

# Prototipe Layanan Video on Demand (VoD) Pada Jaringan OpenFlow

Eki Ahmad Zaki Hamidi  
Program Studi Teknik Elektro  
UIN Sunan Gunung Djati Bandung  
ekiahmadzaki@uinsgd.ac.id

Nanang Ismail  
Program Studi Teknik Elektro  
UIN Sunan Gunung Djati Bandung  
nanang.is@uinsgd.ac.id

Rizka Jalaludin  
Program Studi Teknik Elektro  
UIN Sunan Gunung Djati Bandung  
rizkajalaludin32@gmail.com

**Abstrak** – Sistem distribusi *Video on Demand (VoD)* saat ini umumnya masih menggunakan sistem transmisi konvensional. Perkembangan jaringan yang makin kompleks menuntut manajemen yang semakin handal. *Software-Defined Networking (SDN)* hadir untuk mengatasi kompleksitas sistem dengan memisahkan *control plane* dan *data plane*. Salah satu protokol SDN adalah *OpenFlow*. Makalah ini membahas tentang simulasi layanan VoD melalui jaringan *OpenFlow*. Realisasi *OpenFlow* menggunakan *OpenvSwitch*. Sistem disimulasikan dengan 1 PC server, 1 PC untuk switch dengan *OpenvSwitch*, dan 3 PC client. Hasil simulasi menunjukkan QoS VoD melalui jaringan *OpenFlow* sudah memenuhi standar ITU-T G.1010 dengan *delay* sebesar 3,31 ms, *jitter* sebesar 0 ms, *packet loss* sebesar 0%, dan *throughput* sebesar 2,71 Mbit/s. Penilaian *Mean Opinion Score (MOS)* juga sudah memenuhi standar ITU-T P.800 dengan nilai 3417,06 dan termasuk kategori baik. **Kata Kunci:** MOS, *OpenFlow*, *OpenvSwitch*, QoS, *Software-Defined Networking*, *Video On Demand*.

## I. PENDAHULUAN

Perkembangan yang sangat pesat pada teknologi berbasis IP mendorong munculnya berbagai inovasi layanan multimedia. Kemajuan ini ditandai dengan berkembangnya berbagai produk teknologi baik dari sisi media maupun perangkat. Salah satu layanan multimedia yang memanfaatkan perkembangan teknologi berbasis IP adalah *Video on Demand (VoD)* [1][2].

VoD adalah istilah untuk layanan penyajian video yang bisa diakses *online* secara *streaming* melalui jaringan berbasis IP [1][3]. Layanan ini didasarkan pada permintaan *end user* [4]. VoD saat ini menjadi pesaing dalam bisnis televisi berlangganan [1][2]. VoD memiliki berbagai kelebihan, salah satunya adalah memberikan kontrol terhadap para penggunanya. Melalui konsep ini pengguna akan memiliki kebebasan penuh untuk memilih apa yang ingin ia lihat. *User* dapat melakukan *pause*, *fastforward*, *rewind*, dan lain-lain). Semuanya berjalan secara interaktif dan menggunakan tombol serta perintah yang sederhana [1][4][5].

Beberapa contoh *video server* yaitu Youtube, Metacafe, Dailymotion, dan sebagainya. Bahkan layanan IPTV yang

dipromosikan Indihome Telkom juga memperkenalkan layanan VoD. Adanya *video server*, memudahkan *user* untuk saling berbagi (*share*) video baik secara *online* maupun *non-online*.

VoD mempunyai beberapa kelemahan, diantaranya adalah bahwa aplikasi VoD membutuhkan alokasi *bandwidth* yang lebih besar daripada aplikasi *streaming* suara. Pada VoD, ketika *bandwidth* multimedia lebih besar daripada kecepatan transmisi rata-rata yang diterima maka akan terjadi kongesti/tubrukan paket data sehingga menyebabkan beberapa paket hilang [5]. Menurut ITU-T G.1010, aplikasi *video streaming* membutuhkan nilai <1% PLR (*Packet Loss Ratio*) [6].

Untuk mendukung jaringan internet masa depan, maka diperlukan suatu "*fully managed virtualization network framework*" untuk menyediakan koneksi dari beberapa komponen yang ada dengan dilengkapi *platform* layanan virtual yang memenuhi semua persyaratan dari layanan tersebut. Virtualisasi jaringan akan berusaha menjalankan beberapa jaringan dalam infrastruktur yang sama (*shared*), tanpa adanya penurunan performansi atau kestabilan jaringan karena interferensi antar jaringan. Untuk memberikan inovasi yang cepat dan menyembunyikan kompleksitas dalam desain jaringan, maka diperlukan suatu paradigma baru, yaitu *Software-Defined Networking (SDN)* [7].

