

# **IMPLEMENTASI DAN ANALISIS KINERJA *OPENFLOW* PADA *RASPBERRY PI***



## **TUGAS AKHIR**

*Disusun dalam rangka memenuhi salah satu persyaratan  
Untuk menyelesaikan program studi Strata-1 Teknik Informatika*

*Fakultas Teknik Universitas Hasanuddin*

*Makassar*

**Disusun Oleh :**

**AURAEVADNE KUSUMADIARTO**

**D421 14 503**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS T/EKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2018**

**LEMBAR PENGESAHAN SKRIPSI**

**“IMPLEMENTASI DAN ANALISIS KINERJA *OPENFLOW*  
PADA *RASPBERRY PI*”**

**OLEH :**

**AURAEVADNE KUSUMADIARTO**

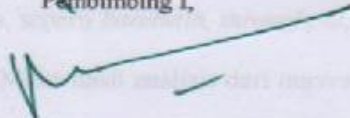
**D421 14 503**

Skrripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 2 Januari 2019.  
Diterima dan disahkan sebagai salah satu syarat memperoleh gelar Sarjana Teknik  
(S.T) pada Program Studi Strata-1 Teknik Informatika Fakultas Teknik  
Universitas Hasanuddin.

Gowa, 2 Januari 2019

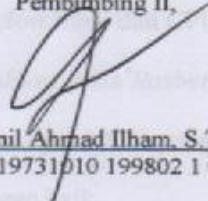
Disetujui Oleh :

Pembimbing I,



Dr.Eng. Muhammad Niswar, ST., M.IT  
Nip. 19730922 199903 1 001

Pembimbing II,



Dr. Amil Ahmad Ilham, S.T., M.IT  
Nip. 19731010 199802 1 001

Diterima dan disahkan oleh:  
Ketua Departemen SI Teknik Informatika



Dr. Amil Ahmad Ilham, S.T., M.IT  
Nip. 19731010 199802 1 001

## ABSTRAK

*Software Defined Network* adalah salah satu konsep baru pada arsitektur yang paling penting untuk manajemen jaringan. *Software Defined Network* berbeda dengan jaringan konvensional, *Software Defined network* memisahkan *control plane* dan *data plane*. Pada arsitektur *Software Defined Network* terdapat protokol yang bersifat *opensource* yaitu *OpenFlow*.

Namun, untuk membuat jaringan *Software Defined Network* menggunakan switch asli tidaklah murah. Maka terdapat *firmware open interface* seperti *OpenvSwitch* sebagai implementasi *OpenFlow* yang akan dipasangkan ke *Raspberry PI* sebagai pengganti *switch* asli untuk menekan biaya pembuatan jaringan *Software Defined Network*.

Pada penelitian ini akan mengimplementasikan dan menganalisis *OpenFlow* pada *Raspberry PI* menggunakan *OpenvSwitch*. Penelitian ini juga menyajikan kerangka pengukuran untuk mendukung berbagai tugas pengukuran jaringan, seperti *bandwidth*, *throughput*, *delay*, serta *flow table* dan *CPU Load* pada *switch*. Maka hasil analisis dari *implementasi OpenFlow* pada *Raspberry PI* sesuai dengan nilai *delay* 0-1Mbit/Sec dan *throughput* mencapai 100% yang di hasilkan artinya *openvswitch* pada *Raspberry PI* bekerja dengan baik.

**Kata kunci :** *Software-Defined Networking, OpenFlow, OpenvSwitch, POX*

## KATA PENGANTAR

Assalamu Alaikum Wr. Wb.

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT karena berkat Rahmat dan Karunia-Nya sehingga Tugas Akhir yang berjudul **“IMPLEMENTASI DAN ANALISIS KINERJA *OPENFLOW* PADA *RASPBERRY PI*”** ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penyusunan penelitian ini disajikan hasil suatu penelitian yang menyangkut judul yang telah diangkat dan telah melalui proses pencarian dari berbagai sumber baik jurnal penelitian, *prosiding* pada seminar-seminar nasional/internasional, buku maupun dari situs-situs di internet.

Penulis menyadari bahwa dalam penyusunan dan penulisan skripsi ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir, sangatlah sulit untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis dengan senang hati menyampaikan terima kasih kepada:

- 1) Kedua Orang tua penulis, Bapak H. Kusumadiarto, S.E. dan Ibu Mince Mahmud Pakaya serta saudara-saudara penulis (Aurapurifi,, S.Si., dan Sujatorago) yang selalu memberikan dukungan, doa, dan semangat serta selalu sabar dalam mendidik penulis sejak kecil;

- 2) Bapak Dr.Eng. Muhammad Niswar, S.T., M.IT., selaku pembimbing I dan Bapak Dr. Amil Ahmad Ilham, S.T., M.IT., selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan perhatian yang luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir;
- 3) Bapak Dr. Amil Ahmad Ilham, S.T., M.IT., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan penulis;
- 4) Bapak Adnan, S.T., M.T. Ph.D., Dr.Ir. Ingrid Nurtanio, M.T., dan Dr.Eng. Zulkifli Tahir, S.T., S.Sc. yang telah menyempatkan waktunya memberikan saran kepada penulis;
- 5) Kanda Novianto Padaunan, S.T., M.T. dan Rizky Eka Arlin yang telah rela meluangkan waktu dan sabar membantu mengatasi masalah dalam pengerjaan tugas akhir ini;
- 6) Kanda Rusdiansyar Haerus yang tidak henti-hentinya memberikan semangat, perhatian yang tulus dan bantuan selama penyusunan tugas akhir ini;
- 7) Kepada teman seperjuangan Fathurrahman dan Hidayatul Jum'ah. Akhirnya bisa selesaikan tugas akhir ini;
- 8) Kepada squad giils (Inka G. Malissa ,Cindy O. Lolo Bulan, Anastasia Yuki Aprilia S., Maqhfira Putri R., S.T.) yang selalu memberi semangat menyelesaikan tugas akhir ini;

- 9) Para teman-teman dan kakak-kakak lab CCIE, SEIS, CBS, IOT, AIMP yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data dan diskusi *progress* penyusunan Tugas Akhir;
- 10) Teman KKN 96 Bontobahari Squad ( Hidayat Nazardi, SE., Rusnianti, S.Si., A. Diar Nur Azika, S.P., Muh. Yusuf Pratama, S.T. dan Sri Rafika ) yang selalu memberi semangat kepada penulis;
- 11) Keluarga angkatan 2014 Departemen Teknik Informatika FT UH atas semua bantuan dan semangat yang diberikan selama ini;
- 12) Teman-teman Rectifier FT UH atas dukungan dan semangat yang diberikan selama ini;
- 13) Segenap Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu penulis.
- 14) Orang-orang berpengaruh lainnya yang tanpa sadar telah menolong dan menjadi inspirasi penulis.

Akhir kata, penulis berharap semoga Allah SWT. berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. Semoga Tugas Akhir ini dapat memberikan manfaat bagi pengembangan ilmu. Aamiin.

