Server Cloud Proxmox VE sebagai Intrusion Detection System (IDS) dalam Pengamanan Jaringan"*. Prosiding Sentrinov, Hal TE178-TE179.

Athailah. (2013). "*Panduan Singkat Menguasai Router*." Jakarta: PT. TransMedia, hal. 6-15.

Atmojo, Y. P. (2018). *"Bot Alert Snort dengan Telegram Bot API pada Intrusion Detection System."* Studi Kasus IDS pada Server Web, 176–180.

Badrul, M. (2012). "*Teknik Komputer Jaringan*." Jakarta Timur: Inti Prima Promosindo, hal. 64-66.

Eka P, Ricky. Rachman, Andy. dan Wahyu H, Tri, (2010). *"Virtual Private Server (VPS) sebagai Alternatif Pengganti Dedicated Server."* Surabaya: Institut Teknologi Sepuluh Nopember, 2010.

Fajrin, T. (2012). "*Analisis Sistem Penyimpanan Data Menggunakan Sistem. Cloud Computing Studi Kasus SMK N 2 Karanganyar*." 1 (10), hal. 31–35.

Khadafi, S., Meilani, D. B., dan Arifin, S. (2017). "*Sistem Keamanan Open Cloud Computing Menggunakan IDS (Intrusion Detection System) Dan IPS (Intrusion Prevention System)". Jurnal IPTEK, 21*(2), hal. 67–76.

Mukmin, C. (2017). "*Efisiensi Maintenance Laboratorium Komputer Berbasis Jaringan*."

Nazir, M. (2014). "*Metode Penelitian*." Bogor: Ghalia Indonesia. 79-154.

Nugraha, B. (2016). "*Analisis Teknik-Teknik Keamanan Pada Cloud Computing dan NEBULA (Future Cloud)"* Survey Paper. *Teknosi*, *02*(02), hal. 35–42.

Nazwita, Dan Ramadhani., S. (2017). "*Analisis Sistem Keamanan Web Server Dan Database Server Menggunakan Suricata.*" hal. 18–19.

Pratama, I. P. A. E. (2014). "*Handbook Jaringan Komputer Teori Dan Praktik Berbasiskan Open Source*." Bandung : Informatika Bandung, hal. 12-21.

Samuel Pardosi, Rudy (2015). "*Kali Linux Top Hacking.",* Sitoluama, hal. 75-96.

Sofana, I. (2013). "*Membangun Jaringan Komputer*." Bandung: Informatika.

Widayanto, B., Munadi, R. dan Mayasari, R. (2015). "*Implementasi dan Analisis Perbandingan Performansi VoIP Server pada VPS Berbasis OpenVZ dan Cloud Computing.",* 2(2), hal. 3195–3202.

HALAMAN LAMPIRAN

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

PALCOMTECH

HALAMAN PENGESAHAN PEMBIMBING SKRIPSI

NAMA	:	SATRIA BAGUS PRIBADI
NOMOR POKOK	:	011150048
PROGRAM STUDI	:	TEKNIK INFORMATIKA
JENJANG PENDIDIKAN	:	STRATA SATU (S1)
KONSENTRASI	:	JARINGAN
JUDUL SKRIPSI	:	IMPLEMENTASI SURICATA UNTUK MENINGKATKAN KEAMANAN PADA CLOUD COMPUTING

Tanggal : 23 Januari 2019

Pembimbing,

Surahmat, S.Kom., M.Kom.

NIDN : 0217058703

Mengetahui,

Ketua,

Benedictus Effendi, S.T.,M.T.

NIP : 09.PCT.13

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

PALCOMTECH

HALAMAN PENGESAHAN PENGUJI SKRIPSI

NAMA	:	SATRIA BAGUS PRIBADI
NOMOR POKOK	:	011150048
PROGRAM STUDI	:	TEKNIK INFORMATIKA
JENJANG PENDIDIKAN	:	STRATA SATU (S1)
KONSENTRASI	:	JARINGAN
JUDUL SKRIPSI	:	IMPLEMENTASI SURICATA UNTUK MENINGKATKAN KEAMANAN PADA CLOUD COMPUTING

Tanggal : 06 Februari 2019

Penguji 1,

Guntoro Barovih, S.Kom., M.Kom.

NIDN : 0201048601

Tanggal : 07 Februari 2019

Penguji 2,

Mahmud, S.Kom., M.Kom.

NIDN : 0229128602

Menyetujui,

Ketua,

Benedictus Effendi, S.T., M.T.

NIP : 09.PCT.13

BAB I

PENDAHULUAN

1.1. Latar Belakang

Keamanan menjadi suatu hal penting yang melindungi suatu informasi atau data, demi keamanan data masyarakat akan teknologi dan ancaman informasi pada *Cloud Computing* sangat beragam mulai dari *scanning Port*, *Backdoor*, *Brute Force* hingga penyerangan *Denial Of Service* (DOS). Ancaman informasi tersebut menyebabkan server akan mati dan tidak dapat beroperasi lagi sehingga otomatis tidak dapat memberikan pelayanan. Berbagai ancaman dalam teknologi *Cloud Computing* dapat dihindari dengan memaksimalkan identifikasi celah keamanan.

Cloud Computing (komputasi awan) merupakan salah satu teknologi yang saat ini banyak dikembangkan dan digunakan oleh perusahaan yang membutuhkan sumber daya komputasi yang besar dan efisien. Seiring perkembangan teknologi tersebut maka ancaman keamanan pada layanan *Cloud Computing* semakin meningkat.

Serangan yang terjadi dalam *Cloud Computing* dapat mengakibatkan kerusakan bahkan kehilangan data maupun kerusakan *hardware*. Ancaman informasi pada *Cloud Computing* menuntut adanya pengamanan jaringan dan layanannya dari kemungkinan adanya serangan. Berbagai serangan atau penyusupan dapat terjadi seperti *Denial Of Service, Port Scanning, Brute Force, dan Backdoor*. Serangan-serangan tersebut dapat bertujuan untuk mengakses sistem aplikasi atau mencoba masuk ke dalam jaringan dengan

hak akses khusus hingga mengakses sumber daya atau layanan yang disediakan baik dalam sistem atau jaringan.

Menurut Sofyan, Periyadi, Anang (2016:1172), Suricata merupakan IDS yang dapat mendeteksi aktifitas ancaman serangan pada jaringan yang dibantu dengan *rules* yang telah ada. Cara kerja dari suricata adalah ketika adanya penyerangan suricata akan melakukan pengecekan paket/serangan yang ada melalui *rules* yang dibuat. Ketika serangan terdeteksi maka suricata akan membuat *log* serangan yang dilakukan, Suricata juga dapat melakukan deteksi otomatis pada *layer* 7 yaitu aplikasi seperti *dns, http, imap, ftp, dan smtp*. Sehingga Suricata dapat memberikan solusi untuk meningkatkan keamanan dalam *Cloud Computing.* Berdasarkan uraian diatas maka penulis mengambil judul **"Implementasi Suricata Untuk Meningkatkan Keamanan Pada *Cloud Computing*".**

1.2. Perumusan Masalah

Berdasarkan pengamatan dan sesuai dengan uraian dari latar belakang diatas serta judul yang diangkat maka penulis merumuskan masalah yang ada yaitu, Bagaimana membuat *rules* yang efektif agar dapat menangkap ancaman serangan di server dengan Suricata dan pengaruh penerapan Suricata *rules* yang ada pada *resource server* dimana IDS diimplementasikan.

1.3. Batasan Masalah

Ruang lingkup yang dibahas dalam penelitian ini adalah sebagai berikut :

Penelitian ini menguji Sistem *Suricata* pada *Virtual Private Server* (VPS) dengan spesifikasi *processor Intel Xeon Gold* 6140 CPU @ 2.30Hz, *memory* 4 gb, dan *SSD* 80gb. berbasis *Virtual Enviroment (KVM)* dan Sistem Operasi *Ubuntu Server* untuk mengetahui serangan *port scanning*, *brute force, denial of service*, dan *backdoor*.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah untuk mendapatkan hasil dari sistem kerja *Suricata* untuk membuat sebuah sistem IDS yang dapat mendeteksi ancaman pada Cloud Computing dan mengimplementasikan rules ke sistem IDS Suricata untuk pendeteksian serangan *Port scanning, Brute force, Denial Of Service, Backdoor* dan mengukur efektifitas penerapan *rules-rules* serangan tersebut.

1.5. Manfaat Penelitian

1.5.1. Manfaat Bagi Penulis

Manfaat penelitian untuk penulis adalah :

a. Menambah pengetahuan bagi penulis khususnya pada keamanan *Cloud Computing*.
b. Menerapkan ilmu pengetahuan yang telah didapatkan selama menjalani studi perkuliahan.

1.5.2. Manfaat Bagi Akademik

Manfaat yang didapat bagi akademik adalah menjadi sebagai salah satu acuan bagi akademik untuk kelanjutan penelitian di masa yang akan datang dan diharapkan dapat menambah pengetahuan bagi pihak yang berkepentingan dalam menerapkan keamanan pada *Cloud Computing*.

1.5.3. Manfaat Bagi Umum

Manfaat bagi umum adalah sebagai ilmu pengetahuan dan ketika ingin menerapkan keamanan di *Cloud Computing* bagaimana cara sistemnya bekerja karena dapat membantu perusahaan dan individual.

1.6. Sistematika Penulisan

Demi mewujudkan suatu hasil yang baik dalam penyusunan skripsi ini penulis menggunakan pembahasan yang sesuai dengan ketentuan yang diberikan, sistematika penulisan tersebut meliputi antara lain :

BAB I PENDAHULUAN

Bab ini berisi uraian latar belakang, rumusan masalah, batasan masalah, tujuan peneltian, manfaat penelitian, dan sistematika penulisan.

BAB II GAMBARAN UMUM PENELITIAN

Bab ini diuraikan mengenai fenomena tentang penelitian yang dilakukan.

BAB III TINJAUAN PUSTAKA

Bab ini berisi teori berdasarkan penulisan skripsi ini yang terdiri dari landasan teori, penelitian terdahulu, dan kerangka penelitian.

BAB IV METODE PENELITIAN

Bab ini penulis membahas lokasi dan waktu penelitian, jenis data, teknik pengumpulan data, dan jenis penelitian dan alat serta teknik pengembangan sistem.

BAB V HASIL DAN PEMBAHASAN

Bab ini memuat hasil-hasil yang diperoleh dalam penelitian dan pembahasan terhadap hasil yang telah dicapai maupun masalah-masalah yang telah ditemukan selama penelitian, serta pengujian sistem yang dibuat.

BAB VI PENUTUP

Bab ini berisi kesimpulan dan saran dari pembahasan dalam penelitian yang telah dilakukan.

BAB II

GAMBARAN UMUM PENELITIAN

2.1. Fenomena Penelitian

Perkembangan teknologi yang sangat pesat saat ini, sangat mempengaruhi kebutuhan digital pada seluruh dunia. Adanya internet pengguna dapat dengan mudah mendapatkan informasi yang sifatnya tidak terbatas. Tetapi kemajuan internet memiliki potensi bahaya bagi pengguna. *Cloud computing* atau komputasi awan saat ini telah memiliki pengguna yang sangat luas termasuk di Indonesia. Kehadiran komputasi awan *(Cloud Computing*) merupakan bagian dari salah satu perkembangan era komputasi modern. Dalam jaman modern ini, dimana teknologi informasi dan komunikasi telah merasuki seluruh aspek kegiatan dan kehidupan manusia, para generasi muda semakin akrab dengan keberadaan teknologi ini.

Keuntungan komputasi awan adalah pengguna bisa menyewa kemampuan dan kapasitas komputasi tersebut sesuai kebutuhan, tidak ada keharusan untuk membeli dan memasang *server* sendiri sehingga lebih terjangkau dari sisi biaya. Walaupun *cloud computing* memberikan manfaat yang sangat menarik dan dapat menghemat biaya serta memberikan berbagai pilihan penyimpanan data, tetapi hal ini juga memberikan risiko dan peluang baru untuk eksploitasi bidang keamanan data yang disimpan dalam *cloud*. Semakin maraknya *virus* dan peretas data dalam jaringan internet menjadikan ketakutan bagi pengguna layanan, terutama perusahaan yang bergerak di bidang keuangan.

Bagi setiap pengguna, data dalam *cloud computing* tentunya merupakan hal yang penting dan sensitif. Oleh karena hal inilah muncul kehawatiran baru bagi pengguna yaitu privasi. Dalam hal keamanan, faktor privasi juga menjadi isu yang menjadi perhatian dikarenakan tingkat privasi yang diinginkan setiap orang berbeda-beda sesuai kebutuhannya masing-masing. Dengan kemampuan privasi data, setiap orang bisa menentukan siapa yang berhak mengakses atau mengubah suatu informasi berdasarkan identifikasi digital. Setiap orang tentunya memiliki standar privasi yang berbeda-beda, namun dengan adanya penyimpanan data dalam bentuk digital (dalam *Cloud Computing*) membuat khawatir dengan kemungkinan bahwa datanya dapat dengan mudah dilihat oleh orang lain yang tidak berhak mengakses data pribadi kita.

Faktor keamanan dan privasi pengguna memang merupakan dua isu penting pada penerapan sistem kerja *cloud computing* di seluruh dunia, bahkan Indonesia. Sebenarnya data yang disimpan pada *platform cloud* jauh lebih aman, karena ada aturan yang mengharuskan setiap penyelenggara layanan *cloud computing* patuh terhadap regulasi dan aturan yang terkait. Salah satu regulasi dan aturan yang mengatur sitem keamanan untuk *platform cloud* adalah *ISO* 27002, yang merupakan standar praktik keamanan informasi dan tingkat keamanan yang wajib dimiliki oleh penyedia jasa layanan *cloud computing.*

Namun, minimnya pengetahuan pengguna terkait keamanan di internet seakan menjadi salah satu isu yang menyebabkan tersendatnya proses adopsi teknologi *cloud computing*. Ancaman ini memang harus

disikapi dengan serius oleh seluruh *stakeholder cloud computing*. Sebab, Indonesia yang tengah memulai era komputasi awan ini, mau tak mau harus siap dengan segala macam kemungkinan ancaman. ada sedikitnya ada tujuh risiko mengenai *security* di *cloud computing*. Tujuh risiko itu adalah *Privilege User Access, Regulatory Compliance, Data allocation, Data Secure, Recovery, Investigative support*, dan terakhir Longterm *Viability.*

Meneropong isu keamanan internet dalam aspek teknis, bisnis, dan sosial mengungkapkan bahwa dilihat dari perspektif teknis, terjadi *trend* dimana jumlah dan variasi *malicious software* bertambah dari masa ke masa. Hal yang sama terjadi pula dengan total kasus *vulnerabilities* yang ditemui dalam berbagai produk perangkat keras maupun perangkat lunak teknologi informasi. Dari segi ancaman atau serangan, data memperlihatkan adanya peningkatan tajam pula terhadap pertumbuhan *spam* maupun *spyware.*

Begitu pula halnya dengan kecenderungan terjadinya peningkatan yang berarti terhadap tindakan kriminal seperti *phishing* maupun *identity theft*, yang telah mengakibatkan terjadinya kerugian ekonomis maupun politis. Yang menarik untuk dicermati adalah terlepas dari adanya *trend* peningkatan dari seluruh komponen atau entitas di atas, waktu bagi seorang kriminal untuk mengeksploitasi berbagai kelemahan sistem komputer atau jaringan semakin sedikit – alias proses untuk membobol sebuah jaringan komputer menjadi semakin cepat dari hari ke hari. Tentu saja kenyataan menakutkan ini harus diwaspadai secara serius bagi mereka yang keberlangsungan hidup bisnisnya sangat ditentukan oleh kinerja teknologi informasi yang dimilikinya.

