BAB II
TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Kajian pustaka ini menjadi acuan penulis untuk melakukan penelitian yang dapat membantu penulis dalam mengevaluasi penelitian yang dilakukan. Penelitian yang penulis lakukan memiliki perbedaan dengan penelitian sebelumnya, yaitu dari segi metode dan juga parameter yang digunakan untuk menghitung hasil dari penelitian ini. Namun, penelitian penulis tidak terlepas dari topik penelitian keamanan jaringan, serangan siber, *Intrusion Detection System (IDS), Intrusion Prevention System (IPS)* OSSEC, Snort, dan *Quality of Service (*QoS*).* Berikut beberapa topik terkait dengan penelitian yang akan penulis lakukan :

2.1.1 Tati Ernawati, Fikri Faiz Fadhlur Rachmat pada tahun 2021 dengan judul "Keamanan Jaringan dengan Cowrie Honeypot dan Snort Inline-Mode sebagai Intrusion Prevention System" [15]

Penelitian ini menggunakan serangan *DDoS* dan *brute force attack.* Pengujian dilakukan dengan melakukan serangan kepada server menggunakan *hydra* untuk melakukan *brute force* dan *slowloris* untuk melakukan *DDoS.* Saat dilakukan serangan, Snort menghasilkan peringatan yang dapat mendeteksi serangan *port HTTP, Telnet,* dan *SSH* dengan menggunakan *port ICMP*. Hasil dari penelitian ini menunjukkan bahwa Snort lebih unggul dibandingkan dengan Honeypot berdasarkan uji parameter integritas. Penelitian ini lebih merekomendasikan *Snort* untuk digunakan dalam sistem keamanan jaringan dibandingkan dengan Honeypot.

2.1.2 Suliman, Andani Achmad, Adnan pada tahun 2021 dengan judul "Implementasi *Honeypot* dan *Port Knocking* dalam Mendeteksi Serangan *DDoS Attack* pada Server Jaringan [4]

Penelitian ini diuji dengan melakukan simulasi sebelum mengimplementasikan Honeypot dan *port knocking* saat terjadi serangan *DDoS* pada server ujian *online*. Dari simulasi ini diperoleh bahwa *attacker* berhasil membuat *traffic* pada jaringan dengan cara mengirimkan *request* pada server secara terus menerus sehingga membuat server *down* dan *attacker* juga berhasil menampilkan *username* dan *password* mikrotik serta *attacker* dapat menampilkan *password* dan *username* pada aplikasi web. Kemudian saat telah mengimplementasikan Honeypot dan *port knocking*, didapatkan hasil bahwa Honeypot dapat mendeteksi 100% adanya serangan *DDoS* yang masuk dari hasil uji 4 jenis serangan *DDoS*. Sedangkan untuk *port knocking* hanya mampu mendeteksi 2 serangan *DDoS* dan 2 serangan *DDoS* lainnya tidak dapat terdeteksi.

2.1.3 Eka Stephani Sinambela pada tahun 2020 dengan judul "Evaluasi Performansi Deteksi Serangan Pada HIDS OSSEC" [14]

Pada penelitian ini, dilakukan pengujian OSSEC untuk dapat mendeteksi beberapa serangan. Diperoleh hasil bahwa OSSEC dapat mendeteksi serangan secara *realtime* yang dilakukan sebanyak 15 kali percobaan dengan penggunaan rata-rata *CPU* rendah. Serangan yang dilakukan pada penelitian ini adalah *DNSEnum, port scanning, SSH brute force.* Berdasarkan hasil pengujian *response time,* konsumsi *CPU* dan memori OSSEC akan semakin tinggi selaras dengan peningkatan jumlah penyerangnya.