*Wassalam*

Makassar, Januari 2019

Penulis

## DAFTAR ISI

	<b>halaman</b>
HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN .....	ii
ABSTRAK .....	iii
KATA PENGANTAR .....	iv
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Manfaat Penelitian .....	3
1.5 Batasan Masalah Penelitian .....	3
1.6 Sistematika Penulisan .....	3
BAB II TINJAUAN PUSTAKA .....	5
2.1 Modul Praktikum .....	5
2.2 Software Definde Network .....	7
2.3 <i>Raspberry PI</i> .....	10
2.4 <i>OpenFlow</i> .....	12

2.5 <i>OpenvSwitch</i> .....	14
2.6 Controller SDN .....	17
2.7 iPerf.....	18
2.8 Oracle VM Virtual Box .....	19
2.9 CentOS .....	22
2.10 Angry IP Scanner .....	24
2.11 QoS(Qualitiy of Service).....	25
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>29</b>
3.1 Tahapan Penelitian .....	29
3.2 Waktu dan Lokasi Penelitian .....	31
3.3 Instrumen Penelitian.....	31
3.4 Perancangan Implementasi Sistem.....	32
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>41</b>
4.1 Hasil Penelitian .....	41
4.2 Pembahasan .....	49
<b>BAB V PENUTUP</b> .....	<b>91</b>
A. Kesimpulan .....	52
B. Saran .....	52
<b>DAFTAR PUSTAKA</b> .....	<b>54</b>
<b>LAMPIRAN</b> .....	<b>94</b>



## DAFTAR GAMBAR

Nomor	halaman
<b>Gambar 2.1</b> .....	8
<b>Gambar 2.2</b> .....	11
<b>Gambar 2.3</b> .....	11
<b>Gambar 2.4</b> .....	14
<b>Gambar 2.5</b> .....	12
<b>Gambar 2.6</b> .....	16
<b>Gambar 2.7</b> .....	19
<b>Gambar 2.8</b> .....	20
<b>Gambar 2.9</b> .....	26
<b>Gambar 3.1</b> .....	31
<b>Gambar 3.2</b> .....	34
<b>Gambar 3.3</b> .....	35
<b>Gambar 3.4</b> .....	36
<b>Gambar 3.5</b> .....	36
<b>Gambar 3.6</b> .....	37
<b>Gambar 3.7</b> .....	37
<b>Gambar 3.8</b> .....	38
<b>Gambar 3.9</b> .....	39
<b>Gambar 3.10</b> .....	39
<b>Gambar 4.1</b> .....	42

<b>Gambar 4.2</b> .....	43
<b>Gambar 4.3</b> .....	43
<b>Gambar 4.4</b> .....	43
<b>Gambar 4.5</b> .....	48
<b>Gambar 4.6</b> .....	50
<b>Gambar 4.7</b> .....	51
<b>Gambar 4.8</b> .....	53

## **DAFTAR TABEL**

<b>Nomor</b>	<b>halaman</b>
<b>Tabel 2.1</b> .....	12
<b>Tabel 2.2</b> .....	27
<b>Tabel 2.3</b> .....	27
<b>Tabel 2.4</b> .....	28
<b>Tabel 2.5</b> .....	29
<b>Tabel 4.1</b> .....	44
<b>Tabel 4.2</b> .....	44
<b>Tabel 4.3</b> .....	45
<b>Tabel 4.4</b> .....	46

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Pada era modern saat ini, teknologi jaringan sangat dibutuhkan di berbagai tempat termasuk dalam dunia pendidikan. Namun, teknologi jaringan saat ini terus berkembang sesuai zamannya. Sehingga industri jaringan menciptakan arsitektur baru yang dikenal dengan *Software Defined Network* (SDN).

*Software Defined Network* (SDN) adalah pendekatan yang muncul pada jaringan komputer yang memisahkan *control plane* dan *data (forwarding) plane* dalam perangkat jaringan tradisional (Adel Nadjaran Toosi, dkk., 2017). Protokol yang digunakan oleh *Software Defined Network* (SDN) adalah *OpenFlow*. *OpenFlow* adalah protokol yang bertujuan untuk mengontrol *data plane switch*, yang telah dipisahkan secara fisik dari control plane menggunakan perangkat lunak *controller* pada sebuah server (Eko Pramono, dkk., 2014). *Controller* yang digunakan untuk *OpenFlow* bermacam-macam. Seperti *ryu*, *pox controller*, *Beacon*, dll. Tetapi hanya beberapa perangkat lunak *controller* yang mendukung *OpenFlow*.

Namun, untuk mengimplementasi jaringan menggunakan *Software Defined Network* (SDN)/*OpenFlow switch* akan menambah beban biaya yang tidak sedikit untuk switch IBM seri G8264 dengan jumlah port 48 port, diberi harga \$29999 atau sekitar Rp. 350.000.000,- (Eko Pramono, dkk., 2014).

Oleh karena itu untuk mengganti perangkat *OpenFlow switch*, kami menggunakan *Raspberry PI* sebagai *switch* virtual.

Jadi, dengan menggunakan *Raspberry PI* ini dapat menekan biaya pembuatan perangkat *Software Defined Network (SDN)* sehingga perangkat ini dapat menjadi media pembelajaran tentang *Software Defined Network (SDN)*. Maka dari itu, tugas akhir yang akan dilakukan berjudul “Perancangan *Software Defined Network (SDN) Switch* Menggunakan *Raspberry PI*”

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang, maka rumusan masalah pada tugas akhir ini adalah:

1. Bagaimana mengimplementasikan *OpenFlow* pada *Raspberry PI* sebagai *switch*?
2. Bagaimana merancang *switch* yang mendukung *Software Defined Network (SDN)* ?
3. Bagaimana kinerja *OpenFlow* pada *Raspberry PI*?

## **1.3. Tujuan Penelitian**

Tujuan dari tugas akhir ini adalah :

1. Untuk Mengimplementasi *OpenFlow* pada *Raspberry Pi* sebagai *switch*
2. Untuk membuat *switch* yang mendukung *Software Defined Network (SDN)* yang berbiaya rendah.
3. Untuk mengetahui kinerja *OpenFlow* pada *Raspberry pi*

#### **1.4. Manfaat Penelitian**

Manfaat dari tugas akhir ini adalah :

- a. Manfaat untuk peneliti
  1. Memberikan kontribusi kepada para peneliti selanjutnya untuk mengembangkan ke arah sistem yang lebih terintegrasi.
- b. Manfaat untuk instansi
  1. Memberikan media dan bahan pembelajaran mengenai *Software Defined Network (SDN)* dan *OpenFlow*.
- c. Manfaat untuk mahasiswa
  1. Memberikan referensi dan acuan untuk memahami proses *Software Defined Network (SDN)*.

