BAB 2
LANDASAN TEORI

Pada bab ini akan membahas tentang teori dasar yang melandasi permasalahan dan penyelesaian yang diangkat dalam penelitian ini. Dasar teori yang di berikan meliputi keamanan jaringan, konsep dari IPS, *Honeypot*, FWSNORT, dan PSAD.

2.1 Jaringan Komputer

Jaringan komputer adalah sekelompok komputer otonom yang saling berhubungan antara satu dengan lainnya menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, program, penggunaan bersama perangkat keras seperti printer, harddisk, dan sebagainya (Suyanto, 2004 : 1). Sedangkan menurut Andreas Handojo dan Justinus Andjarwirawan (2002 : 96), jaringan komputer adalah komunikasi data dari suatu komputer ke komputer lain. Berdasarkan penjelasan tersebut, maka dapat di simpulkan bahwa jaringan komputer adalah interaksi komputer dengan komputer lainnya yang dapat saling bertukar data dan informasi, serta dapat menggunakan perangkat keras yang terhubung secara bersama-sama baik printer, harddisk, scanner, dan sebagainya.

Tujuan dibangunnya suatu jaringan komputer adalah membawa informasi secara tepat dan tanpa adanya kesalahan dari sisi pengirim menuju kesisi penerima melalui media komunikasi. Komunikasi data sendiri memiliki tujuan yaitu :

1. Memungkinkan pengiriman data dalam skala jumlah yang besar, efisien, ekonomis, dan tanpa adanya kesalahan dari suatu tempat ke tempat lainnya.
2. Memungkinkan penggunaan sistem komputer dan peralatan pendukung dari jarak jauh (remote).
3. Mempermudah pengelolaan dan pengaturan data.
4. Mendapatkan data langsung dari sumbernya.
5. Mempercepat penyebar luasan informasi.

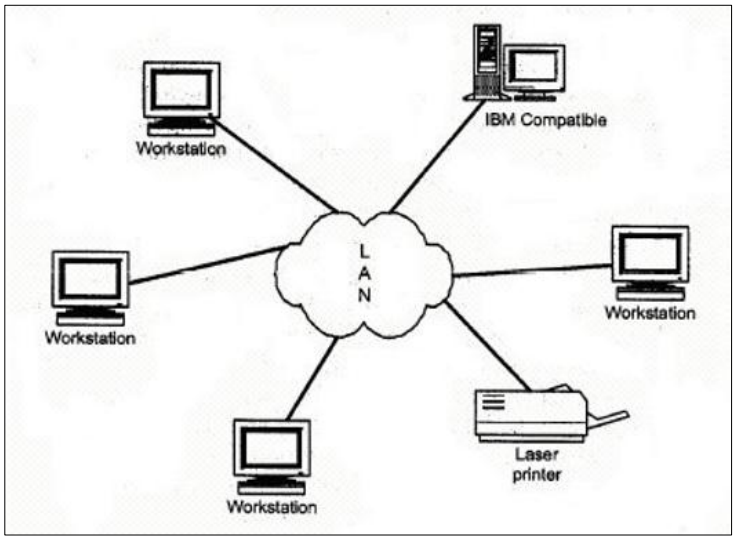
2.1.1 Berdasarkan Area Cakupan

Area adalah suatu wilayah geografis yang digunakan untuk keperluan khusus, sedangkan cakupan adalah lingkup jangkauan sesuatu. Berdasarkan pengertian tersebut area cakupan dapat diartikan jarak jangkauan suatu benda atau hal secara wilayah geografisnya. Jaringan komputer dapat dibedakan berdasarkan area cakupannya. Ada tiga kategori utama jaringan komputer menurut cakupannya, yaitu:

1. LAN (*Local Area Network*)

   Local area network merupakan jaringan milik pribadi didalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik untuk memakai bersama sumberdaya dan saling bertukar informasi (Suyanto, 2004 : 2).

   Berikut ini adalah gambar skema jaringan LAN yang dapat dilihat pada gambar 2.1.



Gambar 2.1 Skema Jaringan LAN

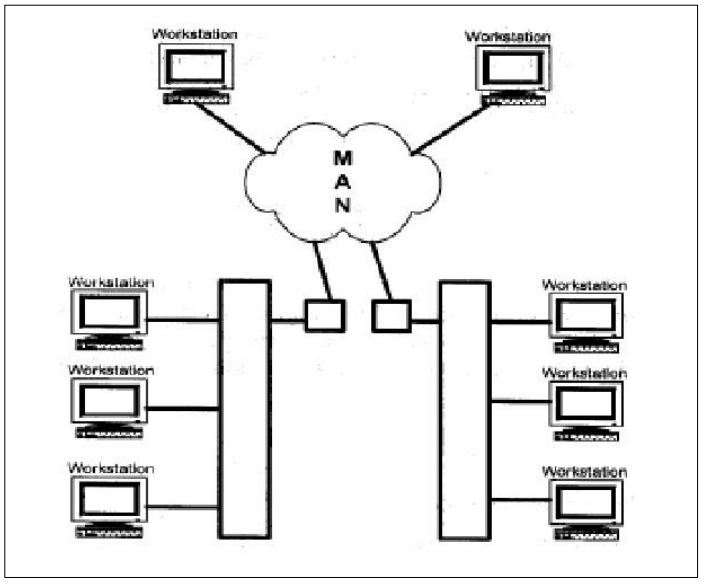
(Sumber : Wahana Komputer , 2003 : 6)

2. MAN (*Metropolitan Area Network*)

   Metropolitan area network pada dasaranya merupakan versi LAN yang berukuran lebih besar dan biasanya menggunakan teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang letaknya

berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi atau umum. MAN mampu menunjang data dan suara, bahkan dapat berhubungan dengan jaringan televisi kabel.

Berikut ini adalah gambar skema jaringan MAN yang dapat dilihat pada gambar 2.2.



Gambar 2.2 Skema Jaringan MAN

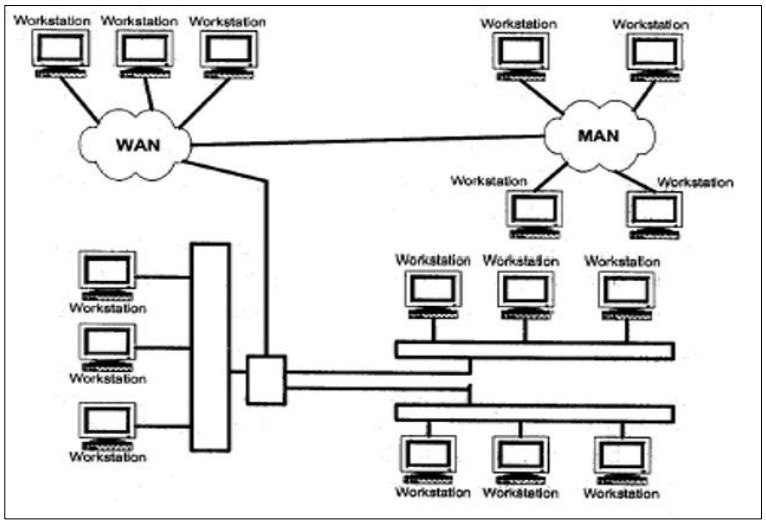
(Sumber : Wahana Komputer , 2003 : 11)

3. WAN (*Wide Area Network*)

Wide Area Network (WAN) merupakan jaringan yang lebih besar dari MAN dan mencakup daerah geografis yang luas, seringkali mencakup sebuah negara atau benua. WAN terdiri dari kumpulan mesin yang bertujuan unuk menjalankan program-program (aplikasi) pemakai.

Pada sebagian besar WAN, jaringan terdiri dari sejumlah banyak kabel atau saluran telepon yang menghubungkan sepasang *router*. Bila dua *router* yang tidak menggunakan kabel yang sama akan melakukan komunikasi, keduanya harus berkomunikasi secara tidak langsung melalui *router* lainnya. Ketika sebuah paket dikirimkan dari sebuah *router* ke *router* lainnya melalui sebuah router perantara atau lebih, maka paket akan diterima router perantara dalam keadaan lengkap, disimpan sampai saluran output menjadi bebas, dan kemudian baru diteruskan. (Tanenbaum, 1997 : 8-9).

Berikut ini adalah gambar skema jaringan WAN yang dapat dilihat pada gambar 2.3.



Gambar 2.3 Skema Jaringan WAN

(Sumber : Wahana Komputer , 2003 : 11)

2.1.2 Berdasarkan Media Transmisi

Media adalah suatu bentuk atau benda yang digunakan untuk menyampaikan informasi atau pesan, sedangkan transmisi adalah komunikasi data melalui penyebaran dan pemrosesan sinyal-sinyal. Sehingga media transmisi dapat diartikan perangkat yang digunakan sebagai jalur transmisi dari data yang dikirimkan. Berdasarkan media transmisinya, jaringan komputer dapat dibagi menjadi dua jenis, yaitu:

1. *Wire Network (Guided Media)*

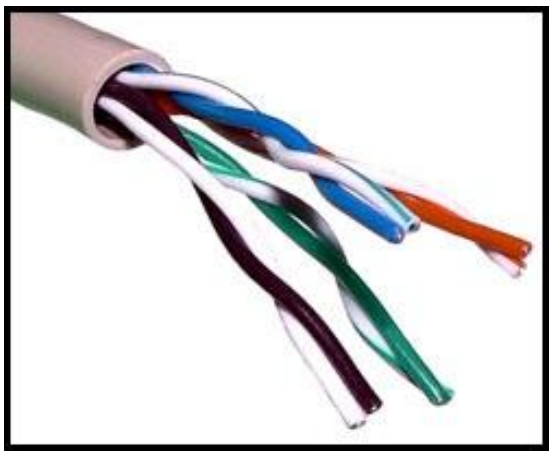
   Wire Network adalah jaringan komputer yang menggunakan kabel sebagai media transmisinya, kabel yang digunakan pun bermacam-macam, antara lain:

a. Kabel *Unshielded Twisted Pair*

   unshielded twisted pair yaitu jenis kabel ini terbuat dari bahan penghantar tembaga, mempunyai isolasi dari plastik & terbungkus oleh bahan isolasi yang dapat melindungi dari api dan juga kerusakan fisik, kabel UTP sendiri

terdiri dari 4 pasang inti kabel yang saling berbelit dimana masing-masing pasang mempunyai kode warna berbeda.

Berikut ini adalah gambar kabel UTP yang dapat dilihat pada gambar 2.4.

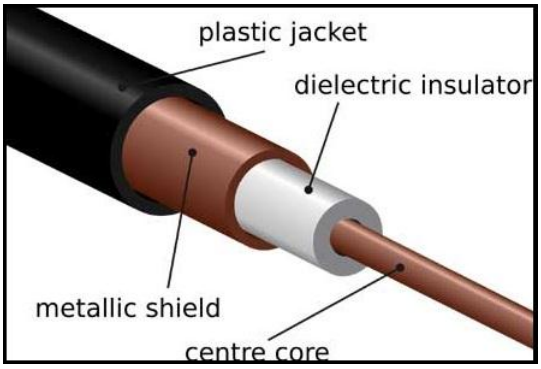


Gambar 2.4 Kabel UTP

b. Kabel Coaxial

Coaxial adalah suatu jenis kabel yang menggunakan dua buah konduktor. Pusatnya berupa inti kawat padat yang dilingkupi oleh sekat yang kemudian dililiti lagi oleh kawat berselaput konduktor. Jenis kabel ini biasa digunakan untuk jaringan dengan *bandwith* yang tinggi. Kabel coaxial mempunyai pengalir tembaga di tengah (*centre core*). Lapisan plastik (*dielectric insulator*) yang mengelilingi tembaga berfungsi sebagai penebat di antara tembaga dan lapisan metal (*metallic shield)*. Lapisan metal berfungsi untuk menghalang sembarang gangguan luar, sedangkan lapisan paling luar adalah lapisan plastik yang disebut *jacket plastic*. Lapisan ini berfungsi seperti jaket yaitu sebagai pelindung bagian terluar.

Berikut ini adalah gambar kabel coaxcial yang dapat dilihat pada gambar 2.5.

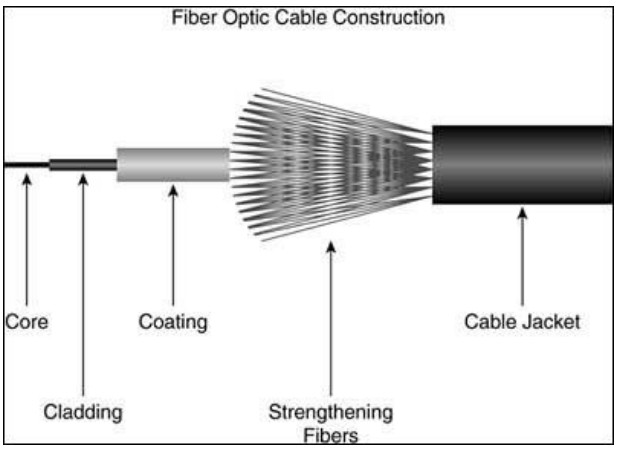


Gambar 2.5 Kabel Coaxcial

c. Kabel *Fiber Optic*

Kabel *Fiber optic* adalah sebuah kabel yang terbuat dari serat kaca dengan teknologi canggih dan mempunyai kecepatan transfer data yang lebih cepat daripada kabel biasa, biasanya *fiber optic* digunakan pada jaringan *backbone* (Tulang Punggung) karena dibutuhkan kecepatan yang lebih dalam jaringan ini,namun pada saat ini sudah banyak yang menggunakan *fiber optic* untuk jaringan biasa baik LAN, WAN maupun MAN karena dapat memberikan dampak yang lebih pada kecepatan dan bandwith karena *fiber optic* ini menggunakan bias cahaya untuk mentransfer data yang melewatinya dan sudah barang tentu kecepatan cahaya tidak diragukan lagi namun untuk membangun jaringan dengan *fiber optic* dibutuhkan biaya yang cukup mahal dikarenakan dibutuhkan alat khusus dalam pembangunannya.

