BAB II
DASAR TEORI

2.1 KAJIAN PUSTAKA

Muhammad Farradhika Muntaha, Primantara Hari Trisnawan, Rakhmadhany Primanda melakukan penelitian pada tahun 2019 yang berjudul tentang "*Implementasi Intrusion Prevention System* (IPS) Berbasis Athena untuk Mencegah Serangan DDOS Pada Arsitektur *Software Defined Network* (SDN)" yang membahas tentang perancangan sistem keamanan IPS dan penerapanya pada arsitektur jaringan SDN dengan menggunakan Athena sebagai SDN *Controller*nya dan kelebihan Athena yaitu mampu terintegrasi penuh pada komponen SDN lain dimana instansi yang ter*Host* di atas SDN *Controller*nya. Penelitian ini menggambarkan implementasi dari IPS berbasis yang mampu mencegah serangan TCP SYN *Flood* dan UDP *Flood* dibuktikan dengan normalnya througput saat terjadinya penyerangan tersebut karena sistem dapat memblokir serangan tersebut [7].

Mochamad Bagus Firmansyah, Ridha Muldina Negara, Danu Dwi Sanjoyo melakukan penelitian pada tahun 2019 yang berjudul "Mengimplementasikan Sistem Keamanan Jaringan *Instrusion Prevention System* Berbasis Snort Pada Arsitektur *Software Defined Network*" yang membahas tentang perancangan dan penggunaan sistem keamanan IPS dengan berbasis Snort serta penggunaan Ryu *Controller* sebagai salah satu *Controller* pada jaringan SDN. Penelitian ini menggambarkan bahwa Snort dapat mendeteksi serangan dengan parameter serangan ukuran paket yang dikirim dengan interval kedatangan paket yang cepat. Sistem kemudian memblokir tersebut dengan cara mengambil data *packet source, packet destination* dan *packet prokol* dari *log file Alert* yang sudah dimodifikasi ke dalam bentuk CSV sehingga mampu untuk meningkatkan keamanan jaringannya [8].

Muhammad Arief Nugroho, Novian Anggis Sumastika melakukan penelitian pada tahun 2018 yang berjudul "Perancangan *Intrusion Prevention System* pada Jaringan *Software Defined Network*" yang membahas tentang perancangan sistem keamanan IPS dan penerapannya pada arsitektur jaringan SDN dengan menggunakan Ryu sebagai SDN *Controller*nya dan Snort sebagai *firewall*. Penelitian ini menggambarkan pengujian sistem keamanan IPS pada paket yang dikirim ke jaringan serta pengujian keamanan dengan didapatkan hasil kecepatan *troughput* jaringan sebelum dan setelah intregasi sistem keamanan IPS ke jaringan SDN [9].

Pande Putu Kika Adi Saputra, Ridha Muldina Negara, S.T., Danu Dwi Sanjoyo, S.T., M.T melakukan penelitian pada tahun 2018 yang berjudul "Integrasi *Intrusion Prevention System* dan Analisa Performasi Pada *Software Defined Network*" yang membahas tentang perancangan sistem keamanan IPS dan menganalisa performa hasil dari penerapan tersebut di jaringan SDN. Perancangan dengan menggunakan Ryu sebagai SDN *Controller*nya dan Snort sebagai *Firewall*. Penelitian ini menggambarkan bahwa pada jaringan SDN yang terintegrasi sistem keamanan IPS untuk parameter QOS hasilnya cenderung lebih stabil karena sistem keamanan IPS mampu memblokir paket serangan sehingga meminimalisir terjadinya penurunan performansi jaringan berupa turunnya kecepatan *troughput* akibat adanya serangan yang masuk ketika pengiriman paket layanan [2].

Riverta Fierre Dwiputra Purba melakukan penlitian pada tahun 2018 yang berjudul "Simulasi Pencegahan Serangan *Denial Of Service* (DoS) Pada *Software Defined Network*ing (SDN) Menggunakan *Intrusion Prevention System* (IPS) dan Algoritma Genetika" yang membahas tentang perancangan sistem keamanan jaringan IPS dan penggunaan pada jaringan SDN untuk melakukan pencegahan serangan jaringan berupa DoS pada jaringan SDN yang diintregasikan dengan algoritma Genetika agar durasi blokir dapat beradaptasi dengan serangan yang datang. Penelitian ini mengambarkan bahwa pencegahan serangan jaringan DoS pada sistem ini dilakukan secara *real-time* dengan membandingkan paket yang masuk terhadap *rules* yang telah didefinisikan penyerang dapat melemahkan *Controller* jaringan SDN [10].