Secara teknis, cara untuk menanggulangi ancaman tersebut adalah melalui instalasi berbagai produk pengamanan internet maupun komputer untuk mencegah kemungkinan dieksploitasinya berbagai kelemahan yang dimiliki oleh sebuah sistem. Misalnya adalah instalasi *firewall* untuk melindungan jaringan internal perusahaan dari akses pihak yang berada pada jejaring eksternal (baca: internet), atau dilibatkannya program anti *virus* dan anti *spyware* untuk mencegah berbagai program jahat masuk ke dalam sistem komputer, atau pemasangan *software patches* untuk menambal lubang-lubang kerawanan yang ada pada sistem aplikasi, atau melakukan proses *encryption* untuk mencegah pihak yang tidak berwenang mengerti isi dari suatu pesan atau informasi rahasia. Keseluruhan usaha yang bertujuan untuk mengurangi probabilitas terjadinya eksploitasi terhadap kerawanan sistem ini (mitigasi) dilakukan pada level teknis operasional, dalam arti kata dikembangkan dengan cara mengadakan sejumlah piranti lunak/keras yang kemudian dipasang atau diinstalasi pada sistem komputer atau jaringan yang ingin dilindungi.

BAB III

TINJAUAN PUSTAKA

3.1. Landasan Teori

3.1.1. Jaringan Komputer

Menurut Pratama (2014:12), jaringan komputer adalah hubungan dari sejumlah perangkat yang dapat saling berkomunikasi satu sama lain (*a network is a interconnection of a set of devices capable of communication*). Perangkat yang dimaksud pada definisi ini mencakup semua jenis perangkat komputer.

Jaringan komputer yang menghubungkan komputer-komputer pada lokasi berbeda dapat di manfaatkan untuk mengirim surat elektronik (*e-mail*), mengirim file data (*upload*), dan mengambil file data dari tempat lain (*download*), serta berbagai kegiatan akses informasi pada lokasi yang terpisah.

3.1.2. Topologi Jaringan

Menurut Pratama (2014:18), topologi jaringan komputer didefinisikan sebagai suatu teknis, cara, dan aturan di dalam merangkai dan menghubungkan berbagai computer dan perangkat terhubung lainnya kedalam sebuah jaringan komputer, sehingga membentuk sebuah hubungan yang bersifat geometris. Topologi ini bersifat sebuah rancangan (desain), yang kemudian dapat

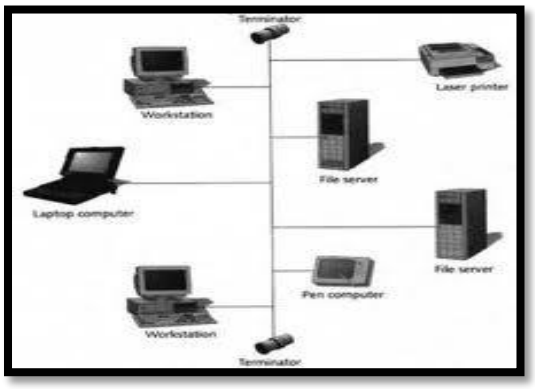
diimplementasikan secara langsung melalui sejumlah perangkat keras penghubung pada jaringan komputer.

Jenis-jenis topologi yang biasanya digunakan sebagai berikut :

a. Topologi *Bus*

Menurut Athailah (2013:9), topologi ini merupakan jenis topologi yang banyak dipergunakan pada masa penggunaan kabel sepaksi. Dengan menggunakan T-Connector dan terminator 50 ohm pada ujung *Network*, maka komputer atau perangkat jaringan lainnya dapat dengan mudah dihubungkan satu sama lainnya. Penerapan jenis topologi ini memiliki kesulitan utama, karena jenis topologi ini menggunakan jenis kabel sepaksi, maka kita kan sulit mengukur panjang kabel yang digunakan, apakah kabel tersebut sudah *matching* (sama) atau belum.

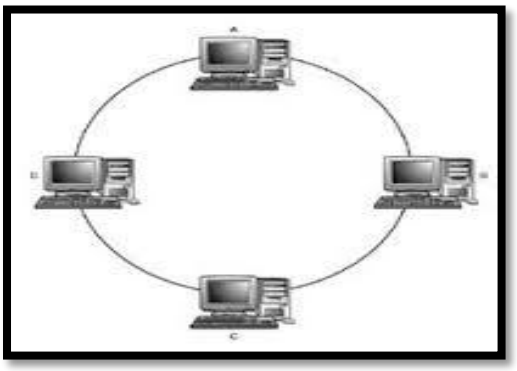
Jika panjang kabel tidak sama, dapat merusak *Network Intercae Card* (NIC) yang digunakan, dan kinerja serta kecepatan jaringan menjadi terhambat karena tidak mencapai kinerja maksimal. Topologi *bus* dapat dilihat pada gambar 3.1.



Sumber : Athailah (2013:9)
Gambar 3.1. Topologi *Bus*

b. Topologi *Ring*

Menurut Athailah (2013:10), topologi ini adalah jaringan komputer yang dibentuk seperti lingkaran atau dalam bahasa Inggris disebut *Ring*, dimana komputer dalam topologi jaringan ini terhubung masing-masing di dua titik dari komputer lainnya. Pada tipe topologi *Ring* ini masing-masing node atau komputer dapat menjadi *repeater* yang memperkuat sinyal di sepanjang sirkulasi. Dengan demikian, masing-masing node pada jaringan yang ber-Topologi *Ring* ini akan saling menguatkan sinyal dari node sebelumnya dan akan meneruskan sinyal tersebut ke node seterusnya. Hal ini dapat terjadi berkat bantuan TOKEN. Topologi *ring* dapat dilihat pada gambar 3.2.



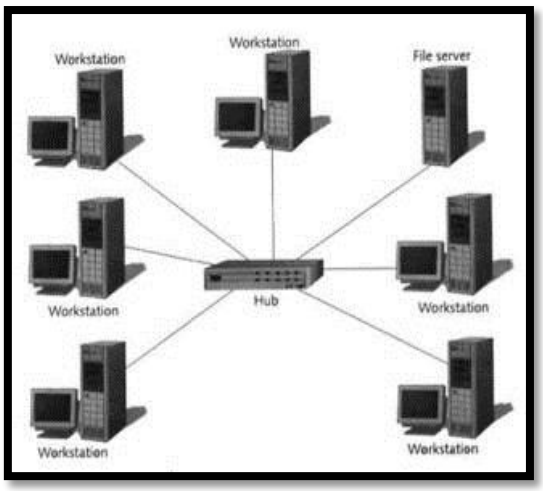
Sumber: Athailah (2013:10)
Gambar 3.2. Topologi *Ring*

c. Topologi Bintang atau *Star*

Menurut Pratama (2014:21), topologi *star* adalah topologi di dalam jaringan komputer, di mana terdapat sebuah komputer (ataupun perangkat jaringan komputer berupa *hub* atau *switch*) yang menjadi pusat dari semua komputer yang terhubung kedalamnya. Komputer pusat ini bertindak sebagai server. Komputer-komputer lainnya, yang dalam hal ini bertindak sebagai *client*, tidak dapat berkomunikasi satu sama lain. Mereka harus melalui komputer pusat (ataupun berupa *hub* dan *switch*) terlebih dahulu, untuk dapat bertukar data dengan sesama komputer *client* lainnya.

Menurut Athailah (2013:12), topologi ini merupakan topologi yang paling banyak digunakan saat ini, dapat dikatakan hampir semua jaringan komputer menggunakan topologi jenis ini. Dalam topologi jaringan

Star, jaringan terpusat pada perangkat yang dinamakan *HUB* dan *SWITCH*, dimana perangkat ini akan menghubungkan node-node yang ada dalam jaringan. Topologi bintang atau *star* dapat dilihat pada gambar 3.3.

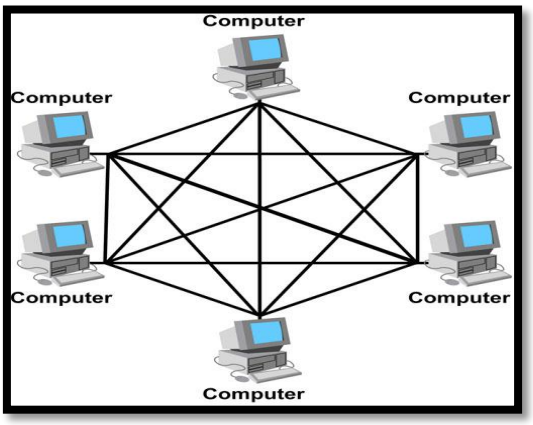


Sumber: Athailah (2013:12)

Gambar 3.3. Topologi *Star*

d. Topologi *Mesh*

Menurut Athailah (2013:13), topologi ini juga dinamakan dengan Topologi Jala atau Topologi Net. Topologi Mesh adalah sebuah Topologi jaringan komputer dimana sebuah node dalam jaringan dapat berkomunikasi secara langsung dengan node lainnya. Topologi mesh dapat di lihat pada gambar 3.4.

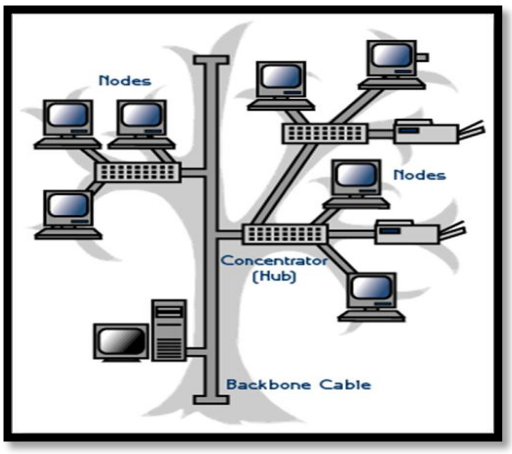


Sumber: Athailah (2013:13)
Gambar 3.4. Topologi *Mesh*

Topologi mesh ini memang di desain untuk memiliki tingkat restorasi dengan berbagai alternatif rute atau penjaluran yang biasanya disiapkan dengan dukungan perangkat lunak atau software.

e. Topologi Pohon atau *Tree*

Menurut Athailah (2013:15), topologi tree atau topologi pohon adalah penggabungan dari dua topologi sebelumnya, yaitu topologi *bus* dan topologi *star* atau bintang. Secara kasat mata topologi ini memang berbentuk seperti pohon yang bercabang-cabang, demikian juga topologi jaringan komputer ini akan memliki cabang yang banyak juga. Topologi tree dapat di lihat pada gambar 3.5.



Sumber: Athailah (2013:15)
Gambar 3.5. Topologi Pohon atau *Tree*

Bentuk dari topologi ini adalah sekolompok node yang terhubung satu sama lainnya dengan menggunakan topologi *star* tersebut terhubung ke kelompok jaringan yang lain dengan menggunakan topologi bus.

3.1.3. Kelas IP *Address*

Menurut Sofana (2013:108), Untuk memudahkan pengaturan IP *address* seluruh komputer pengguna jaringan internet, dibentuklah suatu badan yang mengatur pembagian IP *address*. Dengan kata lain, tanpa IP Address, komputer tidak akan dapat saling berkomunikasi dengan komputer lain dalam sebuah jaringan. Badan tersebut bernama InterNIC (*Internet Network Information Center*). InterNIC membagi-bagi IP *address* menjadi beberapa kelas. Kelas-kelas tersebut meliputi:

3.1.3.1. Kelas A

Menurut Badrul (2012:64), Alamat-alamat kelas A diberikan untuk jaringan skala besar. Nomor urut bit tertinggi di dalam alamat IP kelas A selalu diset dengan nilai 0 (nol). Tujuh bit berikutnya-untuk melengkapi octet pertama-akan membuat sebuah *network identifier*, 24 bit sisanya (atau tuga octet terakhir) merepresentasikan *host identifier.*

3.1.3.2. Kelas B

Menurut Badrul (2012:65), Alamat-alamat kelas B dikhususkan untuk jaringan skala menengah hingga skala besar. Dua bit pertama di dalam octet pertama alamat IP kelas B selalu diset ke bilangan biner 10. 14 Bit berikutnya (untuk melengkapi dua oktet pertama), akan membuat sebuah *network identifier*. 16 bit sisanya (dua oktet terakhir) merepresentasikan *host identifier*. Kelas B hanya memiliki 16,384 *network,* dan 65,534 *host* untuk setiap *network*-nya. Kelas B hanya menggunakan 16 oktet pertamanya sebagai *Network ID* dan 16 sisanya adalah *host id.*

3.1.3.3. Kelas C

Menurut Badrul (2012:66), Alamat IP kelas C digunakan untuk jaringan berskala kecil. Tiga bit pertama di dalam oktet pertama alamat kelas C selalu diset ke nilai

biner 110. 21 bit selanjutnya (untuk melengkapi tiga oktet pertama) akan membentuk sebuah *network identifier*. 8 bit sisanya (sebagai oktet terakhir) akan merepresentasikan *host identifier*. Ini memungkin pembuatan total 2,097,152 buah *network*, dan 254 *host* untuk setiap *Network Id* dan 8 sisanya adalah *host id*. Ini memungkinkan untuk dapat mengkoneksikan komputer *client* yang sedikit dalam satu jaringan tapi *network* yang dapat digunakan banyak.

3.1.4. Sistem Operasi Ubuntu

Menurut (Athailah, 2012:6), Ubuntu merupakan sistem operasi yang terus berkembang sehingga akan ada versi-versi terbaru dari Ubuntu. Secara resmi, versi terbaru Ubuntu akan dirilis setiap enam bulan sekali kecuali untuk versi yang diberi label LTS (*Long Term Support*) yang rilisnya lebih lama yaitu 3 tahun untuk *desktop* dan 5 tahun untuk *server*.

3.1.5. *Cloud Computing*

Menurut (Widayanto, Munadi dan Mayasari, 2015), *Cloud computing* merupakan teknologi komputasi dimana semua *resource* dan sumber daya komputer baik itu memori, aplikasi, *processor*, *network*, dan *operating system* yang digunakan secara *virtual* dengan pola akses *remote* sehingga bisa mengakses layanan tersebut kapanpun, dimanapun selama terhubung dengan jaringan internet.

Menurut (Fajrin, 2012), layanan-layanan yang bisa dipilih dari *Cloud Computing*, yaitu :

a. *Infrastucture as a Service* (IaaS)

Layanan ini diberikan dengan cara menyediakan komponen-komponen berupa *server*, *hardware*, dan jaringan yang dibutuhkan dengan harga tertentu. Pengguna *cloud computing* dapat melakukan instalasi aplikasi yang digunakan pada infrastruktur tersebut.

b. *Platform as a Service* (PaaS)

Layanan yang menyediakan *system software* dan *software* pendukung yang diperlukan untuk membangun aplikasi yang akan dipasang pada *server* tersebut sesuai kebutuhan organisasi atau instansi. Organisasi atau instansi kemudian membangun aplikasi yang dibutuhkan pada *platform* ini dan menggunakannya.

c. *Software as a Service* (SaaS)

Layanan yang diberikan dengan menyediakan *software* maupun aplikasi yang dapat diakses pelanggan via internet. Penyedia layanan *cloud computing* berinteraksi dengan pengguna dan pelanggan melalui sebuah *front-end panel*.

