# BAB II
# TINJAUAN PUSTAKA

## 2.1 *Intrusion Detection System* (IDS)

IDS (*Intrution Detection System*) adalah sebuah sistem yang melakukan pengawasan terhadap *traffic* jaringan dan pengawasan terhadap kegiatan-kegiatan mencurigakan berhubungan dengan *traffic* jaringan maka IDS akan memberikan peringatan kepada sistem atau administrator jaringan. Dalam banyak kasus IDS juga merespon terhadap *traffic* yang tidak normal/anomali melalui aksi pemblokiran seorang *user* atau alamat IP (*Internet Protocol*) sumber dari usaha pengaksesan jaringan (Ginting et al., 2015).

Ada tiga jenis IDS: *Host Based, Network Based dan Hybrid. Host Based Intrusion Detection System* (HIDS) adalah IDS yang memantau aktivitas sistem untuk mendeteksi serangan lokal. *Network Based Intrusion System* (NIDS) adalah IDS yang memonitor lalu lintas jaringan untuk mendeteksi serangan. *Hybrid Intrusion Detection System* adalah IDS yang merupakan kombinasi dari *host* dan IDS berbasis jaringan. Berikut beberapa komponen yang terdapat pada IDS:

1. Sensor:

   Dapat mengenali adanya security event.
2. Console:

   Dapat memonitor event, alert dan mengontrol sensor
3. Central Engine:

   Berguna untuk menyimpan *events logged* yang dilakukan oleh sensor kedalam database dan menggunakan aturan-aturan keamanan yang berguna untuk menagani *event* yang terjadi (Silva et al., 2016).

Apabila ada aktivitas yang dianggap mencurigakan di dalam jaringan makanIDS akan mendeteksi dan mencatat semua aktivitas yang mencurigakan tersebut. Namun IDS ini tidak dapat melakukan tindakan atau pencegahan jika terjadi serangan atau penyususpan didalam jaringan. Namun IDS mempunyai peran yang cukup membantu dalam hal yang berkaitan dengan keamanan jaringanan, diantaranya:

a. Secara aktif mengamati segala macam kegiatan yang mencurigakan.
b. Memeriksa *audit logs* dengan sangat cermat dan seksama.
c. Mengirim *alert* kepada administrator saat adanya serangan yang terdeteksi.
d. Memberikan laporan segala macam kerentanan yang ditemukan.

Namun kembali lagi kepada kemampuan IDS yang terbatas, yang hanya bisa mencatat dan memberikan laporan kepada administrator, karena IDS tidak dapat menghentikan serangan tersebut.

## 2.2 Snort

Snort adalah IDS yang bekerja dengan menggunakan *signature detection*, berfungsi juga sebagai *sniffer* dan *packet logger*. Snort merupakan suatu perangkat lunak untuk mendeteksi penyusup dan mampu dapat menganalisis lalu lintas *real-time*, hal ini dapat mendeteksi berbagai jenis serangan. Snort bukanlah sebatas *protocol* analisis atau sistem pendeteksi penyusupan (Intrusion Detection System) IDS, melainkan sedikit gabungan diantara keduanya, dan bisa sangat berguna dalam merespons insiden-insiden peyerangan terhadap host-host jaringan. Fitur snort dapat menjadi penolong administrator sistem dan jaringan, dimana mampu memperingatkan kita atas penyusup yang berpeluang berbahaya.

Fitur-fitur inilah yang menjadikan snort sebuah sistem pendeteksi gangguan dan serangan jaringan yang sangat berguna bagi tim penanggulangan insiden. Snort sekarang dapat dioperasikan dengan empat (4) buah mode:

*1.* *Paket sniffer*:
   Praktis membaca paket-paket dari jaringan dan memperlihatkan pada kita dalam bentuk aliran tak terputus pada layar.
2. *Packet logger*:
   Untuk mencatat semua paket yang lewat di dalam disk.
3. NIDS (*Network Intrusion Detection System*):
   Deteksi penyusup pada network: pada mode ini snort akan berfungsi untuk mendeteksi serangan yang dilakukan melalui jaringan komputer.
4. *Inline Mode*:

Mengambil paket dari *iptable* dan menginstruksikan iptable untuk meneruskan paket tersebut berdasarkan jenis *rule* dari snort yang digunakan (Ginting et al., 2015).

