BAB II
DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian berjudul "*Implementasi Metode IPS (Intrusion Prevention System) dan IDS (Intrusion Detection System) untuk Meningkatkan Keamanan Jaringan*" pada tahun 2020 yang dilakukan oleh Rifky Kurniawan dan Fajar Prakoso secara garis besar membahas mengenai pendeteksian dan pencegahan serangan DoS. *Tools* yang digunakan yaitu *Snort* dikombinasikan dengan *Basic Analysis and Security Engine* (BASE) yang berguna untuk melihat dan membaca hasil log dari setiap paket data yang masuk. Penelitian ini diimplementasikan pada sistem operasi Ubuntu *server* sebagai *server* IDS. Hasil dari penelitian ini yaitu serangan yang diujikan dapat terdeteksi dan tercatat lognya pada BASE serta serangan dapat di*drop* atau di*block* sehingga tidak dapat mengakses *server* kembali[1].

Penelitian berjudul "*Deteksi Penyusupan Pada Server Menggunakan Metode Intrusion Detection System (IDS) Berbasis Snort*" yang dilakukan oleh Benny Wijaya dan Arie Pratama pada tahun 2020 membahas mengenai permasalahan keamanan yang masih rendah dan rentan terkena serangan DoS. Penelitian ini menggunakan *Snort* sebagai *tools* IDS yang dapat mendeteksi apabila terjadi serangan, serangan yang diuji coba yaitu *Ping of Death*, serangan SSH dan percobaan telnet menggunakan PUTTY. Hasil dari penelitian ini yaitu semua serangan yang diujikan dapat terdeteksi oleh *Snort* dan tersimpan pada *log* sehingga dapat mengetahui waktu serangan dan IP pengirim[4].

Penelitian berjudul "*Analisa Penerapan Intrusion Prevention System (IPS) Berbasis Snort Sebagai Pengaman Server Internet Yang Terintegrasi Dengan Telegram*" yang diteliti oleh Abdul Muhaimi, I Putu Hariyadi, dan Akbar Juliansyah pada tahun 2019 secara garis besar membahas menganai IPS *Snort* yang diintegrasikan dengan aplikasi Telegram untuk mengirimkan peringatan. Skenario penyerangan meliputi *Ftp attack, Telnet Attack, Bruteforce attack, Remote File Incusion attack, dan Http Bruteforce attack*. Hasil dari penelitian ini yaitu notifikasi *alert* berhasil dikirimkan ke telegram dan *snort* mampu mendeteksi serta memblokir skenario serangan yang diujikan secara otomatis[8].

Penelitian berjudul "*Implementasi Intrusion Prevention System (IPS) Menggunakan Snort dan IPTables Pada Monitoring Jaringan Lokal Berbasis Website*" yang diteliti oleh Rudy Suwanto, Ikhwan Ruslianto, dan Muhammad Diponegoro pada tahun 2019 secara garis besar membahas mengenai monitoring *website* dengan mengimplementasikan IPS untuk mencegah terjadinya serangan yang masuk. Penelitian ini menggunakan *tools Snort* yang dikombinasikan dengan IPTables untuk aksi *accept*, *reject*, dan *drop*. Hasil dari penelitian ini yaitu *Snort* mampu mendeteksi adanya serangan yang masuk dan dapat mencegah dengan fitur yang disediakan oleh IPTables. Keberhasilan sistem dalam mendeteksi serangan yaitu 90% untuk serangan *Ping of Death* dan 85% untuk serangan *Port Scanning*[9].

Penelitian berjudul "*Implementasi Intrusion Detection System (IDS) Menggunakan Snort Untuk Mendeteksi Serangan Pada Server*" yang diteliti oleh Ali Farhan, L.A Syamsul Irfan Akbar, dan A. Sjamsjiar Rachman pada tahun 2021 membahas mengenai impelementasi *Snort* sebagai IDS dalam mendeteksi serangan yang masuk. Serangan yang diuji yaitu *Denial Distributed of Service* dan DNS *Spoof* menggunakan *tools* LOIC dan *Ettercap*. Hasil dari penelitian ini yaitu serangan DDos dengan protokol TCP dan DNS *Spoof* mampu dideteksi oleh IDS Snort, pendeteksian serangan berdasarkan jumlah packet yang diterima server dalam satuan waktu tertentu[5].