#### **1.5. Batasan Masalah Penelitian**

Yang menjadi batasan masalah dalam tugas akhir ini adalah :

1. Pembuatan *OpenFlow* dalam *Raspberry PI* menggunakan *Open Vswitch 2.10.0*.
2. *Switch* dirakit menggunakan *Raspberry PI 3* model B.
3. *Controller* yang digunakan adalah *POX Controller*.
4. Operasi sistem yang digunakan di *Raspberry PI* adalah *Rasbian Jessie*
5. Operasi sistem yang digunakan untuk controller *POX* adalah *CentOS7*

## **1.6. Sistematika Penulisan**

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu :

### **BAB I PENDAHULUAN**

Bab ini menguraikan secara umum mengenai hal yang menyangkut latar belakang, perumusan masalah dan batasan masalah, tujuan, manfaat, dan sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori tentang hal-hal yang berhubungan dengan *Android*, Pemrosesan Citra dan metode yang digunakan.

### **BAB III METODOLOGI PENELITIAN**

Bab ini berisi tentang perencanaan dan penerapan algoritma serta teknik pengolahan data.

### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil pengolahan data serta pembahasan yang disertai tabel hasil penelitian.

### **BAB V PENUTUP**

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Modul Praktikum**

Menurut Kamus Besar Bahasa Indonesia (KBBI), modul adalah kegiatan belajar-mengajar yang dapat dipelajari oleh murid dengan bantuan yang minimal dari guru pembimbing, meliputi perencanaan tujuan yang akan dicapai secara jelas, penyediaan materi pelajaran, alat yang dibutuhkan, serta alat untuk menilai dan mengukur keberhasilan murid dalam penyelesaian pelajaran. Secara umum, modul praktikum merupakan bahan ajar yang disusun secara sistematis dan menarik yang mencakup isi materi, metode dan evaluasi yang dapat digunakan secara mandiri untuk mencapai kompetensi yang diharapkan.

Pedoman penulisan modul yang dikeluarkan oleh Direktorat Pendidikan Menengah Kejuruan, Direktorat Jendral Pendidikan Dasar dan Menengah, Departemen Pendidikan Nasional Tahun 2003, agar modul mampu meningkatkan motivasi dan efektifitas penggunaannya harus memiliki kriteria sebagai berikut:

1. *Self Instructional*, yaitu melalui modul tersebut seseorang atau peserta didik mampu belajar secara mandiri dan tidak tergantung pada pihak lain. Untuk memenuhi karakter self instruction, maka modul harus:
  - a. Membuat tujuan yang jelas, dan dapat menggambarkan pencapaian Standar Kompetensi dan Kompetensi Dasar.



- b. Memuat materi pembelajaran yang dikemas dalam unit-unit kegiatan yang kecil/spesifik, sehingga memudahkan dipelajari secara tuntas.
  - c. Tersedia contoh dan ilustrasi yang mendukung kejelasan pemaparan materi pembelajaran.
  - d. Terdapat soal-soal latihan, tugas, dan sejenisnya yang memungkinkan untuk mengukur penguasaan siswa.
  - e. Kontektual, yaitu materi yang disajikan terkait dengan suasana, tugas atau konteks kegiatan dan lingkungan siswa.
  - f. Menggunakan bahasa yang sederhana dan komunikatif
  - g. Terdapat rangkuman materi pembelajaran
2. *Self Contained*, yaitu seluruh materi pembelajaran dari satu unit kompetensi atau subkompetensi yang dipelajari terdapat di dalam satu modul secara utuh. Tujuan dari konsep ini adalah memberikan kesempatan kepada peserta didik untuk mempelajari materi pembelajaran secara tuntas, karena materi dikemas dalam satu kesatuan yang utuh. Jika harus dilakukan pembagian atau pemisahan materi dari satu unit kompetensi harus dilakukan dengan hati-hati dan memperhatikan keluasan kompetensi yang harus dikuasai.
3. *Stand Alone*, yaitu modul yang dikembangkan tidak tergantung kepada media lain atau tidak harus digunakan bersama-sama dengan media pembelajaran lain. Dengan menggunakan modul, peserta didik tidak

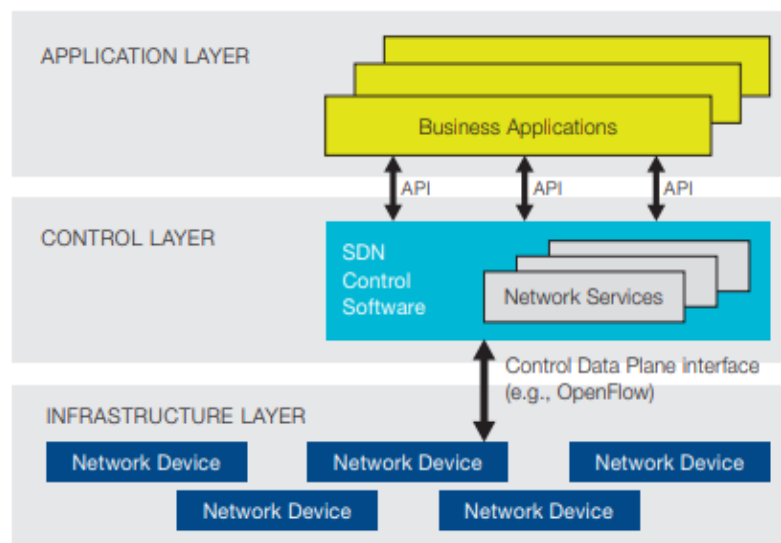
tergantung dan harus menggunakan media yang lain untuk mempelajari dan atau mengerjakan tugas pada modul tersebut. Jika masih menggunakan dan bergantung pada media lain selain modul yang digunakan, maka media tersebut tidak dikategorikan sebagai media yang berdiri sendiri.

4. *Adaptive*, yaitu modul hendaknya memiliki daya adaptif yang tinggi terhadap perkembangan ilmu dan teknologi. Dikatakan adaptif jika modul dapat menyesuaikan perkembangan ilmu pengetahuan dan teknologi, serta fleksibel digunakan. Dengan memperhatikan percepatan perkembangan ilmu dan teknologi, pengembangan modul multimedia hendaknya tetap bersifat kekinian (*up to date*). Modul yang adaptif adalah jika isi materi pembelajaran dapat digunakan sampai dengan kurun waktu tertentu.
5. *User Friendly*, yaitu setiap instruksi dan informasi yang ditampilkan dalam modul hendaknya bersifat membantu dan bersahabat dengan penggunanya, termasuk kemudahan pengguna dalam merespon dan mengakses sesuai dengan keinginan. Penggunaan bahasa yang sederhana, mudah dimengerti serta menggunakan istilah yang umum digunakan merupakan salah satu bentuk *user friendly*.

## 2.2. Software Defined Network (SDN)

Software Defined Network (SDN) adalah jaringan komputer yang memisahkan *controller* dan *data plane* dalam perangkat jaringan konvensional. Berkat pemisahan ini, SDN dapat memberikan tampilan jaringan terpusat secara logis dalam satu titik pengelolaan. Hal ini dicapai melalui *open interface* dan abstraksi fungsi tingkat rendah dan mengubah jaringan menjadi *platform* yang dapat diprogram secara dinamis menyesuaikan tingkah lakunya (Adel Nadjaran Toosi, dkk., 2017).