2.1.4 Saleh Dwiyatno, Ayu Purnama Sari, Agus Irawan, Safig pada tahun 2019 dengan judul "Pendeteksi Serangan *DDoS (Distributed Denial of Service)* Menggunakan Honeypot di PT.Torini Jaya Abadi" [3]

Penelitian ini untuk mendeteksi adanya serangan *DDoS*, dilakukan penambahan *software* Honeypot. Pada saat pengujian,

serangan *DDoS* dilakukan dengan menggunakan *tools* LOIC sehingga nantinya *tools* Honeyd akan dapat mendeteksi adanya serangan *DDoS* yang dilakukan oleh *attacker*. Penelitian ini menggunakan tiga *attacker* dengan percobaan yang dilakukan sebanyak 10 kali percobaan serangan. Dengan mendapatkan status penyerang oleh sistem Honeypot, admin akan mengetahui informasi penyerang yang terjadi pada server sehingga admin dapat menutupi celah kemungkinan target penyerangan kedepannya. Dengan sistem keamanan menggunakan Honeypot dapat membantu mendeteksi serangan *DDoS* secara *real time.*

2.1.5 Risa Eri Susanti, Arif Wirawan Muhammad, Wahyu Adi Prabowo pada tahun 2020 dengan judul "Implementasi *Intrusion Prevention System (IPS)* OSSEC dan Honeypot Cowrie" [8]

Penelitian ini menggunakan serangan *SSH brute force, Man In The Middle (MITM) attack,* dan *Distributed of Service (DDoS).* OSSEC akan mengetahui adanya serangan dengan mengubah alamat MAC pada *gateway* dan server. OSSEC ini dapat mendeteksi adanya *ICMP Flood , UDP Flood,* dan *TCP Flood.* Saat OSSEC diintegrasikan dengan Honeypot dapat mencegah serangan *SSH brute force.*

Tabel 2. 1 Penelitian Sebelumnya

No	Judul	Tahun	Penulis	Isi Penelitian	Perbedaan
1	Keamanan Jaringan dengan Cowrie Honeypot dan Snort Inline-Mode sebagai Intrusion Prevention System	2021	Tati Ernawati, Fikri Faiz Fadhlur Rachmat [15]	Penelitian ini menggunakan serangan *DDoS* dan *brute force attack*. Hasil dari penelitian ini menunjukkan bahwa Snort lebih unggul dibandingkan dengan Honeypot berdasarkan uji parameter integritas.	1. Peneliti melakukan perbandingan performansi *tools OSSEC, Honeypot,* dan Snort 2. Metode yang dilakukan dengan metode *Intrusion Detection System* dan *Prevention (IDPS)* 3. Parameter performansi yang digunakan adalah parameter *Quality of Service (*QoS*)*
2	Implementasi *Honeypot* dan *Port Knocking* dalam Mendeteksi Serangan *DDoS Attack* pada Server Jaringan	2021	Suliman, Andani Achmad, Adnan [4]	Penelitian ini diuji dengan melakukan simulasi sebelum mengimplementasikan Honeypot dan *port knocking* saat terjadi serangan *DDoS* pada server ujian *online*. Dari penelitian didapatkan hasil bahwa Honeypot dapat mendeteksi 100% adanya serangan *DDoS.*	1. Peneliti melakukan perbandingan *tools* OSSEC, Honeypot, dan Snort untuk mendeteksi adanya serangan 2. Metode yang dilakukan dengan metode *Intrusion Detection System* dan *Prevention (IDPS)* 3. Parameter performansi yang digunakan adalah parameter *Quality of Service (*QoS*)*
3	Evaluasi Performansi Deteksi Serangan Pada HIDS OSSEC	2020	Eka Stephani Sinambela [14]	Pada penelitian ini, dilakukan pengujian OSSEC untuk dapat mendeteksi beberapa serangan. Diperoleh hasil bahwa OSSEC dapat mendeteksi serangan secara	1. Peneliti melakukan perbandingan performansi *tools* Honeypot dan Snort untuk mendeteksi serangan yang masuk ke server