Berikut ini adalah gambar kabel *fiber optic* yang dapat dilihat pada gambar 2.6.



Gambar 2.6 Kabel Fiber Optic

2. *Wireless Nerwork (Unguided Media)*

   Wireless Network adalah jaringan komputer yang menggunakan media gelombang radio, *infrared*, ataupun *microwave* sebagai media transmisinya. Spektrum yang digunakan pada *wireless network* antara 3kHz hingga 900 THz.

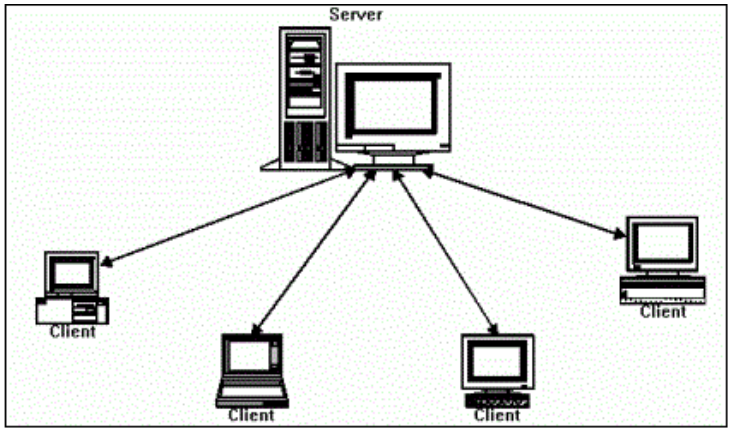
2.1.3 Berdasarkan Fungsi

Berdasarkan fungsinya, jaringan komputer dibagi menjadi dua jenis, yaitu:

1. *Client Server*

   Jenis jaringan ini terdapat komputer yang bertindak sebagai *server* dan komputer yang berperan sebagai *client* (*workstation*). Secara normal komputer *server* dapat mengontrol sepenuhnya komputer *client*. *Server* biasanya mempunyai kemampuan yang lebih besar dari pada *client* (Wahidin, 2007 : 1).

   Berikut ini adalah gambar contoh jaringan *client server* yang dapat dilihat pada gambar 2.7.

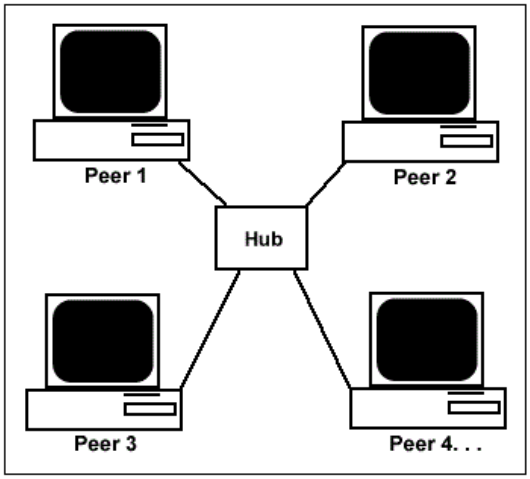


Gambar 2.7 Jaringan Client Server

2. *Peer to Peer*

   Peer to Peer merupakan jenis jaringan yang tidak memerlukan *server* secara khusus, karena komputer yang terhubung pada jaringan dapat bertindak sebagai *server* ataupun *client*. Contoh: hubungan komputer ke komputer (Wahidin, 2007 : 1).

Berikut ini adalah contoh jaringan *peer to peer* yang dapat dilihat pada gambar 2.8.



Gambar 2.8 Jaringan Peer to Peer

2.2 Topologi Jaringan Komputer

Topologi menggambarkan struktur dari suatu jaringan atau bagaimana sebuah jaringan didesain. Dalam definisi topologi terbagi menjadi dua, yaitu topologi fisik (Topologi menggambarkan struktur dari suatu jaringan atau bagaimana sebuah jaringan didesain. Dalam definisi topologi terbagi menjadi dua, yaitu topologi fisik (*physical topology*) yang menunjukan posisi pemasangan kabel secara fisik dan topologi logic (*logical topology*) yang menunjukan bagaimana suatu media diakses oleh *host* (Suyanto ,2004 : 3).

2.2.1 Topologi Fisik

Topologi fisik adalah desain suatu jaringan komputer dengan melihat bentuk nyata pada posisi dan letak *node* atau komputer, topologi fisik memerlukan kabel jaringan dan peralatan pendukung seperti kondensator, *hub*, *switch*, dan lainnya.

Berikut ini adalah jenis-jenis topologi fisik yang digunakan pada sebuah jaringan, yaitu :

1. Topologi *Bus*

Topologi *Bus* terdiri dari satu jalur kabel utama dimana pada setiap ujungnya dipasang sebuah *terminator*. Semua *node* pada jaringan terkoneksi kabel utama (*backbone*). Contoh model topologi bus dapat dilihat pada gambar 2.9.



Gambar 2.9 Topologi Bus

Berikut ini adalah keuntungan-keuntungan dan kerugian-kerugian dalam menggunakan topologi bus yang dijabarkan pada tabel 2.1, yaitu sebagai berikut :

Table 2.1 Kelebihan dan Kekurangan Topologi Bus

Kelebihan	**Kekurangan**
1. Mudah dikembangkan. 2. Jarak LAN tidak terbatas. 3. Keterandalan jaringan tinggi. 4. Kecepatan pengiriman tinggi. 5. Jumlah terminal dapat ditambah ataupun dikurangi tanpa mengganggu operasi yang telah berjalan. 6. Tidak diperlukan pengendalian pusat. 7. Kondusif untuk konfigurasi jaringan pada gedung bertingkat	1. Jika *traffic* data terlalu tinggi atau padat dapat terjadi kemacetan (*congestion*). 2. Diperlukan *repeater* untuk menguatkan sinyal pada pemasangan jarak jauh. 3. Oprasional jaringan LAN bergantung pada setiap terminal. 4. Jika terjadi gangguan atau kerusakan pada salah satu *node* dalam jaringan maka akan mempengaruhi jaringan secara keseluruhan, bahkan ada kemungkinan jaringan akan terhenti.

2. Topologi *star*

Topologi ini dirancang agar setiap *node* yang terhubung pada jaringan terkoneksi melalui konsentrator seperti *hub* atau *switch*. Semua data yang akan dikirim akan melawati konsetrator terlebih dahulu sebelum dikirim ke

tujuan. Konsetrator ini berfungsi untuk mengendalikan jaringan itu da bertindak juga sebagai *repeater*. Contoh model topologi star dapat dilihat pada gambar 2.10.



Gambar 2.10 Topologi Star

Berikut ini adalah keuntungan-keuntungan dan kerugian-kerugian dalam menggunakan topologi star yang dijabarkan pada tabel 2.2, yaitu sebagai berikut :

Table 2.2 Kelebihan dan Kekurangan Topologi Star

Kelebihan	**Kekurangan**
1. Kehandalan topologi ini lebih besar dari topologi lainnya. 2. Mudah dikembangkan. 3. Keamanan data tinggi. 4. Kemudahan akses ke jaringan LAN lainnya.	1. *Traffic* data yang padat menyebabkan jaringan lambat. 2. jaringan tergantung pada terminal pusat atau konsentrator.

3. Topologi *Ring*

Topologi *ring* mempunyai konfigurasi yang mirip dengan topologi *star*, namun tanpa adanya konsentrator. Sehingga media transmisi akan menyerupai cincin. Contoh model topologi ring dapat dilihat pada gambar 2.11.



Gambar 2.11 Topologi Ring

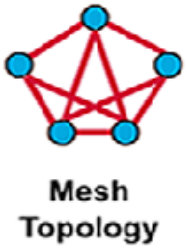
Berikut ini adalah keuntungan-keuntungan dan kerugian-kerugian dalam menggunakan topologi ring yang dijabarkan pada tabel 2.3, yaitu sebagai berikut :

Table 2.3 Kelebihan dan Kekurangan Topologi Star

Kelebihan	**Kekurangan**
1. Laju data (*transfer rate*) tinggi. 2. Dapat melayani *traffic* data yang padat. 3. Tidak diperlukan *host*, relatif lebih mudah. 4. Dapat melayani berbagai media pengiriman. 5. Komunikasi antar terminal mudah. 6. Waktu yang diperlukan untuk mengakses data optimal.	1. Penambahan atau pengurangan terminal cukup sulit. 2. Kerusakan pada media pengiriman data menghentikan kerja seluruh jaringan. 3. Harus ada kemampuan untuk mendeteksi kesalahan dan metode pengisolasian kesalahan. 4. Tidak kondusif untuk pengiriman suara, gambar, dan data.

4. Topologi *Mesh*

   Topologi *mesh* memberi setiap perangkat koneksi khusus *point-to-point* terhadap semua perangkat lain yang berada dalam satu jaringan. Maksud dari koneksi khusus ini adalah bahwa satu koneksi hanya membawa paket antara dua perangkat yang saling terkoneksi. Contoh model topologi mesh dapat dilihat pada gambar 2.12.



Gambar 2.12 Topologi Mesh

Berikut ini adalah keuntungan-keuntungan dan kerugian-kerugian dalam menggunakan topologi ring yang dijabarkan pada tabel 2.4, yaitu sebagai berikut :

Table 2.4 Kelebihan dan Kekurangan Topologi Mesh

Kelebihan	**Kekurangan**
1. Dapat melayani berbagai media pengiriman. 2. Tidak diperlukan pengendalian pusat.	1. Penambahan atau pengurangan terminal sangat sulit. 2. Kerusakan pada salah satu terminal dapat mengakibatkan kelumpuhan jaringan. 3. Kerusakan pada salah satu terminal membuat *traffic* data terhambat.

2.2.2 Topologi Logikal

Topologi logikal mempunyai hubungan dengan bagaimana data ditransmisikan pada sebuah jaringan. Berikut adalah topologi logikal berdasarkan metode aksesnya.

1. Topologi *Broadcast*

   Secara sederhana dapat digambarkan yaitu suatu *host* yang mengirimkan data kepada seluruh *host* lain paa media jaringan.

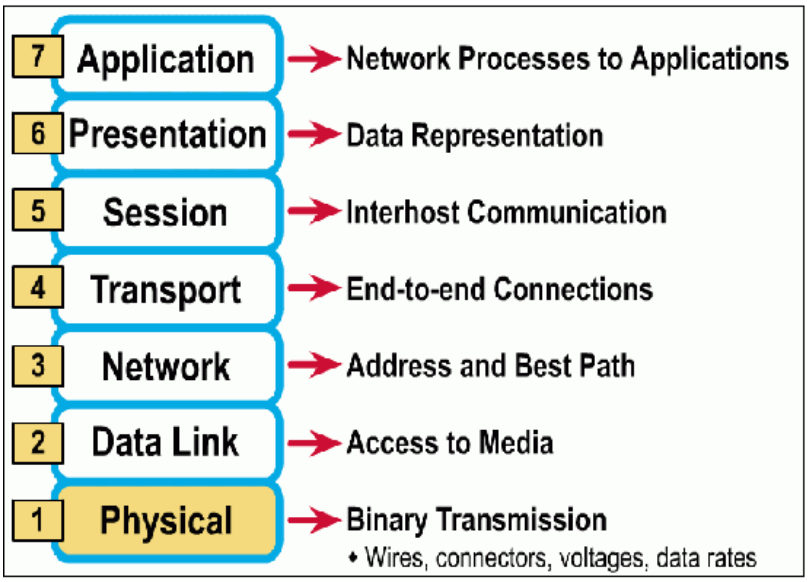
2. Topologi *Token Passing*

   Mengatur pengiriman data pada *host* melalui media dengan menggunakan *token* yang secara teratur berputar pada seluruh *host. Host* hanya dapat mengirimkan data hanya jika *host* tersebut memiliki *token*. Dengan *token* ini, *collision* dapat dicegah.

2.3 *Open System Interconnection* (OSI)

Open system interconnection (OSI) merupakan salah satu standar dalam protokol jaringan yang dikembangakan oleh ISO yang memberikan gambaran tentang fungsi, tujuan, dan kerangka kerja tentang struktur model referensi untuk proses yang bersifat logis dalam sistem komunikasi (Suyanto, 2008 : 14).

OSI adalah suatu standar komunikasi antar entitas yang terdiri atas tujuh lapisan. Ketujuh lapisan tersebut mempunyai peranan dan fungsi yang berbeda antara satu terhadap yang lainnya. Setiap *layer* bertanggung jawab secara khusus pada proses komunikasi data. Tujuan utama model OSI adalah membantu desainer jaringan memahami fungsi dari tiap-tiap *layer* yang berhubungan dengan aliran komunikasi data, termasuk jenis protokol jaringan dan metode transmisi. Model OSI dibagi menjadi tujuh *layer*, dengan karekteristik dan fungsinya masing-masing. Tiap *layer* harus dapat berkomunikasi dengan *layer* di atas maupun di bawahnya secara langsung melayani beberapa protokol dan standar. Berikut ini adalah layer pada model OSI dapat dilihat pada gambar 2.13



Gambar 2.13 Layer Pada Model OSI

Berikut adalah penjelasan dan fungsi-fungsi setiap *layer* pada model OSI :

1. *Physical Layer* (*Layer* 1)

Layer ini berfungsi untuk mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan (seperti *Ethernet* atau *token ring*), dan penglabelan. Selain itu, *layer* ini juga mendefinisikan bagaimana *Network Interface Card* (NIC) dapat berinteraksi dengan media kabel atau radio.