## 2.3 Komponen Snort

Snort merupakan bagian dari *Intrusion Detection System* yang terdiri dari beberapa komponen, yaitu:

1. *Packet Decoder*

   *Packet Decoder* mengambil paket dari beberapa jenis perangkat jaringan dan mempersiapkan paket data untuk dapat masuk ke *preprocessed* atau dikirim ke mesin deteksi (*Detektion Engine*).

2. *Preprocessor*

   *Preprocessor* yaitu komponen atau *plug-ins* yang dapat digunakan dengan snort untuk mengatur atau memodifikasi paket data sebelum *Detection Engine* melakukan beberapa operasi untuk mengetahui apakan paket sedang digunakan oleh penyusup.

3. *Detection Engine*

   *Detection Engine* merupakan bagian terpenting dari snort yang berfungsi untuk mendeteksi jika ada aktifitas intrusi dalam sebuah paket. Beban pada *Detection Engine* tergantung pada beberapa faktor, yaitu:

   a) Jumlah *rule*,
   b) Kekuatan mesin yang menjalankan snort,
   c) Kecepatan bus internal yang digunakan dalam mesin snort,
   d) Beban pada jaringan.

4. *Logging* dan *Alert System*

   Tergantung pada yang ditemukan didalam *Detection Engine* dalam sebuah paket, paket yang digunakan untuk mencatat aktivitas atau menghasilkan *alert*.

5. *Output Modules*

   *Output modules* atau *plug-in* dapat melakukan operasi yang berbeda-beda tergantung pada bagaimana ingin menyimpan *output* yang dihasilkan oleh *logging* dan sistem *alert* dari snort (Sobari, 2015).

## 2.4 *Scanning*

*Scanning* merupakan aktivitas yang dilakukan untuk mendapatkan informasi target. Adapun informasi yang ditemukan oleh penyerang antara lain IP address, sistem operasi, arsitektur sistem, service running di tiap komputer. Scanning dapat dibagi menjadi tiga jenis yaitu:

1. Port Scanning merupakan scanning yang bertujuan untuk menemukan port-port yang terbuka dari suatu host.
2. Network Scanning merupakan scanning yang bertujuan untuk menemukan host atau komputer yang aktif pada suatu jaringan.
3. Vulnerability Scanning merupakan scanning yang bertujuan menemukan kelemahan dari suatu sistem.

Tujuan dari scanning antara lain:

1. Untuk mendeteksi live sistem yang berjalan di jaringan.
2. Untuk menemukan port yang aktif/running.
3. Untuk menemukan sistem operasi yang berjalan di sistem target.
4. Untuk menemukan service yang berjalan.
5. Untuk menemukan IP address sistem target.

## 2.5 DoS (*Denial of Services*)

DoS (*Denial of Services*) menempati *bandwidth server* korban dengan membanjiri *server* dengan sejumlah besar paket data dan juga dikenal sebagai serangan bandwidth. Mereka bertujuan melelahkan sumber daya *server* kritis seperti kapasitas tautan internet, ruang stack dalam jaringan dll. Sehingga *server* tidak dapat memberikan layanan kepada pengguna asli. (Gupta & Sharma, 2019).

## 2.6 Apache

**Gambar 2.1** Apache
(Sukma,2015)

Apache adalah sebuah nama *web server* yang bertanggung jawab pada *request-response* HTTP dan *logging* informasi secara detail (kegunaan basicnya). Apache juga diartikan sebagai suatu *web server* yang kompak, modular, mengikuti standar protokol HTTP, dan tentu saja sangat digemari. Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah.

Menurut Firdaus (2007:5) *web server* merupakan sebuah bentuk server yang khusus digunakan untuk menyimpan halaman website atau homepage. Apache merupakan turunan dari web server yang dikeluarkan oleh NSCA yaitu NSCA HTTPd sekitar tahun 1995-an. Pada dasarnya, Apache adalah "APatCHy" (*patch*) dan pengganti dari NCSA HTTPd. Apache *web server* merupakan tulang punggung permintaan dari *client* yang menggunakan *browser*, seperti Netscape Navigator, Internet Explorer, Modzilla, lynx dan lain-lain. *Web server* dalam berkomunikasi dengan kliennya menggunakan protokol HTTP (*Hyper Text Transfer Protocol*). Apache berada di bawah GNU, *General Public Licensi* yang bersifat *free* sehingga Apache dapat didownload gratis. Adapun pertimbangan dalam memilih Apache adalah:

1. Apache termasuk dalam kategori *free* software (software gratis).
2. Instalasi apache sangat mudah.
3. Mampu beroperasi pada banyak platform sistem operasi, seperti Linux, Windows dan lain-lain.