No	Judul	Tahun	Penulis	Isi Penelitian	Perbedaan
				realtime yang dilakukan sebanyak 15 kali percobaan dengan penggunaan rata-rata CPU rendah.	2. Metode yang dilakukan dengan metode *Intrusion Detection System* dan *Prevention (IDPS)* 3. Serangan yang dilakukan menggunakan *Distributed Denial of Service* 4. Parameter performansi yang digunakan adalah parameter *Quality of Service (*QoS*)*
4	Pendeteksi Serangan *DDoS (Distributed Denial of Service)* Menggunakan Honeypot di PT.Torini Jaya Abadi	2019	Saleh Dwiyatno, Ayu Purnama Sari, Agus Irawan, Safig [3]	Penelitian ini untuk mendeteksi adanya serangan *DDoS*, dilakukan penambahan *software* Honeypot. Pada saat pengujian, serangan *DDoS* dilakukan dengan menggunakan *tools LOIC* sehingga nantinya *tools* Honeyd akan dapat mendeteksi adanya serangan *DDoS* yang dilakukan oleh *attacker*.	1. Peneliti melakukan perbandingan performansi *tools* OSSEC dan Snort untuk mendeteksi serangan yang masuk ke server 2. Metode yang dilakukan dengan metode *Intrusion Detection System* dan *Prevention (IDPS)* 3. Parameter performansi yang digunakan adalah parameter *Quality of Service (*QoS*)*
5	Implementasi *Intrusion Prevention System (IPS)* OSSEC dan Honeypot Cowrie	2020	Risa Eri Susanti, Arif Wirawan Muhammad, Wahyu Adi Prabowo [8]	Penelitian ini menggunakan serangan *SSH brute force, Man In The Middle (MITM) attack,* dan *Distributed of Service (DDoS).* OSSEC akan mengetahui adanya	1. Peneliti menggunakan metode *Intrusion Detection System* dan *Prevention (IDPS)* 2. Peneliti menambahkan *tools*

No	Judul	Tahun	Penulis	Isi Penelitian	Perbedaan
				serangan dengan mengubah alamat MAC pada *gateway* dan server.	Snort untuk mendeteksi serangan yang masuk ke server 3. Parameter performansi yang digunakan adalah parameter *Quality of Service (*QoS*)*

2.2 Dasar Teori

Dalam penelitian ini, penulis membutuhkan ide yang berkaitan dengan topik penelitian yang dilakukan. Baik itu yang bersumber dari buku, jurnal, artikel maupun *website.* Nantinya pada dasar teori ini akan dijabarkan teori keamanan jaringan, metode keamanan jaringan, serangan siber, *tools* yang berkaitan dengan penelitian ini, dan parameter yang digunakan QoS. Berikut penjabaran dari teori tersebut :

2.2.1 Keamanan Jaringan

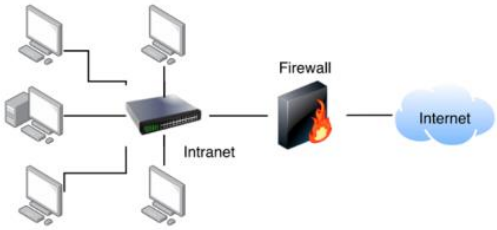
Keamanan jaringan dapat diartikan sebagai sebuah sistem yang dapat melakukan pencegahan aktivitas yang tidak diinginkan dan juga yang tidak memiliki akses ke suatu jaringan. Menghubungkan satu komputer dengan komputer lain menggunakan satu jaringan akan dapat membuat orang lain untuk mengakses data, mengubah isi, memindahkan data hingga menghapus data dalam jaringan tersebut [16]. Untuk itu dibutuhkan pengamanan jaringan agar dapat melindungi data dari segala bentuk ancaman *cyber.* Keamanan jaringan ini bekerja dengan memblokir semua ancaman yang ingin masuk ke jaringan pengguna.

Pada keamanan jaringan harus dapat membedakan antara ancaman, kerentanan, dan serangan. Ancaman merupakan suatu hal yang mengganggu pengoperasian jaringan yang diakibatkan oleh niat jahat atau tidak disengaja ataupun karena kegiatan alam. Kerentanan merupakan kelemahan yang terdapat pada desain, konfigurasi, implementasi ataupun manajemen jaringan yang dimanfaatkan terhadap ancaman. Serangan merupakan kegiatan yang memanfaatkan kerentanan sehingga dapat menimbulkan ancaman bagi pengguna jaringan [17].