2. *Data Link Layer* (*Layer* 2)

   Layer ini mendefinisikan bagaimana cara mengirimkan data melalui suatu media, baik media kabel maupun nirkabel dengan *physical addressing*. Tugas utama dari *layer* ini adalah *error checking, flow control,* dan *media access control*untuk mengatur paket yang akan dikirim.

3. *Network Layer* (*Layer* 3)

   Layer ini berfungsi untuk mendefinisikan alamat-alamat IP, membuat *header* untuk paket-paket dan mencari jalur terbaik kemudian melakukan *routing* melalui *internetworking* dengan menggunakan *router* dan *switch* agar data sampai ditujuan dengan benar.

4. *Transport Layer* (*Layer* 4)

   Layer ini mengatur keutuhan data, menerima data dari *session layer* dan meneruskannya ke *network layer* juga sebaliknya. Lapisan ini berfungsi untuk memecah data ke dalam paket-paket data serta memberi nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima.

5. *Session Layer* (*Layer* 5)

   Layer ini menyiapkan saluran komunikasi dan terminal dalam hubungan antar terminal, menkoordinasikan proses pengiriman dan penerimaan serta mengatur pertukaran data. Lapisan ini membangun, mempertahankan dan menyelaraskan interaksi antara sistem komunikasi.

6. *Presentation Layer* (*Layer* 6)

   Layer ini bertugas untuk mendefinisikan format data, menampilkan data dan menangani kompresi dan enkripsi data.

7. *Application Layer* (*Layer* 7)

Layer paling atas ini mengatur interaksi pengguna komputer dengan program aplikasi yang dipakai. *Layer* ini juga mengatur bagaimana aplikasi dapat mengakses jaringan.

2.3.1 Konsep Dasar TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) adalah bukan sebuah protokol tunggal tetapi satu kesatuan protokol dan *utility*. Setiap protokol dalam kesatuan ini memiliki aturan yang spesifik. Protokol ini dikembangkan oleh ARPA (Advance Research Project Agency) untuk departemen pertahanan Amerika Serikat pada tahun 1969. ARPA menginginkan sebuah protokol yang memiliki karakter sebagai berikut:

1. Mampu menghubungkan berbagai jenis sistem operasi.
2. Dapat diandalkan dan mampu mendukung komunikasi kecepatan tinggi.
3. *Routabel* dan *scalable* untuk memenuhi jaringan yang kompleks dan luas.

2.4 TCP/IP dan Subnet Mask

TCP/IP bukanlah sebuah protokol tunggal tetapi satu kesatian protokol dan *utility*. Setiap protokol dalam kesatuan ini memiliki aturan yang spesifik. Protokol ini dikembangkan oleh ARPA (*Advanced Research Projects Agency*) untuk departemen pertahanan Amerika Serikat pada tahun 1969 (Suyanto, 2008 : 8). Sebuah alamat TCP/IP adalah nilai biner berukuran 32 bit yang diberikan kesetiap *Host* dalam sebuah jaringan. Nilai ini digunakan untuk mengenali jaringan dimana *host* tersebut dan mengenali nomor unik *host* bersangkutan dijaringan tertentu. Setiap *host* yang terhubung menjadi satu pada sebuah *Internet* harus memiliki satu alamat unik TCP/IP.

Setiap alamat TCP/IP jatuh pada satu kelas alamat. Kelas mewakili sebuah grup alamat yang segera dapat dikenali komponen *software* sebagai bagian dari sebuah jaringan fisik. Dengan memperhatikan tiga nilai biner pertama, bisa dikatakan bahwa alamat termasuk kedalam kelas yang mana.

Table 2.5 Urutan Dari Alamat IP dan Host

Class	**Jumlah *Host***	**Jumlah Oktet Pertama**

A	16.777.216	1-126
B	16.536	128-191
C	256	192-223

Setiap komputer di sebuah jaringan biasanya ingin mengirim data langsung ke komputer lainnya. Komputer pengiriman harus memastikan bahwa si penerima berada di jaringan yang sama atau di luar itu. Subnet mask digunakan oleh protokol TCP/IP untuk menentukan bahwa *host* yang akan berkomunikasi berada di jaringan local yang sama atau berada di jaringan remote.

Table 2.6 Klasifikasi Subnet Mask

Class	***Subnet Mask***
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

2.5 *Internet Protocol Version* 4 (IPv4)

IP *address* merupakan pengenal atau tanda yang digunakan untuk memberikan alamat pada tiap-tiap komputer dalam suatu jaringan. Format IPv4 *address* adalah 32 bit yang tiap 8 bit dipisahkan oleh tanda titik, model format bakunya adalah seperti xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx, dengan x yang merupakan bilangan biner. Atau dengan bentuk tiga bilangan dengan titik sebagai pemisah seperti xxx.xxx.xxx.xxx, dimana xxx adalah nilai bilangan decimal dari 8 bit bilangan biner.

2.5.1 *Network* ID dan *Host* ID

Dalam 1 alamat ip dibagi menjadi 2 bagian yaitu *network* ID dan *host* ID yang di tandai oleh *subnet mask*. Untuk menandai suatu *network* ID seluruh bit biner pada *subnet mask* diset ke angka 1 dan untuk menandai suatu *host* ID seluruh bit biner pada *subnet mask* diset ke angka 0, apabila alamat 192.168.1.100 dengan subnet 255.255.255.0 yang di binerkan 11111111.11111111.11111111.00000000 menandakan bahwa 192.168.1 merupakan alamat *network* dan .100 merupakan alamat *host*.

Alamat *host* ID dalam 1 *network* ID harus unik atau tidak boleh sama, karena *host* ID harus merupakan satu-satunya alamat yang dimiliki oleh suatu *host* dalam satu jaringan, jika terdapat 2 *host* ID dalam 1 *network* ID maka akan terjadi konflik dalam jaringan tersebut.

2.6 *Internet Protocol version* 6 (IPv6)

Internet protocol version 6 atau IPv6 merupakan *internet protocol* yang dirancang untuk menggantikan IPv4. Dengan jumlah 4.3 miliyar alamat yang dimiliki IPv4 dirasa akan mengalami kekurangan untuk memenuhi jumlah kebutuhan alamat IP. IPv6 mulai dikembangakan pada 14 juli 1999, selain untuk memenuhi jumlah alamat yang diperlukan, IPv6 juga didesain untuk memperbaiki kekurangan-kekurangan yang ada pada IPv4.

2.6.1 Format Pengalamatan IPv6

*S*ecara umum format pengalamatan IPv6 adalah x:x:x:x:x:x:x:x yang dipisahkan oleh tandan *double-colon* (titik dua ‘:’) dan x yang merepresentasikan panjang 16 bit atau 2 octec kemudian diikuti dengan tanda *slash* (/) untuk menandakan prefix jaringan. Untuk menuliskan alamat IPv6, dapat dilakukan dengan cara seperti 2001:0716:0250:0000:0000:0000:0000:4/64 atau dengan bentuk-bentuk yang berbeda seperti 2001:716:250:0:0:0:0:4/64 atau 2001:716:250::4/64.

Tanda /64 menunjukan bahwa 64 bit pertama merupakan alamat jaringan yang digunakan. Penulisan dua *double-colon* menunjukan bahwa bit yang ada didalamnya adalah bit 0 dan penggunaan dua *double-colon* tersebut tidak dapat diberikan sebanyak 2 kali pada 1 alamat IPv6 karena dapat menyebabkan alamat IPv6 menjadi tidak valid.

2.7 *Port*

Dalam protokol jaringan TCP/IP, sebuah *port* adalah mekanisme yang mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi dengan komputer lainnya dan program di dalam jaringan. Menurut Ferry Wisnuargo dan

Prihandoko (2013 : 3), *port* dapat mengidentifikasikan aplikasi dan layanan yang menggunakan koneksi di dalam jaringan TCP/IP. Sehingga, *port* juga mengidentifikasi sebuah proses tertentu dimana sebuah *server* dapat memberikan sebuah layanan kepada klien atau bagaimana klien dapat mengakses sebuah layanan yang ada di dalam *server*. *Port* dapat dikenali dengan angka 16-bit (2 byte) yang disebut dengan *port number* dan diklasifikasi dengan jenis protokol transport apa yang digunakan, ke dalam *port* TCP dan *port* UDP .

Karena memiliki angka 16-bit, maka total maksimum jumlah *port* untuk setiap protokol transport yang digunakan adalah 65536 buah. Dilihat dari penomorannya, *port UDP* dan TCP dibagi menjadi tiga jenis yaitu sebagai berikut.

1. *Well-known Port*

   Yang pada awalnya bekisar antara 0 hingga 255 kemudian diperlebar untuk mendukung antara 0 hingga 1023. *Port number* yang termasuk kedalam *well-known port* selalu merepresentasikan layanan jaringan yang sama, dan ditetapkan oleh *Internet Assigned Number Authority* (IANA), beberapa diantara *port-port* yang berada di dalam *range well-known port* masih belum ditetapkan dan direservasikan untuk digunakan oleh layanan yang akan digunakan di masa depan. *Well-known port* didefinisikan dalam RFC 1060.

2. *Registered Port*

   Merupakan port-port yang digunakan oleh vendor-vendor komputer atau jaringan yang berbeda untuk mendukung aplikasi dan sistem operasi yang mereka buat. *Registered port* juga diketahui dan didaftarkan oleh IANA tapi tidak dialokasikan secara permanen, sehingga vendor lainnya dapat menggunakan *port number* yang sama. *Range registerd port* berkisar dari 1024 hingga 49151 dan beberapa *port* diantaranya adalah *Dynamically Assigned Port.*

3. *Dynamically Assigned Port*

   Merupakan *port-port* yang ditetapkan oleh sistem operasi atau aplikasi yang digunakan untuk melayani *request* dari pengguna sesuai dengan kebutuhan.

Dynamically assigned port berkisar dari 1024 hingga 65536 dab dapat digunakan atau dilepaskan sesuai dengan kebutuhan.

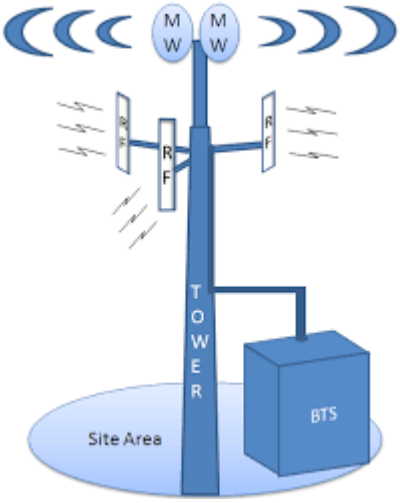
Berikut ini adalah beberapa contoh *port-port* yang sering digunakan yang dapat dilihat pada table 2.7.

Table 2.7 Daftar *port* yang sering digunakan

Port	**Jenis *Port***	***Keyword***	**Digunakan oleh**
20	TCP,UDP	ftp-data	File transfer protocol
21	TCP,UDP	Ftp	File transfer protocol, connection dialog
23	TCP,UDP	telnet	telnet
25	TCP,UDP	Smtp	Simple mail transfer protocol / mail
53	TCP,UDP	Domain	Domain name system server
69	TCP,UDP	Tftp	Trivial file transfer protocol
80	TCP,UDP	www/http	World wide web HTTP
110	TCP,UDP	Pop3	Post office protocol version 3 / post office
123	TCP,UDP	Ntp	Network time protocol
220	TCP,UDP	Imap3	Interactive mail access protocol version 3
443	TCP,UDP	https	Http secure
3306	TCP,UDP	MySQL	MySql Database connection protocol

2.8 *Base Transceiver Station* (BTS)

Base transceiver station atau disingkat BTS adalah sebuah infrastruktur telekomunikasi yang memfasilitasi komunikasi nirkabel antara piranti komunikasi dan jaringan. Piranti komunikasi penerima sinyal BTS bisa telepon, telepon seluler, jaringan nirkabel. Bts mengirimkan dan menerima sinyal radio ke perangkat mobile atau nirkabel dan mengkonversi sinyal-sinyal tersebut menjadi sinyal digital untuk selanjutnya dikirim ke terminal lainnya untuk proses sirkulasi pesan atau data. Model BTS dapat dilihat pada gambar 2.14



Gambar 2.14 Model BTS

2.9 Simulasi

Simulasi berasal dari bahasa inggris, yaitu : *to simulate*, yang artinya untuk memperoleh intisari dari sesuatu tanpa melibatkan kenyataan. Simulasi mempunyai banyak pengertian bila dilihat dari berbagai sudut pandang yang berbeda. Klasifikasi atau pengelompokan dari teknik simulasi dibagi menjadi tiga kelompok, yaitu :

1. Simulasi statis dan dinamis

   Simulasi statis merupakan representasi dari sebuah sistem pada suatu waktu tertentu, atau digunakan pada sistem dimana waktu tidak mempunyai peran. Penerapan paling sederhana pada simulasi ini adalah saat memasukan nilai variable pada suatu rumus untuk memperoleh nilai akhirnya. Sedangkan simulasi dinamis mewakili sistem yang berubah-ubah seturut waktu. Contohnya sistem ban berjalan pada pabrik.

2. Simulasi deterministik dan stochastic

   Simulasi deterministik adalah bila sistem simulasi tidak mempunyai komponen probabilitas. Contoh dari simulasi ini adalah persamaan diferensial yang menghitung reaksi kimia. Simulasi stochastic adalah bila sistem tersebut memiliki beberapa komponen input acak (*random*), hasil akhir dari simulasi

ini juga berupa komponen acak. Kebanyakan sistem antrian (*queueing*) dan persediaan (*inventory*) merupakan model simulasi stochastic.