Penelitian yang dilakukan tidak terlepas dari hasil penelitian yang telah disebutkan diatas, selanjutnya dipaparkan pada tabel 2.1

Tabel 2.1 Tinjauan Pustaka Penelitian Terdahulu

No	Jurnal	Tahun	Keterangan
1	Muhammad Farradhika Muntaha, Primantara Hari Trisnawan, Rakhmadhany Primananda " Implementasi *Intrusion Prevention System* (IPS) Berbasis Athena Untuk Mencegah Serangan DDoS Pada Arsitektur *Software Defined Network* (SDN)"	2019	Sistem keamanan mampu mencegah dan memblokir serangan TCP SYN *Flood* dan UDP *Flood* dibuktikan normalnya jaringan setelah terjadinya penyerangan.

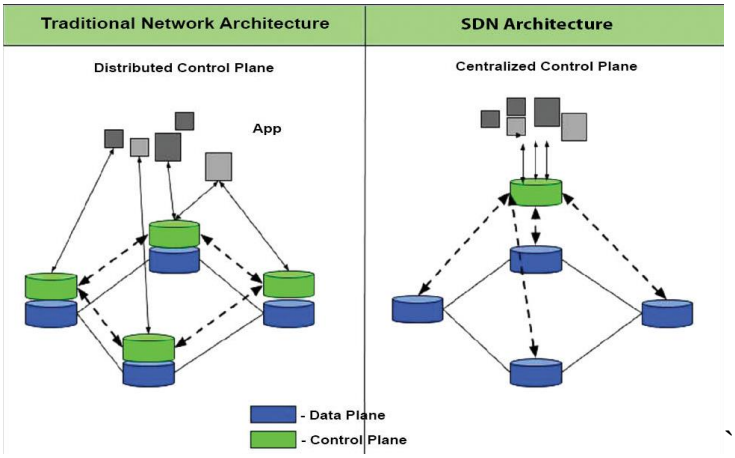
2	Mochamad Bagus Firmansyah, Ridha Muldina Negara, Danu Dwi Sanjoyo "Mengimplementasikan Sistem Keamanan Jaringan *Intrusion Prevention System* Berbasis Snort Pada Arsitektur *Software Defined Network*"	2019	Sistem atau Snort mendeteksi serangan dan memblokir serangan berupa paket yang sangat cepat dengan mengambil data paket serangan dari *log file Alert* ke dalam bentuk CSV.
3	Muhammad Arief Nugroho, Novian Anggis Suwastika "Perancangan *Intrusion Prevention System* Pada Jaringan *Software Defined Networks*"	2018	Pengujian sistem keamanan IPS dengan simulasi penyerangan jaringan berupa DDOS didapatkan hasil sistem dapat memblokir serangan tersebut tetapi *througput* jaringan mengalami penurunan .
4	Pande Putu Kika Adi Saputra , Ridha Muldina Negara, S.T ., M.T, Danu Dwi Sanjoyo, S.T ., M.T "Integrasi *Intrusion Prevention System* Dan Analisa Performansi Pada *Software Defined Network*"	2018	Hasil Parameter QOS pada Jaringan SDN yang terintegrasi dengan IPS cenderung lebih stabil dan aman.
5	Riverta Fierre Dwiputra Purba "Simulasi Pencegahan Serangan Jaringan *Denial Of Service* (Dos) Pada *Software Defined Networking* (SDN) Menggunakan *Intrusion Prevention System* (IPS) Dan Algoritma Genetika"	2018	Pencegahan serangan DOS dilakukan secara *real time* dengan membandingkan paket yang masuk terhadap *rules* yang telah didefinisikan penyerang dapat melemahkan *Controller* jaringan SDN.

2.2 DASAR TEORI

2.2.1 *Software Defined Network* (SDN)

Model arsitektur jaringan traditional memiliki kekurangan karena masing-masing perangkat mempunyai konfigurasi yang berbeda. Sehingga jika ada 2 atau 3 router maka konfigurasi dilakukan dengan sejumlah perangkat router atau perangkat lain seperti switch.