Gambar dibawah ini adalah arsitektur SDN, memiliki 3 layer yaitu :

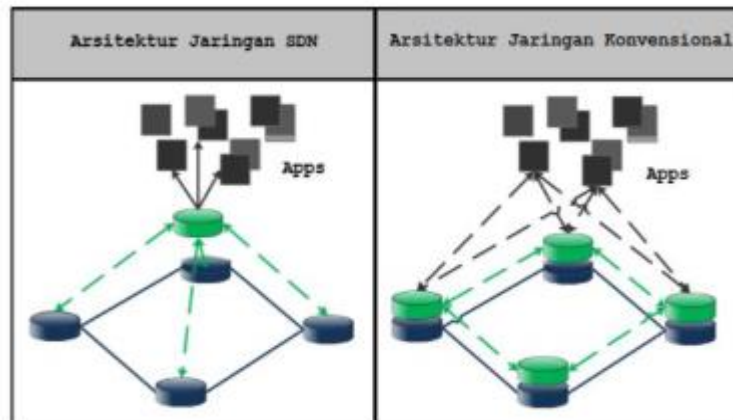


**Gambar 2.1.** Arsitektur SDN (Open Networking Foundation, 2012)

1. Application layer : Berupa aplikasi yang dapat secara langsung dan eksplisit mendefinisikan network requirement dan network behavior yang diinginkan. Layer ini berkomunikasi dengan Control Layer melalui NorthBound Interface (NBI).

2. Control layer : Memiliki tugas mentranslasikan network requirement yang telah didefinisikan oleh application layer menjadi instruksi-instruksi untuk infrastructure layer dan memberikan abstract view yang dibutuhkan application layer (abstract view yaitu informasi statistik dan event yang terjadi di jaringan).
  3. Infrastructure layer : terdiri dari elemen jaringan yang dapat menerima instruksi dari Control Layer. Interface antara Controller layer dan Data Plane disebut SouthBound Interface (SBI) atau Control-To-Data-Plane Interface (CDPI) (Desianto Abdillah, dkk., 2016).
- a. Keuntungan menggunakan Software Defined Network yaitu :
1. Operator jaringan dan administrator dapat secara terprogram mengkonfigurasi jaringan yang disederhanakan. Jadi tidak perlu menkonfigurasi yang ribuan perangkat.
  2. Dapat merubah tingkah laku jaringan secara real time serta dapat menyebarkan aplikasi baru dan layanan jaringan dalam hitungan jam atau hari.
  3. Dapat memberikan fleksibilitas kepada pengelola jaringan untuk mengkonfigurasi, mengelola, mengamankan dan mengoptimalkan sumberdaya jaringan melalui program SDN yang dinamis dan otomatis

4. Mendukung satu set API yang memungkinkan untuk mengimplementasikan layanan jaringan umum, termasuk routing, multicast, keamanan, kontrol akses, manajemen *bandwidth*, rekayasa lalu lintas, kualitas layanan, prosesor dan optimasi penyimpanan, penggunaan energi, dan semua bentuk kebijakan manajemen, kustom disesuaikan untuk memenuhi tujuan bisnis
  
  5. SDN membuat jaringan tidak begitu banyak “*application-aware*” sebagai “*application-customized*” dan aplikasi tidak begitu banyak “*network-aware*” sebagai “*network-capability-aware*”. Akibatnya, komputasi, penyimpanan, dan sumber daya jaringan dapat dioptimalkan (Open Networking Foundation, 2014)
- b. Perbedaan *Software Defined Network* dan Jaringan konvensional
1. Pada Gambar 2.2 memperlihatkan perbedaan arsitektur SDN dan jaringan konvensional. Arsitektur SDN memisahkan *control plane* dan *data plane* sedangkan jaringan konvensional *control plane* dan *data plane* berada di tempat yang sama. Selain itu control plane pada SDN menggunakan struktur terentralisasi sedangkan untuk control plane jaringan konvensional menggunakan struktur terdistribusi.



**Gambar 2.2.** Arsitektur SDN dan Jaringan Konvensional

- Perbedaan terletak pada arsitektur layer yang digunakan. Terlihat pada gambar 2.3 bahwa jaringan dengan arsitektur SDN menggunakan 3 layer yang terdiri dari *Application*, *Control* dan *Physical*. Sedangkan pada arsitektur jaringan konvensional menggunakan 4 layer yang terdiri dari *Application*, *Transport*, *Internet* dan *Network*. (Ahmad Mahardika, 2016)

SDN	TCP/IP
Application	Application
Control Layer	Transport
	Internet
Physical	Network Access

**Gambar 2.3.** Perbedaan Arsitektur *Layer* SDN dan TCP/IP

### 2.3. Raspberry PI

*Raspberry PI* adalah sebuah SBC (*Single Board Computer*) seukuran kartu kredit yang dikembangkan oleh Yayasan *Raspberry PI* di Inggris (UK) dengan

maksud untuk memicu pengajaran ilmu pengetahuan dasar komputer disekolah-sekolah. *Raspberry PI* diproduksi melalui lisensi manufaktur yang berkaitan dengan elemen 14/Premier Farnell dan RS komponen (Yasir Arafad, 2017)



**Gambar 2.4.** Raspberry PI (<https://www.raspberrypi.org/products>)

### 2.3.1. *Raspberry PI 3*

*Raspberry PI 3* adalah generasi ketiga dari *Raspberry PI*, menggantikan *Raspberry PI 2 Model B* pada Februari 2016. *Raspberry PI3* memiliki bentuk yang identik dengan *Raspberry PI 2* sebelumnya (dan *Pi 1 Model B +*) dan memiliki kompatibilitas lengkap dengan *Raspberry PI 1* dan *2*. Pada perangkat terbarunya ini Raspberry menambahkan fitur built- in wireless dan processor yang lebih bertenaga yang belum pernah dimiliki pada versi sebelumnya. Spesifikasi terdapat dilihat pada tabel 2.1 (Yasir Arafad, 2017).