2.2.2 *Firewall*

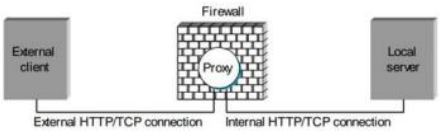
Firewall ini bekerja dengan memeriksa setiap paket yang akan masuk dan juga keluar dari sebuah jaringan. Setiap paket yang lewat

akan dilakukan pengecekan terhadap aturan yang telah dibuat pada *firewall* dan paket yang sesuai dengan aturanlah yang dapat melewati *firewall* ini [18]. Penggunaan *firewall* memiliki beberapa keuntungan, yaitu: dapat mencatat segala aktivitas pada jaringan, dapat menerapkan kebijaksanaan sekuritas, dan dapat digunakan untuk membatasi penggunaan sumberdaya informasi. Sedangkan kelemahannya adalah tidak dapat melindungi jaringan dari serangan yang tidak melewati jaringan ini atau serangan ini melewati pintu lain menuju jaringan tersebut, tidak dapat melindungi serangan yang baru atau belum dikenal oleh *firewall,* dan tidak dapat melindungi dari serangan virus [18].



Gambar 2. 1 Jaringan Komputer dengan *Firewall* [17]

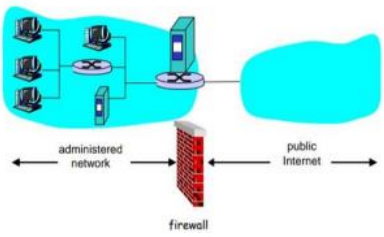
Dari gambar diatas dapat dilihat jika *firewall* diletakkan diantara internet dan intranet atau dikenal dengan jaringan internal. Jadi *firewall* ini nantinya akan membatasi lalu lintas di jaringan internal sehingga hanya dapat memperbolehkan akses ke *email* atau *web* tetapi tidak memperbolehkan jenis lalu lintas lainnya.



Gambar 2. 2 Contoh *proxy firewall* [18]

Dapat dilihat dari gambar diatas jika firewall terletak diantara jaringan internal dan eksternal. Dimana fungsinya dapat memfilter

apa saja yang masuk ke jaringan internal yang berasal dari jaringan eksternal.



Gambar 2. 3 Contoh Kerja *Firewall* [18]

Gambar diatas menunjukkan bahwa untuk masuk ke jaringan internal harus melewati *firewall* terlebih dahulu dan semua hal yang ada di jaringan internal dikelola oleh administrator jaringan. Jadi nantinya untuk masuk ke jaringan internal ini harus melewati *firewall* terlebih dahulu untuk memastikan sesuatu yang masuk ke jaringan internal tersebut aman.

2.2.3 *Intrusion Detection and Prevention System (IDPS)*

Intrusion Detection and Prevention (IDPS) ini merupakan gabungan antara *Intrusion Detection System (IDS)* dan *Intrusion Prevention System (IPS)* serta merupakan salah satu pendekatan untuk mengatasi masalah keamanan jaringan. *Intrusion Detection System (IDS)* adalah sistem yang dapat mengawasi lalu lintas jaringan mengawasi kegiatan mencurigakan di dalam jaringan. *Intrusion Detection System (IDS)* ini akan menganalisa suatu serangan dengan memberikan peringatan kepada administrator jaringan. *Intrusion Prevention System (IPS)* adalah aplikasi yang dapat memantau lalu lintas jaringan, mendeteksi aktivitas mencurigakan, dan melakukan pencegahan terhadap kejadian yang membuat jaringan terganggu. Secara umum, *Intrusion Prevention System (IPS)* ini terbagi kedalam

dua jenis, yaitu *Host-based Intrusion Prevention System (HIPS)* dan *Network-based Intrusion Prevention (NIPS)* [17].

Intrusion Prevention System (IPS) ini merupakan gabungan antara *firewall* dan *Intrusion Detection System (IDS).* Teknologi *Intrusion Detection and Prevention (IDPS)* ini digunakan untuk mencegah serangan yang akan masuk pada jaringan dengan memeriksa serta mencatat semua paket data yang masuk serta mengenali paket yang masuk dengan sensor. Kemudian jika serangan telah terdeteksi, maka *Intrusion Prevention System (IPS)* akan menolak akses *(block)* dan mencatat *(log)* semua paket data yang terdeteksi. Jadi, *Intrusion Prevention System (IPS)* ini bertindak seperti *firewall* yang dapat menerima ataupun memblokir paket data yang masuk dan nantinya *Intrusion Detection System (IDS)* yang akan mendeteksi paket data yang masuk secara mendetail.