3. Simulasi kontinyu dan diskrit

   Simulasi kontinyu merupakan model sistem yang kondisi status variablenya berubah-ubah terus menerus sesuai dengan waktu, simulasi model ini melibatkan persamaan diferensial yang merupakan relasi dari tingkat perubahan status variable sistem terhadap waktu, contohnya simulasi aliran air dari beberapa sungai ke danau. Simulasi diskrit adalah dimana status variabelnya berubah seketika pada satu titik waktu yang terpisah, dengan kata lain sistem berubah hanya pada waktu tertentu. Contoh dari simulasi ini adalah sistem antrian, sistem sirkulasi parker.

2.10 Perangkat Jaringan

Perangkat jaringan merupakan alat atau piranti keras atau lunak yang digunakan untuk membangun suatu sistem jaringan. Masing-masing perangkat jaringan memiliki fungsi dan tujuan tersendiri didalam suatu sistem jaringan. Pemilihan perangkat-perangkat jaringan yang diperlukan dapat disesuaikan dengan kebutuhan sistem jaringan yang akan dibangun.

2.10.1 *Switch*

Switch adalah alat penghubung jaringan dengan *forwarding* berdasarkan alamat MAC. *Switch* membagi *collision domain*tetapi tidak membagi *broadcast domain. Switch* bekerja pada *layer 2* pada model OSI (*Data Link Layer*) da nada juga yang bekerja pada *layer* 3 (*Network layer*) pada model OSI. Perbedaan yang mendasar antara *switch layer* 2 dan *switch layer* 3 adalah kemampuan *switch layer* 3 dapat melakuka proses *routing.* Berikut ini adalah standar proses IPS dapat dilihat pada gambar 2.15.



Gambar 2.15 Switch

2.10.2 *Hub*

Hub mirip dengan *switch* yaitu sebagai konsentrator. Namun *hub* tidak "secerdas" *switch*. Jika informasi dikirim ke *host* target melalui hub maka informasi akan mengalir ke semua *host*. Kondisi semacam ini dapat menyebabkan beban *traffic* yang tinggi. Oleh sebab itu, sebuah *hub* biasanya hanya digunakan pada jaringan berskala kecil. Berikut ini adalah hub dapat dilihat pada gambar 2.16.



Gambar 2.16 Hub

2.10.3 Perbedaan Antara *Hub* dan *Switch*

Perbedaan *hub* dan *switch* terletak dari bagaimana *packet data* / informasi yang dikirim kepada mereka diproses. Ketika data masuk atau datang ke *hub*, *hub* akan mengambil data tersebut dan akam mengtransmisikannya ke setiap komputer yang terhubung dalam jaringan. Tetapi lain halnya dengan *switch*, *switch* akan

menerima data tersebut dan hanya mengirimkan ke komputer yang berkepentingan menerima data tersebut.

Penggunaan *switch* akan memotong penggunaan *bandwith* jaringan secara signifikan, terutama bila memiliki jaringan dengan banyak komputer dan semua komputer tersebut sibuk untuk mengirim dan menerima data disaat yang bersamaan.

2.10.4 *Network Interface Card*

Kartu jaringan (NIC) merupakan salah satu dari perangkat keras yang menyediakan media untuk menghubungkan antar komputer. Kebanyakan kartu jaringan adalah kartu internal. Yaitu kartu jaringan yang dipasang pada slot ekspansi didalam komputer. Kartu jaringan yang banyak terpakai saat ini adalah kartu jaringan *Ethernet*, *local talk* konektor, dan kartu jaringan *token ring*. Berikut ini adalah *network interface card* dapat dilihat pada gambar 2.17.



Gambar 2.17 Network Interface Card

2.10.5 *Router*

Router merupakan perangkat keras jaringan yang dapat digunakan untuk menghubungkan beberapa jaringan yang sama atau berbeda. *Router* juga dapat disebut sebagai alat untuk mengirimkan paket data melalui jaringan atau *internet* untuk dapat menuju tujuannya atau proses *routing*. *Routing* itu sendiri terjadi pada lapisan 3 dari OSI *layer*. *Router* terkadang digunakan untuk menghubungkan 2

buah jaringan yang menggunakan media berbeda, seperti dari *Ethernet* menuju ke token *ring*. Berikut ini adalah *router* dapat dilihat pada gambar 2.18.

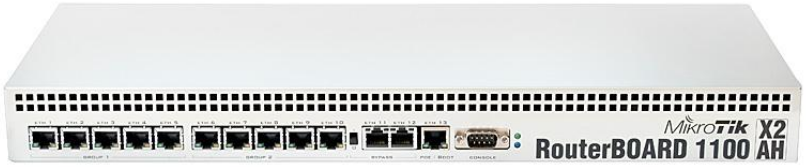


Gambar 2.18 Router

Router memiliki fungsi utama untuk membagi atau mendistribusikan IP *address*, baik secara statis ataupun dinamik kepada semua komputer yang terhubung ke *router* tersebut. Dengan adanya IP *address* unik yang dibagikan oleh *router* ke setiap komputer dapat memungkinkan setiap komputer untuk saling terhubung dan melakukan komunikasi, baik pada jaringan local ataupun *internet.*

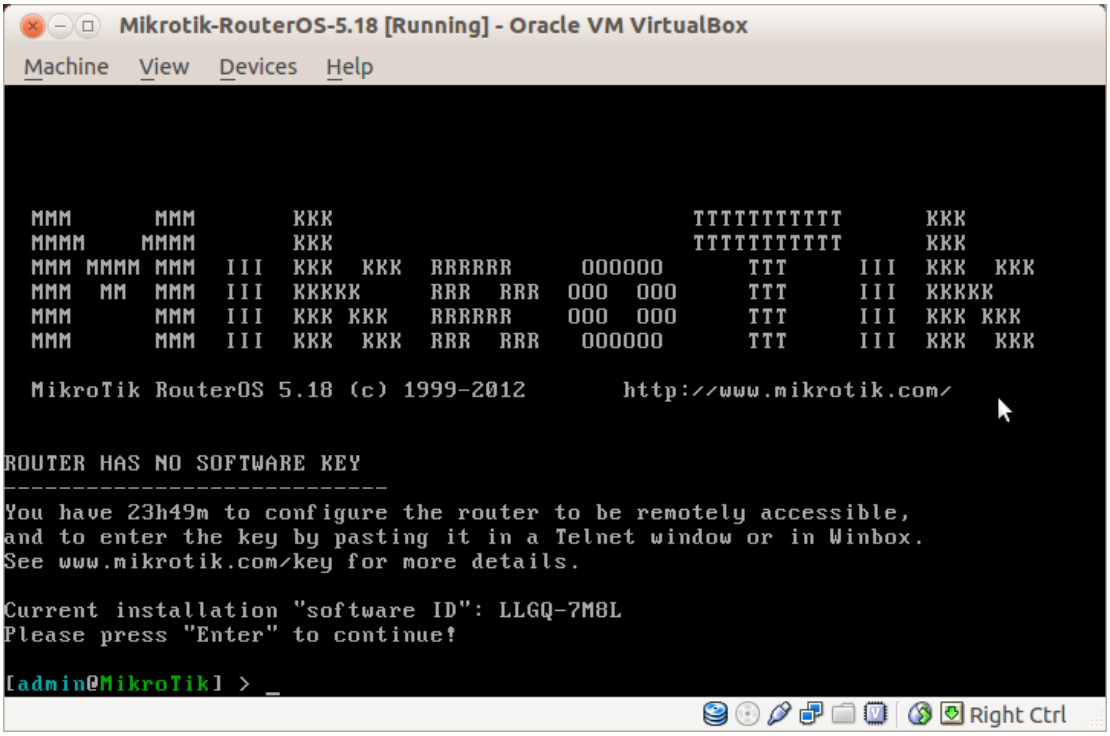
2.10.6 Mikrotik *Router*-OS

Mikrotik *router*-OS adalah sebuah sistem operasi termasuk didalamnya perangkat lunak yang dipasang pada suatu komputer sehingga komputer tersebut dapat berperan sebagai jantung *network*, pengendali atau pengatur *traffic data* antar jaringan. Intinya mikrotik adalah salah satu sistem operasi khusus untuk *router*, mikrotik dikenal sebagai salah satu *router* OS yang handal dan memiliki banyak sekali fitur untuk mendukung kinerja jaringan. Berikut ini adalah perangkat mikrotik dapat dilihat pada gambar 2.19.



Gambar 2.19 Perangkat Mikrotik

Fungsi utama mikrotik adalah menjadikan sebuah komputer menjadi sebagai *network router*, selain itu mikrotik juga memiliki fungsi untuk menjalankan aplikasi, meliputi manajemen *bandwith, firewall, wireless access point, backhaul link, virtual private network* (VPN), dan lainnya. Berikut ini tampilan awal mikrotik *router*-OS dapat dilihat pada gambar 2.20.

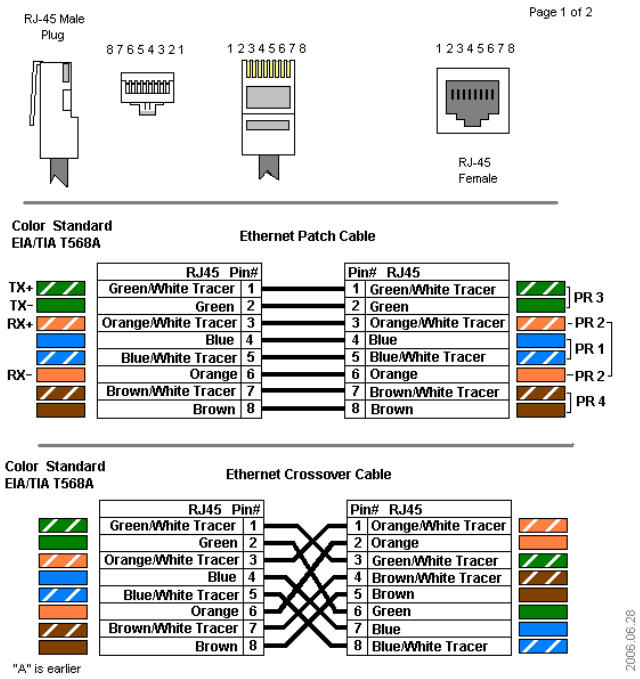


Gambar 2.20 Tampilan Mikrotik Router-OS

2.10.7 Konektor RJ45

Konektor RJ45 adalah penghubung atau konektor kabel *Ethernet* yang digunakan dalam jaringan. Konektor RJ45 biasanya terdapat pada ujung kabel UTP (*unshielded twisted pair*) yang menghubungkan ke pemancar penerima

(*transciever*) atau kartu jaringan.dalam meng konfigurasikan konektor RJ45 terdapat dua konfigurasi yaitu *straight through* dan *cross over*. Berikut ini adalah Konektor RJ45 dan susunan Konfigurasi kabel UTP dapat dilihat pada gambar 2.21.



Gambar 2.21 Konektor RJ45 & Susunan Konfigurasi Kabel UTP

Konfigurasi *straight through* menghubungkan antara pc dengan perangkat jaringan yang berbeda seperti pc ke *hub* ataupun *hub* ke *router*. Sedangkan konfigurasi *cross over* menghubungkan perangkat jaringan dengan perangkat yang sama atau sejenis seperti pc ke pc atau *router* ke *router.*

2.11 Keamanan Jaringan Komputer

Suatu komputer yang terhubung dengan sebuah jaringan komputer mempunyai ancaman keamanan yang lebih besar dibandingkan dengan komputer yang tidak terhubung dengan jaringan komputer. Keamanan jaringan komputer biasanya berbanding terbalik dengan akses jaringan dimana akses jaringan semakin mudah maka keamanan jaringan komputer semakin rentan tetapi apabila

jaringan komputer semakin aman maka akses jaringan akan semakin tidak nyaman (sukirmanto, 2013 : 4).

Keamanan atau *security* adalah mekanisme dan teknik untuk melindungi sesuatu yang dapat berupa data atau informasi didalam sistem. Pada dasarnya *security* adalah sistem yang digunakan untuk melindungi sistem dalam suatu jaringan keamanan agar tetap terjaga.

Pada dasarnya keamanan adalah proteksi perlindungan atas sumber-sumber fisik dan konseptual dari bahaya alam dan manusia. Keamanan terhadap sumber konseptual meliputi data dan informasi. Namun disamping itu banyak sekali upaya manusia untukmenjaga keamanan atas sumber fisik maupun konseptual dengan berbagai macam cara.

Keamanan haruslah memiliki beberapa bagian penting didalamnya, yaitu :

1. *Availability*

   Availability adalah menjaga akses untuk masuk ke dalam informasi.

2. *Confidentianlity*

   Confidentianlity adalah menjaga informasi secara rahasia dan hanya dapat dibuka oleh yang memiliki hak resmi untuk mengaksesnya.

3. *Anonymity*

   Anonymity adalah menyembunyikan identitas dari entitas yang terlibat dalam prosesnya.

4. *Privacy*

   Privacy adalah memiliki hak dan kewajiban yang mengatur akusisi rahasia pribadi, dan informasi rahasia lainnya.

5. *Identification and authentication*

   Identification and authentication adalah cara mengetahui identitas user dalam jaringan komputer.

Tujuan keamanan dapat dicapai dengan suatu metode keamanan jaringan yang dapat melindungi sistem baik dari dalam maupun luar jaringan. Namum bukan hanya dapat melindungi tetapi harus dapat bertindak apabila terjadi suatu serangan yang terjadi didalam sistem ataupun jaringan.

2.12 Tahapan-Tahapan Dalam Pola Serangan Jaringan

Sebelum seorang *hacker* atau penyerang melakukan sebuah serangan pada targetnya, terdapat beberapa tahapan-tahapan yang harus diketahui oleh seorang penyerang sebelum merencanakan sebuah serangan yaitu sebagai berikut (Indrajit, 2013 : 2-4).