**Tabel 2.1** Spesifikasi *Raspberry PI Model B*

Spesifikasi	Keterangan
<b>SOC (System On Chip)</b>	BCM2837
<b>Prosesor</b>	1.2GHz 64-bit quad-core ARMv8 CPU
<b>Memory/RAM</b>	1 GB SDRAM 400MHz

<b>Spesifikasi</b>	<b>Keterangan</b>
<b>GPU</b>	VideoCore IV 3D graphics core
<b>Wireless Adapter/LAN</b>	802.11n Wireless LAN
<b>Bluetooth</b>	Bluetooth 4.1 (built in), Bluetooth Low Energy(BLE)
<b>GPIO</b>	40 Pin
<b>Port USB</b>	4 USB Port
<b>Card Storage</b>	Micro SD card slot (now push-pull rather than pushpush)
<b>Port Ethernet</b>	1 Ethernet Port

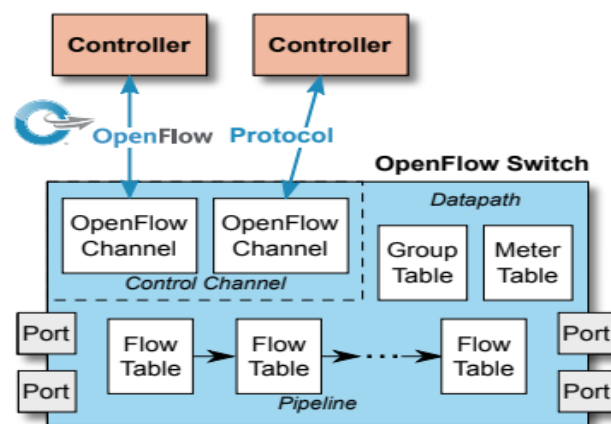
### 2.3.2. Sistem Operasi Raspberry PI

Sistem operasi yang digunakan oleh *Raspberry PI* biasa disebut Raspbian. Sistem operasi bebas berbasis Debian GNU / LINUX dan dioptimalkan untuk perangkat keras *Raspberry PI* (arsitektur prosesor ARMHF). Raspbian dilengkapi dengan lebih dari 35.000 paket, atau perangkat lunak precompiled paket dalam format yang bagus untuk kemudahan instalasi pada *Raspberry PI*. Awal di rilis sejak Juni 2012, menjadi distribusi yang terus aktif dikembangkan dengan penekanan pada peningkatan stabilitas dan kinerja sebanyak mungkin. Meskipun Debian menghasilkan distribusi untuk arsitektur lengan, Raspbian hanya kompatibel dengan versi yang lebih baru dari yang digunakan pada *Raspberry PI* (ARMv7 CPU-A dan vs *Raspberry PI* ARMv6 CPU yang lebih tinggi) (Yasir Arafad, 2017).



## 2.4. OpenFlow

Pada tahun 2008 Stanford University menyimpulkan bahwa jaringan menjadi infrastruktur yang sangat penting. Namun, inovasi jaringan sangat terhambat oleh karena itu sebagai solusi ide virtualisasi jaringan di usulkan di bagian produksi dan eksperimental untuk membuat penelitian *OpenFlow*.



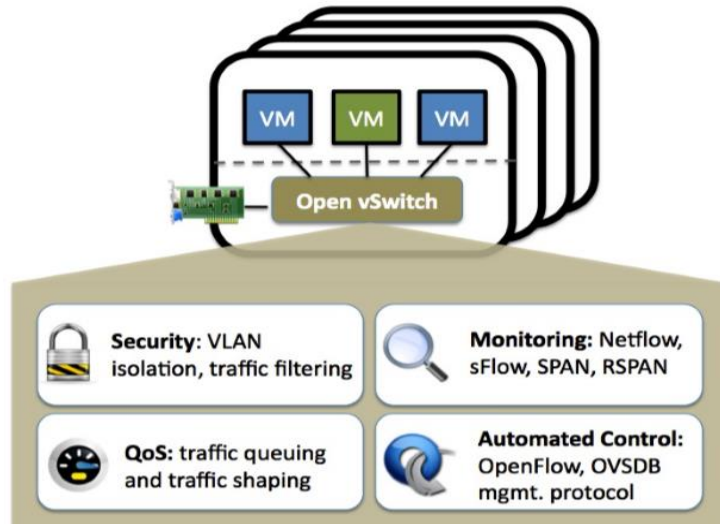
**Gambar 2.5.** Komponen utama dari Switch *OpenFlow* (Open Networking Foundation, 2015)

Komponen utama switch *OpenFlow* v.1.5.1 adalah sebagai berikut

- a. *OpenFlow* Channel : Interface yang menghubungkan setiap *OpenFlow* switch ke *OpenFlow* controller. Melalui interface ini dapat mengkonfigurasi controller, mengelola switch, menerima event dari switch dan mengirim paket keluar. Pengontrol channel switch dapat mendukung single *OpenFlow* channel dengan singgel controller atau beberapa *OpenFlow* channel yang memungkinkan beberapa controller untuk berbagi management switch.

- b. Group Table : Sebuah group table terdiri group entri. Dimana flow entri memiliki ke mampuan untuk mengarahkan group untuk memungkinkan *OpenFlow* mewakili metode forwarding.
- c. Meter Table : Terdiri dari entri meter yang berfungsi untuk mendefinisikan flow meter. Flow meter memungkinkan *OpenFlow* untuk mengimplementasikan berbagai simple operasi QoS, seperti rate-limiting dan dapat di kombinasikan port queue untuk mengimplementasikan kerangka kerja QoS yang kompleks, seperti Diffserv.
- d. Flow Table : Terdiri dari match fields berfungsi untuk mencocokkan paket, priority untuk mencari kecocokan dengan flow entry, counter untuk melakukan update saat pencocokan, instruction untuk untuk memodifikasi set action atau pemrosesan pipeline, timeout adalah jumlah waktu maksimum atau idle sebelum flow kadaluarsa oleh switch, cookie adalah nilai data opaque yang dipilih controller, dan flag berfungsi untuk mengubah manajemen flow entrie (Open Networking Foundation, 2015).

## 2.5. OpenvSwitch



**Gambar 2.6.** OpenvSwitch (OpenvSwitch, 2018)

OpenvSwitch adalah perangkat lunak multilayer yang berlisensi dibawah lisensi opensource apache2. Tujuannya adalah untuk mengimplementasikan platform switch yang mendukung standar interface management dan membuka fungsi forwarding ke ekstensi yang terprogram dan terkontrol.

Openvswitch sangat berfungsi untuk switch virtual di VM. Selain mengekspos standar control dan interface visibilitas ke virtual networking layer, juga di rancang untuk mendukung distribusi di beberapa server fisik. OpenvSwitch mendukung beberapa teknologi virtualisasi berbasis linux termasuk Xen/Xen Server, KVM dan Virtual Box.

Sebagian besar kode ditulis dalam bahasa C dan mudah dipindah ke bahasa lain. OpenvSwitch yang rilis saat ini mendukung fitur-fitur berikut :

- a. Standard 802.1Q model VLAN dengan trunk dan access ports
- b. NIC bonding dengan atau tanpa LACP pada upstream switch

- c. NetFlow, sFlow(R), dan pencerminan untuk peningkatan visibility
- d. QoS (Quality of Service) konfigurasi, dan policing
- e. Geneve, GRE, VXLAN, STT, dan LISP tunneling
- f. 802.1ag manajemen kesalahan konektivitas
- g. *OpenFlow* 1.0 ditambahkan banyak ekstensi
- h. Konfigurasi database transactional dengan bahasa C dan Python bindings
- i. Kinerja forwarding menjadi lebih baik dengan menggunakan modul kernel linux ( hanya mendukung versi linux 3.10 dan versi lebih baru)

OpenvSwitch juga dapat beroperasi sepenuhnya di userspace tanpa bantuan dari modul kernel. Implementasi userspace harus lebih mudah ke port daripada switch berbasis kernel. OpenvSwitch didalam userspace dapat mengakses linux atau perangkat DPDK (Data Plane Development Kit). OpenvSwitch dengan userspace data path dan perangkat non DPK dianggap sebagai eksperimental.