Penelitian yang dilakukan tidak lepas dari hasil penelitian yang telah disebutkan pada Tinjauan Pustaka, kemudian disajikan pada tabel 2.1 dibawah ini.

Tabel 2. 1 Tinjauan Pustaka Penelitian Terdahulu

No	Jurnal	Tahun	Metode	Hasil
1	Implementasi Metode IPS (Intrusion Prevention System) dan IDS (Intrusion Detection System) untuk Meningkatkan Keamanan Jaringan, oleh Rifky Kurniawan, Fajar	2020	Intrusion Detection System dan Intrusion Prevention System	Penelitian ini bertujuan pengimplementasian IDS dan IPS untuk keamanan jaringan. Tools yang digunakan sebagai IDS dan IPS yaitu Snort yang dikombinasikan dengan BASE yang berguna untuk melihat dan

No	Jurnal	Tahun	Metode	Hasil
	Prakoso[1].			membaca hasil log dari setiap paket data yang masuk agar lebih mudah dianalisis.
2	Deteksi Penyusupan Pada Server Menggunakan Metode Intrusion Detection System (IDS) Berbasis Snort, oleh Benny Wijaya dan Arie Pratama[4].	2020	Intrusion Detection System	Penelitian ini membahas permasalahan keamanan yang rentan terkena serangan DoS, agar meminimalisir terjadinya serangan yang terjadi maka menggunakan IDS. Tools yang digunakan untuk mendeteksi serangan adalah Snort, serangan yang diujikan meliputi Ping of Death, SSH, dan Telnet menggunakan PUTTY.
3	Analisa Penerapan Intrusion Prevention System (IPS) Berbasis Snort Sebagai Pemangan Server Internet Yang Terintegrasi Dengan Telegram, oleh Abdul Muhaimi, I Putu Hariyadi, dan Akbar Juliansyah[8].	2019	Intrusion Prevention System	Penelitian ini bertujuan untuk mendeteksi dan mencegah serangan yang diujikan menggunakan tools Snort dan aplikasi Telegram yang digunakan untuk menerima alert apabila terjadi serangan yang menuju server. Skenario penyerangan meliputi Ftp attack, Telnet Attack, Bruteforce attack, Remote File Incusion attack, dan Http

No	Jurnal	Tahun	Metode	Hasil
				Bruteforce attack.
4	Implementasi Intrusion Prevention System (IPS) Menggunakan Snort dan IPTables Pada Monitoring Jaringan Lokal Berbasis Website, oleh Rudy Suwanto, Ikhwan Ruslianto dan Muhammad Diponegoro[9].	2019	Intrusion Prevention System	Penelitian ini bertujuan untuk memonitoring keamanan website dengan mengimplementasikan IPS Snort dan IPTables untuk accept, reject, atau drop paket yang datang. Serangan yang diujikan yaitu Ping of Death dan Port Scanning dengan tingkat akurasi diatas 80% dalam pendeteksian serangan.
5	Implementasi Intrusion Detection System (IDS) Menggunakan Snort Untuk Mendeteksi Serangan Pada Server, oleh Ali Farhan, L.A Syamsul Irfan Akbar, dan A. Sjamsjiar Rachman[5].	2021	Intrusion Detection System	Penelitian ini bertujuan untuk mengimplementasikan IDS yang berguna untuk mendeteksi adanya serangan yang menuju server. Serangan yang masuk akan terdeteksi oleh Snort sesuai dengan rules yang telah dibuat oleh peneliti, serangan yang diujikan meliputi serangan DDoS dan DNS Spoof menggunakan LOIC dan Ettercap.