1. *Reconnaissance*

   Reconnaissance adalah suatu tahap persiapan dimana penyerang berusaha mencari informasi sebanyak-banyaknya mengenai target sasaran sistem yang akan diserang sebelum rangkaian proses penyerangan dilaksanakan. Terdapat dua jenis model *reconnaissance* yang dikenal, yaitu yang bersifat pasif dan aktif. Dikatakan model pasif apabila tidak ada interaksi langsung antara pihak penyerang dengan target sasaran, sementara apabila dikatakan model aktif, jika dilakukan aktivitas interaksi secara langsung dengan sistem.

2. *Scanning*

   Scanning merupakan sebuah proses dimana penyerang dengan menggunakan berbagai alat atau *tools* berusaha mencari celah masuk untuk melakukan serangan. Seorang penyerang melalui aktivitas ini berusaha mencari lubang-lubang kerawanan tempat serangan masuk, biasanya yang akan di-*scan* pertama kali adalah *port* dalam sistem komputer, atau pemetaan jaringan (*network mapping*), atau melalu pencarian kerapuhan (*vulnerability scanning*)

3. *Gaining Access*

   Gaining access adalah proses dimana usaha penetrasi aktif mulai dilaksanakan. Pada dasarnya yang dilakukan oleh seorang penyerang adalah melakukan eksploitasi terhapad kelemahan, kerawanan, dan kerapuhan yang terdapat pada sistem. Cara mendapatkan akses yang dimaksud sangatlah beraneka ragam, tergantung karakteristik dan hasil yang didapat dari proses *scanning* yang sebelumnya telah dilakukan. Tahapan *gaining access* lainnya adalah melakukan apa yang dinamakan sebagai *session hijacking* atau melakukakan pembajakan hak akses seseorang oleh penyerang, sehingga penyerang dapat masuk kedalam sistem yang bukan teritorinya.

4. *Maintaining Access*

Maintaining access adalah sebuah periode dimana setelah penyerang berhasil masuk kedalam sistem, penyerang berusaha untuk tetap bertahan memperoleh hak akses tersebut. Pada saat inilah diistilahkan bahwa sistem yang diserang telah berhasil diambil alih oleh penyerang. Ketika tahap ini berlangsung, kendali sepenuhnya berada ditangan seorang penyerang dan penyerang tersebut dapat melakukan apa saja yang dingininkannya seperti melakukan perubahan pada tampilan utama *website,* memanipulasi informasi, mengacaukan konfigurasi, dan lainya.

5. *Covering Tracks*

   Covering tracks adalah tahapan akhir yang dianggap paling sulit dan sering dilupakan oleh seorang penyerang. Dikatakan sulit karena selain banyak hal yang harus dilakukan penyerang dan cukup memakan waktu, selain membutuhkan sumber daya yang tidak sedikit, diperlukan juga pengetahuan dan keahlian yang sangat baik dari seorang penyerang. Banyak jenis penghapusan jejak yang dikenal di kalangan penyerang seperti *steganography*, *tunneling, log files altering,* dan lainnya.

2.13 *Denial of Service* (DoS) *Attack*

Denial of Service (*DoS*) *attack* adalah tindakan untuk membuat sistem, servis, jaringan yang digunakan oleh *client* dan *server* tidak dapat berfungsi atau menurunkan performa secara signifikan dengan meng-*overloading* sumberdaya tersebut (Ramesh , 2014 : 117).

Dalam kebanyakan kasus, jika penyerang tidak dapat mendapatkan hak akses legal pada sistem yang ditarget, pada akhirnya penyerang akan melakukan serangan *DoS* dengan mencoba untuk membuat sumberdaya (*resource*) tersebut *crash*. Dampak dari serangan *DoS* dapat menyebabkan kehilangan finansial terutama pada *website* atau *server* yang bergerak dalam aktivitas bisnis atau *e-commerce*. Terkadang dampaknya pun mempengaruhi perusahaan atau organisasi yang pernah menjadi korban serangan, sehingga mereka akan kehilangan kepercayaan untuk menggunakan servis itu kembali.

Tujuan dari *DoS attack* tidak untuk mendapatkan hak akses yang legal untuk masuk ke dalam sistem, melainkan untuk mencegah user untuk mengakses servis. Agar serangan tersebut dapat berjalan, penyerang akan melakukan berbagai cara antara lain :

1. Mencoba untuk membajiri *traffic* data pada jaringan yang ditargetkan, sehingga membuat jaringan tidak dapat memberi user konektivitas.
2. Mencoba untuk mengganggu koneksi antara dua mesin dalam suatu jaringan yang berdampak servis tidak dapat dipergunakan.
3. Mencoba untuk mencegah beberapa individu dari menakses servis atau mengganggu salah satu servis yang spesifik agar tidak dapat diakses.

2.14 *Scanning*

A *Scanning* adalah proses dimana penyerang akan menggali informasi tentang alamat IP target, sistem operasi yang digunakan, arsitektur jaringan yang digunakan, dan servis yang sedang berjalan pada komputer pun dapat di peroleh. Tidak seperti *footprinting* yang hanya mengali informasi secara pasif dari pihak ketiga dalam berbagai sumber, *scanning* secara aktif berhadapan dengan target untuk menperoleh informasi (Ramesh, 2014 : 51).

Dalam melakukan *scanning*, terdapat beberapa tipe *scanning* diantaranya sebagai berikut :

1. *Port scanning*

   Port scanning meliputi pengiriman beberapa pesan pada komputer target untuk memperoleh tipe servis jaringan apakah yang sedang berjalan pada jaringan tersebut. Dikarenakan servis tersebut berkaitan dengan nomor *port*, maka dengan melakukan *port scan* pada target akan menampilkan seluruh *port* yang terbuka.
2. *Network scanning*

   Network scanning adalah suatu prosedur untuk mengidentifikasi *host* yang sedang aktif pada jaringan target yang bertujuang untuk melakukan serangan pada target ataupun untuk menilai keamanan jaringan. Dalam langkah ini akan memungkinkan penyerang untuk membuat daftar *host-host* yang dapat

diserang secara langsung atau menggunakan *host* tersebut untuk menyerang *host* lain secara tidak langsung.

3. *Vulnerability scanning*

   Vulnerability scanning berhubungan dengan penggunaan *tools* otomastis yang dikenal dengan *vulnerability scanner* yang secara otomatis mengidentifikasi kelemahan-kelemahan keamanan pada sistem komputer di jaringan tersebut. *Tools* ini akan melacak target untuk mencari tahu celah keamanan manakah yang dapat di eksploitasi.

2.15 *Malware*

Malware adalah perangkat lunak (*software*) yang diciptakan untuk menyusup atau merusak sistem komputer. Perangkat lunak dianggap sebagai perangkat perusak berdasarkan maksud yang terlihat dari pencipta dan bukan berdasarkan ciri-ciri tertentu. *Malware* mencakup virus komputer, *worm, Trojan horse, rootkit, spyware, adware,* dan perangkat lunak lainnya yang berniat jahat dan tidak diinginkan. *Malware* tidak sama dengan perangkat lunak cacat (*defective software*), yaitu, perangkat lunak yang mempunyai tujuan sah tetapi berisi *bug* yang berbahaya.

2.16 *Honeypot*

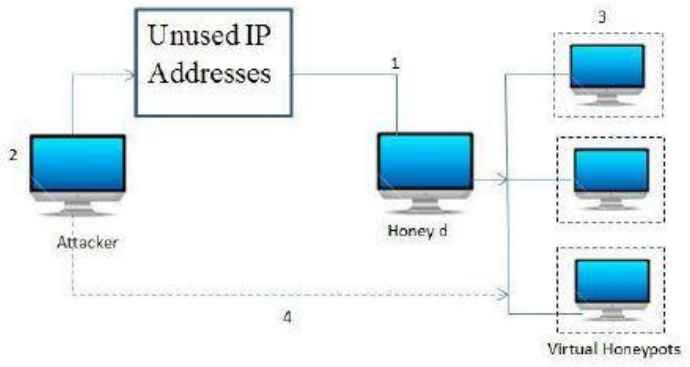
Honeypot merupakan teknologi keamanan yang bertujuan mengidentifikasi, mencari celah keamanan dan berkomprimasi aktif ketika terjadi aktifitas penyusupan keamanan teknologi informasi (Anggeriana , 2012 : 3). Sedangkan menurut Muhammad Arief *honeypot* merupakan suatu alat untuk membantu *administrator* jaringan mendapatkan informasi tentang penyerangan sistem dan membantu *administrator* jaringan untuk melihat, menganalisa, serta mempelajari aktifitas-aktifitas yang mempunyai kecenderungan membahayakan sistem. Oleh karena itu dapat disimpulkan *honeypot* adalah suatu teknologi atau alat yang berfungsi untuk mencari informasi tentang aktivitas-aktivitas yang dapat membahayakan atau menyerang suatu sistem dengan membuat *honeypot* itu

sendiri sebagai *dummy* penyerangan, agar dapat mencari informasi serta menganalisa tindak-tindak serangan yang terjadi.

Secara fungsional dari teknologi *honeypot* terdiri dalam banyak variasi dan umumnya terbadi dalam dua kategori yaitu *honeypot* interaksi rendah (*low interaction honeypot*) dan *honeypot* berinteraksi sensitif (*high interaction honeypot*).

2.16.1 *Low Interaction Honeypot*

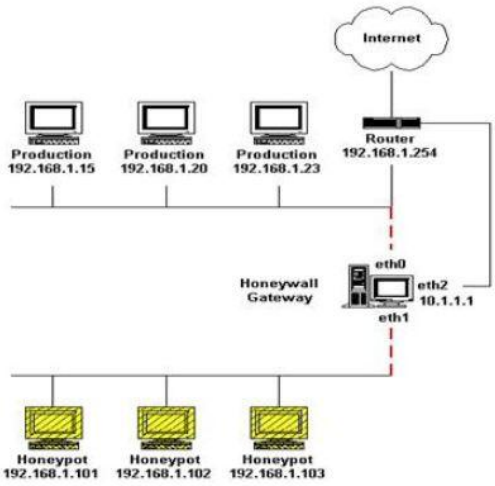
Low interaction honeypot adalah sebuah *honeypot* yang dedesain untuk menyerupai jaringan infrastruktur pada server asli. Penyerang hanya mampu memeriksa dan terkoneksi ke satu atau beberapa *port*. Contoh sederhana dari *honeypot* jenis ini dalah pembuatan sebuah *service* yang mendengarkan dan mencatat setiap koneksi yang terjadi pada sebuah *port*. Pada *low interaction honeypot* penyerang tidak berinteraksi dengan sistem operasi secara langsung. Hal ini akan mengurangi resiko karena tidak akan ada sistem yang akan diambil alih. *Honeypot* jenis ini bersifat koneksi satu arah karena dari satu sisi (sisi *honeypot)* hanya mendengarkan dan mencatat koneksi yang terjadi tanpa memberika balasan kepada koneksi tersebut. Jika pada *honeypot* disediakan program yang dapat mengemulasikan suatu layanan, maka *intruder* akan menerima respon seperti hanya respon yang diberikan oleh layanan aslinya. Berikut ini adalah arsitektur *low interaction honeypot* dapat dilihat pada gambar 2.22.



Gambar 2.22 Arsitektur Low Interaction Honeypot

2.16.2 *High Interaction Honeypot*

Pada *high interaction honeypot* terdapat sistem operasi dimana penyerang dapat berinteraksi secara langsung dan tidak ada batasan yang membatasi interaksi tersebut. Dengan dihilangkannya batasan-batasan tersebut, maka tingkat resiko yang dihadapi semakin tinggi karena penyerang dapat memiliki akses *root*. Pada saat yang sama, kemungkinan pengumpulan informasi semakin meningkat dikarenakan kemungkinan serangan yang tinggi. Informasi tersebut dapat berupa pola serangan, program yang digunakan, motivasi dan lain-lain. *High interaction honeypot* harus diawasi secara terus menerus, pengawasan ini diperlukan karena apabila *high interaction honeypot* telah diambil alih dan dimanfaatkan oleh penyerang maka *honeypot* tersebut dapat menjadi ancaman bagi jaringan yang ada. Berikut ini adalah arsitektur *high interaction honeypot* dapat dilihat pada gambar 2.23.



Gambar 2.23 Arsitektur High Interaction Honeypot

2.16.3 Perbandingan *Low Interaction* dan *High Interaction Honeypot*

Masing-masing kategori dari *honeypot* tersebut memiliki keunggulan, keunggulan yang dimiliki *low interaction honeypot* yaitu, skala pembangunan dan skala pemeliharaan kategori sederhana, karena *honeypot low interaction* didesain hanya untuk mengumpulkan informasi penyerang dan menjadi target utama penyerang sehingga dampak resiko minimum dari sistem utama dapat dicapai.

Sedangkan keunggulan dari *high interaction honeypot* yaitu, merekam dan mengumpulkan informasi lebih spesifik dari *low interaction honeypot*, *behavior* penyerang dan penelusuran jaringan protocol secara spesifik dari penyerang saat terjadinya insiden penyerangan atau penyusupan. Berikut ini adalah perbandingan antara *low interaction* honeypot dan *hight interaction honeypot* dapat dilihat pada gambar 2.24.

Low Interanction	**High Interaction**
Mensimulasi sistem operasi dan servis	Sistem operasi dan servis sungguhan tanpa emulasi
• Mudah diinstall dan deploy. Konfigurasi *software* biasanya sederhana. • Resiko minimal, emulasi mengontrol apa yang bias dilakukan penyusup/ • Menangkap jumlah informasi terbatas.	• Menangkap informasi lebih banyak • Bisa cukup kompleks • Resiko tinggi, penyusup bisa berinteraksi dengan sistem operasi sungguhan.