Komponen utama OpenvSwitch yaitu :

- a. ovs-vswitchd : Sebuah daemon yang mengimplementasikan switch, bersama dengan modul kernel linux untuk flowbased switch.
- b. ovsdb-server : Server data base yang ringan bahwa ovs-vswitchd memiliki pertanyaan untuk memperoleh konfigurasinya.
- c. ovs-dpctl : alat untuk mengkonfigurasi modul kernel switch
- d. script and spec: Membangun RPM untuk Citrix XenServer dan Red Hat Enterprise Linux. RPM XenServer memungkinkan OpenvSwitch untuk di instal pada host Citrix XenServer sebagai pengganti drop-in untuk switch.

- e. `ovs-vsctl` : Kegunaanya untuk query dan memperbarui konfigurasi `ovs-vswitchd`.
- f. `ovs-appctl` : Mengirimkan perintah untuk menjalankan daemon `openvswitch`

OpenvSwitch juga menyediakan beberapa tool, yaitu :

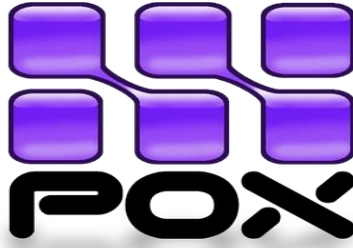
- a. `ovs-ofctl` : Kegunaanya untuk query dan mengontrol switch *OpenFlow* dan controller
- b. `ovs-pki` : Kegunaanya untuk membuat dan mengelola infrastruktur public-key (openvswitch, 2018).

## **2.6. Controller SDN**

Entitas perangkat lunak yang memiliki kontrol eksklusif atas serangkaian abstraksi dari sumber daya data plane. Controller SDN juga dapat menawarkan contoh model informasi yang diabstraksi, setidaknya ke satu klien. Kontroler SDN dapat diimplementasikan dengan sejumlah komponen perangkat lunak, yang berada pada sejumlah platform fisik (Open Networking Foundation, 2014)

### **2.6.1. POX controller**

POX adalah platform yang digunakan untuk pengembangan dan pemodelan pada network kontrol software. POX menggunakan python dalam bahasa pemrogramannya. POX bekerja pada layer control plan sebagai sebuah network controller. Dalam arsitektur RouteFlow POX berada pada RFPProxy yang bertanggung jawab untuk interaksi dengan *OpenFlow* switch (datapath) melalui protokol *OpenFlow* (Rohmat Tulloh, dkk., 2015).



**Gambar 2.7.** *POX Controller* (Rohmat Tulloh, dkk., 2015)

## **2.7. iPerf**

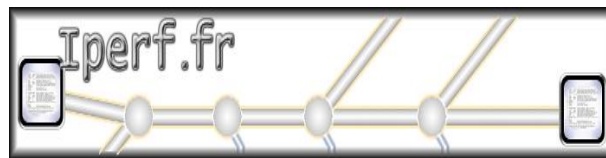
iPerf dikembangkan secara oronal oleh NLANR atau DAST. Namun pada prinsipnya, iPerf dikembangkan oleh ESnet atau Lawrence Berkeley National Laboratory dan dirilis di bawah lisensi BSD three-clause.

iPerf adalah perangkat lunak untuk pengukuran aktif dari bandwidth maksimum yang dapat dicapai pada jaringan. Mendukung penyetelan berbagai parameter yang terkait dengan pengaturan waktu, buffer dan protokol (TCP, UDP, SCTP dengan IPv4 dan IPv6). Setiap tes itu melaporkan bandwidth, kerugian, dan parameter lainnya.

Berikut beberapa fitur iPerf :

1. TCP dan SCTP
  - a. Mengukur bandwidth
  - b. Melaporkan ukuran MSS/MTU dan mengukur ukuran baca
  - c. Mendukung ukuran TCP melalui socket buffer.
2. UDP
  - a. Klien dapat membuat UDP dari bandwidth yang ditentukan.
  - b. Menghitung packet loss
  - c. Menghitung delay jitter

- d. Mampu mengirim data dalam waktu yang bersamaan
3. Dapat digunakan di Windows, Linux, Android, MacOS X, FreeBSD, OpenBSD, NetBSD, VxWorks, Solaris, dll.
4. Klien dan server dapat memiliki beberapa koneksi simultan (opsi -P).
5. Server menangani banyak koneksi, dalam kali satu tes.
6. Dapat menentukan waktu pengiriman data (opsi -t), dan jumlah data untuk ditransfer (-n atau opsi -k) (<https://iperf.fr/>).



**Gambar 2.8.** iPerf (<https://iperf.fr/>)

## **2.8. Oracle VM Virtual Box**

Oracle VM VirtualBox adalah perangkat lunak virtualisasi cross-platform yang dapat menjalankan beberapa sistem operasi seperti windows, mac, dan linux pada saat yang bersamaan sehingga memperluas kemampuan komputer yang ada. Didesain untuk digunakan pada sistem mulai dari ultra-book hingga hardware kelas server high-end, Oracle VM VirtualBox ringan dan mudah dipasang dan digunakan (Oracel Virtualitation, 2018).

Berikut beberapa keuntungan menggunakan Oracle virtual box :

1. Menjalankan beberapa sistem operasi secara bersamaan : VirtualBox dapat menjalankan lebih dari satu sistem operasi secara bersamaan tanpa harus reboot untuk menggunakannya

2. Instalasi perangkat lunak lebih mudah : Dengan VirtualBox, pengaturan yang begitu kompleks dapat dimasukkan ke dalam mesin virtual.
3. Pengujian dan pemulihan kerusakan : Dengan cara ini dapat dengan bebas bereksperimen dengan lingkungan komputasi. Jika ada yang salah dapat dengan mudah beralih kembali ke pengaturan sebelumnya dan menghindari backup dan restores yang terlalu sering.
4. Konsolidasi infrastruktur : Virtualisasi dapat secara signifikan mengurangi biaya perangkat keras dan listrik ((Oracle Virtualization, 2018).)

Fitur utama Oracle VM virtual box :

1. Portabilitas : VirtualBox berjalan pada sejumlah sistem operasi 32-bit dan 64-bit. VirtualBox adalah apa yang disebut "host" hypervisor atau hypervisor tipe 2. VirtualBox memerlukan sistem operasi yang sudah ada untuk diinstal. Dengan demikian dapat berjalan bersama aplikasi yang ada pada host itu.
2. Tidak diperlukan virtualisasi hardware : VirtualBox tidak memerlukan fitur prosesor yang dibangun ke dalam perangkat keras yang lebih baru seperti Intel VT-x atau AMD-V. Berbeda dengan banyak virtualisasi lainnya, dapat menggunakan VirtualBox bahkan pada perangkat keras yang lebih lama di mana fitur-fitur ini tidak ada.
3. Guest Addition : VirtualBox Guest Additions adalah paket perangkat lunak yang dapat diinstal di dalam sistem tamu yang didukung untuk



meningkatkan kinerja mereka dan untuk memberikan integrasi dan komunikasi tambahan dengan sistem host.