*Software-Defined Networking (SDN)* adalah satu jaringan komputer yang sangat fleksibel karena ia dikonfigurasi dan dikendalikan melalui *software* terpusat. SDN ini dikembangkan oleh *Stanford University* yang mengeluarkan teknologi *OpenFlow*. Dengan pengaplikasian *OpenFlow*, pengguna dapat menganalisis gambaran kebutuhan pertumbuhan jaringan, yang dapat diamati langsung hanya melalui sebuah aplikasi. SDN memungkinkan *administrator* sistem untuk mempercepat koneksi penyediaan jaringan [7] [8].

Artikel ini membahas tentang simulasi VoD melalui jaringan *OpenFlow* untuk melihat sejauh mana dukungan jaringan *OpenFlow* dalam mendukung transmisi *Video on Demand (VoD)*.

## II. LANDASAN TEORI

### A. *Video on Demand*

*Video on demand* adalah layanan berdasarkan permintaan *end-user*, yang memungkinkan *user* untuk memilih dan

melihat konten video yang ingin ditonton, dimana *end-user* dapat mengontrol konten video yang ditonton (misal: dapat melakukan *pause*, *fastforward*, *rewind*, dan lain-lain) [4]. Namun pada sistem yang menggunakan metode *streaming*, hal ini akan membebani *server* dan memerlukan pemakaian *bandwidth* yang lebih besar.

Konsep dasar dari *VoD* adalah menyimpan program/konten dan kemudian dikirimkan ke penonton ketika diminta oleh penonton tersebut. Penyimpanannya berupa *server* tersentralisasi yang menggunakan perangkat untuk mengirimkan pemrograman secara simultan ke ratusan penonton, atau dapat pula menggunakan penyimpanan lain yang terdistribusi ke seluruh jaringan. Untuk membatasinya, perangkat penyimpanan individu untuk tiap penonton dapat diletakkan di masing-masing *set top box* [2][4].

*VoD* memiliki beberapa tipe layanan, antara lain: *True Video on demand* (VoD), *Near Video on demand* (NVoD), *Subscription Video on demand* (SVoD), *Free Video on Demand* (FVoD), *Everything on Demand* (EoD), *Personal Video Recorders* (PVRs), *Network Personal Video Recorders* (NPVRs), dan *Pay Per View* (PPV) [2].

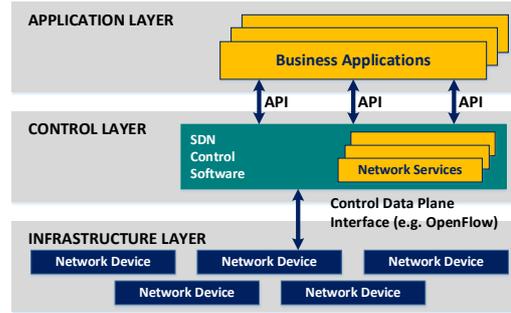
B. Internet Protocol (IP)

*Internet Protocol* atau protokol internet didesain untuk interkoneksi sistem komunikasi komputer pada jaringan paket *switched*. *Internet Protocol* menyediakan skema pengalamatan yang seragam sehingga komputer pada satu jaringan dapat berkomunikasi dengan komputer pada jaringan yang lain [9][10].

C. Software-Defined Networking (SDN)

SDN adalah sebuah pendekatan baru dalam mendesain, membangun, dan mengelola jaringan komputer. Pada dasarnya SDN memisahkan data dan fungsi kontrol perangkat jaringan seperti *router*, *packet switch*, dan *switch LAN*. Secara umum dalam perangkat *networking* terdapat dua bagian, yaitu *control plane* dan *data plane*. Dengan demikian SDN dapat didefinisikan sebagai *Application Programming Interface* (API) [8][11]. *Control plane* adalah bagian yang berfungsi untuk mengatur logika pada perangkat *networking* seperti *routing table*, pemetaan jaringan dan sebagainya. *Data plane* adalah bagian yang berfungsi untuk meneruskan paket-paket yang masuk ke suatu *port* pada perangkat *networking* menuju *port* keluar dengan berkonsultasi kepada *control plane* [10][11][12].

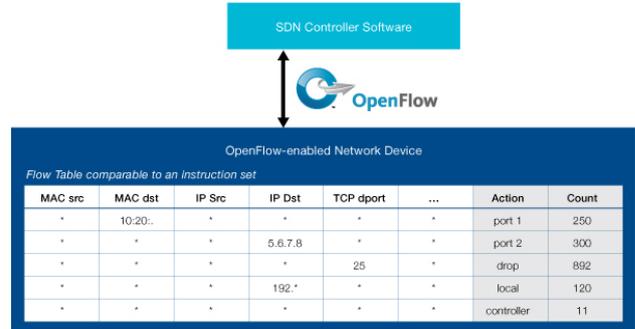
Gambar 1 menggambarkan pandangan logis dari arsitektur SDN. Jaringan intelijen (logis) dipusatkan di perangkat lunak berbasis pengendali SDN, yang mempertahankan pandangan global dari jaringan. Akibatnya, jaringan muncul ke aplikasi dan kebijakan mesin yang tunggal (satu). Dengan SDN, perusahaan dan operator mendapatkan kontrol vendor-independen di seluruh jaringan dari titik logis tunggal, yang sangat menyederhanakan desain jaringan dan operasi. SDN juga sangat menyederhanakan perangkat jaringan sendiri, karena mereka tidak lagi perlu memahami dan memproses ribuan standar protokol tetapi hanya menerima instruksi dari pengendali SDN [12].