Konsep pada SDN yaitu *Control plane* dan *forwarding Plane*nya dipisah dalam suatu perangkat yang berbeda. Artinya perangkat jaringan yang terhubung akan dikendalikan oleh aplikasi *Plane* yang berkolaborasi dengan *Control plane* untuk mengintruksikan arah suatu *Traffic* atau paket. Konsep SDN beroperasi menggunakan protokol OpenFlow. OpenFlow seperti sebuah penghubung yang mengendalikan perangkat berdasarkan MAC, IP *Address*, atau TCP *Port* untuk melakukan kegiatan tertentu. Secara logika *Control plane* diletakan secara terpusat yang membutuhkan sebuah sistem operasi jaringan yang mampu membentuk peta digital dari seluruh jaringan dan dipresentasikan melalui sejenis API (*Application Programming Interace*). Contoh perbedaan penggunaan Arsitektur Jaringan SDN dan Arsitektur Jaringan Tradisional seperti gambar 2.1.



Gambar 2.1 Perbedaan Arsitektur Jaringan Tradisional dan SDN [10].

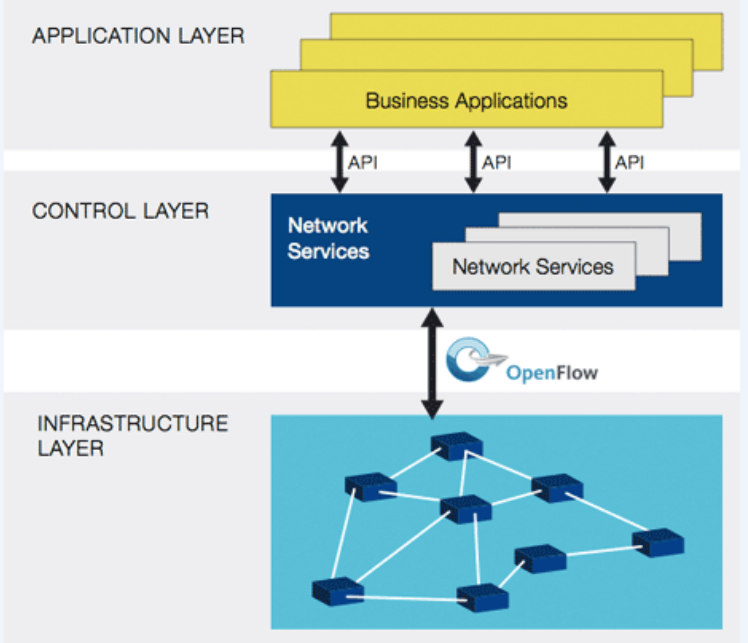
Gambar 2.1 merupakan perbedaan Arsitektur tanpa SDN dan dengan SDN. Arsitektur jaringan SDN dibagi menjadi 3 berdasarkan layernya yaitu :

1. Infrastruktur (Data *Plane* / *Infrastructure Layer*)

   Terdiri dari elemen jaringan yang dapat mengatur SDN *Datapath* sesuai dengan instruksi yang diberikan melalui *Control Data Plane Interface* (CDPI) Kontrol (*Control plane* /*Layer*).

2. Entitas kontrol (SDN *Controller*) mentranslasikan kebutuhan aplikasi dengan infrastruktur dengan memberikan intruksi yang sesuai untuk SDN *Datapath* serta memberikan informasi yang relevan dan dibutuhkan oleh SDN *Application*.
3. Aplikasi (*Apllication Plane* / *Layer*)

   Berada pada lapis teratas, berkomunikasi dengan sistem via *NorthBound Interface* (NBI).

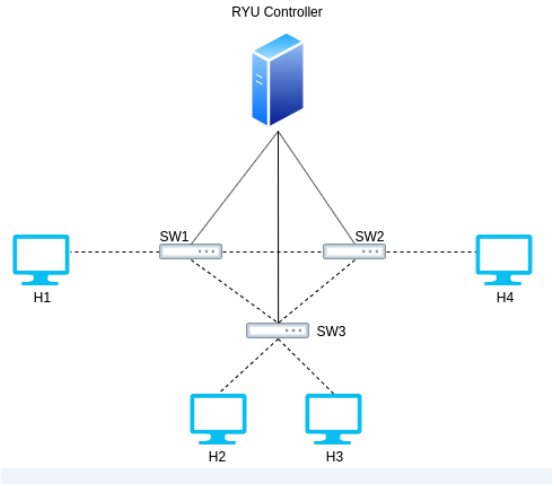


Gambar 2.2 Arsitektur *Layer*[6].