2.2 DASAR TEORI

2.2.1 Keamanan Jaringan

Keamanan jaringan berperan untuk mencegah dan mengidentifikasi penyerang yang mencoba mengakses sistem jaringan komputer. Keamanan jaringan diperlukan untuk mengantisipasi ancaman yang mengganggu. Dalam melindungi keamanan jaringan, perlu diterapkan hukum dasar yang meliputi aspek *Confidentiality*, *Integrity*, dan *Availability* (CIA).

Gambar 2. 1 Aspek Keamanan Jaringan [10]

Gambar 2.1 menunjukan aspek keamanan jaringan Semua aspek yang terdapat pada CIA akan menjadi komponen yang diperlukan dalam keamanan jaringan atau informasi. Berikut penjelasan dari aspek CIA:

1. *Confidentiality* (kerahasiaan)

Aspek yang menjamin kerahasiaan data atau informasi, data atau informasi yang dikirim, diterima, atau disimpan hanya dapat diakses oleh pihak yang berwenang.

2. *Integrity* (integritas)

Aspek yang menjamin keakuratan dan keutuhan data atau informasi, data atau informasi tidak dapat diubah apabila tidak mendapat izin dari pihak yang berwenang.

3. *Availability* (ketersediaan)

Aspek yang menjamin ketersediaan data atau informasi, data atau informasi akan selalu tersedia ketika dibutuhkan oleh pihak yang berwenang[10].

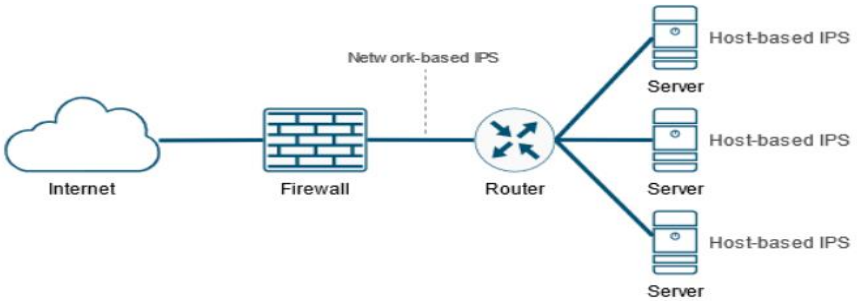
2.2.2 *Intrusion Detection System* (IDS)

Intrusion Detection System (IDS) adalah metode yang berguna untuk

mendeteksi aktivitas mencurigakan dalam sebuah jaringan. Apabila terdapat aktivitas yang mencurigakan maka IDS akan memberikan peringatan,. *Intrusion Detection System* (IDS) merupakan sebuah *software* atau *hardware* yang dapat mendeteksi aktivitas mencurigakan yang dianggap tidak normal dalam sebuah jaringan. IDS dapat melakukan pemeriksaan terhadap paket yang masuk atau keluar, menganalisis aman atau tidaknya paket tersebut. Apabila terdapat paket yang mencurigakan, maka IDS akan menganggap paket tersebut sebagai intrusi atau serangan. Kemampuan dari IDS yaitu memberikan peringatan atau alert kepada administrator selaku penanggung jawab suatu sistem apabila terdapat aktivitas yang mencurigakan, IDS hanya bertugas mendeteksi dan mengirimkan peringatan karena IDS bersifat pasif[11].

2.2.3 *Intrusion Prevention System* (IPS)

Intrusion Prevention System (IPS) adalah metode yang berguna untuk memantau trafik jaringan, mendeteksi aktivitas yang mencurigakan, serta melakukan tindakan yang lebih lanjut atau pencegahan dini terhadap aktivitas yang dianggap mencurigakan sehingga membuat sistem tidak berjalan secara semestinya. IPS bertindak seperti *firewall* yang berguna untuk aksi *accept*, *block*, dan *drop* paket yang datang. Peran utama IPS yaitu untuk mencegah atau menghentikan serangan yang sedang berlangsung[1].