Gambar 1. Arsitektur *Software-Defined Networking* [12].

D. OpenFlow

Untuk mengubah konsep SDN ke implementasi praktis *OpenFlow*, dua persyaratan yang harus dipenuhi. Pertama, harus ada arsitektur logis umum di semua *switch*, *router*, dan perangkat jaringan lainnya yang akan dikelola oleh pengontrol SDN. Arsitektur logis ini dapat diimplementasikan dengan cara yang berbeda pada peralatan *vendor* yang berbeda dan di berbagai jenis perangkat jaringan, asalkan *controller* SDN melihat fungsi saklar seragam. Kedua, standar protokol aman diperlukan antara *controller* SDN dan perangkat jaringan. Kedua persyaratan ini ditangani oleh *OpenFlow*, yang merupakan sebuah protokol antara SDN *controller* dan perangkat jaringan, serta spesifikasi struktur logis dari fungsi *switch* jaringan [11][12][13][14][15].



Gambar 2. Contoh kumpulan perintah *OpenFlow* [15].

Berikut ini adalah format *Flow* yang digunakan:

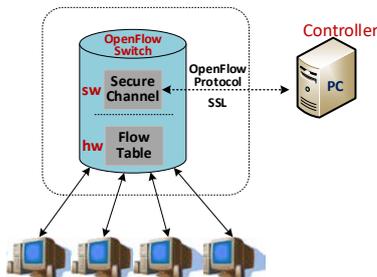
Ingress Port	MAC DA	MAC SA	Ether type	VLAN ID	P-Bits	IP Src	IP Dst	IP Protocol	IP DSCP	TCP/UDP src port	TCP/UDP dst port
--------------	--------	--------	------------	---------	--------	--------	--------	-------------	---------	------------------	------------------

Gambar 3. Format *Flow* [15].

E. Switch OpenFlow

*Switch OpenFlow* atau yang biasa disebut *OpenvSwitch* merupakan implementasi *switch OpenFlow* yang dapat digunakan baik sebagai saklar virtual murni dalam lingkungan *virtual* dan sebagai saklar *software* tujuan umum yang menghubungkan *node* secara fisik yang terpisah. *Switch OpenFlow* adalah referensi yang awalnya dikembangkan oleh *Stanford University* dan kemudian

dilanjutkan *Open Networking Foundation (ONF)*. *OpenFlow Switch* meliputi komponen dan fungsi dasar yang dibutuhkan oleh perangkat *networking* untuk mendukung *OpenFlow* [16][17][18].



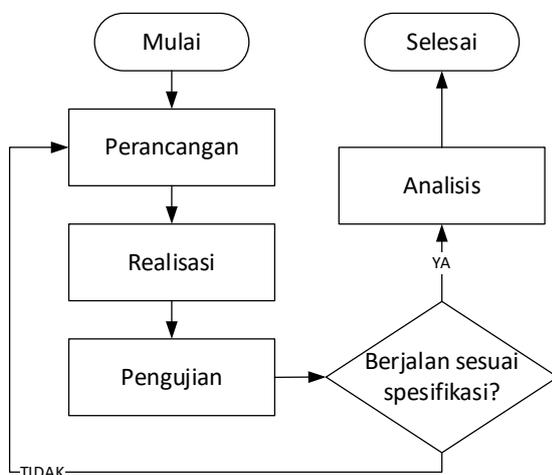
Gambar 4. Arsitektur *OpenVswitch* [17].

Gambar 4 menunjukkan arsitektur *Switch OpenFlow* yang mendukung *OpenFlow* (yang selanjutnya disebut *OpenFlow switch*) sesuai referensi *OpenFlow Switch Specification* versi 1.3.0. *OpenFlow switch* berkomunikasi dengan *controller* melalui protokol *OpenFlow*. *OpenFlow switch* terdiri dari tiga bagian yaitu [17]:

- Flow Table*  
*OpenFlow Switch* terdiri dari satu atau lebih *flow table* yang berfungsi untuk memproses paket yang datang.
- Secure Channel*  
*Secure channel* merupakan sebuah *interface* yang menghubungkan *OpenFlow switch* dan *controller*.
- OpenFlow Protocol*  
*OpenFlow protocol* menyediakan sebuah cara yang bersifat terbuka dan standar untuk bisa berkomunikasi dengan *OpenFlow switch*.