Intrusion Detection and Prevention (IDPS) ini memiliki 3 metode untuk melakukan pendeteksian, yaitu [18]:

2.2.3.1 *Signature-based Detection*

Metode ini sangat efektif pada saat *Intrusion Detection and Prevention (IDPS)* mendeteksi serangan yang sudah dikenal tapi tidak efektif saat serangannya baru atau tidak dikenal oleh *Intrusion Detection and Prevention (IDPS).* Metode ini membandingkan paket data dan kemudian akan didaftarkan menggunakan operasi perbandingan. Kelemahan metode ini adalah tidak bisa melacak kejadian pada komunikasi yang lebih kompleks.

2.2.3.2 *Anomaly-based Detection*

Metode ini membandingkan kegiatan yang sedang dipantau dengan kegiatan yang normal dengan tujuan untuk mendeteksi adanya penyimpangan. Kelebihan metode ini dapat mendeteksi adanya serangan baru atau serangan yang tidak

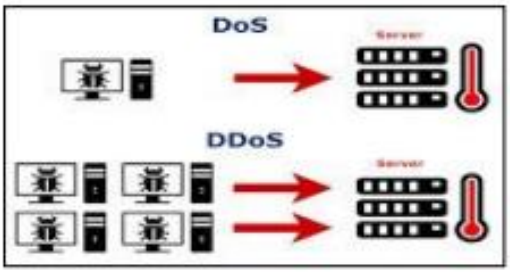
dikenal. Kelemahannya adalah sulit untuk mendeteksi secara akurat.

2.2.3.3 *Stateful Protocol Analysis*

Pada metode ini, sistem *Intrusion Detection and Prevention (IDPS)* akan memahami dan melacak situasi pada protokol, *transport*, dan *application network.* Kelebihan metode ini adalah dapat mendeteksi perintah yang dilakukan secara berulang. Kelemahannya adalah sulit membedakan implementasi *client* dan *server* pada interaksi protokol.

2.2.4 Serangan *Distributed Denial of Service (DDoS)*

Ada banyak sekali jenis ancaman keamanan jaringan, salah satunya adalah serangan dengan perangkat lunak jahat yang ditujukan untuk merusak atau menghancurkan sistem jaringan tersebut. Diantaranya berupa virus, *worm, backdoors, Denial of Service (DoS),* dan *Distributed Denial of Service (DDoS). Distributed Denial of Service (DDoS)* merupakan serangan terhadap server dalam jaringan internet dengan menggunakan banyak *host* yang dilakukan untuk membanjiri jaringan dengan *request* sehingga jaringan server tersebut tidak dapat berfungsi. Adapun tipe dari serangan keamanannya adalah interupsi *(interruption).* Tujuan dari serangan ini adalah agar sistem/server jaringan tidak dapat lagi diakses oleh penggunanya.



Gambar 2. 4 Contoh Penyerangan pada *Interruption* [18]

Ada beberapa jenis *Distributed Denial of Service*, diantaranya [19]:

2.2.4.1 *ICMP Flood*

ICMP Flood merupakan serangan yang membanjiri komputer target dengan paket *ICMP echo request* dan serangan ini tidak perlu menunggu respon dari komputer target. Serangan ini menghabiskan *bandwith* hal ini karena komputer merespon paket *ICMP echo reply*.

2.2.4.2 *TCP Flood*

TCP Flood merupakan serangan yang membanjiri jalur *TCP* yang membuat target kehabisan *bandwidth* pada *port* yang sedang di *Flood*.

2.2.4.3 *UDP Flood*

User Datagram Protocol (UDP) merupakan protokol internet yang mengirim pesan ke komputer lain melalui jaringan tanpa perlu menunggu konfirmasi dari komputer tujuan. Serangan ini akan membanjiri target dengan paket *User Diagram Protocol (UDP).* Dengan tujuan untuk membanjiri *port* acak pada komputer target, sehingga menyebabkan komputer harus berulang kali memeriksa *port* yang di *request* dan jika tidak ditemukan komputer target akan merespon jika *port* tidak ditemukan. Proses ini akan memakan banyak *CPU* dan *memory* sehingga komputer tidak dapat diakses.