3.1.6. Intrusion Prevention System (IPS)

Menurut Khadafi, Meilani, Arifin (2017:69), *Intrusion Prevention System* (IPS) adalah suatu metode yang mengkombinasikan teknik firewall dan metode *Intrusion Detection System* (IDS). Perangkat lunak *Intrusion Detection System* adalah

aplikasi berbasikan Linux yang dapat memantau sistem atau trafik jaringan dari penyalah gunaan atau aktivitas jahat yang kemudian dapat menghasilkan laporan ke dalam sistem. Sistem pada IPS dapat mencegah serangan yang akan masuk ke jaringan lokal dengan memeriksa dan mencatat semua paket data serta mengenali paket data sensor, disaat *attack* telah teridentifikasi, IPS akan menolak akses *block* dan mencatat log semua paket data yang telah teridentifikasi.

3.1.7. Suricata

Menurut Yuri, Budi Yan (2017 : 178) Suricata merupakan IDS yang mampu mendeteksi sebuah aktifitas jaringan dan mengidentifikasi ancaman serangan dibantu dengan rules yang terintegrasi.

3.1.8. *Virtual Private Server* (VPS)

Menurut Ricky, Eka, Tri Wahyu (2010:3) *Virtual private server* adalah *virtualisasi server*. Sebuah *physical server* dibagi menjadi beberapa *virtual private server* sehingga setiap VPS terlihat dan bekerja seperti sebuah *server* mandiri yang sebenarnya. Setiap VPS memiliki Full Root Acces, Sistem Operasi, dan pengaturan sendiri untuk init script, users, pemrosesan, filesystem, dan sebagainya termasuk *resources server* seperti CPU dan RAM yang berdiri sendiri. Berbeda dengan *shared hosting* yang menggunakan *resource server* bersamasama dan saling mempengaruhi, proses yang berjalan

pada suatu VPS tidak akan mempengaruhi VPS yang lain dalam satu *server*. VPS memungkinkan beberapa system operasi dijalankan pada satu mesin Server Fisik tunggal secara bersamaan. Hal ini dapat dilakukan tanpa melakukan partisi ulang dan boot ulang. Pada VPS yang disediakan akan dijalankan sistem operasi sesuai dengan yang diinginkan. VPS memungkinkan beberapa system operasi dijalankan pada satu mesin Server Fisik tunggal secara bersamaan. Hal ini dapat dilakukan tanpa melakukan partisi ulang dan *boot* ulang. Pada VPS yang disediakan akan dijalankan sistem operasi sesuai dengan yang diinginkan. Dengan cara ini maka pengguna dapat memboot suatu sistem operasi *linux*) sebagai sistem operasi tuan rumah (*host*) dan menjalankan sistem operasi lainnya. Sistem operasi yang dijalankan di dalam sistem operasi tuan rumah dikenal dengan istilah sistem operasi tamu (guest)

3.1.9. Jenis Serangan *Cyber*

Menurut Khadafi, Meilani, Arifin (2017:70), Beberapa jenis serangan yang umum terjadi pada system keamanan diantaranya *port scanning, sniffing, ICMP flood*, dan *hijacking*. *Port scanning* merupakan suatu proses untuk mencari dan membuka pada *port* komunikasi pada sebuah celah jaringan komputer. Dari hasil serangan tersebut akan didapatkan celah atau lubang kelemahan sebuah *server* yang diserang. Packet

sniffing merupakan pencegatan data paket-paket yang mengalir pada jaringan. Dengan sebuah aplikasi yang beroperasi pada lapisan ke 2 OSI dan juga kombinasi dari NIC yang berada pada mode *promiscuous* (mode mendengar) untuk menangkap semua *traffic* yang mengalir dari dan menuju ke jaringan internet pada suatu jaringan. *ICMP flood* dilakukan oleh seorang *hacker* dengan cara melakukan eksploitasi ke *system server* dengan tujuan untuk membuat suatu target menjadi hang, yang disebabkan oleh pengiriman sejumlah paket yang besar ke arah *target server*. *Exploiting* sistem ini dilakukan dengan mengirimkan suatu *command ping* dengan tujuan *broadcast* ataupun multicast dimana si pengirim dibuat seolah-olah adalah *target host*. *Hijacking* atau yang disebut dengan *man in the middle attack* (MITM) sebuah teknik serangan yang memanfaatkan kelemahan dari protokol TCP atau IP. Serangan dilakukan ketika terdapat diantara 2 *user* yang sedang berkomunikasi, tetapi terdapat seseorang yang lain yang secara aktif memonitor, men-*capture*, dan mengontrol komunikasi tersebut secara transparan.

a. *Backdoor*

Menurut Rudy (2015:75), *Backdoor* yang bila diterjemahkan secara bebas berarti pintu belakang ini digunakan juga sebagai istilah untuk menggambarkan adanya jalan tikus atau pintu tersembunyi yang disiapin oleh hacker agar bias tetap

menguasai computer korban tanpa perlu susah payah melakukan eksploitasi.

b. *Brute Force*

Menurut Rudy (2015:96), *Brute Force* adalah jenis serangan yang bertujuan untuk mencoba segala kemungkinan kombinasi karakter untuk mencari *account* yang *valid***.**

c. *Port Scanning*

Menurut Sofyan, Periyadi, Anang (2016:1173) Port Scanning merupakan metode mendeteksi port pada suatu target untuk melihat port apa saja yang aktif. Port scanning biasanya digunakan untuk memulai suatu serangan pada target yang akan diserang

d. *Denial of Service*

Menurut Beny (2016:37), *DoS attack* adalah serangan yang bertujuan untuk membuat sebuah *server* atau *website* tidak dapat diakses oleh user lain. Salah satu contoh serangan yang dapat dilakukan adalah dengan membanjiri jaringan dengan paket-paket sampah. Dengan menerapkan serangan ini, *server* penyedia *cloud* akan menjadi *down*, sehingga dapat dengan mudah dimasuki oleh seorang penyerang.

3.1.10. Linux

Menurut Hariyanto (2009:15), Linux adalah tiruan *(clone)* UNIX. Mulanya, pegembangan Linux dilakukan Linus Benedict

Torvalds, universitas Helsinki, Finlandia sebagai proyek hobby. Seluruh kode sumber Linux termasuk kenel, *device driver*, pustaka, program, dan kakas pengembangan disebarkan secara bebas dengan lisensi GNU GPL *(General Public Licence)* versi kedua. Kernel pertama yang dirilis pada publik adalah versi 0,01, pada tanggal 14 mei 1991. Kernel saat itu masih sangat sederhana. Kemudian, pada 14 maret 1994 dirilis versi 1.0.

3.1.11. Ubuntu

Menurut Sofana (2010:7), Ubuntu merupakan salah satu distro turunan Debian. Debian atau lengkapnya Debian GNU/Linux GNU merupakan singkatan dari "GNU is Not Unix". Proyek GNU dimulai pada tahun 1984, bertujuan untuk menghasilkan sebuah sistem operasi mirip Unix atau Unix-like yang bersifat *free*. Debian atau GNU/Linux adalah hasil dari proyek tersebut. Saat ini, debian telah dikembangkan menjadi berbagai distro turunan. Salah satunya bernama Ubuntu.

3.1.12. *Cloud Computing*

Menurut Sofana (2012:3), *Cloud Computing* merupakan sebuah model *client-server*, dimana *resource* seperti *server*, *storage*, *network* dan *software* dapat dipandang sebagai layanan yang dapat diakses oleh pengguna secara *remote* dan setiap saat.

Menurut Pratama (2014:60), Berdasarkan layanan *cloud computing* dibedakan menjadi tiga model yaitu:

1. *Infrastructure As a Service* (IAAS)

   Infrastructure As a Service (IAAS) merupakan jenis layanan pada *cloud computing* yang menekankan kepada layanan penyediaan sarana jaringan komputer *(computer network)*, perangkat keras jaringan, komputer *server*, media penyimpanan *(storage)*, *processor*, beserta dengan proses virtualisasi, yang menunjang proses komputasi.

2. *Platform As a Service* (PAAS)

   Platform As a Service (PAAS) atau *cloud* PAAS merupakan jenis layanan pada *cloud computing* yang menekankan kepada penyediaan *platform* untuk membantu proses pengembangan perangkat lunak secara cepat dan mudah. Layanan *platform* yang disediakan oleh *cloud* PAAS umumnya juga berbasis web, dimana didalamnya telah tersedia banyak fitur yang memudahkan *programmers* dan pengguna awam didalam mengembangkan aplikasi tanpa memerlukan banyak proses penulisan sumber kode *(coding).*

3. *Software As a Service* (SAAS)

Software As a Service (SAAS) merupakan jenis layanan yang diberikan oleh teknologi *cloud computing* kepada para penggunanya dalam bentuk pemakaian bersama perangkat lunak (aplikasi). Umumnya layanan SAAS disediakan dalam bentuk tatap muka berbasis web. Bisa dikatakan SAAS merupakan jenis layanan *cloud computing* yang paling banyak digunakan dan paling mudah digunakan oleh para pengguna komputer, khususnya pengguna akhir yang tidak terlalu membutuhkan pengetahuan teknis di dalam *instalasi* dan konfigurasi. Cukup dengan sebuah komputer/perangkat *mobile*, sistem operasi, aplikasi *web browser,* dan koneksi internet, seorang pengguna komputer dapat dengan mudah menggunakan layanan *cloud computing* tipe SAAS ini.

Berdasarkan model *deployment cloud computing* dibagi menjadi empat:

1. *Private Cloud*

Private Cloud dimaksudkan sebagai model *deployment cloud computing* yang ditujukan untuk penggunaan yang terbatas pada kalangan tertentu saja *(private),* model *deployment* ini umumnya banyak diterapkan untuk lingkungan laboratorium riset, sekolah, perpustakaan, gedung atau bangunan (kantor atau perusahaan), dan lain-lain.

2. *Public Cloud*

   Public Cloud merupakan model *deployment* pada teknologi *cloud computing*, dimana layanan *cloud computing* diletakan dilokasi *public* (misalkan dijaringan internet dan memiliki *ip public*) sehingga layanan data, informasi didalamnya dapat digunakan dan dibagikan dengan mudah keseluruh pengguna. Dari sisi para pengguna, *public cloud* tidak seperti *private cloud. Public cloud* menyediakan akses sebanyak mungkin kepada siapapun yang terhubung kedalam jaringan *cloud* yang menyediakan layanan *public cloud*.

3. *Community Cloud*

   Community Cloud merupakan model *deployment* pada *cloud computing* yang dibangun oleh satu atau beberapa buah komunitas. Komunitas yang tergabung biasanya memiliki tujuan, visi dan misi yang sama. Misalkan saja dalam contoh ini komunitas sistem operasi linux dan aplikasi-aplikasi *open source* dari berbagai kota atau daerah di Indonesia. Komunitas dalam hal ini juga mencakup instasi, organisasi, lembaga, maupun suatu kelompok tertentu.

4. *Hybrid Cloud*

   Hybrid Cloud adalah model *deployment cloud computing* yang merupakan gabungan dari *private cloud* dan *public cloud* pada model *deployment hybrid* ini, digunakan aturan atau SLA yang merujuk kepada data mana saja yang akan diletakan dimedia penyimpanan

(storage public cloud (internet dan data mana saja yang akan diletakan di *storage private cloud* (internet). Hal ini bertujuan untuk memudahkan didalam manajemen keamanan dan manajemen data. *Hybrid Cloud* menggabungkan kelebihan yang dimiliki oleh *private cloud* dan *public cloud*. Oleh karena itu, saat ini hingga kedepan nanti model *deployment hybrid cloud* inilah yang akan banyak dipilih dan digunakan.

Menurut Sofana (2012:20), Dalam layanan *cloud computing* ada beberapa komponen yang diperlukan, yaitu:

1. *Cloud Client*

   Ini karena *hardware*, aplikasi dan semua yang berkaitan dengan *cloud computing* dikembangkan untuk klien. Tanpa adanya *client* atau pengguna *software cloud computing*, semuanya akan sia-sia. *Client* untuk *cloud computing* ada dua jenis, yaitu komponen *hardware* atau kombinasi di dua tempat, yaitu kapasitas *hardware* lokal dari *security software*. Melalui optimasi *hardware* dengan *security*, aplikasi akan bisa dijalankan dengan mulus.

2. *Cloud Service*

   Salah satu alasan kenapa *cloud computing* menjadi populer adalah karena layanan ini diperlukan oleh dunia bisnis. Ini karena bisnis memerlukan cara untuk mengefisienkan proses bisnis yang berarti keuntungan akan meningkat.

3. *Cloud Applications*

   Service kadang dianggap sebagai aplikasi. Ini memang setengah benar karena *service* menyediakan fungsi. Adapun aplikasi adalah apa yang dikembangkan oleh *software developer* atau *programmer* dimana mereka harus fokus untuk memastikan aplikasi berjalan dengan benar.

4. *Cloud Platform*

   Di *Website* atau aplikasi normal yang tidak berhubungan dengan *cloud computing*, aplikasi akan berhubungan secara langsung dengan *server*. Namun di *cloud computing*, aplikasi dijalankan ke aplikasi lain yang disebut *platform*. *Platform* ini biasanya bahasa pemrograman seperti *AJAX*, *PHP* atau *Ruby on Rails.*

5. *Cloud Storage*

   Semua aplikasi dan fungsi harus di simpan pada media simpan. Media simpan *cloud* ini akan menyimpan data dan informasi sehingga fungsi bisa diimplementasikan dengan baik.

   Optimasi *storage* berkaitan dengan bagaiman fasilitas *storage* diproteksi dari berbagai ancaman serta serangan. Selain itu *cloud storage* juga berkaitan dengan konsisten serta nilai *uptime.* Semakin lama nilai *uptime* akan semakin andal media *storage cloud* ini.

6. *Cloud Infrastructure*

   Semua fungsi, *service* dan kemampuan *storage* untuk menyediakan data hanya bisa diakses jika infrastruktur optimal.

Infrastruktur ini bisa dianggap sebagai *platform* akhir yang memungkinkan semuanya bisa dijalankan. Setiap komponen-komponen diatas harus dioptimasikan sehingga aplikasi *cloud* bisa berjalan dengan baik dan aman.

3.1.13. Virtualisasi

Virtualisasi menurut Zaida (2013:4), adalah teknologi yang mengizinkan sistem komputer untuk membuat suatu sistem komputer bayangan didalam sistem komputer tersebut. *Virtualisasi server* adalah penggunaan perangkat lunak yang memungkinkan suatu perangkat keras untuk menjalankan beberapa sistem operasi dan *services* pada saat yang sama, sedangkan *virtual server* adalah penggunakaan perangkat lunak yang memungkinkan banyak perangkat keras untuk menjalankan suatu sistem secara terpadu.

3.1.14. Proxmox

Menurut Purbo (2012:37), Proxmox adalah sebuah distro linux *virtualisasi* berbasis Debian yang mengusung OpenZV dan KVM. Proxmox memungkinkan untuk melakukan manajemen terpusat dari banyak server fisik.

Menurut Athaila (2016:147), Proxmox adalah sebuah merek yang sudah cukup terkenal, terutama buat yang bergelut dengan solusi open source. Hal ini dikarenakan Proxmox dikembangkan berbasiskan sistem operasi Linux dari distribusi

Debian, sehingga keamanan sistem operasi virtualisasi yang satu ini tidak perlu diragukan lagi.