### III. METODOLOGI PENELITIAN

Metodologi yang digunakan dalam penelitian ini mengacu pada tahapan pada Gambar 5.



Gambar 5. Diagram Alir Penelitian.

#### A. Perancangan Sistem

Pada tahapan ini akan dilakukan pemodelan topologi jaringan. Selain itu pada tahapan ini akan dirancang antar muka halaman web VoD.

#### B. Realisasi Sistem

Rancangan yang sudah dibuat selanjutnya direalisasikan untuk mendapatkan prototipe VoD pada jaringan *OpenFlow*. Dimana sistem akan melibatkan satu *PC server*, *Switch*, dan 3 *client*. Spesifikasi perangkat keras yang digunakan adalah sebagai berikut:

- Satu unit Komputer dengan *operating system* Windows 8 (*trial*) berfungsi sebagai *server* dengan spesifikasi sebagai berikut:
  - *Processor*: Intel (R) Pentium (R) Dual CPU E2200 @ 2.20 GHz
  - *RAM*: 1.00 GB
- Tiga unit Komputer dengan *operating system* Windows 8 (*trial*) berfungsi sebagai *client* dengan spesifikasi sebagai berikut:
  - *Processor*: Intel (R) Pentium (R) Dual CPU E2200 @ 2.20 GHz
  - *RAM*: 1.00 GB
- Satu unit komputer dengan *operating system* Linux Debian berfungsi menjalankan *switch OpenFlow (OpenVswitch)* dengan spesifikasi sebagai berikut:
  - *Processor*: Intel Core2Duo
  - *RAM*: 1.00 GB
- Kabel LAN sebagai media transmisi data dari *server* menuju *client*.

#### C. Pengujian dan Analisis Sistem

Skenario pengujian terdiri dari beberapa tahap, yaitu:

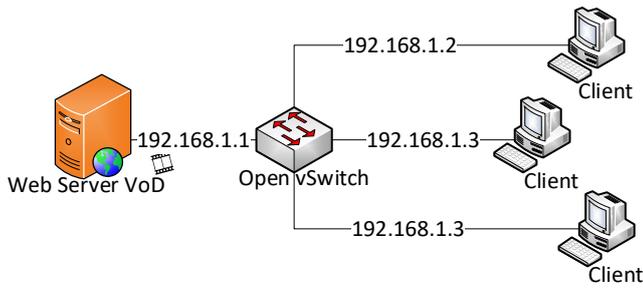
- Pengujian konektivitas dengan menguji *flow* antara *server – client*
- Pengujian QoS yang mencakup parameter: *delay*, *jitter*, *packet loss*, dan *throughput*,
- Pengujian *Mean Opinion Score (MOS)* untuk menguji kualitas VoD menurut *user*.

Masing-masing data hasil uji akan dianalisis ketercapaiannya.

### IV. HASIL DAN PEMBAHASAN

#### A. Perancangan Arsitektur Sistem *Video on Demand*

Pada perancangan skema sistem VoD ini, sistem dibuat agar *user* dapat dengan mudah mengaksesnya. Dengan perangkat yang mudah untuk diimplementasikan. Sistem *Video on Demand (VoD)* dirancang berbasis *Web (Web Base)*. Topologi jaringan yang dibuat mengacu pada Gambar 6.



Gambar 6. Topologi Jaringan.

Web server menggunakan XAMPP versi 1.8.1. XAMPP dipilih karena merupakan *opensource* dan mampu mendukung banyak sistem operasi.

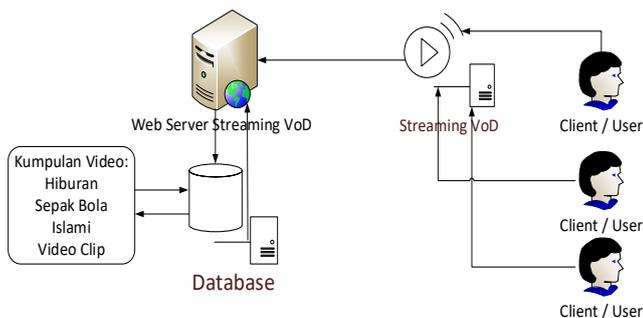
Streaming VoD server dibuat sebagai penyimpan *file-file* video yang akan di-streaming-kan kepada *client/user*. *File-file* video akan disimpan dalam *folder* Htdocs. VoD ini berisikan empat kategori, yaitu:

- Hiburan, berisikan video-video hiburan
- Islami, berisikan video pembacaan Surah Al Quran, ceramah keagamaan dan pembacaan shalawat,
- Sepak Bola, berisikan video cuplikan-cuplikan dokumentasi pemain sepak bola dunia, dan
- Video Clip, berisikan video klip musik.