Gambar 2.24 Perbandingan Low Interaction dan High Interaction Honeypot

2.16.4 Tipe *Honeypot*

Honeypot terbagi menjadi dua tipe, yaitu sebagai berikut :

1. *Production honeypot*

   Production honeypot mudah digunakan, hanya menangkap informasi terbatas, dan digunakan terutama oleh perusahaan atau korporasi. *Production honeypot* ditempatkan dalam jaringan produksi dengan *server* produksi lainnya oleh organisasi utuk memperbaiki keadaan keamanan secara keseluruhan. Mereka memberika sedikit informasi tentans serangan atau penyerangan dari *research honeypot.* Tujuan dari *production honeypot* adalah untuk membantu mengurangi resiko dalam sebuah organisasi. Sebuah *honeypot* memberikan nilai tambah langkah-langkah keamanan dair suatu organisasi.

2. *Research honeypot*

   Research honeypot dijalankan oleh para sukarelawan, penelitian nirlaba organisasi atau lembaga pendidikan untuk mengumpulka informasi tentang motif atau taktik *blackhat* penargetan komunikasi jaringan yang berbeda. *Honeypot* ini tidak menambah nilai langsung ke organisasi tertentu. Sebaliknya mereka digunakan untuk meneliti organisasi menghadapi ancaman, dan belajar bagai mana lebih melindungi terhadap ancaman-ancaman. Informasi ini kemudian digunakan untuk melindungi ancaman-ancamana tersebut.

2.16.5 Macam-macam Sistem Honeypot

Terdapat banyak sekali sistem-sistem *honeypot* baik untuk sistem operasi windows, linux, maupun yang dapat melakukan *cross-platform*, berikut ini adalah beberapa macam sistem *honeypot* yang sering digunakan :

1. Symantec Decoy Server

   Ini adalah *honeypot* bertipe *high interaction* sistem deteksi intrusi yang dapat mendeteksi, mengadung dan monitor akses tidak sah dan penyalah gunaan sistem time nyata. Symantec decoy server sendiri bertipe *host* dan *network based IDS*, selain itu Symantec decoy server dapat mengalihkan penyerang dari sumberdaya kunci dan juga memberikan deteksi awal baik serangan internal maupun eksternal.

2. Kippo SSH

   Kippo SSH adalah *honeypot* bertipe *medium interaction* yang didesain untuk merekam aktivitas dari penyerang, seperti *brute force* dan paling penting adalah melakukan interaksi pada penyerang dengan cara meniru *service* ssh.

3. HoneyD

   Honey adalah *low interaction honeypot* client yang menciptakan virtual host dalam jaringan. *Honeypot* ini dapat dikonfigurasi untuk bertindak seperti sebuah sistem operasi yang nyata. Pada saat yang sama dapat mengkonfigurasi sistem-sistem operasi untuk mengaktifkan layanan seperti FTP, HTTP, Telnet, dan lain-lain.

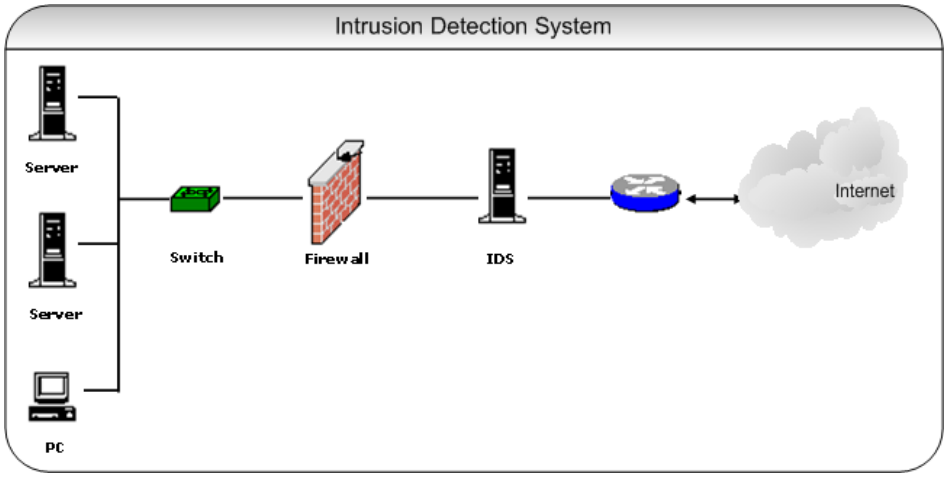
Dari setiap sistem-sistem *honeypot* itu pasti memiliki kelebihan dan kekurangan masing-masing yang dapat dilihat pada table 2.8.

Table 2.8 Kelebihan dan Kekurangan Dari Sistem *Honeypot*

Honeypot	**Kelebihan**	**Kekurangan**
Symantec Decoy Server	1. Tampilan sistem sudah berbasis GUI. 2. *Alert* dan *log* dapat langsung dikirim via *e-mail* ke *administrator*.	1. Sistem *honeypot* ini berbayar. 2. Hanya dapat di install pada platform linux.
Kippo SSH	1. Bekerja tidak hanya sebagai pendeteksi serangan, namun bekerja layaknya SSH *server*.	1. Resiko yang besar untuk digunakan sebagai lompatan untuk menyerang sistem yg lain.
HoneyD	1. Dapat digunakan baik di platform windows dan linux. 2. Dapat mensimulasikan ribuan *virtual host* pada saat yang sama. 3. Dapat memantau semua port berbasis UDP dan TCP.	1. Keterbatasan lingkup pandangan serangan, karena hanya bisa menangkap aktivitas yang diarahkan pada honeyd.

2.17 *Intrusion Detection System* (IDS)

Intrusion detection system (IDS) adalah sebuah sistem yang melakukan pengawasan terhadap *traffic* jaringan dan pengawasan terhadap kegiatan-kegiatan yang mencurigakan didalam sebuah sistem jaringan (hermawan, 2012 : 6). Jika ditemukan kegiatan-kegiatan yang 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/anomaly melalui aksi penolakan seorang user atau alamat IP. Berikut ini contoh arsitektur *intrusion detection system* dapat dilihat pada gambar 2.25.



Gambar 2.25 Arsitektur Intrusion Detection System (IDS)

Istilah *intrusion detection system* merupakan visi suatu alat yang diletakan pada perimeter jaringan untuk memberitahu akan adanya penyusup. IDS juga mempunyai peranan penting untuk mendapatkan arsitektur *defence-in-depth* (pertahanan yang mendalam) dengan melindungi akses jaringan internal sebagai tambahan dari perimeter *defence*. Banyak dari fungsi jaringan internal yang bisa dimonito demi keamanan yang maksimal. Terdapat dua jenis *intrusion detection system*, yaitu :

1. *Network-based intrusion detection system* (NIDS)

   Network-based intrusion detection system adalah semua *traffic* yang mengalir ke sebuah jaringan akan dianalisis untuk mencari apakah ada percobaan serangan atau penyusupan ke dalam sistem jaringan. Kelemahan dari NIDS adalah bahwa NIDS agak rumit diimplimentasikan dalam sebuah jaringan yang menggunakan *switch ethernet*, meskipun beberapa vendor *switch ethernet* telah menerapkan fungsi IDS di dalam *switch* buatannya untuk memonitor port atau koneksi.

2. *Host-based intrusion detection system* (HIDS)

   Host-based intrusion detection system adalah aktivitas sebuah *host* jaringan individual akan dipantau apakah terjadi sebuah percobaan serangan atau penyusupan kedalamnya atau tidak. Tipe ini bekerja untuk melindingi sisi *host*, dengan cara meletakan *rules* yang lebih spesifik sesuai dengan kondisi *host.*

2.17.1 Cara Kerja IDS

IDS bekerja dengan beberapa cara, terdapat dua cara kerja yang dapat dilakukan oleh IDS yaitu *knowledge-based* dan *behavior-based.* penjelasan cara kerja keduanya adalah sebagai berikut :

1. *Knowledge-based* (*missuse detection*)

   Knowledge-based pada IDS adalah cara kerja dengan mengenali adanya penyusupan atau serangan dengan cara menyadap paket data, kemudian membandingkannya dengan *database rule* pada IDS tersebut. *Databse rule* tersebut berisi *signature-signature* paket serangan. Jika pola paket data tersebut terdapat kesamaan dengan *rule* yang terdapat pada *database rule* IDS, maka paket data tersebut dianggap sebagai sebuah serangan dan demikian pula sebaliknya, jika paket data tersebut tudak memiliki kesamaan, maka paket data tersebut tidak akan dianggap sebuah serangan

2. *Behavior-based* (*anomaly based*)

   Behavior-based adalah cara kerja IDS dengan mendeteksi adanya serangan atau penyusupan dengan mengamati adanya keanehan-keanehan pada sistem atau adanya keanehan dari kondisi pada saat sistem berjalan dengan normal. Sebagai contoh : adanya penggunaan memori yang berlebihan atau melonjak secara terus menerus atau terdapat koneksi secara parallel dari satu alamat IP dalam jumlah yang banyak di dalam waktu yang bersamaan. Kondisi tersebut dianggap keanehan yang selanjutnya oleh *anomaly based* ini dianggap sebagai sebuah serangan.

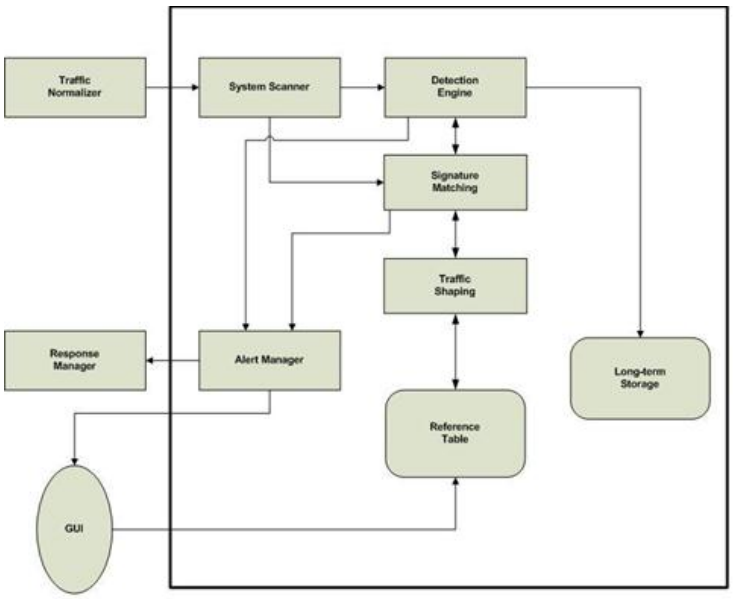
2.18 *Intrusion Prevention System* (IPS)

Intrusion prevention system (IPS), juga dikenal sebagai sistem deteksi intrusi dan pencegahan merupakan peralatan pengamanan jaringan yang bekerja memonitor kegiatan pada suatu jaringan dari aktivitas yang mengancamkegiatan sebuah jaringan (hermawan, 2012 : 7). *Intrusion prevention system* dianggap sebagai pengembangan dari *intrusion detection system*, keduanya bekerja untuk memonitor *traffic* jaringan dari aktivitas yang membahayakanjaringan. Tidak seperti IDS, karena IPS mampu mencegah serangan yang datang dengan bantuan

administrator secara minimal atau bahkan tidak sama sekali. Secara logik IPS akan menghalangi suatu serangan sebelum terjadi eksekusi pada memori.

IPS didesain sebagai sebuah *embedded system* yang membuat banyak *filter* untuk mencegah bermacam-macam serangan seperti *worm, virus, denial of service (DoS)* dan *traffic* berbahaya lainnya agar suatu jaringan tidak menderita banyak kerugian. Penggunaan *multiple filter* pada IPS mebuatnya secara signifikan lebih efektif ketika menginspeksi, mengidentifikasi dan memblokir serangan berdasarkan urutan waktu. Bentuk umum yang dapat menjelaskan IPS adalah : IPS = IDS + *Firewall*.

Berikut ini adalah standar proses IPS dapat dilihat pada gambar 2.26.



Gambar 2.26 Standar Proses IPS

Intrusion prevention system (IPS) bertugas untuk memonitor paket-paket data jaringan dari adanya aktivitas yang mencurigakan dan mencoba untuk melakukan aksi-aksi tertentu menggunakan kebijakan-kebijakan (*policy*) tertentu. IPS akan mengirimkan sebuah peringatan *(alert)* kepada *network* atau *system administrator* ketika suatu hal yang mencurigakan terdeteksi, memungkinkan *administrator* dapat memilih sebuah tindakan untuk diambil ketika terjadi sebuah *event*. IPS juga dapat memonitor seluruh jaringan, *wireless network protocol, network behavior* dan *traffic* sebuah komputer. Setiap IPS menggunakan metode

deteksi tertentu untuk menganalisis resiko. Terdapat 4 jenis *intrusion prevention system*, yaitu :

1. *Network-based intrusion prevention system*

   Network-based intrusion prevention system memiliki fungsi yaitu untuk memantau seluruh jaringan dari *traffic* yang mencurigakan dengan menganalisis aktivitas protokol.

2. *Wireless intrusion prevention system*

   Wireless intrusion prevention system memiliki fungsi yaitu untuk memonotor jaringan nirkabel dari *traffic* yang mencurigakan dengan menganalisa protokol dari jaringan nirkabel.

3. *Network behavior analysis*

   Network behavior analysis (NBA) memiliki fungsi memeriksa *traffic* jaringan untuk mengidentifikasi ancaman yang menghasilkan arus *traffic* yang tidak biasa, seperti *Distributed Denial of Service attacks* (DDoS attack), beberapa bentuk *malware*, dan pelanggaran kebijakan.

4. *Host-based intrusion prevention system*

   Host-based intrusion prevention system yaitu perangkat lunak yang di *install* untuk memonitor sebuah *host* untuk kegiatan yang mencurigakan dengan menganalisis peristiwa yang terjadi didalam *host* tersebut.