Pada gambar 2.2 merupakan layer pada aristektur SDN. Pada perangkat konvesioal, *packet forwarding* (jalur data) dan *routing* (jalur kontrol) terjadi pada perangkat yang sama. OpenFlow membedakan dan memisahkan jalur data dari jalur kendali. Bagian jalur dara berada pada perangkat tersendiri, *controler* terpisah membuat keputusan routing. Perangkat dan *Controller* berkomunikasi melalui protokol OpenFlow dan metodologi ini dikenal sebagai SDN [11].

2.2.2 Mininet

Mininet merupakan emulator jaringan yang menciptakan SDN yang terdiri dari *virtual Host*, switch dan *controler*. Mininet dijalankan pada sistem operasi Linux dan mendukung protokol OpenFlow [5].



Gambar 2.3 Contoh Topologi Pada Mininet [12].

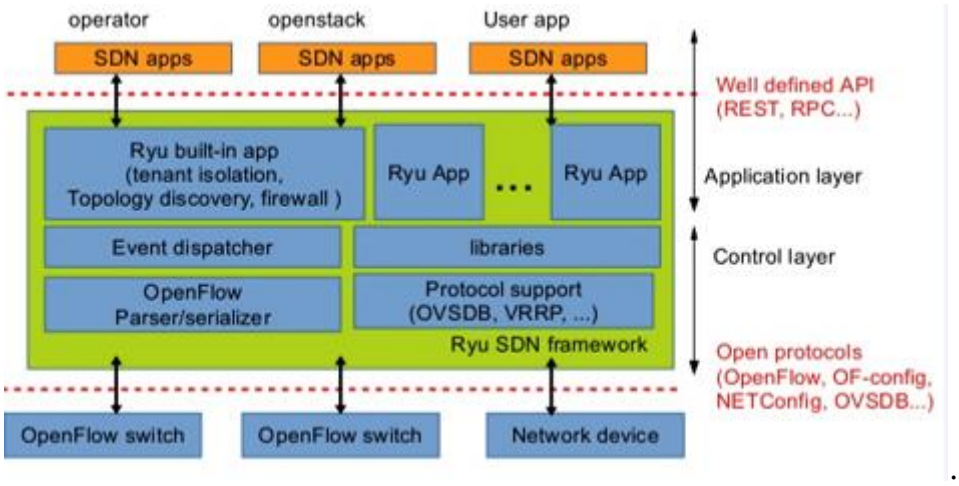
2.2.3 Ryu *Controller*

Ryu dalam bahasa jepang dilafalkan *ree-yooh* yang artinya mengalir [13]. merupakan salah satu *Controller* pada SDN yang dirancang untuk meningkatkan kemampuan dalam jaringan yang bermanfaat untuk mempermudah, Secara umum *Controller* merupakan fungsi otak dari SDN. Ryu merupakan open source yang dikembangkan oleh NTT *Communication* dengan linsensi dari apache 2.0 serta memiliki API (*Application Program interface)* yang sudah didefinisikan dengan sangat baik sehingga mempermudah pengguna untuk membuat dan mengatur suatu jaringan yang baru dengan target penggunaan untuk riset,developer dan operator jaringan..

Ryu menggunakan bahasa pemrograman python dan memiliki dokumentasi yang banyak sehingga lebih mudah solusi jika terdapat permasalahan. *Controller* Ryu mendukung prokol pada SDN seperti OpenFlow, Netconf, Of Config dan lainnya. Sebenarnya Ryu bukanlah *Controller* tetapi merupakan framework yang disebut *Controller* dan gabungan dari framerwork. Berikut Kelebihan Ryu yaitu:

1. Ryu menyediakan banyak komponen yang berguna untuk jaringan SDN.
2. Komponen Lama pada Ryu dapat dimodifikasi sesuai dengan kebutuhan dan menerapkan komponen yang baru.
3. Dapat menggabungkan komponen untuk membangun Aplikasi.