Jaringan *OpenFlow* menggunakan *software* *OpenVSwitch* (*Switch* *OpenFlow*), yang bertindak sebagai *switch* penghubung antara *client* ke *server* dengan media transmisi kaber RJ45. *Streaming video* dilakukan secara *multicast* maupun *unicast*.

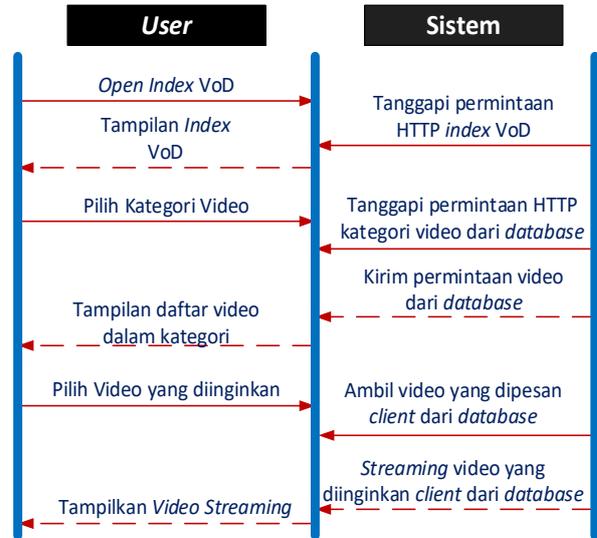
B. Perancangan Web Server VoD

Web server berfungsi menerima permintaan HTTP dari *client* melalui *web browser* dan akan mengirimkan hasilnya dalam bentuk halaman-halaman *web* yang berbentuk dokumen HTML. Web server VoD yang menggunakan *software* XAMPP 1.8.1 ini merupakan perangkat untuk *server* yang berfungsi menyimpan *file-file* video dan mengirimkan *streaming video* ke *client*. *Server* ini akan menerima permintaan *streaming* dari *client*, lalu *server* akan mendistribusikan data dari *database* ke halaman *web* VoD dan di-streaming-kan ke *client* yang meminta layanan VoD tersebut. Arsitektur sistem VoD server diperlihatkan pada Gambar 7.



Gambar 7. Arsitektur Sistem VoD Server.

Pada penelitian ini digunakan *personal computer* yang digunakan sebagai *server* yang menyimpan dan *streaming*-kan video-video yang ada di *server* terhadap *client*. Gambar 8 menunjukkan cara *client* melakukan *request* dan menonton video.



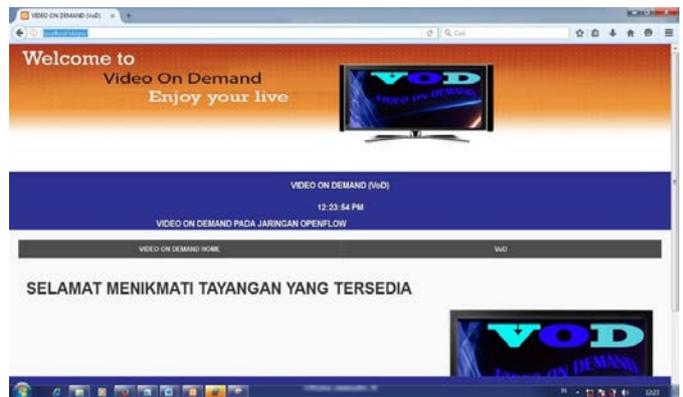
Gambar 8. Sequence diagram untuk request VoD.

C. Perancangan Tampilan Homepage VoD

Homepage VoD terdiri dari 2 halaman utama, yaitu:

i. Layar Home/Menu Utama

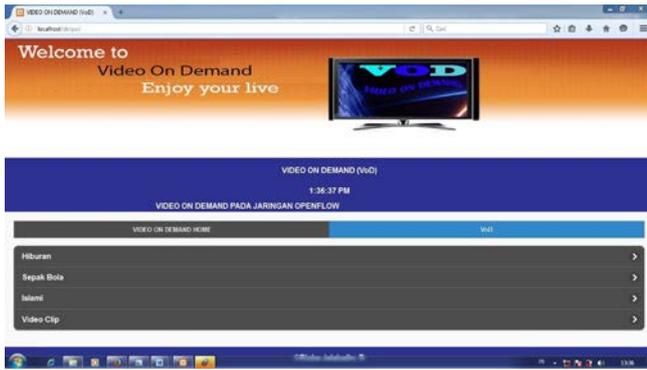
Halaman *Home* merupakan halaman awal yang tampil saat *user* mengakses *website* VOD. Tampilan layar *home* dapat dilihat pada Gambar 9.



Gambar 9. Halaman Menu Utama VoD.

ii. Layar VoD

Halaman *Video on Demand* (VoD) berisikan empat kategori VOD yang tersedia. Layar VoD dapat dilihat pada Gambar 10.



Gambar 10. Tampilan Kategori VoD.