Ada perbedaan yang mendasar antara *Intrusion Detection System* (*IDS*) dan *Intrusion Prevention System* (IPS) seperti pada tabel 2.9 dibawah ini :

Table 2.9 Perbedaan IDS dan IPS

IDS	IPS
Install pada segmen jaringan (NIDS) dan pada host (HIDS)	*Install* pada segmen jaringan (NIPS) dan pada *host* (HIPS)
Berada pada jaringan sebagai sistem yang pasif	Berada pada jaringan sebagai sistem yang aktif
Tidak bisa menguraikan *traffic* enkripsi	Lebih baik untuk melindungi aplikasi
Manajemen kontrol terpusat	Manajemen kontrol terpusat
Baik untuk mendeteksi serangan	Ideal untuk membloking perusakan web
Alerting (reaktif)	*Blocking* (proaktif)

2.19 Sistem Operasi Komputer

Sistem operasi komputer merupakan bagian yang terpenting dalam suatu sistem komputer, merupakan jantung dari komputer itu sendiri. Sistem operasi komputer merupakan program yang figunakan untuk mengendalikan sistem *input/output*, seperti *keyboard, diskdrive*, dan juga digunakan untuk membaca dan menjalankan program atau aplikasi. Sistem operasi juga merupakan sekumpulan mekanisme dan kebijakan yang membantu mendefinsikan pengendalian sumberdaya yang di-*share* (Sidik, 2004 : 1).

Sistem operasi dibedakan berdasarkan kemampuan untuk menangani proses dan pemakaian komputer pada saat yang bersamaan menjadi sistem operasi *standalone* dan *multiuser*. Pada sistem operasi *standalone*, komputer hanya dapat digunakan untuk satu pemakai saja pada saat yang bersamaan, tetapi proses yang di tanganinya bias lebih dari satu pada saat yang bersamaan. Sedangkan dengan sistem operasi *multiuser*, komputer dapat digunakan untuk melayani proses dari banyak pemakai pada saat yang bersamaan.

2.19.1 Linux

Linux adalah varian dari sistem operasi Unix. Linux dikembangkan oleh Linus Torvlads, pada saat itu masih merupakan mahasiswa di Finlandia, tahun 1990, yang berkeinginan memiliki sistem operasi sendiri untuk komputernya yang bias dengan mudah dimodifikasi, seolah program komputer biasa (Sidik, 2004 : 11). Berikut ini adalah logo sistem operasi linux dapat dilihat pada gambar 2.27.



Gambar 2.27 Logo Sistem Operasi Linux

Versi awal dari linux kemudian disebarluaskan secara bebas, untuk mendapatkan masukan dari pengguna dan pengembang *software* yang memang mencari sistem operasi alternatif. Sambutan yang sangat antusias diterima dari komunitas yang mencoba sistem operasi baru ini, ternyata kemudian mereka menyokong dan membantu mengembangkan sistem operasi ini.

2.19.1.1 Debian

Debian pertama kali diperkenalkan oleh Ian Murdock, seorang mahasiswa dari universitas Purdue, Amerika Serikat. Debian dikenal karena pilihannya yang beragam. Rilis stabil saat ini memuat lebih dari 29 ribu paket perangkat lunak untuk 9 arsitektur komputer. Debian menggunakan kernel linux dan juga menggunakan 2 kernel FreeBSD. Arsitektur komputer ini mulai dari Intel/AMD yang umumnya ditemukan di sistem *embedded* dan *server mainframe* IBM Zseries. Berikut ini adalah logo sistem operasi debian dapat dilihat pada gambar 2.28.



Gambar 2.28 Logo Sistem Operasi Debian

Fitur yang menonjol dari debian adalah APT sistem pengaturan paket, repository dengan jumlah paket yang banyak, kebijakan paket yang ketat, dan kualitas rilis, begitupun untuk penghapusan paket. Standar instalasi debian menggunakan GNOME *desktop environtment.*

2.19.1.2 Ubuntu

Salah satu komponen utama dalam jaringan adalah sistem operasi komputer. Sistem operasi ini berfungsi untuk mengatur komunikasi jaringan berupa dokumen, *printer*. *Scanner* dan perangkat-perangkat lainnya. Sistem operasi dapat membedakan arsitektur dalam pemanfaatan fasilitas-fasilitas yang ada di jaringan. Misalkan membedakan perangkat jaringan seperti *Ethernet* dan *token ring* atau arsitektur lainnya. Berikut ini adalah logo sistem operasi ubuntu dapat dilihat pada gambar 2.29.



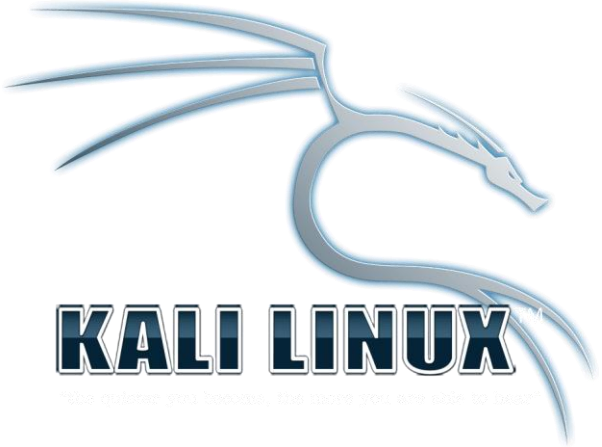
Gambar 2.29 Logo Sistem Operasi Ubuntu

Sistem operasi Ubuntu merupakan turunan dari sistem operasi linux yang lain, yakni Debian. Ubuntu itu sendiri dibuat dengan tujuan selalu gratis tanpa adanya biaya lisensi, bersifat *open source*, dan siap untuk dipergunakan dalam

kondisi yang stabil. Ubuntu didukung oleh perusahaan bernama canonical, Ltd yang memiliki tujuan untuk membantu perkembangan, distribusi, dan promosi dari produk-produk yang bersifat *open source*. Perusahaan ini bermarkas di eropa dan dipimpin oleh seseorang bernama Mark Shuttleworth. Sejak pertama kali diluncurkan, Ubuntu mendapatkan perhatian yang sangat besar dari pengguna linux yang lain. Hai ini disebabkan karena kestabilan yang dimiliki oleh Ubuntu itu sendiri. Selain itu kenyamanan dan kemudahan yang dimiliki Ubuntu menjadi daya tarik yang besar bagi pengguna linux di seluruh belahan dunia.

2.19.1.3 Kali Linux

Kali linux adalah salah satu distribusi linux tingkat lanjut untuk melakukan *penetration testing* dan audit keamanan. Kali linux merupakan pembangunan kembali BackTrack linux secara sempurna, mengikuti sepenuhnya kepada standar pengembangan debian. Semua infrastruktur baru telah dimasukan kedalam satu tempat, *tools*, dan lainnya. Pencipta dari backtrack telah menetapkan untuk berkembang dikarenakan dari tantangan dari *cyber security* dan dibutuhkannya pondasi modern testing yang baru. Kali linux lahir dan dirilis pada 13 maret 2013, kali linux memiliki dasar dari debian dan *FHS-compliant filesystem.* Berikut ini adalah logo sistem operasi kali linux dapat dilihat pada gambar 2.30.



Gambar 2.30 Logo Sistem Operasi Kali Linux

Kali linux memiliki lebih banyak keuntungan dari backtrack pendahulunya, karena kali linux hadir dengan lebih banyak *tools* yang telah di-

update, *tools* yang searah dengan standar repositori debian dan disingkronisasi empat kali dalam sehari. Yang artinya pengguna dapatkan *update* paket dan *security fixes* terbaru. *Compliant filesystem* terbaru menerjemahkan ke hampir seluruh *tools* dari manapun dalam sistem. Kali juga membuat kostumisasi, instalasi *unattended*, dan fitur lingkungan flexible desktop yang kuat.

2.19.2 Windows

Microsoft windows atau yang lebih dikenal dengan sebutan windows adalah salah satu dari keluarga sistem operasi. Sistem operasi windows telah berevolusi dari MS-DOS, sebuah sistem operasi yang berbasis modus teks dan *command-line.* Windows pertama yang menggunakan sistem GUI adalah Windows *Graphic Encirontment* 1.0 atau yang biasa dikenal dengan Windows 1.0 dikenalkan pertama kali pada 10 novembar 1983. Windows 1.0 merupakan perangkat lunak 16-bit tambahan (bukan merupakan sistem operasi) yang berjalan di atas MS-DOS (dan beberapa varian dari MS-DOS), sehingga ia tidak dapat berjalan tanpa adanya sistem operasi DOS. Berikut ini adalah logo sistem operasi windows dapat dilihat pada gambar 2.31.



Gambar 2.31 Logo Sistem Operasi Windows

Dimulai dari DosShell untuk DOS 6 buatan Microsoft dan inginya Microsoft bersaing terhadap larisnya penjualan apple macintosh yang menggunakan GUI, Microsoft menciptakan windows 1.0. nama ini berasal dari kelatahan karyawan Microsoft yang menyebutkan nama aplikasi tersebut sebagai

program windows (jendela program). Windows versi 2 adalah versi windows pertama yang bisa diinstal program.

2.19.2.1 Windows Server 2012

a Windows server 2012 merupakan sistem operasi terbaru yang dirilis oleh Microsoft setelah windows server 2008 windows server 2010. Seperti halnya windows server 2008 R2, windows server 2012 hanya diproduksi dalam arsitektur 64-bit. Windows server 2012 terdiri dari empat edisi yaitu Windows Server 2012 Foundation, Windows Server 2012 Essential, Windows Server 2012 Standard dan Windows Server 2012 Datacenter. Perbedaan antara satu edisi dengan edisi yang lain terletak pada lisensi, fitur, dan harga. Berikut ini adalah logo sistem operasi windows server 2012 R2 dapat dilihat pada gambar 2.32.



Gambar 2.32 Logo Sistem Operasi Windows Server 2012 R2

Saat menginstall Windows Server 2012, anda bisa memilih apakah meng*install* menggunakan *Graphical User Interface* (GUI), atau *Server Core*. *Server Core* lebih baik dari sisi keamanan system (*System Security*), namun bagi anda yang baru mempelajari dan berkenalan dengan Windows Server, sebaiknya meng*install* menggunakan GUI. Salah satu kelebihan Windows Server 2012 dengan Windows Server versi sebelumnya adalah, anda dapat berpindah/*switching* dari *Server Core* ke GUI dan sebaliknya, dimana pada Windows Server versi sebelumnya anda harus menginstall ulang *server* jika ingin melakukan hal tersebut.

2.20 *Shell Script*

Shell script adalah suatu *file* yang berisi perintah-perintah yang harus dikerjakan oleh *shell* dan kemudian dieksekusi dengan cara memanggil *file* tersebut. Perintah dalam suatu *script* bisa berupa perintah yang dapat diberikan dari *prompt* atau perintah kendali alir (*flow control commands*). Perintah kendali alir adalah perintah untuk mengatur urutan eksekusi perintah-perintah dalam suatu *script*.

Shell script umumnya ditulis dengan menggunakan teks editor, misalnya vim, nano atau emacs, yang biasanya terdiri dari perintah-perintah dan komentar dan umumnya diberikan ekstensi .sh. Setiap *file script* biasanya diawali dengan baris seperti : #!/bin/bash yang artinya *script* dibawah baris tersebut akan dijalankan dengan interpreter /bin/bash. Komentar ditandai dengan tanda # dan terdiri dari teks yang memberitahu apa yang terjadi.

2.21 Fwsnort

Fwsnort merupakan *open source* aplikasi baris perintah yang ditulis dalam C dan dirancang untuk menhurai file aturan yang termasuk dalam snort *software intrusion detection system*. Hal ini juga menghasilkan *IPtables* setara *rulesets*. *Features* pada fitur *glancekey* termasu dukungan untuk mendeteksi TCP SYN, NULL, FIN, XMAS *scan,* dan *scan* UDP, beberapa aturan *signature* untuk snort, modus forensic untuk analisis *IPtables file log*, operasi pasif sistem *signature* melalui paket TCP SYN, dan lainnya.

Selain itu , aplikasi ini mendukung validasi kode *header* dan jenis lapangan icmp, tingkat bahaya konfigurasi dan batas memindai tugas, *iptables ruleset parsing, dshield alert, auto-blocking IP address scanning,* dan status yang komprehensif, kemampuan untuk membatasi parser snort untuk menerjemahkan aturan hanya ditemukan dalam aturan *iptables*.

2.22 PSAD

PSAD merupakan *tool* untuk memanfaatkan *iptables log message* untuk mendeteksi, *alert* dana memblokir aktivitas *port scan* serta *traffic* data yang mencurigakan lainnya. PSAD memanfaatkan *Netfilter log message* untuk mendeteksi, mewaspadai, dan memblokir port scan dan *traffic* yang dicurigai. Untuk scan TCP, TCP *flag* PSAD analisis untuk menentukan jenis *scan* (syn,fin,xmas,dll) dan opsi *command line* yang sesuao yang dapat dipasok ken map untuk menghasilkan seperti *scan*. Selain itu , PSAD memanfaatkan banyak TCP, UDP, ICMP, dan *signature* yang terkandung dalam sistem deteksi intrusi snort.

2.23 Iptables

Iptables adalah suatu *tools* dalam sistem operasi linux yang berfungsi sebagai alat untuk melakukan *filter* (penyaringan) terhadap lalulintas data. Iptables sudah menjadi platform untuk membuat *firewall* di kebanyakan distro linux. Dengan menggunakan iptables seorang pengguna atau admin jaringan bisa mengatur lalulintas paket data yang keluar masuk pada *router* atau *server* yang menjadi *gateway* antara jaringan local dengan jaringan publik.

Dibawah ini adalah bentuk umum perintah yang digunakan iptables dapat dilihat pada table 2.10.