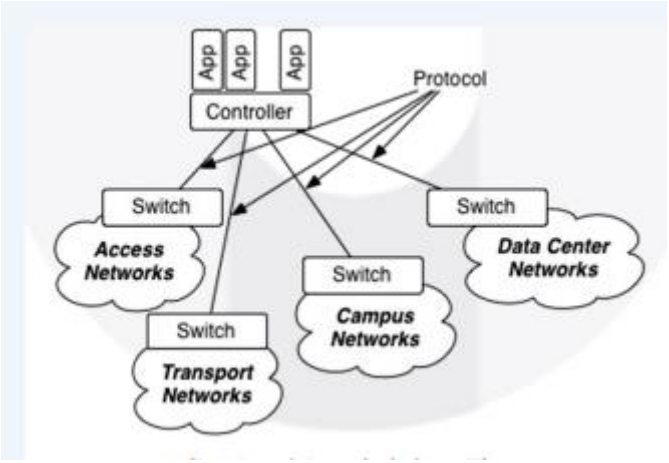
Framework Ryu berada pada *control* layer pada arsitektur SDN dan beberapa aplikasi pada Ryu berada pada application layer yang berfungsi untuk berkomunikasi dengan aplikasi SDN lain menggunakan API [14].



Gambar 2.4 Arsitektur Ryu [14].

2.2.4 Open Flow

OpenFlow merupakan salah satu protokol diantara *Layer* kontrol dan *Layer* infrastruktur. OpenFlow memiliki fungsi sebagai *Control plane* dan melakukan kontrol langsung terhadap perangkat forwarding yang meregulasi bagaimana pergerakan paket didalam sebuah jaringan dan memungkinkan sesbuah *Controller* berinteraksi dengan switch dengan tipe yang berbeda-beda [8].



Gambar 2.5 Protokol OpenFlow [8].

2.2.5 Intrusion Prevention System

IPS atau *Intrusion Prevention System* merupakan salah satu *firewall* atau sistem keamanan yang membuat akses kontrol dengan cara melihat konten aplikasi dibandingkan dengan melihat Ip *Address* atau *port* yang biasanya digunakan *firewall*. Sistem IPS hampir sama dengan sistem setup IDS. Sistem IPS mampu mencegah serangan yang datang dengan adanya bantuan dari administrator atau tidak. Perbedaan utama dari sistem IPS

dengan IDS yaitu IDS digunakan sebagai sistem monitoring sedangkan IPS bertindak sebagai kontrol sistem. Secara umum IPS dibagi menjadi 2 yaitu :

1. *Host*-based *Intrusion Prevention System* (HIPS)

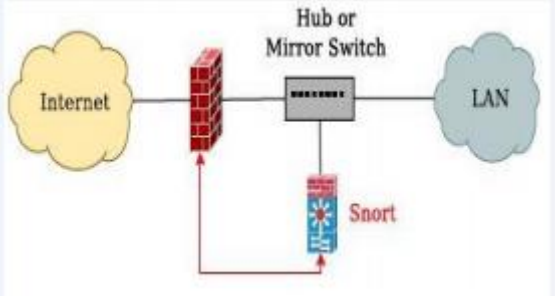
   Merupakan sebuah sistem pencegahan yang terdiri dari banyak lapisan dengan menggunakan paket filtering, inspeksi status dan metode pencegahan yang bersifat *real-time* untuk menjaga *Host* tetap berjalan normal dan efisien. Mekanisme kerjanya dengan mencegah kode berbahaya yang mencoba menyerang atau memasuki *Host*.

2. *Network* Based *Intrusion Prevention System* (NIPS)

   Sistem IPS jenis ini dapat menahan semua trafik jaringan dan memeriksa kegiatan dan kode yang mencurigakan. Menggunakan in-line model sehinga performansi tinggi merupakan sebuah elemen penting dari perangkat IPS untuk mencegah *bottleneck* atau macetnya proses aliran data atau paket pada sebuah jaringan. NIPS menggunakan 3 komponen untuk mengakselerasi performa *Bandwidth*. NIPS melakukan pantaun dan proteksi dalam satu jaringan secara global. NIPS menggabungkan fitur IPS dengan *firewall* sehingga bisa disebut sebagai *in-Line* IDS *atau Gateway Intrusion Detection System* (GIDS) [15].