Gambar 2. 2 Topologi NIPS dan HIPS [10]

Berdasarkan Gambar 2.2 IPS terbagi menjadi 2 berdasarkan dengan jenisnya yaitu:

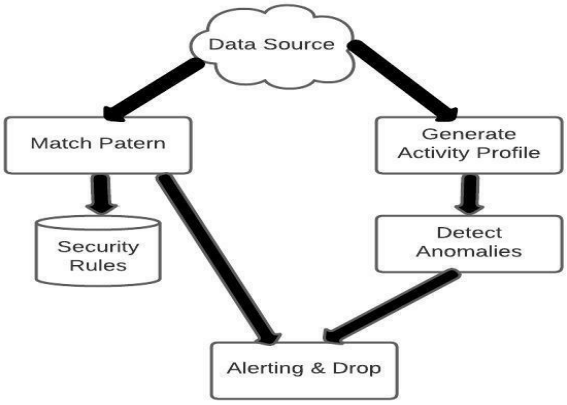
1. *Network Intrusion Prevention System* (NIPS)

Network Intrusion Prevention System (NIPS) bertugas untuk memantau seluruh aktivitas seperti protokol, lalu lintas data yang *in* dan *out* dalam jaringan, dan memeriksa aktivitas yang dianggap mencurigakan. NIPS hampir sama dengan *Network Intrusion Detection System* (NIDS) yang membedakan hanya

tindakannya, NIDS bersifat pasif sehingga hanya mampu memberikan *alert* saja tetapi tidak melakukan tindakan lanjut.

2. *Host Intrusion Prevention System* (HIPS)

Host Intrusion Prevention System (HIPS) bertugas untuk memeriksa seluruh aktivitas dan lalu lintas yang berada di *host*, HIPS mencegah aktivitas mencurigakan yang memasuki *host*. HIPS hampir sama dengan *Network Intrusion Prevention System* (HIDS) yang membedakan hanya tindakannya, HIDS bersifat pasif sehingga hanya mampu memberikan *alert* saja tetapi tidak melakukan tindakan lanjut seperti *block* atau *drop*[10].



Gambar 2. 3 Metode IDS dan IPS

Berdasarkan Gambar 2.3 terdapat 2 metode untuk memeriksa layak atau tidaknya sebuah paket masuk ke dalam jaringan, ada dua metode IDS dan IPS yang bisa digunakan yaitu:

1. *Signature Based*

Metode ini bekerja dengan cara mencocokan pola serangan dengan *signature* atau *rule* yang tersimpan di *database*. Metode ini menjaga sistem dari serangan yang sudah dikenali sebelumnya, untuk menjaga keamanan jaringan maka *signature* atau *rule* di *database* harus rutin ter*update*[12].

2. *Anomaly Based*

Metode ini mengharuskan untuk melakukan konfirgurasi terlebih dahulu pada IDS dan IPS yang sudah tersedia sehingga dapat mengetahui bagai mana pola yang akan ada pada sebuah sistem jaringan komputer. Apabila IDS dan IPS mendapatkan secara *anomaly* pada paket yang diterima atau dikirimkan, maka otomatis IDS dan IPS akan mengenali sebagai serangan dan memberikan peringatan serta menolak paket tersebut[3].

2.2.4 *Intrusion Detection and Prevention System* (IDPS)

Intrusion Detection and Prevention System (IDPS) merupakan kombinasi dari *metode Intrusion Detection System* (IDS) dan *Intrusion Prevention System* (IPS). Metode ini dapat mendeteksi intrusi, mengidentifikasi intrusi, memberikan respon lanjutan terhadap intrusi, dan pencegahan intrusi. Cara IDPS mendeteksi intrusi yaitu dengan memantau lalu lintas jaringan kemudian mencocokan dengan pola yang telah tersimpan, setelah mendeteksi lalu lintas yang mencurigakan, IDPS akan mengintifikasi intrusi jenis serangan yang masuk, serangan yang masuk dapat sama dengan yang sebelumnya atau merupakan serangan baru. IDPS kemudian akan memberikan tindakan yang lebih lanjut terhadap serangan yang terjadi, tindakan ini dapat berupa pemberitahuan dan pemblokiran serangan yang masuk.