3.2. Penelitian Terdahulu

Hasil penelitian terdahulu digunakan sebagai pedoman dasar, acuan, pertimbangan, maupun perbandingan bagi penelitian terbaru yang sejenis, adapun penelitian terdahulu yang penulis gunakan seperti pada tabel 3.1 berikut :

Tabel 3.1 Tabel Penelitian Terdahulu

No	Judul Penelitian	Penulis dan Tahun	Hasil
1.	Analisis Teknik-Teknik Keamanan Pada *Cloud Computing* dan NEBULA (*Future Cloud*) ISSN : 2476-8812	Beny Nugraha (2016) .	Penulis mendapatkan penelitian ini menggunakan nebula cloud dan penyerangan dilakukan dengan 4 cara yaitu Snooping, Traffic analysis, Denial of Service, dan Man-in-the-middle. Dari serangan tersebut mendapatkan serangan traffic analysis dapat diatasi dengan onion routing.

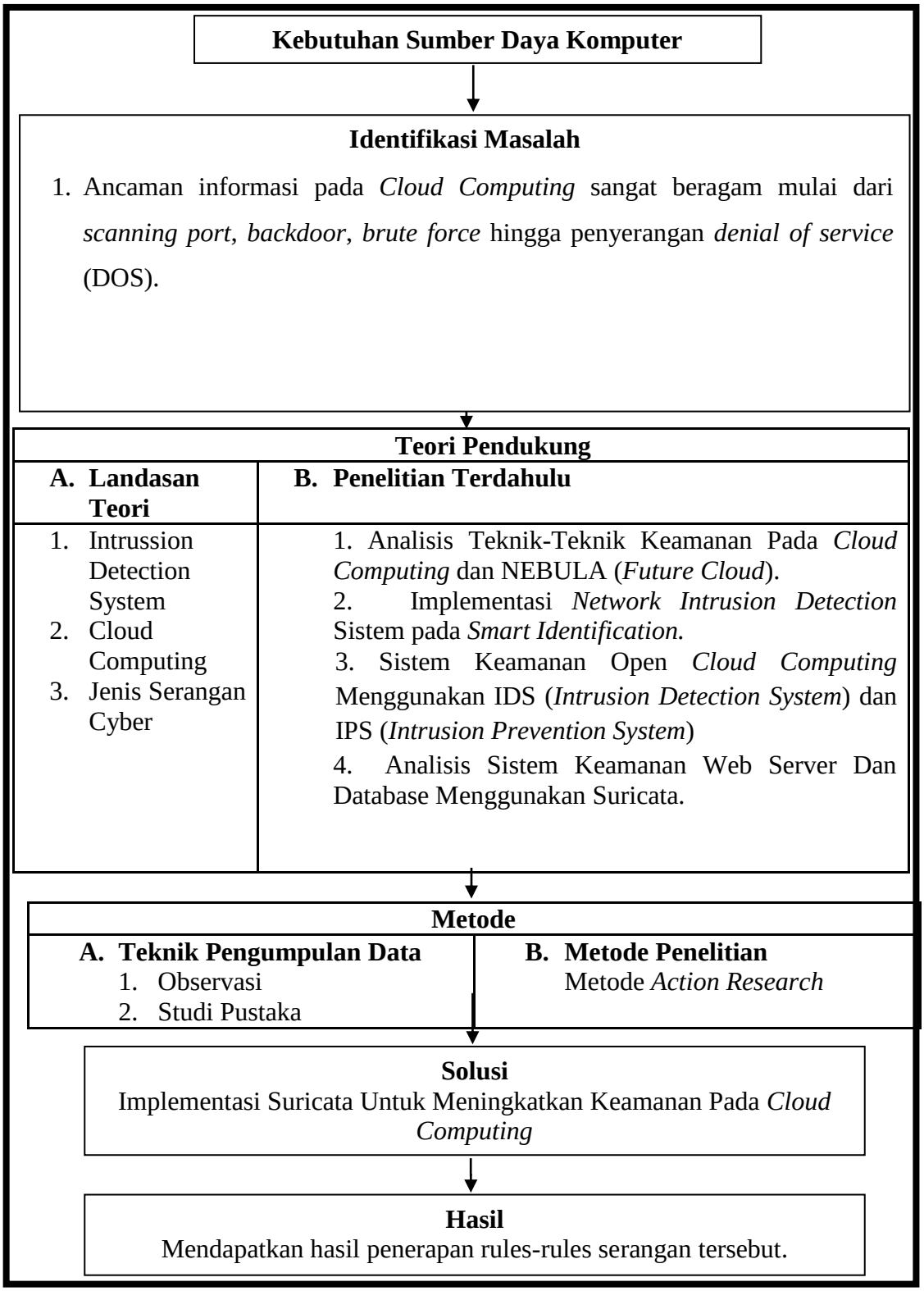
No	Judul Penelitian	Penulis dan Tahun	Hasil
2.	Implementasi *Network Intrusion Detection* Sistem pada *Smart Identification* *ISSN : 2442-5826*	Sofyan Hadi, Periyadi, dan Anang Sularsa (2016).	Penelitian ini menggunakan suricata untuk mendeteksi aktifitas ancaman serangan pada jaringan jaringan yang dibantu dengan rules yang telah ada, selain suricata penelitian ini juga menggunakan snorby untuk web interface, Barnyard2 untuk perekaman data, Gammu untuk sms gateway sehingga dimana ada aktifitas ancaman akan ada notifikasi.

No	Judul Penelitian	Penulis dan Tahun	Hasil
3.	Sistem Keamanan Open *Cloud Computing* Menggunakan IDS (*Intrusion Detection System*) dan IPS (*Intrusion Prevention System*) ISSN : 1411-7010	Shah Khadafi, Budanis Dwi Meilani, dan Samsul Arifin (2017).	Penelitian ini menggunakan perangkat laptop yang jadi *attacker* dan mengunakan 3 *server* IPS yaitu *Snort*, Barnyard, *BASE* menggunakan proxmox host dan *virtual OS* sebagai web server dan serangan yang digunakan yaitu *sniffing, port scanning*, dan *ddos*. sehingga menghasilkan hasil pengujian *scanning* dan hasil pengujian *d-dos* sehingga mendapatkan perbandingan serangan ketika IPS diaktifkan dan IPS dimatikan sehingga mendapatkan perbedaan yang signifikan.

No	Judul Penelitian	Penulis dan Tahun	Hasil
4.	Analisis Sistem Keamanan Web Server Dan Database Menggunakan Suricata. ISSN : 2579-7271	Nazwita dan Siti Ramadhani (2017).	dengan aturan yang telah ditetapkan akan memberikan tindakan Pass, Drop, Reject dan tools yang digunakan nmap untuk port scanning. Hasil yang didapatkan merumuskan rules suricata didapatkan dengan baik.

3.3. Kerangka Pemikiran

3.3.1. Kerangka Penelitian



Gambar 3.6 Kerangka Penelitian
(*Sumber : Sendiri*)

Pada gambar 3.9 kerangka penelitian peneliti menemukan serangan-serangan yang ada di cloud computing seperti *Scanning port, Backdoor, Brute force* dan hingga penerangan *denial of service (DOS)*. mengenai serangan tersebut keamanan informasi akan sangat terancam sehingga penulis juga menemukan software IDS Suricata untuk meningkatkan keamanan pada Cloud Computing.

Oleh karena itu peneliti tertarik melakukan implementasi Suricata untuk meningkatkan keamanan *Cloud Computing* bertujuan untuk mengimplementasikan rules ke sistem IDS Suricata untuk pendeteksian serangan *Port scanning, Brute force, Denial Of Service, Backdoor* dan mengukur efektifitas penerapan *rules-rules* serangan tersebut.

BAB IV

METODE PENELITIAN

4.1. Lokasi dan Waktu Penelitian

4.1.1. Lokasi

Tempat penelitian untuk skripsi ini dilakukan di Laboratorium Komputer di STMIK PalComTech yang berlokasi di jalan Basuki Rahmat No. 05 Palembang.

4.1.2. Jadwal Penelitian

Dalam penelitian ini, penulis menyusun segala kegiatan dalam sebuah jadwal penelitian yang berlangsung selama kurang lebih empat bulan. Dimulai dari bulan oktober 2018 hingga bulan januari 2018. Berikut jadwal penelitian di jabarkan pada tabel 4.1 sebagai berikut :

Tabel 4.1 Jadwal Penelitian

No.	Uraian Kegiatan	Bulan/Tahun															
		Oktober 2018				November 2018				Desember 2018				Januari 2019			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	*Diagnosis*																
	a. Mengumpulkan masalah serangan yang sering terjadi di *Cloud Computing*																
2	*Action Planning*																
	a. *Virtual Private Server (VPS)* Ubuntu dengan spesifikasi *processor Intel Xeon Gold* 6140 CPU @ 2.30Hz, *memory* 4 gb, dan *SSD* 80gb. b. *Tools attack* berupa *port scanning, brute force, denial of service* (DOS), dan *Backdoor.* c. *Software Suricata.*																
3	*Action Taking*																
	Pengguna melakukan serangan melalui *tools* yang telah disediakan.																

No.	Uraian Kegiatan	Bulan/Tahun															
		Oktober 2018				November 2018				Desember 2018				Januari 2019			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
4	*Evaluation*																
	Pada tahap ini penulis mendapatkan hasil dari 4 serangan ke Virtual Private Server *(VPS)* yang dideteksi oleh IDS Suricata dan mendapatkan hasil di log suricata.																
5	*Reflection*																
	Pada tahap ini penulis mendapatkan hasil dari pengujian serangan tersebut apakah suricata dapat mendeteksi serangan tersebut. Dan penulis juga menambah persentasi keamanan untuk meningkatkan keamanan suricata tersebut.																

4.2. Jenis Data

Dalam penelitian ini perlu di uraikan apakah data dalam penelitian ini merupakan data primer atau data sekunder.

4.2.1. Data Primer

Menurut Riadi (2016:48), Data Primer adalah data informasi yang diperoleh tangan pertama yang dikumpulkan secara langsung dari sumbernya. Data primer adalah data yang paling asli dalam karakter dan tidak mengalami perlakuan statistik apapun.

Penulis mengumpulkan data primer dengan menggunakan metode observasi. Data yang didapatkan oleh penulis berupa file.log dari hasil serangan ke *virtual private server* yang telah dipasang *Intrusion Detection System* (IDS) Suricata.

4.2.2. Data Sekunder

Menurut Riadi (2016:48), Data Sekunder adalah informasi tangan kedua yang sudah dikumpulkan oleh beberapa orang (organisasi) untuk tujuan tertentu dan tersedia untuk berbagi penelitian. Data sekunder tersebut tidak murni dalam karakter dan telah menjalani *treatment* setidaknya satu kali. Contoh data sekunder adalah data yang diperoleh dari Biro Pusat Statistik (BPS), buku, laporan, jurnal dan lain-lain.

Penulis mengumpulkan data sekunder melalui penelitian terdahulu, buku refrensi dan jurnal guna menunjang serta memperkaya pengetahuan tentang pemahaman dari penelitian yang akan dilakukan penulis.

4.3. Teknik Pengumpulan Data

4.3.1. Observasi

Menurut Nazir (2014:171), Pengumpulan data dengan *observasi* langsung dengan pengamatan langsung adalah cara pengambilan data dengan menggunakan mata tanpa ada pertolongan alat standar lain untuk keperluan tersebut.

Pada metode ini penulis melakukan pengujian terhadap *platform KVM*, Sistem Operasi *Ubuntu* dan mencatat hasil penyerangan ke *server* yang sudah diberi keamanan oleh Suricata dan yang belum diberi keamanan oleh Suricata.

4.3.2. Studi Pustaka

Menurut Nazir (2014:79), Studi kepustakaan merupakan langkah yang penting dimana setelah seseorang peneliti menetapkan topik penelitian, langkah selanjutnya adalah melakukan kajian yang berkaitan dengan teori yang berkaitan dengan topik penelitian.

Studi pustaka yang di lakukan penulis adalah mencari jurnal refrensi di internet serta mengunjungi perpustakaan untuk mencari buku-buku yang berhubungan dengan masalah yang akan diteliti.

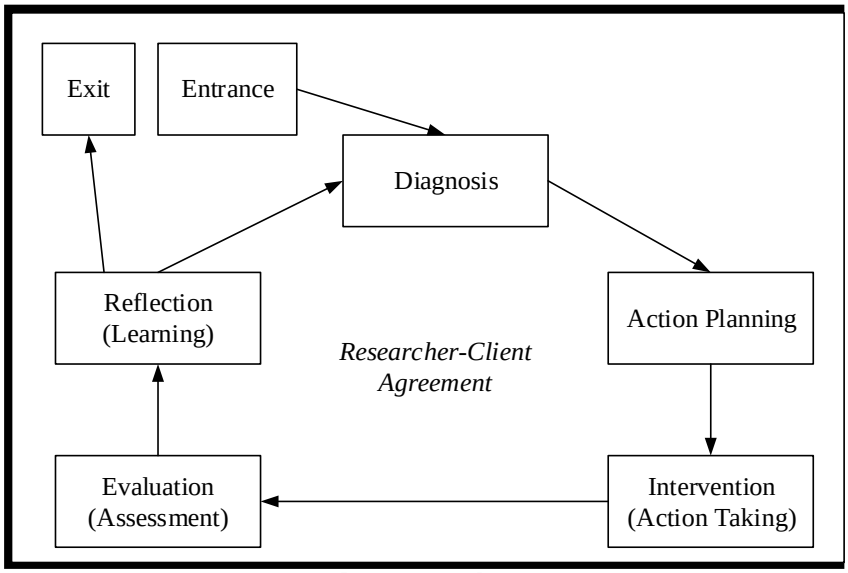
4.4. Jenis Penelitian

Jenis penelitian yang dilakukan pada penelitian ini adalah penelitian eksperimen. Menurut Nazir (2014:51), Metode Eksperimental merupakan metode penelitian yang sering digunakan, lebih-lebih dalam penelitian eksakta. Eksperimen adalah observasi dibawah kondisi buatan (*artifical condition*) dimana kondisi tersebut dibuat dan diatur oleh si peneliti. Dengan demikian, penelitian eksperimental adalah penelitian yang dilakukan dengan mengadakan manpulasi terhadap objek penelitian serta adanya kontrol.

Menurut Sugiyono (2017:72), Metode Eksperimen dapat diartikan sebagai metode penelitian yang digunakan untuk mencari pengaruh perlakuan tertentu terhadap yang lain dalam kondisi yang terkendalikan.

4.5. Metode Penelitian

Menurut Davison, Martinson & Kock dalam Mukmin (2017), menyebutkan penelitian tindakan sebagai metode penelitian, didirikan atas asumsi bahwa teori dan praktek dapat secara tertutup diintegrasikan dengan pembelajaran dari hasil intervensi yang direncanakan setelah diagnosis yang rinci terhadap konteks masalahnya. Terlihat tahapan Metode *Action Research* pada gambar 4.1.