Apache Web Server merupakan *web server* yang bersifat *open source* dan mempunyai performa yang sangat bagus, fleksibel dan mendukung berbagai macam platform sistem operasi seperti Windows NT/9x, UNIX, Netware 5x, OS/2 dan berbagai macam sistem operasi lainnya. Apache sangat cepat sekali mengeluarkan update terbarunya, sehingga mengurangi munculnya *bugs* dan kelemahan program.

Kelebihan:

1. Freeware (*software* gratis)
2. Mudah di-install.
3. Mampu beroperasi pada berbagai platform sistem operasi.

4. Mudah mengkonfigurasinya.
5. Mudah dalam menambahkan periferal lainnya ke dalam platform web servernya, misalnya: untuk menambahkan modul, cukup hanya menset *file* konfigurasinya agar mengikutsertakan modul itu ke dalam kumpulan modul lain yang sudah dioperasikan.
6. Mampu di kompilasi sesuai dengan spesifikasi HTTP yang sekarang.
7. Menyediakan feature untuk multihomed dan virtual server.
8. Secara otomatis menjalankan *file* index.html, halaman utamanya, untuk ditampilkan secara otomatis pada clientnya.
9. Lebih aman karena memiliki level-level pengamanan
10. Apache mempunyai komponen dasar terbanyak di antara *web server* lain, yang berarti bahwa *web server* Apache termasuk salah satu dari *web server* yang lengkap.
11. Performansi dan konsumsi sumberdaya (*resource*) dari *web server* apache tidak terlalu banyak, hanya sebesar 20 MB untuk *files* dasarnya dan setiap daemon nya hanya memerlukan sebesar 950 KB memory per-child.
12. Mendukung transaksi yang aman (*secure transaction*) menggunakan SSL (*Secure Socket Layer*).
13. Mempunyai dukungan teknis melalui web.
14. Mempunyai kompatibilitas platform yang tinggi.
15. Mendukung third party berupa modul-modul tambahan.

## 2.7 MySQL

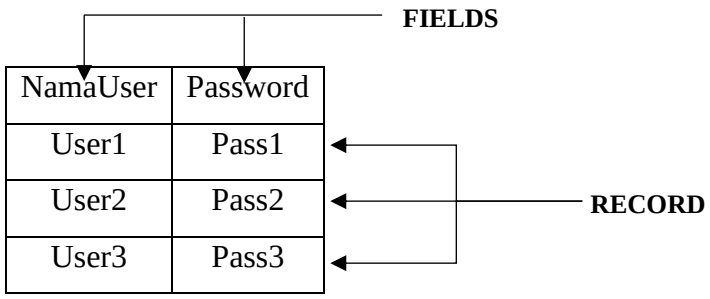
**Gambar 2.2** MySQL

(Sukma,2015)

MySQL adalah sebuah database manajemen sistem (DBMS) popular yang memiliki fungsi sebagai *relational database manajemen system* (RDBMS). Selain itu MySQL *software* merupakan suatu aplikasi yang sifatnya *open source* serta server basis data MySQL memiliki kinerja sangat cepat, *reliable*, dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded systems* (Yuliansyah, 2014).

Database adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur *file* yang menyusun sebuah database adalah *Data Record* dan *Field*.

1. Data adalah satu satuan informasi yang akan diolah. Sebelum diolah, data dikumpulkan di dalam suatu *file* database.
2. RECORD adalah data yang isinya merupakan satu kesatuan seperti NamaUser dan *Password*. Setiap keterangan yang mencakup NamaUser dan *Password* dinamakan satu *record*. Setiap *record* diberi nomor urut yang disebut nomor *record* (*Record Number*).
3. FIELD adalah sub bagian dari *Record*. Dari contoh isi *record* di atas, maka terdiri dari 2 *field*, yaitu: *field* NamaUser dan *Password*.