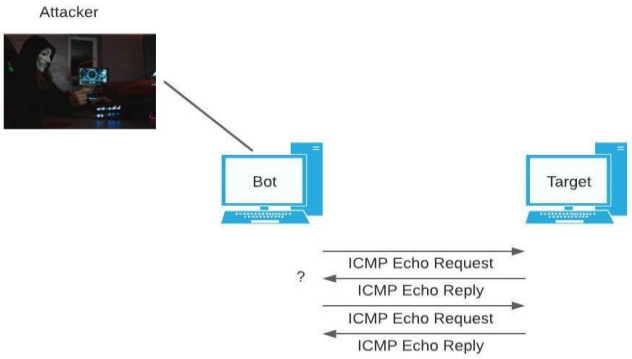
IDPS berperan penting dalam menjaga keamanan jaringan, karena dapat mendeteksi, menghentikan, bahkan mencegah serangan yang masuk sebelum serangan tersebut menyebabkan kerusakan pada server atau target.

2.2.5 *Denial of Service* (DoS)

Denial of Service (DoS) merupakan serangan keamanan pada jaringan yang bertujuan untuk mengikat sumber daya dari *server*, dan menghalangi pengguna untuk mengakses layanan yang tersedia. Cara kerja serangan ini yaitu mengirimkan banyak paket permintaan ke jaringan secara bersamaan, hal ini menyebabkan sumber daya sistem terbebani dan tidak dapat memberikan performansi yang bagus atau bahkan mengalami kegagalan total[6].

2.2.6 *Internet Control Message Protocol Flood (ICMP Flood)*

ICMP *flood* merupakan serangan yang bekerja dengan membanjiri target menggunakan paket permintaan tinggi, sehingga jaringan diharuskan merespons dengan jumlah paket balasan yang sama. Hal ini yang menyebabkan target menjadi tidak dapat diakses oleh lalu lintas normal[13]. Serangan ICMP Flood menghasilkan lalu lintas jaringan yang tinggi dalam waktu singkat, serangan ini menghabiskan *bandwidth* yang tersedia untuk lalu lintas yang sah dan membuat server menjadi tidak responsif. Langkah penyerangan ICMP flood yaitu dimulai dengan attacker Penyerang mengirimkan banyak paket ICMP echo request ke server target, *server* yang ditargetkan mengirimkan paket balasan ICMP echo reply ke setiap alamat IP perangkat yang meminta sebagai tanggapan

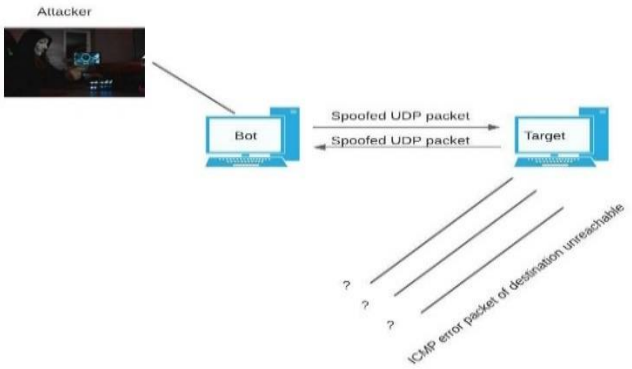


Gambar 2. 4 ICMP Flood

Gambar 2.4 Karena paket yang dikirimkan oleh penyerang berjumlah banyak, maka menyebabkan server tidak bisa membalas satu persatu paket tersebut dan server mengalami *down*.

2.2.7 *User Data Protocol Flood (UDP Flood)*

UDP Flood merupakan serangan yang bekerja membanjiri port dengan paket UDP palsu yang bersifat acak dan tidak valid dalam jumlah besar sehingga menyebabkan target menjadi tidak dapat menanggapi permintaan yang sah atau tak mampu lagi mengelola tingginya *load* port yang masuk[14].