2.2.5 *Open Source Security (OSSEC)*

OSSEC merupakan perangkat lunak yang dapat melakukan *log analysis, file integrity checking, policy monitoring, rootkit detection, real time alerting,* dan *active response.* OSSEC memiliki fungsi yang sama dengan *Security Information and Event Management (SIEM)* dan *Security Threat Response Management (STRM).* Arsitektur dari aplikasi ini menggunakan *client-server.* Antara *client* dan *server*

berkomunikasi menggunakan protokol *UDP* dengan *port* 1514 dan dienkripsi menggunakan algoritma *symmetric key blowfish*.

2.2.5.1 Kategori penginstallasian OSSEC [8]

OSSEC memiliki beberapa kategori instalasi, yaitu :

2.2.5.1.1 *Local Installation*

Fungsinya untuk melindungi dan mengamankan satu komputer tunggal.

2.2.5.1.2 *Agent Installation*

Fungsinya untuk melindungi dan mengamankan sistem komputer di sisi klien yang akan melaporkan keamanan sistemnya ke OSSEC secara terpusat.

2.2.5.1.3 *Server Installation*

Fungsinya untuk mengumpulkan informasi dari OSSEC *agent* terkait keamanan sistem di komputer klien. OSSEC *server* ini akan melakukan monitoring *log* dan juga menerima monitoring *log* dari OSSEC *agent* terkait dengan keamanan sistem. Fungsi dari OSSEC *server* ini adalah untuk menyimpan *log* dan melakukan analisa respon.

2.2.5.2 Fitur-fitur OSSEC [1]

Berikut beberapa fitur yang ada pada OSSEC yang fungsinya untuk mengamankan sistem pada jaringan, diantaranya:

2.2.5.2.1 *File Integrity Checking*

Tujuannya untuk memeriksa integritas file yang penting dan jika ada perubahan pada file, OSSEC akan mendeteksi dan memberikan peringatan. Perubahan file ini dapat terjadi karena

beberapa hal, seperti salah ketik oleh admin, penyalahgunaan oleh karyawan, dan serangan.

Setiap file akan memiliki *digital fingerprint* sebagai bentuk keamanan menggunakan kriptografi *hash* atau *hash value.* Nantinya OSSEC akan melakukan pemantauan terhadap file penting tersebut untuk mendeteksi perubahan ketika ada seseorang yang mengubah isi dari file tersebut. Kemudian OSSEC akan membandingkan dengan *hash value file* sebelum dan setelah diubah.

2.2.5.2.2 *Log Monitoring*

Log monitoring ini dilakukan oleh OSSEC untuk memberitahu kegiatan yang dilakukan pada jaringan. OSSEC akan mengumpulkan, menganalisis, dan menghubungkan *log* untuk memberitahu *administrator* jika ada penyalahgunaan yang terjadi. OSSEC akan dapat mendeteksi jika ada aplikasi yang di-*install* pada *client* yang dipasang OSSEC *agent.*

2.2.5.2.3 *Active Response*

Fitur ini merupakan solusi untuk penyerangan secara otomatis. Dengan tujuan untuk memblokir penyerang sehingga sistem dapat melakukan pertahanan serta dapat memantau yang terjadi pada sistem. Dengan fitur ini, OSSEC juga dapat dikategorikan sebagai *Intrusion Prevention System (IPS).*

2.2.5.2.4 *Decoder* dan *rules*

Untuk mendeteksi adanya serangan, OSSEC akan menggunakan file *log* untuk

selanjutnya dilakukan *predecoding* yaitu, proses pengekstrakan informasi statis pada file *log* lalu dibuat *decoder* atau *rules*.

2.2.6 Snort

Snort adalah perangkat lunak yang dapat mendeteksi penyusup dan juga dapat menganalisis lalu lintas secara *real time* dan mendeteksi berbagai serangan [20]. Snort adalah *NIDS* yang bekerja dengan menggunakan *signature detection* dan bekerja sebagai pelacak dan pencatat paket. Snort dapat dioperasikan dengan tiga mode, yaitu:

1. *Packet Sniffer*

   Ini digunakan untuk membaca paket data dari jaringan dan menampilkan status aliran tanpa henti pada layar.