#### D. Pengujian dan Analisis

##### i. Analisis Flow

Flow yang digunakan adalah sebagai berikut:

```
root@debian:~# ovs-ofctl add-flow ovs0
arp,action=normal
root@debian:~# ovs-ofctl add-flow ovs0
"priority=65535,in_port=1,action=output:2,3,4"
root@debian:~# ovs-ofctl add-flow ovs0
"priority=65535,in_port=2,action=output:1"
root@debian:~# ovs-ofctl add-flow ovs0
"priority=65535,in_port=3,action=output:1"
root@debian:~# ovs-ofctl add-flow ovs0
"priority=65535,in_port=4,action=output:1"
```

Flow tersebut menunjukkan Video On Demand (VoD) telah berhasil diimplementasikan pada jaringan OpenFlow. Server dapat mem-broadcast video kepada ketiga client.

##### ii. Analisis Dump Flow

Dump flow digunakan untuk melihat paket-paket yang digunakan dalam OpenFlow. Analisis hasil dump flow dapat diketahui pada Gambar 11.

```
root@debian:~# ovs-ofctl dump-flows ovs0
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=75.688s, table=0, n_packets=138, n_bytes=15187, idle_age=0,
priority=65535, in_port=3 actions=output:1
cookie=0x0, duration=104.832s, table=0, n_packets=218, n_bytes=26182, idle_age=1,
priority=65535, in_port=1 actions=output:2:3, output:4
cookie=0x0, duration=70.424s, table=0, n_packets=111, n_bytes=9146, idle_age=1,
priority=65535, in_port=4 actions=output:1
cookie=0x0, duration=81.352s, table=0, n_packets=95, n_bytes=8677, idle_age=0,
priority=65535, in_port=2 actions=output:1
cookie=0x0, duration=123.976s, table=0, n_packets=0, n_bytes=0, idle_age=123.arp
actions=normal
root@debian:~#
```

Gambar 11. Dump Flow

Flow dari port 3 ke port 1 atau dari client (IP 192.168.1.2) ke server memiliki durasi pemakaian flow sebesar 75,688 s, banyaknya paket flow adalah 138 paket. Ukuran paketnya sebesar 15187 byte. Lama flow tidak dipakai bernilai 0 karena flow tidak pernah berada pada keadaan off, dengan priority sebesar 65535.

Flow dari port 1 ke port 2, 3, 4 atau koneksi dari server ke semua client memiliki durasi flow sebesar 104,832 s, banyak paket flow sebesar 218 paket, dengan ukuran

paket sebesar 26182 byte. Lama flow tidak digunakan hanya 1 kali.

Flow dari port 4 ke port 1 atau dari client dengan IP 192.168.1.3 ke server, memiliki durasi flow sebesar 70,424 s, dan banyak paket flow adalah 111 paket dengan ukuran paketnya sebesar 9146 byte. Lama flow yang tidak dipakai 0 karena flow tidak pernah berada dalam posisi off.

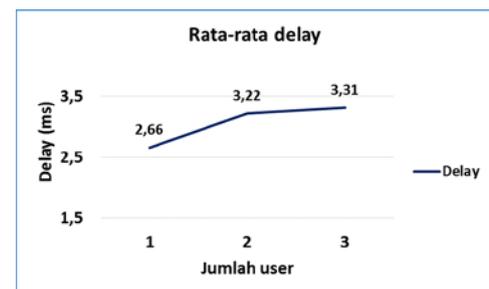
Flow dari port 2 ke port 1 atau dari client dengan IP 192.168.1.4 ke server, memiliki durasi flow sebesar 81,352 s, dan banyak paket flow yang digunakan sebesar 95 paket, dengan ukuran paketnya sebesar 8677 byte. Lama flow yang tidak dipakai 0 karena flow tidak pernah berada dalam keadaan off.

Ketika semua port tidak melakukan interkoneksi dari server ke client, memiliki durasi sebanyak 123,87 s. Lama flow tidak dipakai adalah 123 kali. Pada penelitian ini priority-nya adalah 65535 yang merupakan nilai priority terbesar di dalam OpenSwitch. Ketika priority paling besar maka posisi flow-nya paling tinggi. Namun bukan berarti flow tersebut di-forward paling pertama namun fungsi memasukan sebuah priority bertujuan supaya administrator bisa mengetahui posisi flow itu berada.

##### iii. Quality of Service

###### • Delay

Hasil pengukuran menunjukkan bahwa delay tidak melebihi standar yang ditetapkan oleh ITU-T G.1010 tentang delay yang disarankan yaitu kurang dari 150 ms.