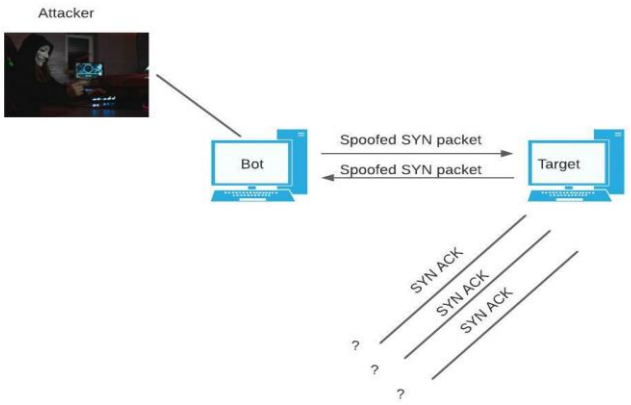


Gambar 2. 5 UDP Flood

Gambar 2.5 menunjukan serangan UDP *Flood*, cara kerja UDP flood yaitu diawali dengan memeriksa untuk melihat apakah ada program yang sedang berjalan yang sedang mendengarkan permintaan di port yang ditentukan, Jika tidak ada program yang menerima paket di *port* tersebut, *server* merespons dengan paket ICMP (ping) untuk memberi tahu pengirim bahwa tujuan tidak dapat dijangkau.

2.2.8 *SYN Flood*

SYN Flood merupakan serangan yang memanfaatkan koneksi 3-Way-Handshake yaitu ketika *attacker* hanya mengirimkan paket SYN berulang-ulang tanpa mengirimkan paket ACK sebagai konfirmasinya.



Gambar 2. 6 SYN Flood

Gambar 2.6 menunjukan serangan SYN *Flood*, cara kerja SYN Flood yaitu Penyerang mengirimkan volume tinggi paket SYN ke server target menggunakan alamat IP palsu. *Server* merespons setiap permintaan koneksi dengan mengirimkan paket SYN ACK dan membiarkan port terbuka menerima respons. Sementara *server* menunggu paket ACK terakhir, yang tidak pernah sampai, penyerang terus mengirim lebih banyak paket SYN. Kedatangan paket SYN baru menyebabkan server sementara mempertahankan koneksi *port* yang terbuka untuk jangka waktu tertentu, dan setelah semua port yang tersedia telah digunakan, *server* tidak dapat berfungsi secara normal[15].

2.2.9 *Snort*

Snort merupakan salah satu *tools open source* keamanan yang dapat berfungsi untuk mendeteksi dan mencegah intrusi jaringan yang bersifat *rule-driven*. Snort digunakan untuk memantau lalu lintas jaringan secara pasif dan dapat memberikan *alert* ketika intrusi muncul[2]. Keunggulan snort yaitu tanggap dalam mendeteksi intrusi yang masuk, konfigurasinya mudah, *support* ke dalam berbagai sistem operasi dan bersifat *open source* atau gratis. Pada *snort* terdapat *plugin* dan aturan yang dapat di *download* dari komunitas pengguna *snort*, sehingga pengguna dapat memodifikasi kemampuan snort sesuai dengan kebutuhan. *Snort* bekerja menggunakan rules yang dapat dimodifikasi oleh pengguna[16].

Secara prinsip snort dapat dioperasikan dalam tiga mode, yaitu:

1. *Packet Sniffer*

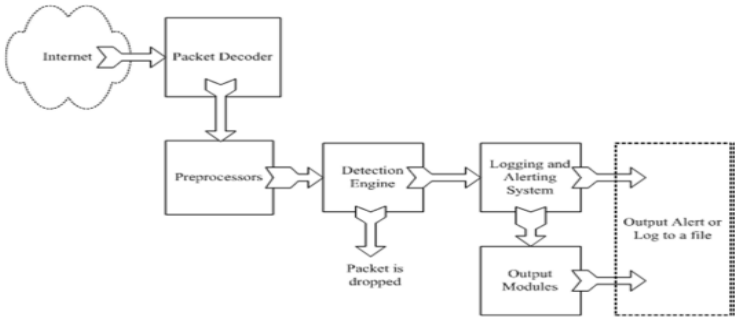
Berguna untuk melihat atau mengamati paket yang lewat di jaringan.