2. *Packet Logger*

   Berfungsi untuk mencatat paket-paket ke dalam *disk.*

3. *Network Intrusion Detection System (NIDS)* mode

   Snort akan mendeteksi serangan dari jaringan komputer.

2.2.7 Wireshark

Wireshark merupakan perangkat lunak yang digunakan untuk memahami struktur dari protokol jaringan yang berbeda [21]. Wireshark dapat mengambil informasi yang sedang berjalan di jaringan. Semua paket data yang berformat *log* akan mudah ditangkap dan dianalisis oleh wireshark. Kekurangan wireshark adalah kesulitan untuk mendeteksi *drive* yang bersifat *wireless* [22]. Wireshark disebut juga sebagai *Network Packet Analyzer* yang berfungsi untuk menangkap paket jaringan dan untuk menampilkan informasi paket secara detail. Wireshark dapat memudahkan memonitoring dan menganalisa paket yang masuk pada jaringan [23].

2.2.8 *Quality of Service (*QoS*)*

*Quality of Service (*QoS*)* merupakan metode pengukuran jaringan untuk mengelola *bandwidth*, *delay*, dan *packet loss* [24]. Tujuan dari *Quality of Service (*QoS*)* untuk menentukan tingkat kepuasan *user* terhadap layanan jaringan. Ada tiga jenis *Quality of Service (*QoS*)* yang sering dipakai, yaitu *best-effort service, integrated service*, dan *differentiated service* [23].

Berikut parameter *Quality of Service (*QoS*)* yang digunakan [25]:

2.2.8.1 *Throughput*

Throughput adalah kemampuan jaringan untuk mengirim data yang efektif dengan satuan *bit per second (bps)*. Persamaan perhitungan *throughput* :

$$Throughput = \frac{Paket\ Data\ Yang\ Diterima(kb)}{Waktu\ Pengiriman\ Data(s)}$$

2.2.8.2 *Delay* atau *Latency*

Delay atau *Latency* merupakan waktu yang dibutuhkan data untuk melakukan perjalanan hingga dari sumber sampai ke tujuan. *Delay* dapat digolongkan menjadi beberapa golongan, diantaranya :

2.2.8.2.1 *Packetization Delay*

Delay ini dikarenakan waktu yang dibutuhkan untuk proses pembentukan paket *IP* dari informasi pengguna. *Delay* hanya terjadi satu kali, yaitu pada sumber informasi.

2.2.8.2.2 *Queuing Delay*

Delay ini karena waktu yang diperlukan router untuk menangani transmisi paket pada jaringan.

2.2.8.2.3 *Delay* Propagasi

Delay ini karena proses perjalanan informasi selama di media transmisi, seperti kabel *SDH* dan *coaxial* atau tembaga.

Persamaan perhitungan *delay* :

$$Delay = \frac{Total\ Delay}{Jumlah\ Total\ Paket}$$

2.2.8.3 *Packet loss*

Packet loss adalah sebuah kegagalan dalam mengirimkan transmisi paket *IP* mencapai tujuannya. Kegagalan tersebut diakibatkan oleh *overload* trafik di jaringan, tabrakan dalam jaringan, *error* media fisik, dan gagal karena terjadi *overflow* pada *buffer*.

Persamaan perhitungan *Packet loss* :

$$Packet\ loss = \frac{(Paket\ Dikirim - paket\ diterima)}{Paket\ Dikirim} \text{ x } 100\ \%$$

2.2.8.4 *Jitter*

Jitter adalah perubahan perbedaan waktu antar paket di jaringan akibat antrian yang panjang saat proses data. Besarnya *jitter* dipengaruhi oleh variasi beban trafik dan besarnya *congestion* pada jaringan. Untuk mendapatkan nilai *Quality of Service (*QoS*)* yang bagus nilai *jitter* harus dipertahankan seminimum mungkin.

Persamaan perhitungan *jitter* :

$$Jitter = \frac{Total\ variasi\ delay}{Total\ Paket\ Yang\ Diterima}$$