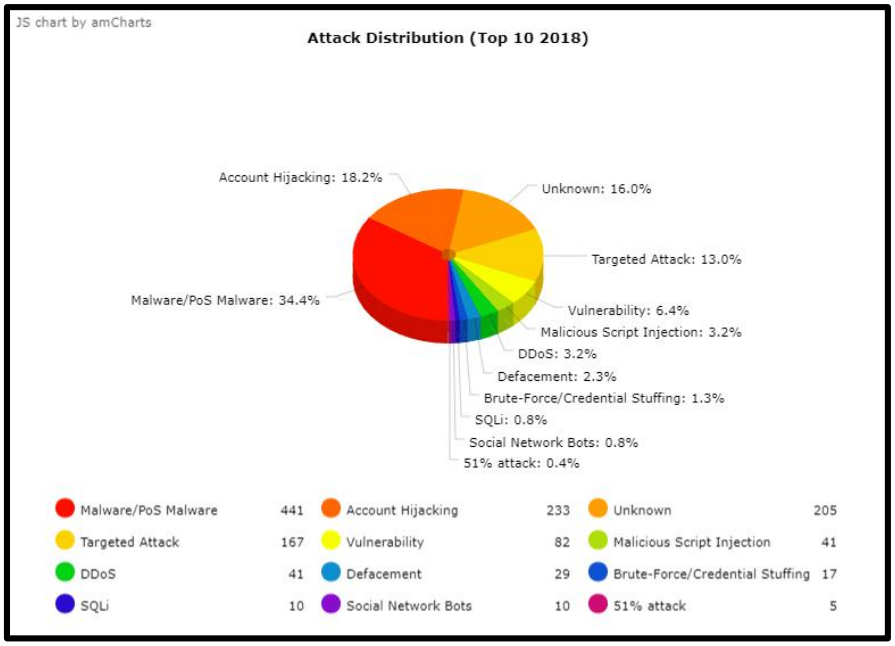
Sumber : (Mukmin, 2017)

Gambar 4.1 Tahapan Action Research

Keterangan tahapan metode diatas adalah :

1. ***Diagnosis***

Pada tahapan ini penulis mengidentifikasi permasalahan mengenai serangan *cloud computing* seperti *port scanning, brute force, denial of service,* dan *backdoor* bisa dilihat pada gambar 4.2 .



Sumber : (Hackmageddon**)**

Gambar 4.2 Serangan Cloud Computing

Penjelasan dari gambar diatas yaitu penulis mendapatkan data berbagai serangan ke *cloud computing*. Penulis sendiri mendapatkan data serangan yang akan diuji di penelitian ini yaitu port scanning masuk di bagian *Vulnerabilty* 6.4 %, *Brute-Force/Credential Stuffing* : 1,3 %, *Denial Of Service* : 3.2 %, dan *Backdoor* masuk di bagian *Malware/PoS Malware* sebesar : 34.4 %.

2. *Action Planning*

Pada tahapan ini penulis melakukan pemahaman terhadap alat apa saja yang dibutuhkan yaitu *:*

a. *Virtual Private Server (VPS)* yang berfungsi sebagai *server virtual* yang akan dipasang *software Intrusion Detection System* (IDS) Suricata, *Server* ini juga akan menjadi target dari 4 serangan yang dari *tools* yang telah disediakan. Penulis memilih *Virtual Private*

Server (VPS) *Digital Ocean* seharga 20$/*month* berlokasi di *Singapore* dan spesifikasi dari *Virtual Private Server* tersebut (VPS) *Processor Intel Xeon Gold* 6140 CPU @ 2.30GHz, *Memory* 4gb, dan *solid state disk* (SSD), *server* ini juga bersistem operasi *Linux Ubuntu 16.04.*

b. *Tools* serangan penulis juga menambahkan 4 *tools* serangan yang akan digunakan ke target *Virtual Private Server* (VPS) yang telah dipasang suricata. *Tools* serangan tersebut yaitu :

1. *Software Nmap* yang berfungsi sebagai *software* serangan *scanning port* yang berfokus mencari pada *port* yang terbuka disuatu *server*.

2. *Software Hydra* yang berfungsi sebagai *software* serangan *Brute Force software* tersebut akan melakukan *login* paksa secara berkala dari *word list password* yang telah disediakan penyerang.

3. *Software Denial Of Service* Ha3MrX berfungsi sebagai *software* serangan, cara kerja *software* serangan tersebut akan mengirimkan *packet* yang berlebihan ke sebuah *server* yang akan menyebabkan server menjadi down ataupun offline.

4. *Software Rootkit-Ninja* berfungsi sebagai *backdoor* apabila suatu penyerang mendapatkan akses *user software backdoor* ini akan mengubah akses *user* tersebut menjadi akses

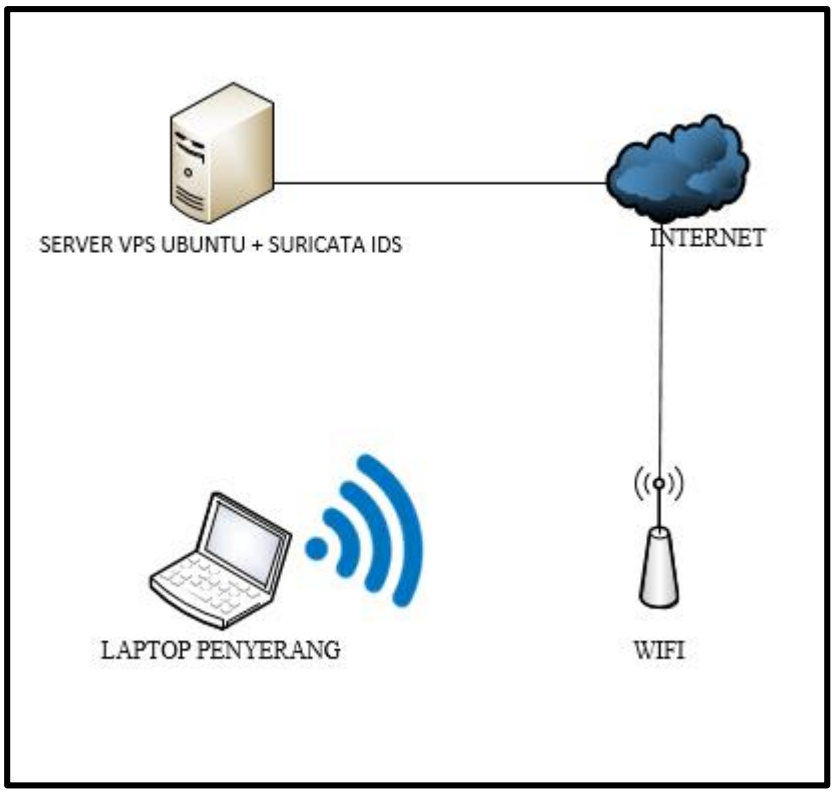
root, dengan cara dipanggil ulang *rootkit* akan merubah *user* biasa menjadi *user root*.

c. Laptop Penyerang penulis juga menambahkan perangkat laptop untuk menyerang. Laptop tersebut menggunakan sistem operasi *linux* yaitu *Parrot OS* yang akan menggunakan *tools* yang telah disediakan.
d. *Software Suricata* berfungsi sebagai *Intrusion Detection System* yang akan mendeteksi serangan apa yang akan datang ke *Virtual Private Server* (VPS) yang telah disediakan oleh penulis.
e. *Rules file suricata* berfungsi sebagai *rules* suricata untuk dipasang di *software Intrusion Detection System* suricata *file rules* tersebut berfungsi sesuai *rules* apa yang akan dipasang untuk mendeteksi serangan ke *Virtual Private Server* (VPS) tersebut.

3. *Intervention* (*Action Taking*)

Tahap ini juga peneliti mulai melakukan serangan menggunakan *tools* yang telah disiapkan dan mengkonfigurasi *IDS* Suricata yang telah disiapkan setelah selesai penguji akan menuliskan bagaimana *System* Suricata bekerja untuk mendeteksi serangan *port scanning, brute force, denial of service,* dan *backdoor* di *Cloud Computing*.

a . Topologi Penyerangan



Gambar 4.3 Topologi Alur Penyerangan

Pada gambar 4.3 topologi alur penyerangan ini menjelaskan *server* dengan PC penyerang berada di internet *gateway* berbeda. karena *server* menggunakan *Virtual Private Server Ubuntu* yang disewa sedangkan *PC* penyerang berada ditempat yang berbeda dengan *server*, *PC* penyerang terhubung dengan jaringan *wireless*.

4. *Evaluation (Assessment)*

Bagian tahap ini penulis mengumpulkan hasil dari penyerangan yang ada di *log* Suricata dan penulis juga mengevaluasi apakah Suricata dapat mendeteksi serangan *port scanning, brute force, denial of service,* dan *backdoor* ke *Cloud Computing.*

5. *Reflection (Learning)*

Pada tahap ini penulis mendapatkan hasil dari pengujian serangan tersebut apakah suricata dapat mendeteksi serangan tersebut. Penulis juga menambah *rule* untuk meningkatkan keamanan suricata tersebut.

4.6. Alat Dan Teknik Pengujian

4.6.1. Alat dan Bahan

Beberapa hal yang perlu diperhatikan dalam implementasi Suricata untuk meningkatkan keamanan pada *Cloud Computing* agar dapat berjalan dengan lancar sesuai dengan kebutuhan.

Implementasi Suricata pada *Cloud Computing* dilakukan menggunakan *Virtual Private Server* (VPS) yang ditentukan oleh penulis. Adapun beberapa spesifikasi yang digunakan adalah sebagai berikut:

Spesifikasi perangkat keras :

1. *Intel Xeon Gold* 6140 CPU @ 2.30Hz
2. *RAM 4 GB*
3. *Solid State Drive (SSD) 80 GB*

Spesifikasi perangkat lunak :

1. *Ubuntu 16.04.5 LTS*
2. *Parrot OS*
3. *Software Suricata*.

4.6.2. Teknik Pengujian

Pada teknik pengujian ini penulis akan menguji kemanan *Cloud Computing* yang telah dipasang *Intrusion Detection System* (IDS) Suricata dengan skenario serangan sebagai berikut:

a. *Port Scanning*

Disini penulis akan melakukan *port scanning* menggunakan *software nmap* untuk melihat *port* yang terbuka di *Virtual Private Server* (VPS) *Ubuntu* yang telah dipasang suricata yang telah dipasang oleh penulis.

b. *Brute Force*

Disini penulis akan melakukan teknik penyerangan *brute force SSH attack* berbasis *script python* yaitu dimana serangan yang bertujuan untuk mencoba segala kemungkinan kombinasi karakter untuk mencari *account* yang *valid* terhadap *login SSH* yang berada di *Virtual Private Server* (VPS) *Ubuntu* telah disediakan oleh penulis**.**

c. *Denial of Service*

Disini penulis akan melakukan teknik penyerangan *Denial of Service* dengan software Denial Of Service berbasis *script python* dimana *denial of service* akan langsung menyerang *Ip address Virtual Private Server (VPS).*

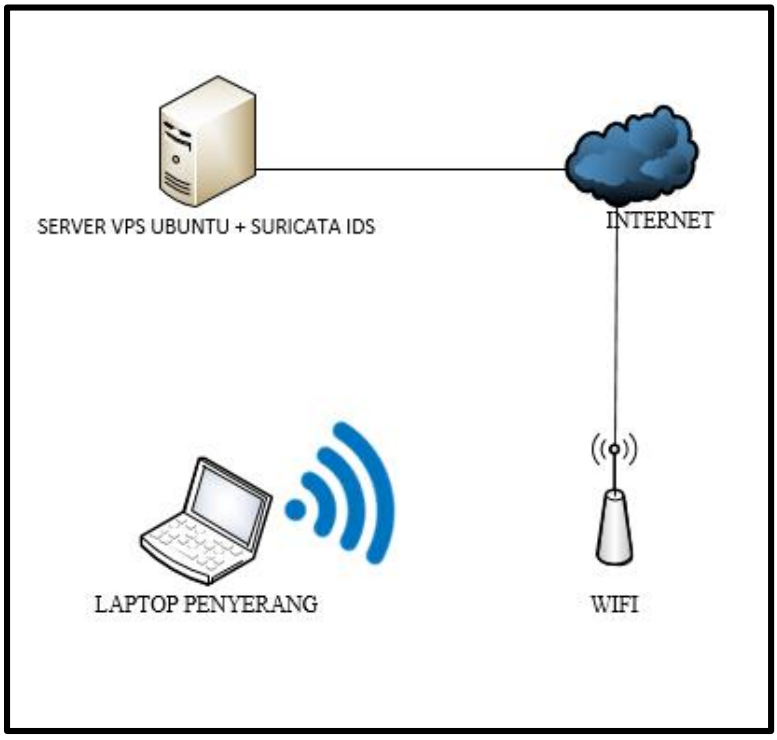
d. *Backdoor*

Disini penulis melakukan teknik *backdoor* melalui *software rootkit server* dimana bila *rootkit* telah di *install* menggunakan *user root* dan ketika *user* biasa memanggil *rootkit* tersebut maka akan langsung berubah hak akses menjadi *root***.**

Hasil penelitian yang akan didapat yaitu :

1. Penulis akan mendapatkan hasil masing-masing *log* dari serangan *Port Scanning, Brute force, Denial of service, Backdoor.*
2. Penulis akan mengoptimalkan sistem suricata untuk memaksimalkan keamanan di *Cloud Computing.*

4.6.3. Topologi Jaringan



Gambar 4.4 Topologi Jaringan

Gambar 4.2 topologi ini menjelaskan *server* dengan *PC* penyerang berada di internet *gateway* berbeda. Karena *server* menggunakan *Virtual Private Server Ubuntu* yang disewa berbeda lokasi sedangkan PC penyerang berada ditempat yang berbeda dengan *server*, *PC* penyerang terhubung dengan jaringan *wireless*.

BAB V

HASIL DAN PEMBAHASAN

5.1. Hasil

Dalam peneliti ini penulis menggunakan metode *Action Research.* berikut merupakan tahapan sistem informasi akademik berdasarkan tahapan dalam metode *Action Research.*

5.1.1. *Diagnosing*

Pada tahap ini penulis akan memilih perangkat apa saja yang akan dibutuhkan untuk melakukan penelitian, serangan yang sering terjadi di *Cloud Computing* seperti *port scanning, bruteforce, denial of service, backdoor*. Berikut adalah perangkat keras yang dapat dilihat pada tabel 5.1.

Tabel 5.1 Perangkat Keras

No	Nama Perangkat	Spesifikasi	Jumlah	Kegunaan
1	*Virtual Private Server (VPS) Singapore*	**Prosesor: Intel Xeon Gold 6140 CPU @ 2.30GHz* **RAM : 4GB DDR3* *Penyimpanan: *80gb SSD*	1 Buah	*Server* SURICATA , yang bersistem operasi *Ubuntu 16.0 TLS*
2	Laptop Acer Travelmate 4350	**Prosesor : Intel(R)Core(TM) i3-2350M (3M Cache, 2.30 Ghz)* **RAM : 4GB DDR3* **Media penyimpanan : 320GB HDD* **GPU : Intel HD*	1 Buah	Laptop Penyerang

	Nama Perangkat	Spesifikasi	Jumlah	Kegunaan
		*Konektivitas : *Wifi 802.11 b/g/n + Bluetooth 2.0, 10/100 Ethernet Port* **Port 3x USB 2.0, 1x*		

Berikut adalah perangkat lunak yang dapat dilihat pada tabel 5.2.

Tabel 5.2 Perangkat Lunak

No	Nama Perangkat	Deskripsi
1	*Suricata 4.1.0-dev*	*Sofware Intrusion Detection System (IDS).*
2	*Ubuntu Server TLS 16.0*	Sistem Operasi *Server* untuk Suricata.
3	*Putty*	*Software* Untuk Mengakses *Virtual Private Server* (VPS).
4	*Parrot OS*	Sistem Operasi yang digunakan untuk menyerang *Virtual Private Server* (VPS) yang telah dipasang Suricata.
5	*Hydra*	*Software* yang digunakan untuk memasukan *password* secara berkala melalui *word list* yang telah disediakan.
6	*Nmap*	*Software attack* untuk melihat *port* yang terbuka pada *server*.
7	*Denial Of Service Ha3MrX*	Sebagai *software* serangan, cara kerja *software* serangan tersebut akan mengirimkan *packet* yang berlebihan ke sebuah *server* yang akan menyebabkan *server* menjadi *down* ataupun *offline*.
8	*Rootkit-Ninja*	Penyerang mendapatkan akses *user software backdoor* ini akan mengubah akses *user* tersebut menjadi akses *root*, dengan cara dipanggil ulang *rootkit* akan merubah *user* biasa menjadi *user root*.