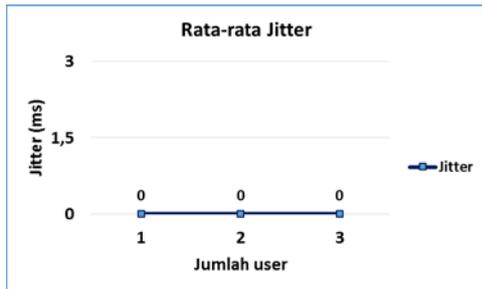


Gambar 12. Rata-Rata Delay Transmisi VoD Pada Jaringan OpenFlow

Dari Gambar 12, dapat dilihat bahwa peningkatan delay pada streaming VoD server dipengaruhi oleh adanya penambahan user. Hal ini disebabkan oleh semakin meningkatnya antrian sehingga paket yang dikirim semakin lama. Delay untuk 3 user sebesar 3,3112 ms. Delay yang ada masih sesuai dengan standar ITU-T G.1010 yaitu kurang dari 150 ms.

###### • Jitter

Hasil pengukuran menunjukkan bahwa tidak ada perubahan nilai jitter (konstan). Tidak adanya perubahan nilai jitter disebabkan karena jarak antara server dan client pada uji coba hanya sejauh 1 m.

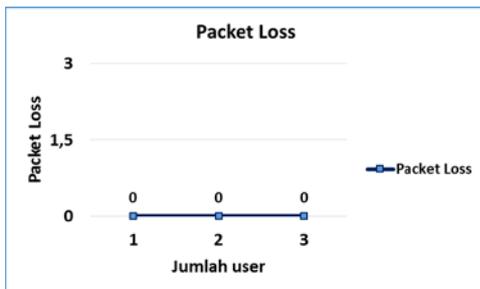


Gambar 13. Rata-rata *Jitter* Transmisi VoD Pada Jaringan *OpenFlow*.

Dapat dilihat bahwa nilai *jitter* adalah konstan 0 ms. Dengan demikian *jitter* untuk jaringan yang dibangun sesuai dengan standar ITU – T G.1010 yaitu dengan nilai *jitter* 0 ms (sangat bagus).

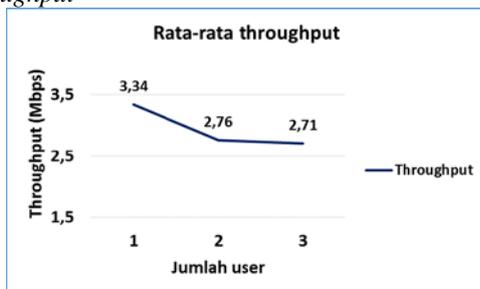
• *Packet Loss*

Penyebab utama *packet loss* adalah hilangnya paket akibat antrian. Pada pengamatan yang telah dilakukan, perubahan jumlah *user* mempengaruhi paket data yang hilang. Namun karena simulasi hanya menggunakan 3 *client* dengan jarak ke *server* hanya 1 meter, tidak ada *packet loss* pada transmisi VoD yang dilakukan. Gambar 14 menunjukkan rerata *packet loss* pada uji coba VoD yang dilakukan.



Gambar 14. *Packet Loss* pada transmisi VoD pada Jaringan *OpenFlow*.

• *Throughput*



Gambar 15. Grafik *Throughput* VoD Pada Jaringan *OpenFlow*.

Hasil analisis menunjukkan bahwa rata-rata *throughput* mengalami penurunan nilai pada saat terjadi penambahan *user*. Nilai *throughput* terkecil terjadi ketika *server*

diakses oleh 3 *user*. Nilai *throughput* akan semakin berkurang seiring bertambahnya *user* yang mengakses layanan VoD. *Throughput* bernilai 3,42 Mbit/s ketika diakses 1 *user*, ketika di akses oleh 3 *user* bernilai 2,71 Mbit/s.

iv. *Mean Opinion Score (MOS)*

Perhitungan nilai MOS (*Mean Opinion Score*) dilakukan berdasarkan pendekatan matematis dengan *E-Model* yang distandarkan kepada ITU-T G.107. Nilai akhir estimasi *E-Model* disebut dengan R faktor menggunakan persamaan (1) di bawah ini [3][19]:

$$R = 94,2 - Id - Ief \tag{1}$$

*R* = Faktor kualitas transmisi.

*Id* = Merupakan faktor penurunan kualitas yang disebabkan oleh pengaruh *delay*

*Ief* = Merupakan faktor penurunan kualitas yang disebabkan oleh *packet loss* yang terjadi.