2.2.6 Snort

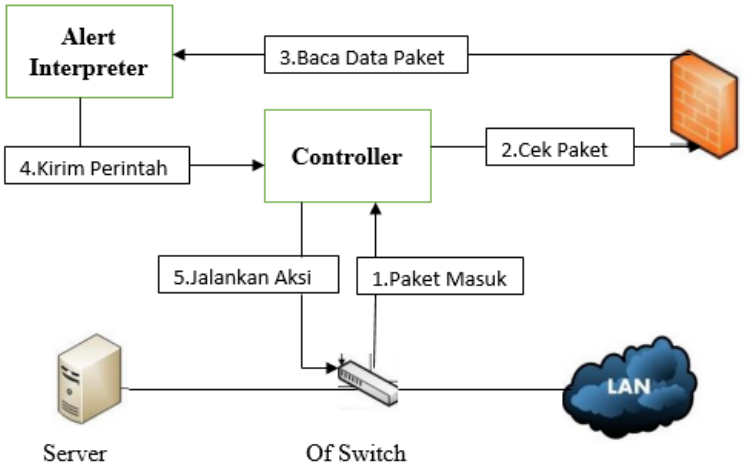
Snort merupakan aplikasi *open source* dari IPS dan berfungsi untuk analisis *trafic* secara *real-time* dan *packet logging* pada jaringan IP. Pada arsitektur jaringan tradisional snort berfungsi dengan cara menduplikat semua paket yang lewat ke snort dan dicocokan dengan *rules* atau aturan yang ada pada snort, jika ada paket yang mencurigakan maka snort akan menyalakan *Alert* atau peringatan.



Gambar 2.6 Cara kerja Snort pada jaringan tradisional [2].

Pada jaringan SDN snort menerapkan cara kerja *mirror function* pada switch OpenFlow. *Controller* akan mengatur satu *port input* dengan dua (2) *port output*, satu *port* sebagai *forwarding* dan *port* lainnya untuk snort. Paket yang lewat akan diteruskan ke tujuan sekaligus ke arah snort untuk dianalisa. Jika terdapat paket yang mencurigakan

maka snort akan mengirim *Alert* ke *Controller* untuk menjalankan mekanisme pemblokiran paket [2].



Gambar 2.7 Cara kerja Snort pada jaringan SDN [2].

2.2.7 Keamanan Jaringan

Keamanan jaringan merupakan bagian dari sebuah sistem informasi yang berfungsi untuk menjaga validitas dan integritas data serta menjamin ketersediaan layanan bagi penggunanya dalam menjaga keamanan jaringan dengan menerapkan konsep CIA yaitu *Confidentiality* (Kerahasiaan), *Integrity* (Integritas), dan *Availability* (Ketersediaan). Kerahasiaan berhubungan dengan hak ases utnuk membaca data atau informasi dari suatu sistem komputer [16].

2.2.8 Denial Of Service

DOS atau *Denial Of Service* merupakan jenis serangan yang dilakukan dengan cara membanjiri trafik atau lalu lintas jaringan internet pada server, sitem, atau jaringan. Serangan ini dilakukan menggunakan beberapa komputer *Host* penyerang sampai komputer target tidak bisa diakses. Berikut cara kerja dan tujuan DOS. *Denial Of Service* dibagi menjadi 3 tipe yaitu :

1. *Request Flooding*

   Merupakan teknik yang digunakan untuk membanjiri jaringan menggunakan banyak requset. Akibatnya pengguna lain yang terdaftar tidak dapat dilayani.

2. *Traffic Flooding*

   Merupakan teknik yang digunakan dengan membanjiri lalu lintas jaringan dengan banyak data yang menyebabkan pengguna lain tidak bisa dilayani.

3. Mengubah dan merusak sistem konfigurasi,komponen dan server.

Teknik ini termasuk tipe DOS, tetapi cara ini tidak banyak digunakan karena cukup sulit untuk dilakukan [17].

Gambar 2.8 Serangan DOS [18].

2.2.9 Hping3

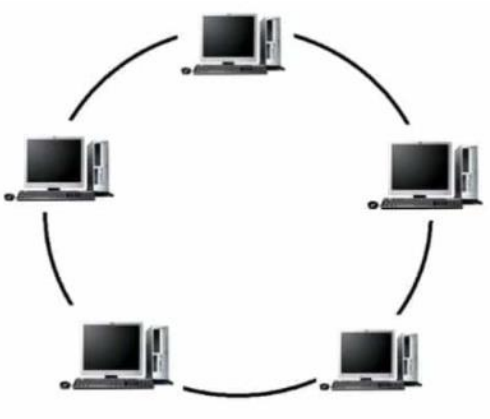
Aplikasi hping3 memungkinkan pengguna untuk mengirim paket yang dimanipuasi seperti ukuran,kuantitas dan fragmentasi paket untuk membebani target dan melewati atau menyerang *firewall*. Aplikasi Hping3 digunakan untuk pengujian keamanan atau kemampuan efektivitas *firewall* dan Server pada suatu jaringan jika dapat menangani paket dalam jumlah besar. Gambar 2.10 merupakan logo dari apliaksi hping3 [19].