5.1.2. *Action Planning*

Pada tahapan ini penulis melakukan pemahaman terhadap alat dan *tools* penyerangan apa saja yang dibutuhkan yaitu :

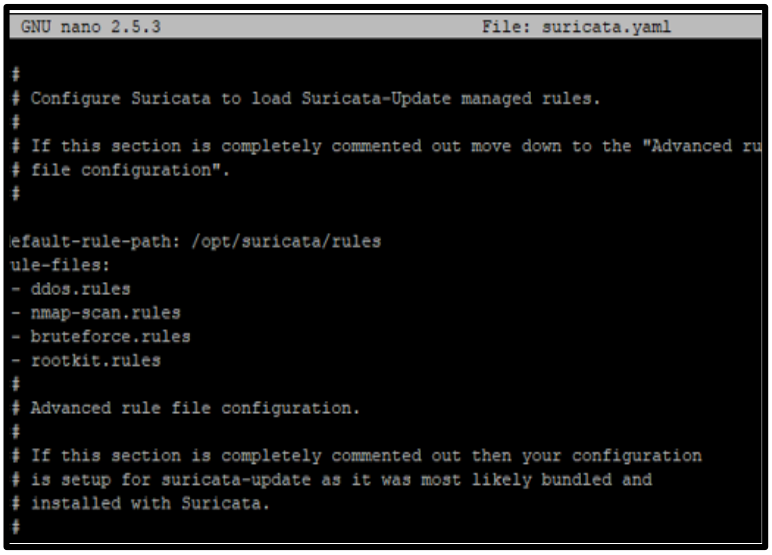
1. Menyiapkan *Virtual Private Server* yang bersistem operasi *Ubuntu Server* TLS 16.0. yang berlokasikan di *Singapore*.
2. Menyiapkan *software* suricata untuk *intrusion detection system* dan menyiapkan *rules file* suricata untuk 4 serangan tersebut.
3. Menyiapkan 4 *tools* untuk serangan yaitu *:*
   a. *Nmap* berfungsi berfungsi sebagai *scanning port*.
   b. *Hydra* berfungsi berfungsi sebagai *software brute force*.
   c. *Denial Of Service Ha3MrX* berfungsi sebagai *denial of service tools*.
   d. *Rootkit-Ninja* berfungsi sebagai *backdoor*.
4. Melakukan serangan *port scanning, brute force, denial of service* dan *backdoor* tersebut ke *Cloud Computing*.
5. Mendapatkan Hasil *log* dari serangan *port scanning, brute force, denial of service* dan *backdoor* pada *Cloud Computing*.

5.1.3. *Intervention (Action Taking)*

5.1.3.1. Konfigurasi Suricata

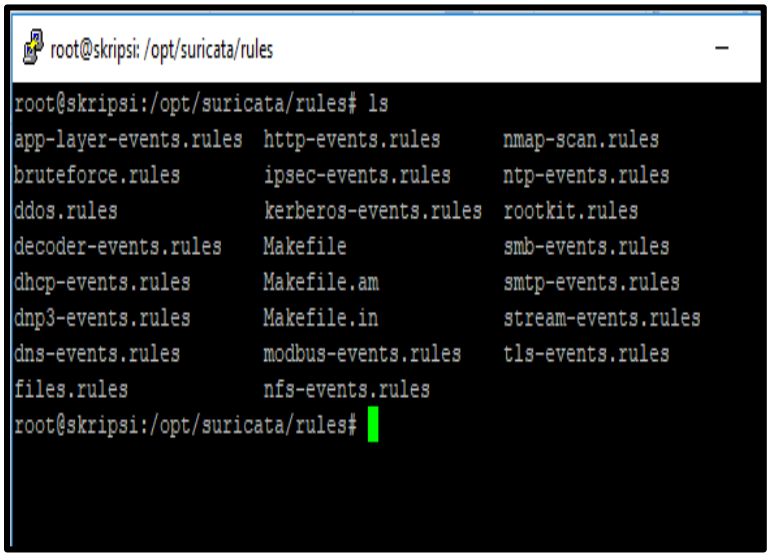
Pada pengujian ini peneliti mulai melakukan konfigurasi *Intrusion Detection System* (IDS) suricata dan menambahkan *file rules* untuk mencegah serangan ke *virtual private server* yang berada direktori /opt/suricata/rules/,

penulis juga menambahkan *list rules* ke *file* konfigurasi suricata.yaml yang berada di dalam direktori /opt/suricata/ di *Virtual Private Server* (VPS) yang telah disediakan oleh penulis dapat dilihat pada gambar 5.1.



Gambar 5.1 : *File* konfigurasi Suricata.yaml.

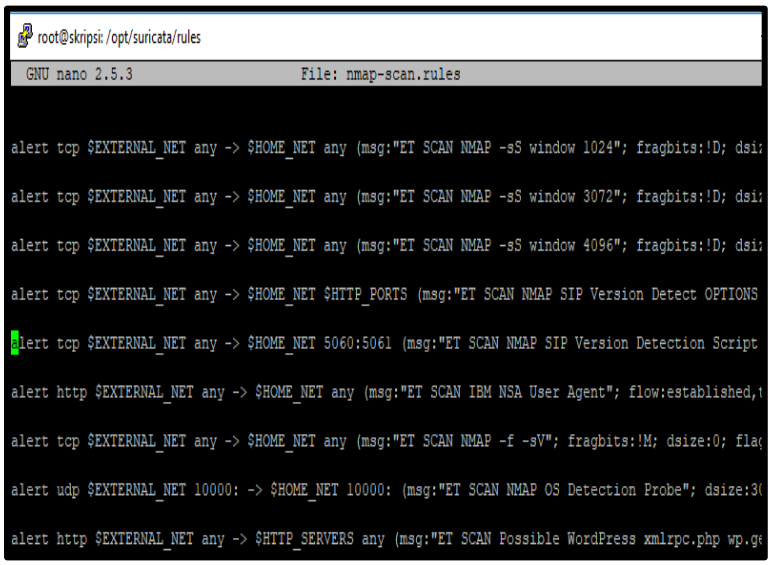
Pada gambar 5.1 penulis melakukan konfigurasi suricata yang berada pada files suricata.yaml, penulis juga melakukan konfigurasi *rules* suricata, *files* konfigurasi suricata tersebut berada pada direktori /opt/suricata/rules dan *list rules files* tersebut ada 4 yaitu *ddos.rules, nmap-scan.rules, bruteforce.rules, dan rootkit.rules*.



Gambar 5.2 : list file rules serangan suricata

Pada gambar 5.2 penulis hanya menggunakan *files rules* yang berfokus pada serangan yang sudah ditentukan seperti *ddos.rules, nmap-scan.rules, bruteforce.rules dan rootkit.rules.*

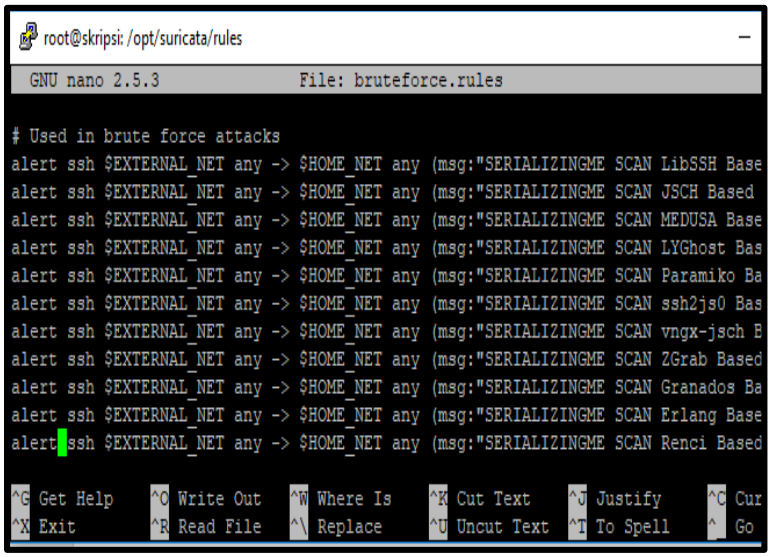
5.1.3.2. *File rules scanning port*



Gambar 5.3 : *file rules scanning port*

Pada gambar 5.3 penulis melakukan penambahan *file rules* untuk serangan *port scanning*, dimana dalam *rules* tersebut akan memberikan *alert Transmission Control Protocol* (TCP) bila ada serangan ke *HOME_NET* akan diberi peringatan ke *log* suricata yang berada pada /var/log/suricata/fast.log .

5.1.3.3. File rules *brute force*



Gambar 5.4 : *file rules brute force*

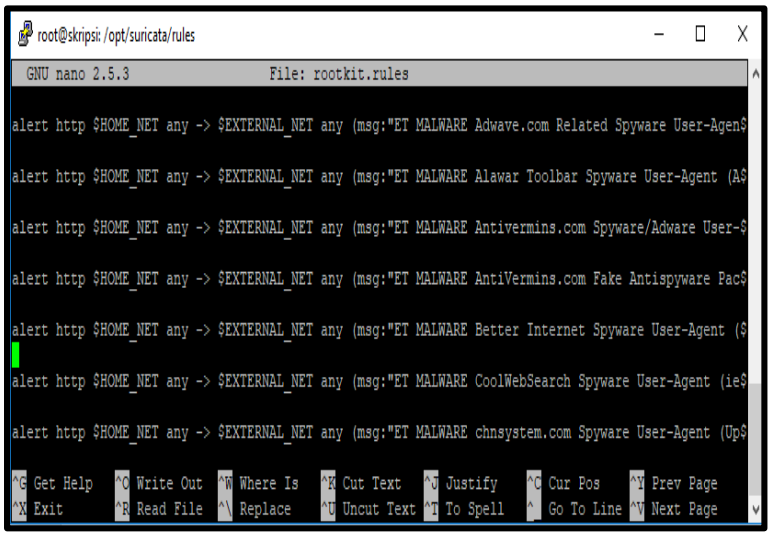
Pada gambar 5.4 penulis menambahkan *files rules bruteforce*, yaitu *alert* berfokus kepada *Secure Shell* (SSH). Yaitu akan memberikan pesan *SERIALIZINGME Scan LibSSH Base*.

5.1.3.4. File rules *denial of service*

Gambar 5.5 : *file rules denial of service.*

Pada gambar 5.5 penulis menambahkan *files rules denial of service*, guna untuk memberikan peringatan bila ada serangan *alert* di *protocol User Datagram Protocol* (UDP) dan *Hypertext Transfer Protocol* (HTTP), Yaitu akan memberikan pesan *ET Dos DOS*.

5.1.3.5. Files rules *backdoor*

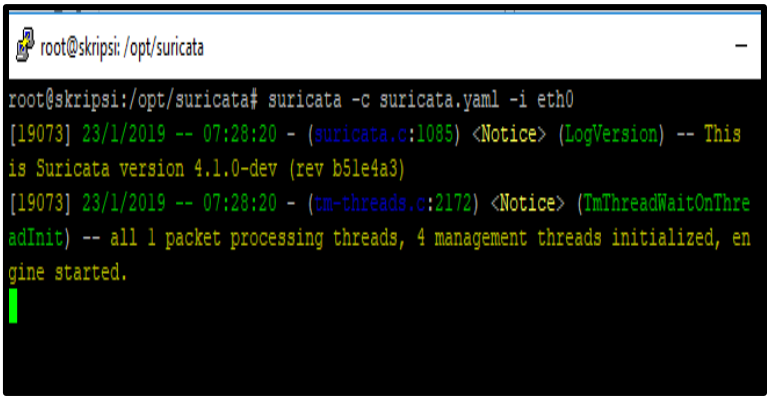


Gambar 5.6 : *file rules backdoor*

Pada gambar 5.6 penulis menambahkan *alert* di *protocol Hypertext Transfer Protocol* (HTTP) apabila ada kegiatan yang mencurigakan datang dan akan memberikan pesan seperti *backdoor, rootkit dan spyware*.

5.1.3.6. Menjalankan *Software Intrusion Detection System* Suricata.

Pada bagian ini penulis mengoperasikan *software Intrusion Detection System* Suricata dengan menggunakan perintah suricata -c suricata.yaml -i eth0, dan juga penulis sudah menambahkan *rules* untuk mendeteksi serangan ke *virtual private server* yang akan dijalankan dapat dilihat pada gambar 5.7.

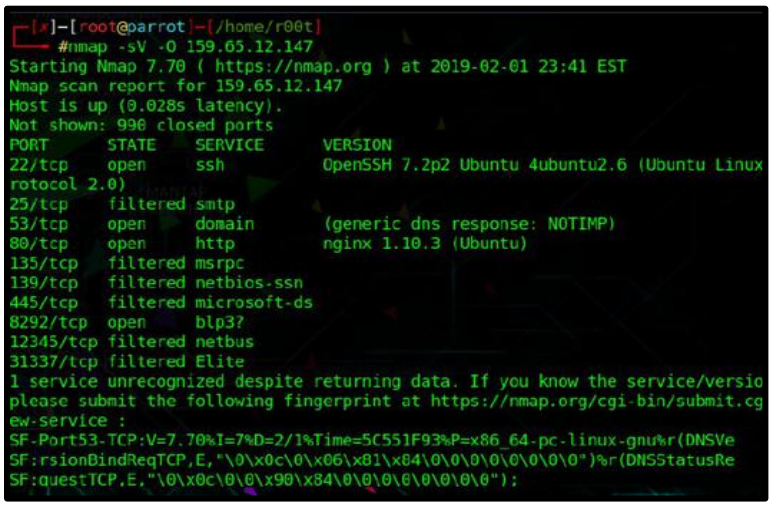


***Gambar 5.7. : Software* suricata dijalankan.**

5.1.3.7. Serangan *scanning port* dengan nmap.

Penulis melakukan serangan ke *Virtual Private Server* dengan menggunakan *tools attack nmap* untuk melakukan serangan *scanning port*, yaitu untuk mengetahui

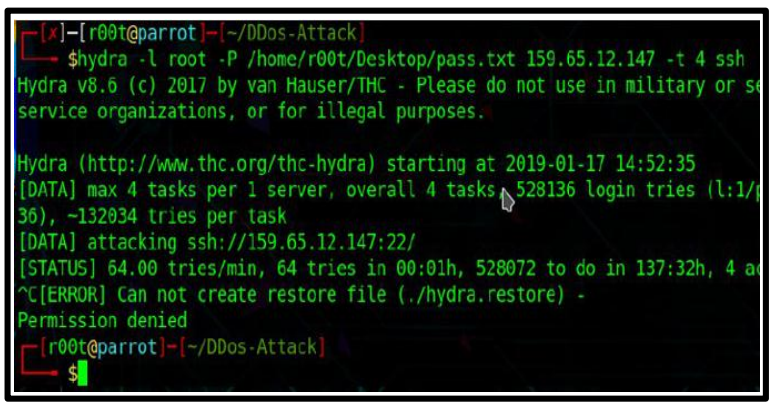
port apa saja yang terbuka di *Virtual Private Server Ubuntu* tersebut. dapat dilihat pada gambar 5.8.