4. Dukungan perangkat keras yang hebat. VirtualBox mendukung:
  - a. Guest Multiprocessing : VirtualBox dapat menyajikan hingga 32 CPU virtual ke setiap mesin virtual, terlepas dari berapa banyak core CPU yang secara fisik ada pada host Anda.
  - b. Dukungan perangkat USB : VirtualBox mengimplementasikan pengendali USB virtual dan memungkinkan untuk menghubungkan perangkat USB ke mesin virtual tanpa harus menginstal driver khusus pada host. Dukungan USB tidak terbatas pada kategori perangkat tertentu.
  - c. Dukungan ACPI : Advanced Configuration and Power Interface (ACPI) didukung penuh oleh VirtualBox. Ini memudahkan kloning gambar PC dari mesin asli sehingga mesin virtual pihak tiga menjadi VirtualBox. Dengan dukungan status daya ACPI yang unik, VirtualBox bahkan dapat melaporkan ke sistem operasi tamu yang mengetahui ACPI status daya dari host. Untuk sistem seluler yang menggunakan baterai, tamu dapat mengaktifkan penghematan energi dan memberi tahu pengguna tentang kekuatan yang tersisa.
  - d. Resolusi Multilayer : VirtualBox mesin virtual mendukung resolusi multilayer dari layar fisik, memungkinkan untuk tersebar di sejumlah besar layar yang terpasang ke sistem host.

- e. Dukungan iSCSI internal : Fitur unik ini memungkinkan untuk menghubungkan mesin virtual langsung ke server penyimpanan iSCSI tanpa melalui sistem host. VM mengakses target iSCSI secara langsung tanpa tambahan biaya yang diperlukan untuk mem-virtualisasi hard disk dalam file container.
- f. PXE Network boot : Kartu virtual network terintegrasi dari VirtualBox sepenuhnya mendukung booting jarak jauh melalui Preboot Execution Environment (PXE) ([Oracle Virtual Box, 2017](#)).

## **.2.9. CentOS**

CentOS (Community ENTERprise Operating System) adalah sebuah operasi sistem bawaan linux. Awalnya dibangun dengan nama Red Hat Enterprise Linux (RHEL). Untuk mengunduh RHEL harus memiliki akun berlangganan RHEL yang aktif dan tidak murah. RHEL sepenuhnya didasarkan pada perangkat lunak open source. Sehingga RHEL berkewajiban untuk melepaskan semua sumber kodenya kepada siapa saja yang berlangganan.

Namun, RHEL membuat sumber kode yang tersedia bagi siapa saja. Dengan menggunakan undang-undang perlindungan merek dagang untuk mencegah orang membangun RHEL versi lain. Kemudian mendistribusikannya dengan berbagai cara dan dilakukan banyak penelitian untuk menghapus semua logo merek dagang Red Hat dan menjadikan namanya CentOS. CentOS dapat diunduh secara gratis dari situs web CentOS.

CentOS dan RHEL memiliki banyak kesamaan, Versi baru RHEL dirilis setiap 18-24 bulan. Sedangkan CentOS sekitar sebulan setelah tanggal rilis RHEL karena proyek CentOS harus membangun dan melakukan pengujian kembali. Proyek CentOS terdiri dari sukarelawan yang melakukan rekondisi ini, menguji pembaruan, dan memberikan dukungan (Peter Membrey, dkk., 2009).

Pembaruan kecil juga dilakukan setiap enam bulan atau lebih, dan ini termasuk semua pembaruan yang dari rilis awal, dukungan perangkat keras tambahan, dan setiap peningkatan teknologi atau fitur baru. CentOS memiliki komunitas dunia yang sangat kuat dengan lebih dari 6 juta sistem terdaftar. Jadi banyak sistem yang diinstal dari disk yang sama dan banyak perusahaan yang mengatur repositori lokal mereka sendiri untuk pembaruan, sehingga diperkirakan ada lebih banyak sistem CentOS di luar perkiraan (Peter Membrey, dkk., 2009).

Berikut perangkat lunak repository yang tersedia :

1. The CentOS Plus repository : berisi paket yang menggantikan paket asli yang dibangun kembali dari kode sumber RHEL. Ini memecah kompatibilitas biner, jadi bisa mendapatkan perangkat lunak baru yang tidak tersedia.
2. The CentOS Extras repository : berisi paket yang digunakan selain paket asli. Menggunakan repositori ini tidak akan merusak kompatibilitas biner, karena tidak akan memperbarui paket asli.

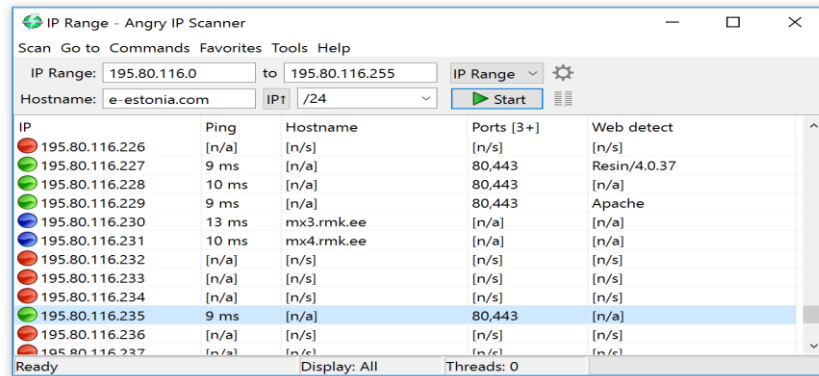
Tentu saja, ada beberapa kerugian ketika menggunakan CentOS di atas RHEL, yaitu :

1. CentOS sebagai proyek komunitas tidak dapat memberikan tingkat dukungan yang sama dengan entitas komersial seperti Red Hat
2. CentOS harus menerima pembaruan RHEL sebelum pembaruan CentOS dapat dibuat. Namun, penundaan ini tidak signifikan, dan meskipun Proyek CentOS memiliki target 72 jam, pembaruan biasanya dirilis dalam 24 jam.
3. CentOS tidak memiliki semua sertifikasi perangkat keras dan perangkat lunak yang tersedia di RHEL  
  
CentOS tidak menawarkan program ganti rugi jika ada IP atau kode yang melanggar (Peter Membrey, dkk., 2009).

## **2.10. Angry IP Scanner**

Angry IP Scanner adalah pemindai alamat IP dan port yang sangat cepat dengan rentang alamat IP apapun. Termasuk cross platform yang sangat ringan, sehingga mudah digunakan dimana saja. Juga memiliki fungsi seperti informasi NetBIOS (nama computer, nama workgroup), rentang alamat ip favorit, deteksi server web, dll.