2.2.10 Topologi Jaringan

Topologi jaringan komputer adalah metode atau cara yang digunakan agar bisa menghubungkan satu komputer dengan komputer lainnya. Struktur atau jaringan yang digunakan untuk menghubungkan satu komputer dengan komputer lainnya bisa dengan menggunakan kabel atau tanpa kabel. Topologi jaringan komputer berfungsi untuk mengetahui bagaimana masing-masing komputer atau host dalam jaringan komputer dapat saling berkomunikasi satu sama lain. Berikut Jenis-jenis topologi jaringan komputer

1. Topologi *Ring*

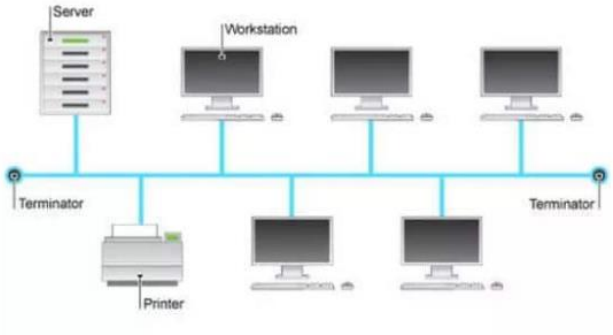
   Topologi *ring* atau sering disebut topologi cincin adalah jenis topologi jaringan yang digunakan untuk menghubungkan sebuah komputer dengan komputer lainnya dalam sebuah rangkaian yang berbentuk melingkar seperti cincin. Umumnya, jenis topologi jaringan ring ini hanya menggunakan LAN *card* agar masing-masing komputer terhubung.



Gambar 2.9 Topologi Ring

2. Topologi Bus

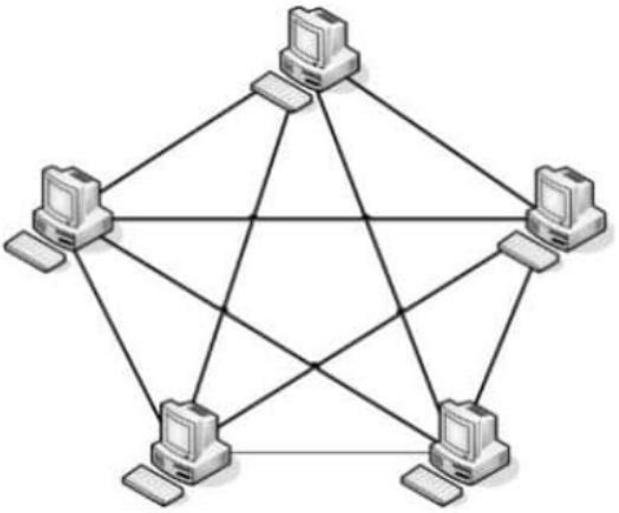
   Topologi bus adalah topologi jaringan yang lebih sederhana. Umumnya topologi jaringan ini dilakukan pada instalasi jaringan berbasi kabel coaxial. Topologi bus menggunakan kabel coaxial pada sepanjang node *client* dan konektor. Jenis konektor yang digunakan adalah BNC, Terminator, dan TBNC.



Gambar 2.10 Topologi Bus

3. Topologi *Mesh*

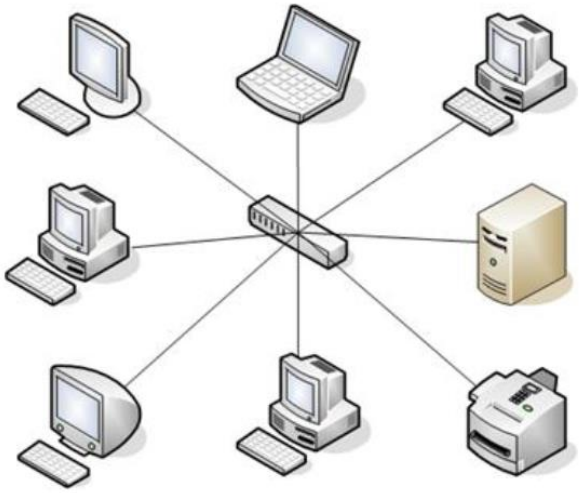
   Topologi *mesh* adalah sebuah topologi yang bisa digunakan untuk rute yang banyak. Jaringan topologi ini menggunakan kabel tunggal sehingga proses pengiriman data menjadi lebih cepat tanpa melalui hub atau switch.