2. *Packet Logger*

Berguna untuk mencatat semua paket yang lewat di jaringan.

3. *Network Intrusion Detection System* (NIDS)

Berguna untuk mendeteksi serangan yang masuk pada jaringan[11].



Gambar 2. 7 Komponen pada Snort

Snort terdiri dari beberapa komponen, komponen ini saling terhubung satu sama lain untuk mendeteksi intrusi dan menghasilkan keluaran dengan format yang dibutuhkan dari deteksi sistem. Gambar 2.7 menunjukan bagaimana komponen snort tersusun, setiap paket data baru yang datang akan masuk melewati *packet decoder*. Dalam perjalanan menuju *output modules*, sebagian paket dibuang dan sisanya menjadi log atau menghasilkan sebuah *alert*[17].

Berikut penjelasan komponen pada *Snort*:

1. *Packet Decoder*

Packet decoder menerima paket dari berbagai jenis *interface* jaringan, bekerja pada *layer data link* dan memilah *interface* tersebut berdasarkan *input*annya melalui *ethernet* atau *wireless* yang kemudian akan dilanjutkan ke tahap *preprocessor*.

2. *Preprocessor*

Komponen yang berguna untuk memodifikasi atau mengatur paket data sebelum masuk ke tahap selanjutnya, yaitu tahap *Detection Engine*.

3. *Detection Engine*

Merupakan komponen inti dari *snort*, paket yang masuk dari tahap sebelumnya akan dibandingkan dengan *rules* yang telah tersimpan, apabila terdapat kecocokan maka akan dianggap sebagai serangan atau intrusi.

4. *Logging and Alerting System*

Komponen ini bertugas menghasilkan paket apa yang ditemukan pada tahap *Detection Engine*, paket akan tercatat pada log aktifitas dan untuk menghasilkan *alert* atau peringatan.

5. *Output Modules*

Tahap ini berguna untuk menyimpan output yang telah diproses oleh sistem *snort*. *Output* yang dihasilkan bermacam variasi seperti *tcpdump, binary format, texs, syslog, database* dan sebagainya[15].

2.2.10 *Wireshark*

Wireshark merupakan *tools network analyser* yang populer digunakan oleh *network administrator* untuk menganalisis dan melacak kinerja jaringan termasuk protokol didalamnya. *Wireshark* dapat menangkap paket yang melewati jaringan dalam beragam *format protocol*. *Tools* ini juga dapat digunakan untuk *sniffing* dengan menangkap paket yang melewati jaringan kemudian di analisa. Keunggulan *wireshark* yaitu gampang digunakan, dan bersifat lintas *platform* dimana pengguna *Macintosh* dan *Linux* dapat menggunakannya. *Wireshark* dapat menganalisis ratusan protokol dan menampilkannya dalam mode *Grapik* (GUI), dapat membaca data secara langsung yang masuk melalui *Ethernet, Bluetooth,* dan *USB*[18].

2.2.11 Hping

Hping merupakan *tools* yang berguna untuk pengujian *firewall*, pengujian jaringan, dan audit keamanan. Hping dapat digunakan untuk membuat paket IP yang berisi TCP, UDP atau ICMP payloads. Semua *fieldheader* dapat dimodifikasi dan dikontrol dengan menggunakan baris perintah (*command line*). Hping dapat dipasang pada berbagai sistem operasi seperti NetBSD, MacOS, Windows, FreeBSD, Linux, Solaris dan OpenBSD[19].

2.2.12 Quality of Service (QoS)

Quality of Service (QoS) merupakan metode yang digunakan untuk mengukur performa atau kinerja jaringan. QoS suatu jaringan merupakan tingkat kecepatan dan kehandalan penyampaian berbagai data di dalam suatu komunikasi. Standar metrik formal dalam mengukur QoS pada jaringan meliputi *Troughput*, *Jitter*, *Latency/ Delay*, dan *Packet Los*[7].