***Gambar 5.8. : Scanning Port* dengan nmap.**

5.1.3.8. Serangan bruteforce dengan hydra

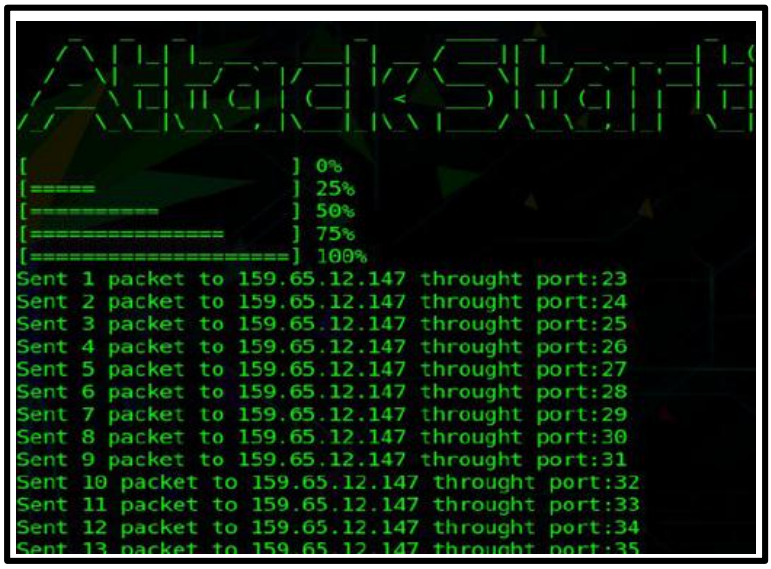
Penulis melakukan serangan *brute force* ke target *virtual private server* tersebut. dengan menggunakan sistem operasi *linux parrot OS* yang berfokus kepada *service port* 22 yaitu *service secure shell* (SSH) *login* metode yang digunakan *bruteforce* dengan *word list password* yang telah disediakan oleh penulis dapat dilihat pada gambar 5.9.



***Gambar 5.9 : Bruteforce* attack dengan hydra.**

5.1.3.9. Serangan *Denial Of Service*

Penulis melakukan serangan *denial of service* dimana serangan tersebut akan mengirimkan *sent packet* yang berlebihan, penulis menggunakan *script python* untuk mengirimkan *Denial Of Service*. Targetnya yaitu *virtual private server* tersebut dapat dilihat pada gambar 5.10.



Gambar 5.10 : *Denial Of Service* attack.

5.1.3.10. Serangan Backdoor

Penulis melakukan *install rootkit-Ninja* yang telah disediakan. *Rootkit-ninja* tersebut dipasang di *folder* /tmp/rootkit-Ninja. Penulis juga merubah hak akses *file* Callback.sh dan .rootkit-ninja. Fungsi *rootkit* ini yaitu sebagai *backdoor* bila sudah dipasang dan dipanggil kembali akses *user* biasa akan berubah menjadi *root user* dapat dilihat pada gambar 5.11.

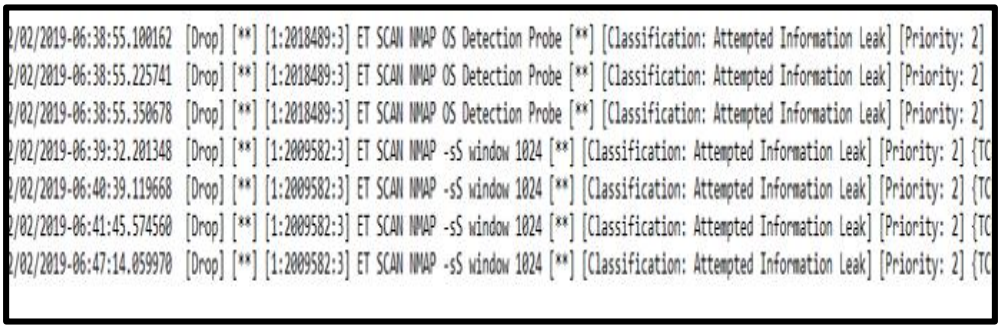
Gambar 5.11 : *Backdoor attack* dengan *rootkit-ninja*.

5.1.4. *Evaluation* (*Assesment*)

Berikut adalah ringkasan hasil *log* dari 4 serangan hasil serangan *log* berada di /var/log/suricata/ file tersebut bernama fast.log, dengan metode *scanning port, brute force, denial of service dan backdoor.*

5.1.4.1. *File log* serangan Nmap

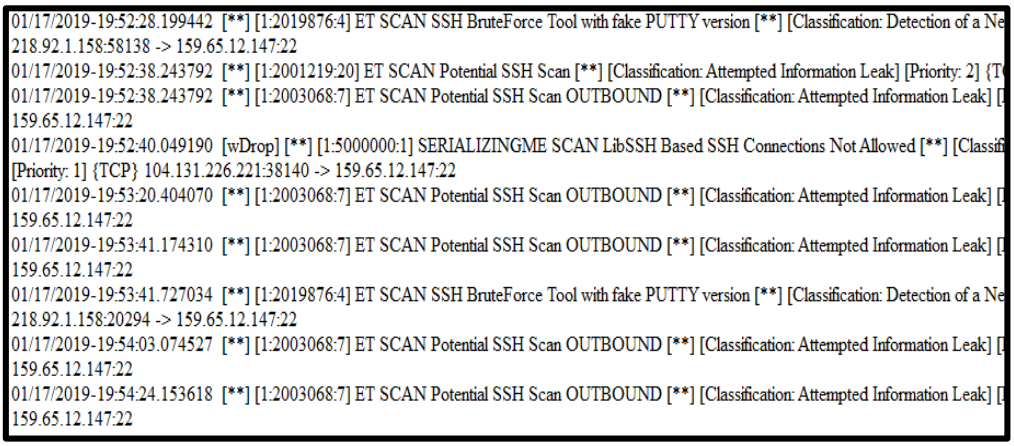
Berikut ini adalah bukti *log* hasil serangan *scanning port* dengan *nmap* yang di deteksi oleh *Intrusion Detection System* (IDS) Suricata, dimana terdeteksi *scan suspicious*, bisa dilihat pada gambar 5.12.



Gambar 5.12 : *log* dari serangan *scanning port.*

5.1.4.2. *File log* serangan *bruteforce*

Berikut ini adalah bukti *log* hasil dari serangan *bruteforce* yang menggunakan aplikasi *hydra* yang telah terdeteksi oleh *Intrusion Detection System* (IDS) Suricata, dimana terdeteksi kegiatan mencurigakan seperti *scan ssh bruteforce tool*, Bisa dilihat pada gambar 5.13.

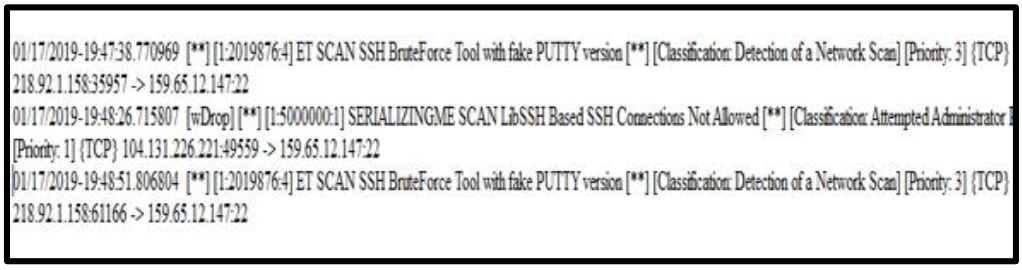


Gambar 5.13 *: log* dari serangan *bruteforce*.

5.1.4.3. File log serangan *Denial of Service*

Berikut ini adalah bukti log hasil dari serangan *denial of service* yang menggunakan *script python* yang telah

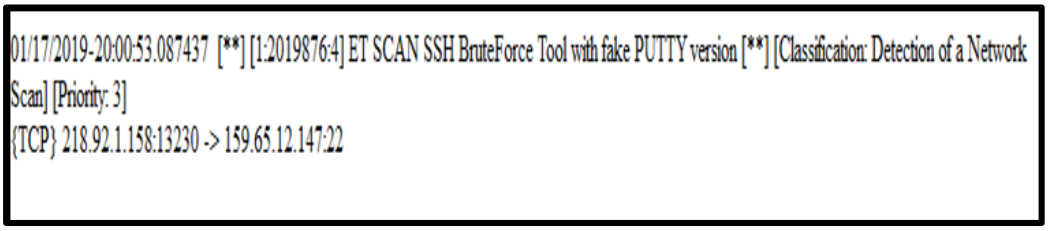
terdeteksi oleh *Intrusion Detection System* (IDS) Suricata, terdapat kegiatan mencurigakan seperti *Scan LibSSH Based SSH Connection not allowed,* bisa dilihat pada gambar 5.14



Gambar 5.14 *: log* serangan *denial of service*.

5.1.4.4. *File log* serangan *backdoor*

Berikut ini adalah bukti *log* hasil serangan dari *backdoor* yang berbeda dari *log* serangan sebelumnya yaitu sedikit hanya mendeteksi satu kegiatan yang mencurigakan yaitu *SSH BruteForce Tool with fake putty version*, Bisa dilihat pada gambar 5.15.



Gambar 5.15 : log serangan *backdoor*.

5.1.4.5. *Alert Log* Suricata ke Telegram

Disini juga penulis menambahkan *log* suricata *to telegram* guna untuk mengoptimalkan keamanan *Cloud Computing* yaitu menggunakan *file swatchdog* untuk melihat *REGEX* yang sesuai dengan serangan tersebut, yaitu

ditambahkan di *file* konfigurasi .swatchdogrc ke *file* fast.log dan juga *software swatchdog* akan menjalankan *file shell* telegram-bot.sh untuk mengirimkan notifikasi suricata tersebut ke telegram seperti gambar 5.16.

```
watchfor /Attempted Information Leak]/
      exec bash telegram-bot.sh  "Serangan Bruteforce bos di VPS MU"
      echo red
      throttle 00:01:00
```

Gambar 5.16 : konfigurasi file .swatchdogrc

Penulis menggunakan *bash script* untuk bot telegram yaitu konfigurasi di apiToken yaitu diisi token dari bot telegram yang sudah disediakan untuk mengirimkan notifikasi ke bot tersebut. Penulis juga menambahkan *userChatid* yaitu adalah token untuk koneksi pesan ke akun telegram bot yang telah dibuat oleh akun penulis. Pada gambar 5.17.

```
#!/bin/bash
message=$1
dt=`date '+%d/%m/%Y %H:%M:%S'`
IP=$(ip a s|sed -ne '/127.0.0.1/!{s/^[ \t]*inet[ \t]*\([0-9.]\
+\)\/.*$/\1/p}')
apiToken=<TELEGRAM BOT TOKEN>
userChatId=<CHAT ID TELEGRAM>
sendTelegram() {
  curl -s -X POST \
  https://api.telegram.org/bot$apiToken/sendMessage \
  -d text="$IP : $message" -d chat_id=$userChatId
  echo  $dt : $IP : $message  \
  >> /var/log/sendTelegramMessage.log
}
if  [[ -z "$message" ]]; then
  echo "Please add message to me!"
else
  sendTelegram
fi
```

Gambar 5.17 : telegram-bot.sh konfigurasi.

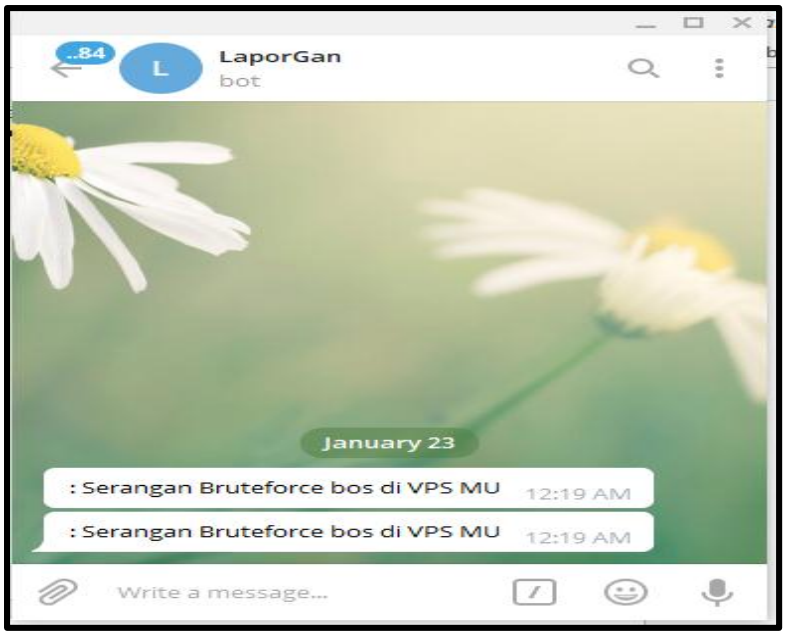
Penulis mencoba serangan ulang *brute force* menggunakan *hydra* untuk menguji apakah notifikasi ke telegram sudah berfungsi dengan baik sesuai *regex* yang telah disetting di *file* .swatchdogrc dan telegram-bot.sh pada *software* IDS suricata seperti gambar 5.18.

```
┌─[root@parrot]─[/home/r00t]
└──╼ #hydra -l root -P /home/r00t/pass.txt 159.65.12.147 -t 4 ssh
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2019-01-22 12:19:11
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip wa
iting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 2151220 login tries (l:1/p:215
1220), ~537805 tries per task
[DATA] attacking ssh://159.65.12.147:22/
```

Gambar 5.18 : Serangan *brute force* ulang dengan *Hydra*

Penulis mendapatkan hasil notifikasi dari bot LaporGan yang telah dibuat oleh penulis untuk mendapatkan notifikasi bila ada serangan yang masuk di *log* suricata yaitu fast.log, Jenis notifikasi juga ditulis berbeda setiap serangan yang dilakukan oleh penulis seperti gambar 5.19.



Gambar 5.19 : Notifikasi Telegram dari serangan *bruteforce*

5.1.5. *Reflection (Learning)*

Pada tahap ini, penulis membuat laporan dengan hasil yang telah di dapatkan setelah melakukan penelitian.

5.1.5.1. Hasil dari pengujian serangan yang di deteksi oleh suricata.

Penulis mendapatkan hasil yang optimal dari hasil *log* serangan yang dideteksi oleh *intrusion detection system* (IDS) Suricata yang berada pada dir /var/log/suricata/fast.log, dan juga *rules* suricata tidak hanya terfokus pada satu serangan tetapi dimana serangan tersebut menyerupai serangan yang lain maka akan sama terdeteksi oleh suricata tersebut. Dari *file log* tersebut juga dijelaskan waktu serangan tanggal jam dan penjelasan deskripsi dari serangan tersebut. Penulis juga menambahkan *fileswatcher* untuk mengirimkan notifikasi bila ada *regex* yang masuk di *file log* suricata yang akan dikirimkan ke telegram bila ada serangan.