Gambar 2.11 Topologi *Mesh*

4. Topologi *Star*

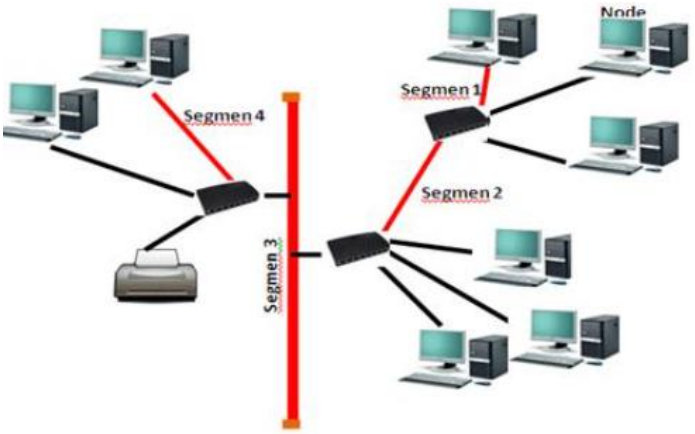
   Topologi *star* atau disebut juga topologi bintang adalah topologi jaringan berbentuk bintang dimana umumnya menggunakan hub atau switch untuk koneksi antar *client*.



Gambar 2.12 Topologi Star

5. Topologi *Tree*

   Topologi tree atau topologi pohon adalah hasil penggabungan dari topologi bus dan topologi star. Umumnya digunakan untuk interkoneksi antara hirarki dengan pusat yang berbeda-beda.



Gambar 2.13 Topologi Tree

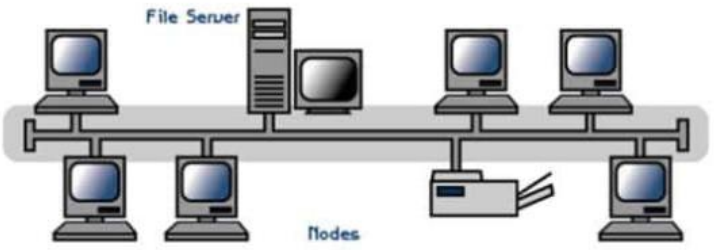
6. Topologi *Peer to Peer*

   Topologi *peer to peer* adalah topologi jaringan yang sangat sederhana karena hanya menghubungkan 2 komputer. Umumnya topologi *peer to peer* menggunakan satu kabel saja untuk menghubungkan kedua komputer agar bisa saling berbagai data.

Gambar 2.14 Topologi *Peer to peer*

7. Topologi Linie*r*

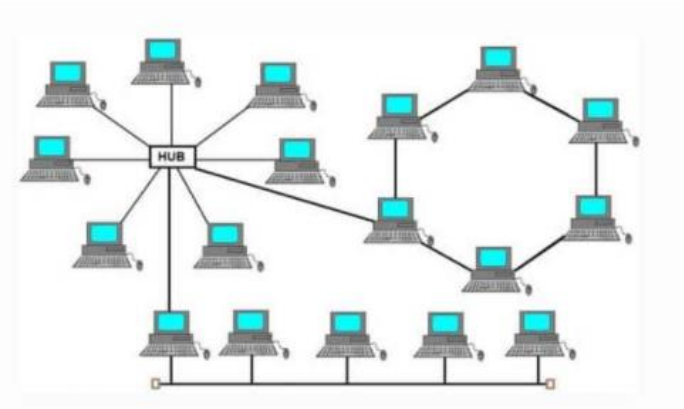
   Topologi linier atau sering disebut dengan topologi bus berurut. Umumnya topologi ini hanya menggunakan satu kabel utama sebagai konektor masing-masing titik sambungan pada setiap komputer.



Gambar 2.15 Topologi Linear

8. Topologi *Hybrid*

   Topologi *Hybrid* adalah gabungan dari beberapa topologi yang berbeda dan membentuk jaringan baru. Dengan kata lain, jika ada dua atau lebih topologi yang berbeda terhubung dalam satu jaringan [20].



Gambar 2.16 Topologi *Hybrid* [20].