1. *Troughput*

Troughput merupakan kemampuan sebenarnya suatu jaringan dalam melakukan transfer data. *Troughput* bersifat dinamis bergantung trafik yang sedang terjadi. Berikut perhitungan nilai *Troughput* dapat menggunakan persamaan (2.2) dan standarisasi nilai *Troughput* menurut TIPHON ditampilkan pada tabel 2.2

$$Troughput = \frac{jumlah\ data\ yang\ dikirim}{waktu\ pengiriman\ data} \qquad (2.2)$$

Tabel 2. 2 Standarisasi nilai Troughput menurut TIPHON

Kategori *Troughput*	*Troughput*	Indeks
Bad	0 – 338 kbps	0
Poor	338 – 700 kbps	1
Fair	700 – 1200 kbps	2
Good	1200 kbps – 2,1 Mbps	3
Excelent	> 2,1 Mbps	4

2. *Jitter*

Jitter merupakan perubahan variasi *delay* pada suatu periode, *jitter* juga disebut gangguan komunikasi digital atau analog yang disebabkan oleh perubahan sinyal karena referensi posisi waktu. Beberapa penyebab *jitter* yaitu: Panjangnya antrian dalam waktu pengolahan data, meningkatnya trafik secara tiba-tiba yang berakibat penyempitan *bandwidth,* kecepatan kirim dan terima paket dari setiap *titik*[20]. Berikut perhitungan nilai *Jitter* dapat menggunakan persamaan (2.3) dan standarisasi nilai *Jitter* menurut TIPHON ditampilkan pada tabel 2.3

$$Jitter = \frac{total\ variasi\ delay}{total\ paket\ yang\ diterima} \qquad (2.3)$$

Tabel 2. 3 Standarisasi nilai Jitter menurut TIPHON

Kategori *Jitter*	*Jitter*	Indeks
Poor	125 – 225 ms	1
Medium	75 – 125 ms	2
Good	0 – 75 ms	3
Perfect	0 ms	4

3. *Latency* (*Delay*)

Latency (*Delay*) merupakan total waktu tunda suatu paket yang disebabkan oleh

proses transmisi dari satu titik menuju titik yang lainnya. *Delay* dalam jaringan terdiri dari *delay processing*, *delay jitter buffer*, dan *delay network*. Berikut perhitungan nilai *Latency* dapat menggunakan persamaan (2.4) dan standarisasi nilai *Latency* menurut TIPHON ditampilkan pada tabel 2.4

$$Latency = \frac{total\ delay}{total\ paket\ diterima} \qquad (2.4)$$

Tabel 2. 4 Standarisasi nilai Latency menurut TIPHON

Kategori *Latency*	*Latency*	Indeks
Poor	> 450 ms	1
Medium	300 – 450 ms	2
Good	150 – 300 ms	3
Perfect	< 150 ms	4

4. *Packet Loss*

Packet Loss merupakan parameter yang menunjukan kondisi jumlah total paket yang hilang. Paket yang hilang dapat disebabkan oleh *collision* dan *congestion* pada jaringan. Beberapa penyebab terjadinya *packet loss* yaitu: *overload* trafik jaringan, tabrakan (*congestion*) pada jaringan, *error* pada media fisik, kegagalan disisi penerima yang disebabkan karena *overflow* pada *buffer*. Berikut perhitungan nilai *Packet loss* dapat menggunakan persamaan (2.5) dan standarisasi nilai *Packet loss* menurut TIPHON ditampilkan pada tabel 2.5[7]

$$Packet\ Los = \frac{paket\ dikirim - paket\ diterima}{paket\ diterima}\ x\ 100\% \qquad (2.5)$$

Tabel 2. 5 Standarisasi nilai Packet Loss menurut TIPHON

Kategori *Packet Loss*	*Packet Loss*	Indeks
Poor	> 25 %	1
Medium	15 – 24 %	2
Good	3 – 14 %	3
Perfect	0 – 2 %	4