Tabel 5.3 : Hasil *log* dan dampak

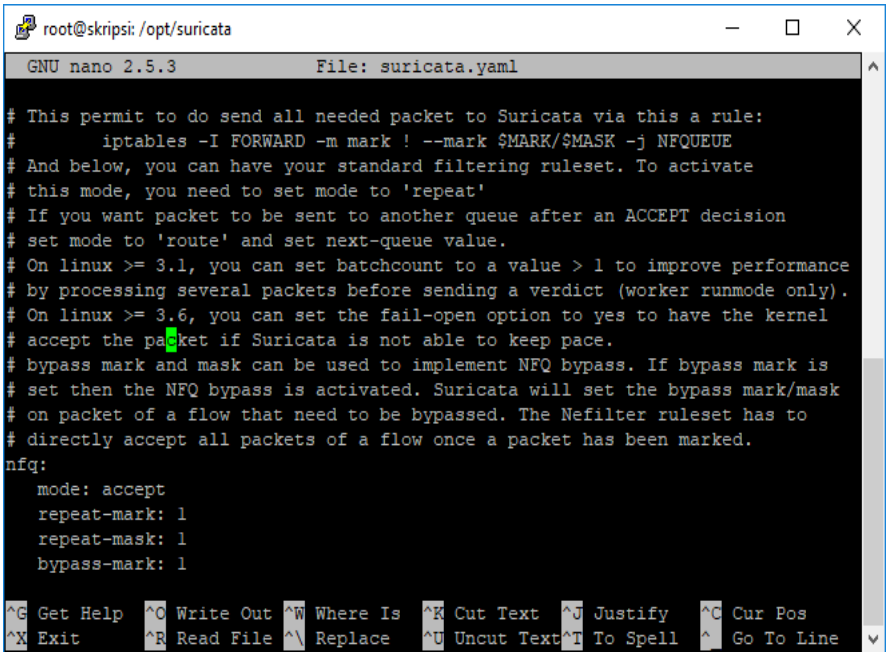
SERANGAN	HASIL *LOG*	DAMPAK
Scanning Port	Dalam serangan *scanning port* didapatkan berupa log yaitu : 02/02/2019-06:38:55.100162 [Drop] [**] [1:2018489:3] ET *SCAN NMAP OS Detection Probe [**] [Classification: Attempted Information Leak]*	Dampak dari serangan tersebut. Penyerang bisa melihat *port service* berapa yang dibuka oleh *server* dan penyerang juga bisa fokus serangan ke *port* tertentu.

SERANGAN	HASIL LOG	DAMPAK
	[Priority: 2] {UDP} 101.128.76.135	
Brute Force	Dalam serangan *Brute force* didapatkan berupa *log* yaitu : *ET SCAN SSH BruteForce Tool with fake PUTTY version [**] [Classification: Detection of a Network Scan] [Priority*: 3] {TCP} 218.92.1.158:58138 -> 159.65.12.147:22	Serangan *brute force* bisa menyebabkan *usage memory* menjadi tinggi dan juga bila *password* kita masuk kedalam *word list* penyerang, maka penyerang akan mendapatkan *password* untuk ke *login ssh*.
Denial Of Service	Dalam serangan *Denial Of Service* didapatkan berupa *log* yaitu : *SERIALIZINGME SCAN LibSSH Based SSH Connections Not Allowed [**] [Classification: Attempted Administrator Privilege Gain] [Priority*: 1] {TCP} 104.131.226.221:49559 -> 159.65.12.147:22.	Serangan ini juga bisa menyebabkan *server* menjadi *down* dan tidak bisa berjalan dengan baik.

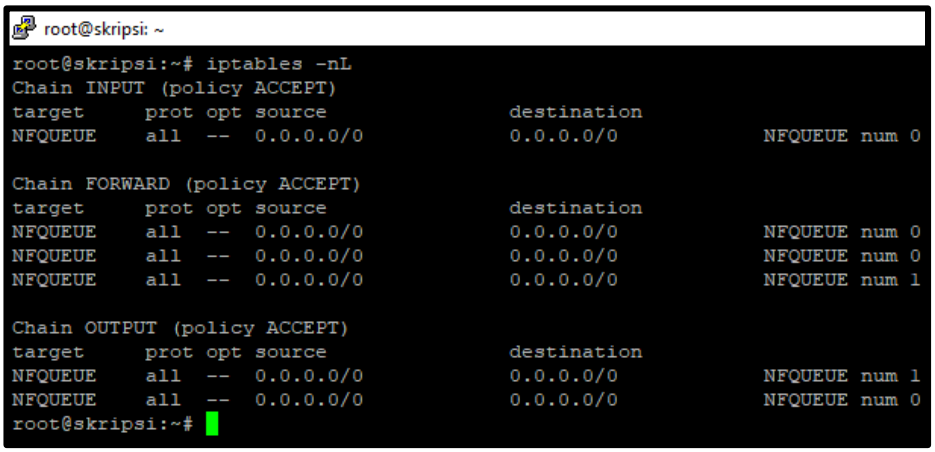
Backdoor	Dalam serangan *Backdoor* didapatkan berupa *log* yaitu : 01/17/2019-20:00:53.087437 [**] [1:2019876:4] *ET SCAN SSH BruteForce Tool with fake PUTTY version [**] [Classification: Detection of a Network Scan] [Priority*: 3] {TCP} 218.92.1.158:13230 -> 159.65.12.147:22	*Backdoor* ini bisa menyebabkan akses *user* biasa menjadi akses *root* tanpa diketahui kalau tidak adanya *log* dari suricata.

5.1.5.2. Hasil Analisa Keamanan Suricata

Penulis menambahkan konfigurasi suricata untuk mengoptimalkan keamanan bila ada serangan tidak hanya di deteksi oleh suricata namun di DROP suricata bila ada kegiatan mencurigakanoleh suricata. Penulis mengaktifkan mode Netfilter yang berada pada suricata.yaml



Gambar 5.20 : Konfigurasi *Netfilter*

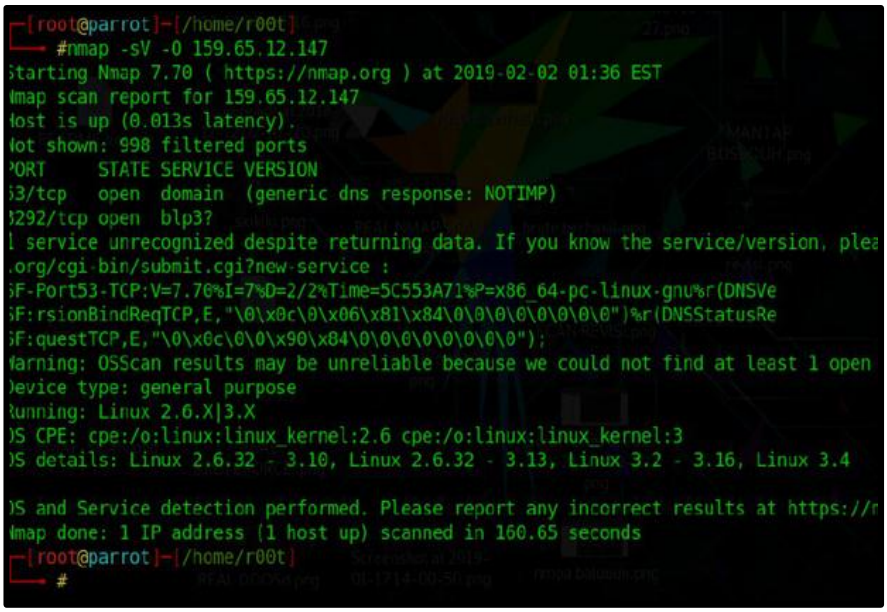


Gambar 5.21 : Konfigurasi *Iptables*

Pada gambar 5.21 Penulis menambhakan konfigurasi di *iptables* yang akan di sambungkan ke *netfilter* yang ada di konfigurasi suricata yang telah di aktifkan oleh penulis sebelumnya.

Gambar 5.22 : Suricata dijalankan

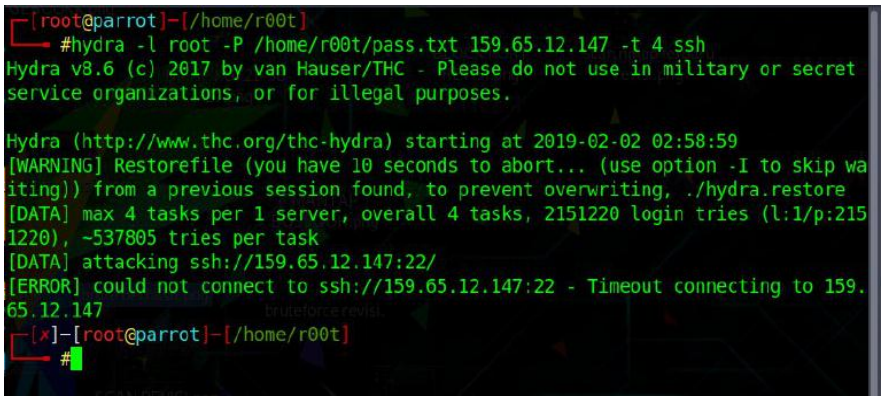
Pada gambar 5.22 penulis menjalankan suricata dengan perintah suricata –c suricata.yaml –q 0 –q 1 –D, yaitu –q pada suricata berfungsi untuk menjalankan *mode netfilter* pada suricata.



Gambar 5.23 : Serangan *Scanning Port* ulang

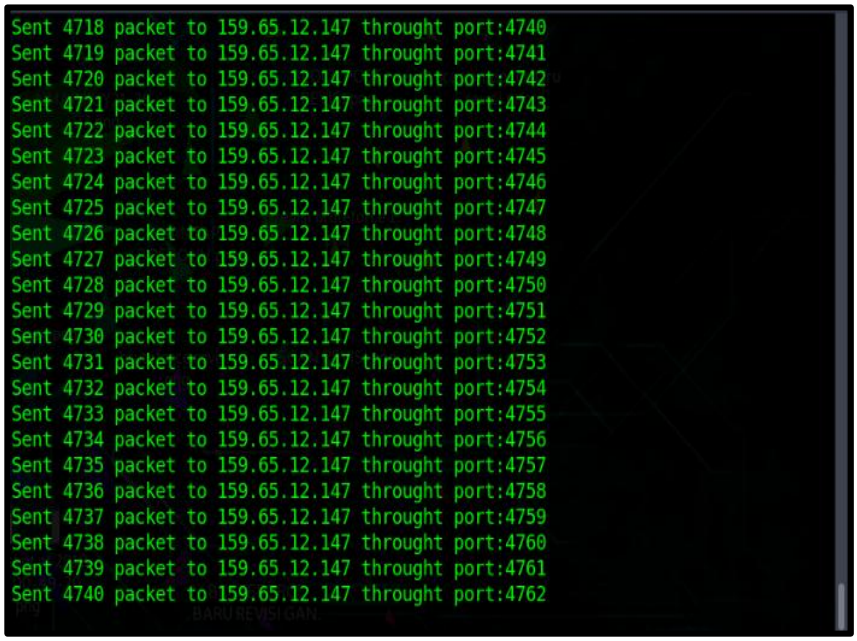
Pada gambar 5.23. penulis melakukan pengujian *scanning port ke Virtual Private Server* (VPS) yang telah dipasang suricata

dengan *mode netfilter*, penulis mendapatkan hasil yang berbeda ketika *scanning port* awal dengan konfigurasi sebelumnya. Ketika *netfilter* yang ada pada suricata sudah dikonfigurasi penulis mendapatkan hasil yaitu hanya terdapat 2 port yang terbuka pada *Virtual Private Server* (VPS) tersebut.



Gambar 5.24 : Serangan *Brute Force* ulang

Pada gambar 5.24. penulis melakukan serangan bruteforce dan mendapatkan hasil *Could not connect to ssh://159.65.12.147:22 – Timeout connecting to 159.65.12.147*, *netfilter* yang telah dikonfigurasi oleh penulis berhasil melakukan *drop* serangan *Brute Force*.

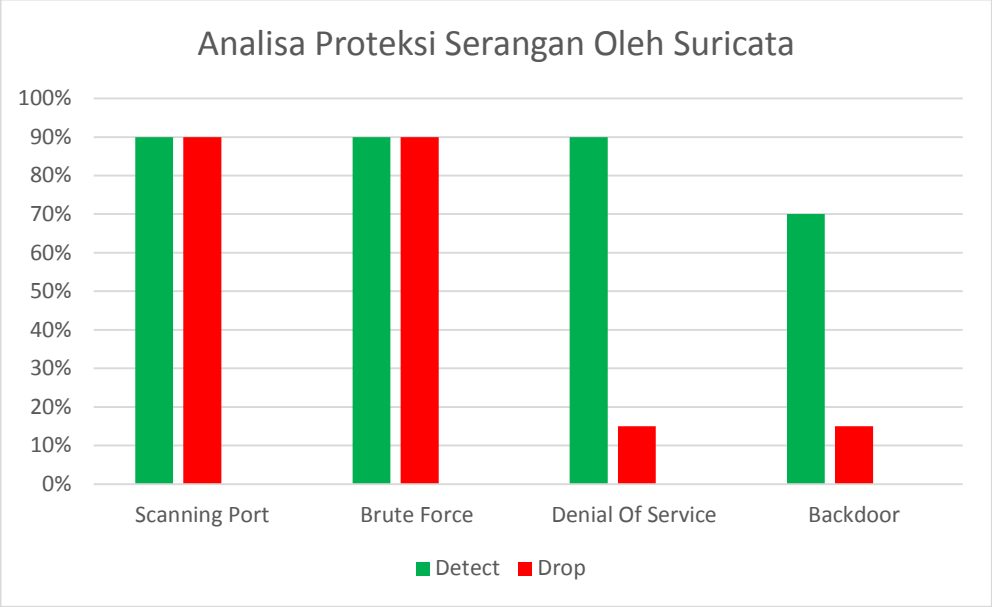


Gambar 5.25. : Serangan *Denial Of Service* ulang

Pada gambar 5.25. penulis melakukan pengujian ulang *Denial Of Service*. Penulis mendapatkan hasil *Denial Of Service* tetap jalan, dan tidak *drop* oleh *netfilter* yang ada pada suricata.

Gambar 5.26 : Serangan *Backdoor* ulang

Pada gambar 5.27. penulis melakukan Serangan *backdoor* ulang. Penulis juga mendapatkan hasil yang sama seperti serangan sebelumnya tidak dapat di *drop* oleh *netfilter* yang ada pada suricata.



Gambar 5.27 : Analisa Proteksi Suricata

Pada gambar gambar 5.27 penulis membuat grafik batang yaitu persentase proteksi serangan 90 % *Scanning Port* mampu di deteksi dan di drop oleh suricata, Persentase proteksi serangan 90 % dapat deteksi dan di *drop* oleh Suricata, Persentase proteksi serangan dari *Denial Of Service* yaitu 90 % dapat dideteksi oleh suricata dan 15 % dapat di *drop* oleh suricata, Dan serangan *backdoor* persentase proteksi serangan 70 % dapat dideteksi oleh suricata dan 15 % dapat di *drop* oleh suricata.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil pembahasan dari bab sebelumnya terhadap implementasi suricata untuk meningkatkan keamanan pada *cloud computing* maka dapat di ambil kesimpulan sebagai berikut :

1. Penulis menambahkan notifikasi telegram untuk mengetahui apabila ada aktifitas mencurigakan masuk ke log suricata di /var/log/suricata/fast.log.
2. Penulis menambahkan netfilter di suricata sehingga suricata tidak hanya mendeteksi tetapi mematikan atau drop bila ada aktifitas mencurigakan pada *Cloud Computing*.

6.2. Saran

Adapun saran yang dapat di kembangkan untuk penelitian selanjutnya :

1. Suricata dapat dikombinasikan dengan software intrusi yang lainnya seperti snorby dan barnyard
2. Suricata juga dapat mengidentifikasi file, MD5 Checksum, dan file ekstrasi.
3. Suricata tidak memiliki shared object rules seperti software intrusi lainnya.