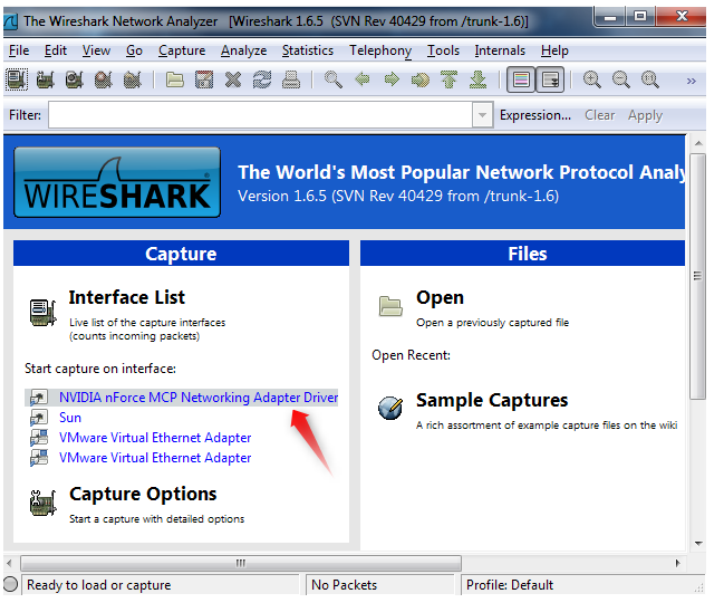
Table 2.10 Bentuk Umum Perintah Iptables

Perintah	Keterangan
-A --append	Perintah ini menambahkan aturan di akhir *chain.*
-D --delete	Menghapus satu aturan *chain.*
-R --replace	Mengganti aturan *chain* dengan aturan baru
-I --insert	Memasukan aturan pada suatu baris di *chain*.
-L --l	Menampilkan semua aturan pada table.
-p --protocol	Mengecek tipe protokol tertentu, seperti TCP,UDP,ICMP, dan ALL.
-s	Menyocokan paket berdasarkan alamat

--src --source	ip asal tunggal seperti 192.168.2.69.
-d --dst --destination	Menyocokan paket berdasarkan alamat tujuan.
-i --in-interface	Menyocokan paket berdasarkan *interface source* paket dan hanya berlaku pada *chain* INPUT,FORWARD,dan PREROUTING
-o --out-interface	Menyocokan paket berdasar *interface destination* paket seperti –in-interface.

2.24 Wireshark

Wireshark merupakan *network protocol analyzer*, atau juga termasuk salah satu *network analysis tool* atau *packet sniffer*. Wireshark mengizinkan pengguna mengamati data dari jaringan yang sedang beroperasi atau dari data yang ada di dalam *disk*, dan langsung melihat atau mensortir data yang tertangkap, mulai dari informasi singkat dan rincian bagi masing-masing paket. Berikut ini adalah tampilan menu wireshark dapat dilihat pada gambar 2.33.



Gambar 2.33 Tampilan Menu Wireshark

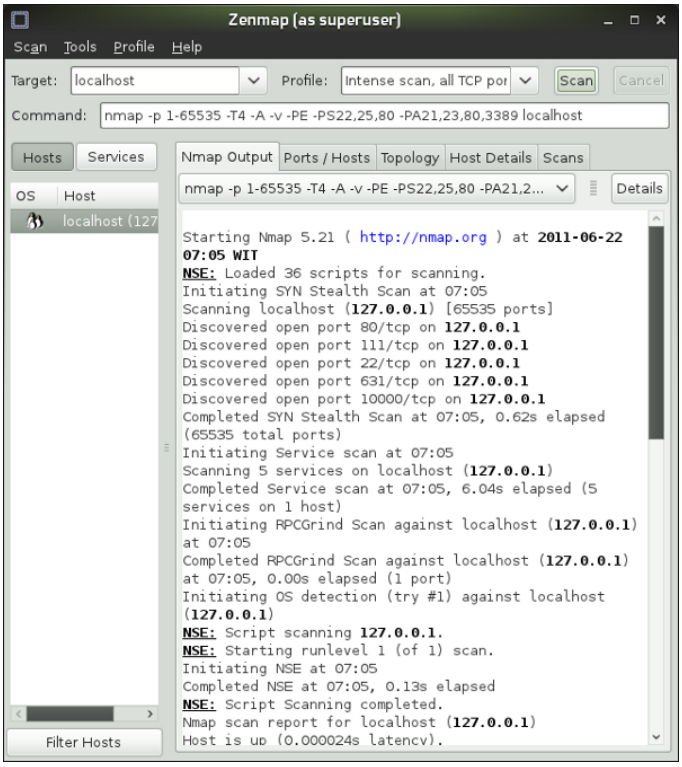
2.25 Nmap

Nmap adalah utilitas keamanan *open-source* kuat yang dapat digunakan untuk meng-audit keamanan dan mengeksplorasi jaringan, tetapi dapat juga digunakan untuk komputer yang tidak terhubung dengan jaringan. Nmap didesain untuk melakukan *scanning* jaringan besar ataupun kecil dengan menggunakan paket *raw* IP untuk mengetahui *host* yang sedang aktif dalam jaringan, *service* yang dijalankan , dan *port* yang terbuka. Berikut ini adalah bentuk umum perintah yang digunakan dalam melakukan *scanning* : nmap [tipe *scan*] [opsi *scan*] [alamat target]. Sebagai contoh berikut adalah perintah dan pengertian opsi untuk *scanning* dasar : nmap –sS –P0 –sV –o 192.168.1.1. Untuk opsi yang digunakan memiliki fungsi sebagai berikut, -sS opsi untuk menggunakan fitur TCP SYN *scan*, -P0 adalah opsi untuk melakukan *scanning* terhadap protocol tertentu berdasarkan nomor protokolnya, sedangkan –sV adalah opsi untuk mencari versi dari *service* yang aktif pada port tertentu, dan –o adalah opsi untuk mengetahui sistem operasi apa yang digunakan oleh target.

Nmap dapat digunakan untuk melakukan *fingerprinting* yang bisa membandingkan dan memberikan estimasi jenis istem operasi apa yang digunakan target. Nmap juga mempunyai banyak kelebihan atau *flags* yang akan memanipulasi cara *scanning*.

2.26 Zenmap

Zenmap adalah aplikasi multi platform sebagai interface sederhana untuk apikasi nmap. Nmap (*Network Mapper*) sendiri adalah sebuah aplikasi *open source* untuk explorasi network dan audit keamanannya. Nmap bekerja dengan melakukan scan terhadap komputer (*host*) *stand alone* ataupun host yang terhubung dalam sebuah jaringan, menentukan *host-host* yang aktif dalam suatu jaringan, menentukan informasi sistem operasi, *port-port* yang terbuka dan jenis *firewall* yang digunakan. Berikut ini adalah tampilan menu zenmap dapat dilihat pada gambar 2.34.



Gambar 2.34 Tampilan Menu Zenmap

Zenmap memiliki beberapa pilihan profil untuk melakukan *scanning* berikut ini adalahabeberapa profil yang sering digunakan dalam zenmap.

1. *Intense scan*

   Intense scan merupakan profil dimana perintah *scan* meliputi *port-port* yang terbuka, model topologi jaringan dari target yang di *scan*, *service* yang sedang aktif, serta *host detail* yang berisikan informasi sistem operasi target dan persentase ketepatan hasil *scan.*

2. *Intense scan plus UDP*

   Profil ini hampir sama dengan *intense scan*, namun ditambah dengan port-port UDP yang sedang aktif.

3. *Intense scan, all TCP ports*

   Profil ini juga sama dengan *intense scan* dan ditambah melakukan *scanning* nomor *port* dimulai dari nomor 1 hingga nomor *port* 65535.

4. *Ping scan*

   Profil *ping scan* adalah cara *scan* target melalui perintah *ping*, apabila balasan *ping* mendapat *reply* berarti target yang di *scan* sedang aktif.

5. *Quick scan*

   Profil ini melakukan *scan* lebih cepat dari *intense scan* namun dan hanya melakukan *scaning* secara cepat dan tidak mendalam.

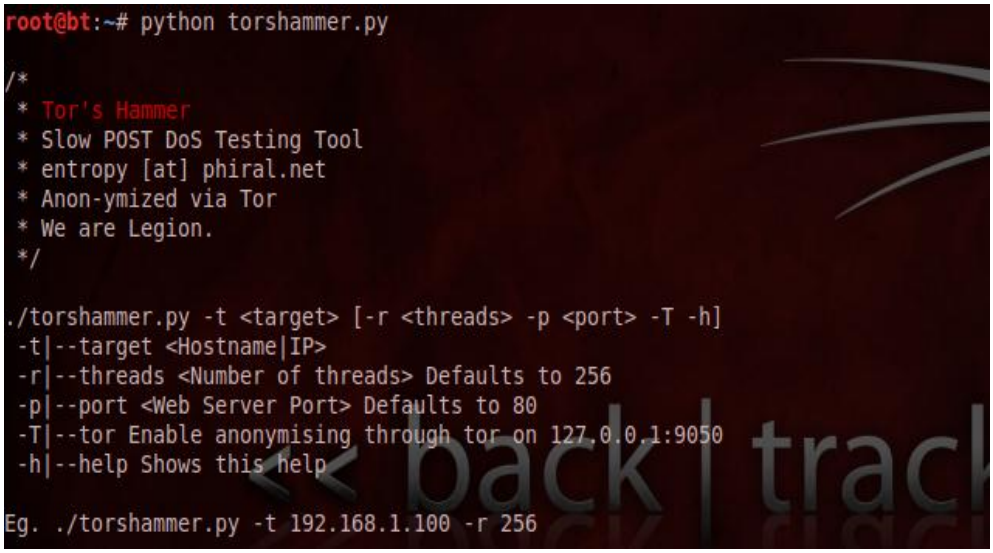
2.27 Hping3

Hping adalah sebuat TCP/IP *assembler* dan juga merupakan *command-line* yang berorientasi pada pemrosesan paket TCP/IP. Hping dapat digunakan untuk membuat paket IP yang berisi TCP. UDP atau ICMP *payloads*. Semua *field header* dapat dimodifikasi dan dikontrol dengan menggunakan baris perintah (*command-line*).

Berikut ini adalah bentuk umum perintah yang digunakan hping untuk menyerang targetnya : hping3 [opsi] [alamat target], dan berikut adalah contoh perintah untuk melakukan serangan pada target yang beralamat 192.168.1.1 : hping3 –c 100000 –d 12000 –S –w 128 –p 21,80,443,3306 –flood 192.168.1.1. dari perintah diatas dapat diartikan hping3 menyerang target yang beralamat di 192.168.1.1 dengan menggunakan opsi –c yang berarti jumlah paket yang akan dikirimnya sebanyak 100000 byte, kemudian –d yang berarti ukuran data yang dikirim sebesar 12000 byte, -S yang berarti menggunakan SYN flag, -w berarti *winsize* yang digunakan berukuran 128, -p adalah opsi ke *port* mana saja serangan akan dituju, dan opsi terakhir --flood yaitu mengirim paket secepat-cepatnya agar target menjadi lebih sulit untuk merespon.

2.28 Torshammer

Torshammer adalah salah satu *tool* untuk menlancarkan serangan DDoS yang diciptakan menggunakan bahasa pyton. Torshammer juga dapat digunakan bersamaan dengan tor *network* agar keamanan penyerang lebih sulit diketahui. Jika tors digunakan pada alamat 127.0.0.1:9050, tors akan menyerang dan mematikan hampir seluruh *web server* yang tidak dilindungi dalam apache dan IIS dalam satu serangan. Berikut ini adalah tampilan menu torshammer dapat dilihat pada gambar 2.35.



Gambar 2.35 Tampilan Menu Torshammer

2.29 Sqlmap

Sqlmap adalah *tools opensource* berbahasa python yang mendeteksi dan melakukan eksploit pada bug SQL *injection* secara otomatis. Dengan melakukan serangan SQL *injection* seorang penyerang dapat mengambil alih serta memanipulasi sebuah *database* didalam sebuah *server*. Bentuk umum dari perintah sqlmap berbentuk sqlmap –u [alamat web yang menjadi celah] [opsi], berikut ini adalah contoh perintah untuk melihat isi database dalam server yang menjadi target penyerang : sqlmap –u http://192.168.0.103/pertanian/login-submit.php --data=”username=test&pass=test” –dbs. Dari perintah tersebut dijelaskan bahwa alamat web yang menjadi celah dari target adalah 192.168.0.103/pertanian/login-submit.php, yang kemudian akan dimasukan data untuk *tampering* atau data untuk negosiasi yang berisikan *username* = test dan pass = test dan –dbs yang berfungsi untuk menampilkan *database* apa saja yang terdapat pada *server* target.

Selain opsi tersebut, sqlmap memiliki beberapa opsi yang sering digunakan untuk melihat data-data pada *database* targetnya. Dibawah ini adalah beberapa opsi yang terdapat pada sqlmap dapat dilihat pada tabel 2.11.

Table 2.11 Bentuk Umum Perintah Sqlmap

Nama opsi	**keterangan**
-u	Alamat target atau alamat dimana menjadi celah untuk disusupi.
--data	Data string yang akan dikirim melalui *POST*.
--dbs	Menampilkan *database* yang terdapat pada target / *server*.
--tables	Menampilkan tabel pada *database* yang terdapat pada target / *server*.
--colums	Menampilkan kolom pada tabel yang terdapat pada target / *server*.
-D	*Database* yang dipilih untuk dilakukan suatu perintah.
-T	Tabel yang dipilih untuk dilakukan suatu perintah .
-C	Kolom yang dipilih untuk dilakukan suatu perintah.
--dump	Menampilkan isi / data dari database dan menyimpannya dalam *file* teks.