Hasil pemindaian dapat disimpan ke file daftar CSV, TXT, XML atau IP-Port. Dengan bantuan plugin, Angry IP Scanner dapat mengumpulkan informasi apa pun tentang IP yang dipindai. Untuk meningkatkan kecepatan pemindai, menggunakan pendekatan multithread. Yaitu sebuah thread pemindaian terpisah yang dibuat untuk setiap alamat IP yang dipindai (<https://angryip.org>).



**Gambar 2.9.** Angry IP Scanner (<https://angryip.org>)

## 2.11. QoS (Quality of Service)

QoS adalah metode pengukuran jaringan atau mendefinisikan karakteristik dan sifat dari satu layanan. QoS digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu layanan. QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada traffic jaringan tertentu melalui teknologi yang berbeda-beda. QoS menawarkan kemampuan untuk mendefinisikan atribut-atribut layanan jaringan yang disediakan, baik secara kualitatif maupun kuantitatif (Eri Prasetyo, dkk., 2016).

Berikut parameter dari QoS :

1. Throughput : kecepatan (rate) transfer data efektif yang berhasil diterima , yang diukur dalam satuan waktu. Kategori Throughput diperlihatkan di Tabel 2.2 (Eri Prasetyo, dkk., 2016).

**Tabel 2.2.** Kategori Throughput

<b>Indeks</b>	<b>Throughput (%)</b>	<b>Kategori</b>
4	100	Sangat Bagus
3	75	Bagus
2	50	Kurang Bagus
1	<25	Jelek

(sumber : TIPHON)

- Rumus menghitung throughput

$$\text{Throughput} = \frac{\text{Jumlah Data yang Dikirim}}{\text{Waktu Pengiriman Data}} \times 100\% (1)$$

2. Paket Loss : parameter yang menunjukkan kondisi jumlah total paket yang hilang, dapat terjadi karena collision dan congestion pada jaringan. Kategori Paket loss diperlihatkan pada Tabel 2.3 (Eri Prasetyo, dkk., 2016).

**Tabel 2.3.** Kategori Paket Loss

<b>Indeks</b>	<b>Paket Loss (%)</b>	<b>Kategori</b>
4	0	Sangat Bagus
3	3	Bagus
2	15	Kurang Bagus
1	25	Jelek

(Sumber : TIPHON)

- Rumus menghitung packet loss

$$\text{Paket Loss} = \frac{\text{Paket Data yang Dikirim} - \text{Paket Data yang di terima}}{\text{Paket Data yang Dikirim}} \times 100\% \quad (2)$$

3. Delay : Waktu yang dibutuhkan paket untuk mencapai tujuan, karena adanya antrian, atau mengambil rute yang lain untuk menghindari kemacetan. . Kategori Delay diperlihatkan pada Tabel 2.4 (Eri Prasetyo, dkk., 2016)

**Tabel 2.4.** Kategori Delay/Latency

<b>Indeks</b>	<b>Besar Delay (ms)</b>	<b>Katerogi</b>
4	<150	Sangat Bagus
3	150-300	Bagus
2	300-450	Kurang Bagus
1	>450	Jelek

- Rumus menghitung Delay

$$\text{Delay} = \frac{\text{Total Delay}}{\text{Total Paket yang diterima}} \quad (3)$$

4. Jitter : Akibat yang terjadi dari variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket di akhir perjalanan. Kategori Jitter diperlihatkan pada Tabel 2.5 (Eri Prasetyo, dkk., 2016).

**Tabel 2.5.** Katergori Jitter

<b>Indeks</b>	<b>Besar Jitter (ms)</b>	<b>Kategori</b>
4	<150	Sangat Bagus
3	150-300	Bagus
2	300-450	Kurang Bagus
1	>450	Jelek

(Sumber : TIPHON)

- Rumus Menghitung Jitter

$$\text{Jitter} = \frac{\text{Total Variasi Delay}}{\text{Total Paket yang Diterima}} \quad (4)$$

$$\text{Total Variasi Delay} = \text{Delay} - (\text{Rata-Rata Delay})$$

## 2.8. Apache Jmeter

Apache JMeter merupakan aplikasi desktop yang berjenis open source. Apache JMeter merupakan aplikasi Java yang didesain untuk membuat traffic jaringan. Apache JMeter biasanya digunakan untuk mensimulasikan pemberian beban yang berat ke server . Apache JMeter dapat melakukan pengujian berbagai protocol sebagai berikut :

1. HTTP (Hyper Text Transfer Protocol) HTTP adalah komunikasi protocol yang memfasilitasi transfer informasi ke internet. HTTP juga merupakan protocol requestresponse (permintaan dan tanggapan) antara klien dan server.



2. HTTPS (Hyper Text Transfer Protocol Secure) Memiliki pengertian yang sama dengan HTTP hanya saja HTTPS memiliki fungsi lebih dibidang keamanan. HTTPS menggunakan Secure Socket Layer (SSL) atau Transport Layer Security (TLS) sebagai sub layer dibawah HTTP. HTTP dienkripsi dan dekripsi dari halaman yang diminta oleh pengguna dan halaman yang dikembalikan oleh web server.
3. SOAP (Simple Object Transfer Protocol) SOAP adalah protocol untuk pertukaran informasi dengan desentralisasi dan terdistribusi. SOAP dibangun dengan menggunakan protocol komunikasi HTTP. Karena HTTP didukung oleh semua browser dan server, maka SOAP dapat berkomunikasi dengan berbagai aplikasi meskipun terdapat perbedaan sistem operasi, teknologi dan bahasa pemrogramannya.
4. FTP (File Transfer Protocol) FTP adalah suatu protocol yang berfungsi untuk tukar-menukar file dalam suatu jaringan yang menggunakan TCP koneksi bukan UDP.
5. LDAP (Light weight Directory Access Protocol) LDAP merupakan bagian dari protocol internet yang digunakan untuk mengakses suatu directory.
6. TCP (*Transmission Control Protocol*) Sebuah *protocol* yang mendukung komunikasi antar jaringan yang memungkinkan antara satu komputer dengan komputer yang lain dapat saling bertukar data dalam suatu jaringan. ( Reza, dkk., 2015).

## **2.9 Load Average**

*Load average* merupakan beban rata-rata sebuah sistem dalam menjalankan banyak proses pada sebuah PC. *Load average* dinyatakan dalam 3 nilai yang mewakili periode waktu 1 menit, 5 menit dan 15 menit. Angka tersebut berkaitan erat dengan beberapa banyak *core* CPU pada sebuah PC. Semakin banyak *core* CPU, maka semakin semakin bagus semua PC menjalankan banyak proses. Beberapa perintah yang menyediakan analisis statik CPU adalah *top* dan *uptime*. Beberapa parameter dari perintah *top* yaitu, *top*, *uptime*, *user*, *task*, *mem* & *swap*. Sedangkan parameter untuk perintah *uptime* hanya waktu, tanggal dan *load average*.