**Gambar 2.3** Contoh Tabel Database NamaUser Password

## 2.8 Banyard2

Barnyard2 adalah tool *open source* sebagai penerjemah *alert unified* dan *log* dari snort. Barnyard2 dapat meningkatkan efisiensi snort dengan cara mengurangi beban pada sensor deteksi. Barnyard2 bekerja dengan membaca *Snort's unified logging output files* dan memasukannya kedalam *database*. Jika *database* tidak

tersedia maka Barnyard2 akan memasukan semua data ketika *database* tersedia kembali sehingga tidak ada *alert* atau *log* yang hilang (Notoatmodjo, 2014).

## 2.9 *Basic Analysis and Security Engine* (BASE)

BASE (Basic Analysis and Security Engine) adalah sebuah *interface web* untuk melakukan analisis dari intrusi yang snort telah deteksi pada jaringan. (Orebaugh, 2008:217) BASE ditulis oleh Kevin Johnson adalah program analisis sistem jaringan berbasis PHP yang mencari dan memproses *database* dari *security event* yang dihasilkan oleh berbagai program monitoring jaringan, firewall atau sensor IDS. (Kuhlenberg, 2007:424). Ini menggunakan otentifikasi pengguna dan sistem peran dasar, sehingga sebagai admin keamanan dapat memutuskan apa dan berapa banyak informasi yang setiap pengguna dapat melihat.

BASE adalah sistem manajemen basis data berbasis *desktop* yang lengkap, didesain untuk memenuhi kebutuhan yang luas dari pengguna, mulai dari melacak koleksi CD pribadi Anda, hingga menghasilkan laporan penjualan bulanan BASE menawarkan panduan untuk membantu pengguna yang baru terhadap desain basis data (baru terhadap BASE) untuk membuat *Tabel*, *Query*, *Form*, dan *Report*, bersama dengan sekumpulan definisi tabel yang sudah didefinisikan untuk melacak Asset, Konsumen, Penjualan, *Invoice*, dan banyak lagi.

Ketika hanya memerlukan basis data personal, BASE menawarkan mesin basis data relasional HSQL, dikonfigurasi untuk pengguna tunggal, dengan data tersimpan pada dokumen BASE, beserta dengan dukungan *native* untuk dokumen BASE. Untuk kebutuhan yang lebih besar, BASE mendukung berbagai basis data yang populer secara native: MySQL, Adabas D, Microsoft Access, dan Postgre SQL. Sebagai tambahan, dukungan untuk driver standar JDBC dan ODBC juga mengijinkan untuk terhubung secara virtual pada sembarang basis data yang ada (Mentang et al., 2015).

## 2.10 Sistem Operasi

Sistem operasi (atau sering disebut *operating system*; OS) adalah seperangkat program yang mengelola sumber daya hardware komputer, dan menyediakan

layanan umum untuk aplikasi *software*. Sistem operasi adalah jenis yang paling penting dari *software* sistem dalam sistem komputer (Fadhilah, 2007).

Sistem operasi mempunyai tiga sasaran utama yang diharapkan dapat menyediakan layanan sesuai kebutuhan *user*nya, antara lain:

1. Kenyamanan, membuat penggunaan komputer menjadi lebih nyaman atau lebih dikenal dengan istilah *User* Friendly.
2. Efisien, penggunaan sumber daya sistem komputer secara efisien.
3. Mampu Berevolusi, sistem operasi harus dibangun sehingga memungkinkan dan memudahkan pengembangan, pengujian serta pengajuan sistem-sistem yang baru (Novianto, 2013).

## 2.11 CentOS

**Gambar 2.4** Centos

(Sumber: wiki.centos.org)

CentOS (*Community Enterprise Operating System*) merupakan Distro Linux yang cocok digunakan dalam skala *Enterprise* selain itu juga gratis. CentOS di buat dari *source code Red Hat Enterprise* (RHEL) yang dikembangkan oleh sebuah komunitas yang disebut CentOS *Project. Software* ini dirilis dibawah *General Public License* (GPL) yang selanjutnya dikembangkan oleh sebuah komunitas yang disebut CentOS *Project*.

Keuntungan linux CentOS adalah *open source*, kompatibilitas, *user* friendly. CentOS tersedia secara gratis, dukungan teknis utamanya disediakan terhadap para pengguna melalui milis, forum berbasis *web*, ataupun *chat*. Proyek CentOS tidak berafiliasi dengan Red Hat. Untuk penggalangan dana, CentOS berbasis donasi dari para pengguna serta sponsor dari perusahaan-perusahaan yang menggunakannya (Amnur et al., 2014).

## 2.12 Keamanan Jaringan

Jaringan komputer (*computer network*) menurut Sutanta (2005) merupakan interkoneksi sejumlah komputer dan peralatan (*peripheral*) yang dihubungkan dengan jalur transmisi dan alat komunikasi membentuk suatu sistem sehingga dapat saling bertukar data, informasi atau menggunakan peralatan secara bersama-sama untuk melaksanakan tugas pengolahan data.

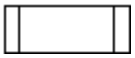
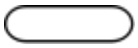
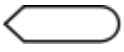
Jaringan komputer adalah sekelompok komputer otonom yang saling berhubungan satu dengan yang lainnya menggunakan *protocol* komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, aplikasi, dan perangkat keras secara bersamasama. Jaringan komputer dapat diartikan juga sebagai kumpulan sejumlah terminal komunikasi yang berada di berbagai lokasi yang terdiri lebih dari satu komputer yang saling berhubungan, oleh (Sukmaaji & Rianto, 2008).

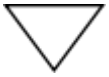
## 2.13 Flowchart

*Flowchart* merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan tipe operasi program yang berbeda. Sebagai *representasi* dari sebuah program, *flowchart* maupun algoritma dapat menjadi alat bantu untuk memudahkan perancangan alur urutan logika suatu program, memudahkan pelacakkan sumber kesalahan program, dan alat bantu untuk menerangkan logika program (Budiutomo, 2017). Simbol *Flowchart* dapat dilihat pada tabel 2.1.

**Tabel 2.1** Simbol Flowchart

| No. | Simbol | Nama Simbol | Keterangan |
|---|---|---|---|
| 1. | | *Alternate Process* | Menyatakan segala jenis operasi yang diproses dengan menggunakan mesin yang memiliki *keyboard.* |
| 2. | | *Decision* | Suatu penyelesaian kondisi dalam program. |

| | | | |
|---|---|---|---|
| 3. | | *Data* | Mewakili data *input* atau *output.* |
| 4. |  | *Predefined Process* | Suatu operasi yang rinciannya di tunjukkan di tempat lain |
| 5. | | *Document* | *Document input* dan *output* baik untuk proses manual, mekanik atau komputer. |
| 6. |  | *Terminator* | Untuk menunjukkan awal dan akhir dari suatu proses. |
| 7. | | *Process* | Proses dari operasi program komputer. |
| 8. | | *Manual Input* | *Input* yang menggunakan *online* keyboard. |
| 9. | | *Conector* | Penghubung ke halaman yang masih sama . |
| 10. | | *Off-Page Connector* | Penghubung ke halaman lain. |
| 11. |  | *Display* | *Output* yang ditampilkan di monitor. |
| 12. | | *Delay* | Menunjukkan penundaan. |

| | | | |
|---|---|---|---|
| 13. | | *Preparation* | Memberi nilai awal suatu besaran. |
| 14. | | *Manual Operation* | Pekerjaan manual. |
| 15. | | *Card* | *Input* atau *output* yang menggunakan kartu. |
| 16. | | *Punch Tape* | *Input* atau *output* menggunakan pita kertas berlubang. |
| 17. |  | *Merge* | Penggabungan atau penyimpanan beberapa proses atau informasi sebagai salah satu. |
| 18. | | *Dirrect Access Storage* | *Input* atau *output* menggunakan drum magnetik. |
| 19. | | *Magnetic Disk* | *Input* atau *output* menggunakan *hard disk.* |
| 20. | | *Sequential Access Storage* | *Input* atau *output* menggunakan pita magnetik. |
| 21. | | *Sort* | Proses pengurutan data di luar komputer. |

| | | | |
|---|---|---|---|
| 22. | | *Stored Data* | *Input* atau *output* menggunakan *diskette.* |
| 23. | | *Extract* | Proses dalam jalur paralel. |
| 24. | | *Arrow* | Menyatakan jalan atau arus suatu proses. |
| 25. | | *Summing Junction* | Untuk berkumpul beberapa cabang sebagai proses tunggal. |
| 26. | | *Or* | Proses menyimpang dalam dua proses. |