Nilai *Id* ditentukan dari persamaan (4) sebagai berikut [3][20]:

$$Id = 0,024d + 0,11(d - 177,3) H(d - 177,3) \tag{2}$$

Dengan nilai *d* (*delay*) sebesar 3,31 ms, dan H adalah fungsi tangga, maka:

$$\begin{aligned} Id &= 0,024(3,3112) + 0,11.(3,31-177,3)(1)(3,31-177,3) \\ Id &= 0,0794 + 0,11.(-173,99) \cdot 1 \cdot (-1773,99) \\ Id &= 0,0794 + (-19,14) (-173,99) \\ Id &= 0,0794 + (-3.329,93) \\ Id &= -3.329,86 \end{aligned}$$

Nilai *Ief* dapat ditentukan dari persamaan (3) sebagai berikut [3] [19]:

$$Ief = 7 + 30 \ln (1 + 15e) \tag{3}$$

dimana *e* adalah *Packet loss* yang didapat pada penelitian, bernilai 0, maka:

$$\begin{aligned} Ief &= 7 + 30 \ln (1 + 15 \cdot 0) \\ Ief &= 7 + 30 \ln (1 + 0) \\ Ief &= 7 + 30 \ln 1 \\ Ief &= 7 + 0 \\ Ief &= 7 \end{aligned}$$

Setelah nilai *Id* dan *Ief* diketahui, maka nilai R dapat diperoleh dengan memasukkannya ke persamaan (1).

$$\begin{aligned} R &= 94,2 - (-3.329,86) - 7 \\ R &= 3424,06 - 7 \\ R &= 3417,06 \end{aligned}$$

Berdasarkan ketentuan ITU-T P.800 bahwa [3] [20]:

- $R < 0$ ; maka  $MOS = 1$
- $R > 100$ ; maka  $MOS = 4,5$
- $0 < R < 100$  maka  $MOS = 1 + 0,035R + 7 \times 10^{-6} R(R-60)(100-R)$

Dengan nilai R sebesar 3417,06, maka didapat nilai MOS sebesar 4,5. Berdasarkan rekomendasi ITU T P.800.2 nilai 4,5 termasuk pada kategori baik.

Hasil dari kuisioner yang diambil dari 10 orang responden yang telah mencoba sistem, didapatkan hasil untuk tampilan *web* sebanyak 60% responden menyatakan bahwa tampilan sistem bagus, 70% menyatakan sistem sangat mudah untuk digunakan, dan 50% menyatakan kualitas gambar *streaming* yang dihasilkan bagus.

## V. KESIMPULAN

Berdasarkan perancangan prototipe layanan *Video on Demand* (VoD) pada jaringan *OpenFlow*, maka dapat disimpulkan bahwa konsep teknologi *OpenFlow* dapat diterapkan dengan baik pada aplikasi *Video on Demand* (VoD). Dan berdasarkan hasil pengujian, parameter QoS yang diamati sudah memenuhi standar yang ditetapkan ITU – T, dengan hasil *delay* sebesar 3,31 ms, *jitter* sebesar 0,0 ms, *packet loss* sebesar 0%, dan *throughput* sebesar 2,71 Mbit/s. Untuk hasil perhitungan matematis, nilai MOS sebesar 3417,06, termasuk pada kategori baik.

## REFERENSI

- [1] Lestaringati, S.I., Zarman, W., & Perdana, D. (2011). Perancangan dan Implementasi Video On Demand Pada Jaringan Lokal. *Majalah Ilmiah UNIKOM*, Vol. 9, No. 1, pp. 11-20.
- [2] Simpson, W. & Greenfield, H. (2012). *IPTV and Internet Video Expanding the Reach of Television*. USA: National Association of Broadcasters, CRC Press.
- [3] W, HP, Susilawati, H. & Noviandono, R.K. (2013). *Analisis Performansi VoIP (Voice over Internet Protocol) Pada Jaringan WiMax (Worldwide Interoperability for Microwave Access) di Wilayah Jakarta*. Jakarta: Universitas Jendral Sudirman.
- [4] Muryanti, S.D.P. & Affandi, A. (2015). *Rancang Bangun Sistem Keamanan Konten Video On Demand (VOD) Pada Internet Protocol Television (IPTV) Menggunakan Video Encryption Algorithm (VEA)*. Surabaya: Jurusan Teknik Elektro, ITS.
- [5] Marpaung, K.W., W, A.P & Ambarwati, R. (2011). *Desain Video on Demand (VOD) Menggunakan High Speed Downlink Packet Access (HSDPA) di Wilayah Urban Kota Malang*. Malang: Unibraw.
- [6] ITU-T. (2011). *ITU-T Recommendation G.1010: End-User Multimedia QoS Categories*. ITU-T.
- [7] Risdianto, A.C. (2012). *Implementation and Analysis of Control and Forwarding Plane for Software Defined Network*. Bandung: ITB.
- [8] Faruqi, N.A., Nurwadi, L., Ismail, N. & Maryanto, D. (2017). Simulasi Kinerja Berbagai Topologi Jaringan Berbasis Software-Defined Network (SDN). *Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung (SENTER 2017)*. Bandung.
- [9] Sardju, A.P. (2012). *Implementasi IPTV (Internet Protocol Television) Berbasis Web Pada Jaringan Wireless*. Makassar: Universitas Hasanudin.
- [10] Ismail, N. (2006). *Sistem Keamanan pada IPTV (Internet Protocol Television)*. Bandung: Bidang Khusus Teknologi Informasi Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.
- [11] McKeown, N. & De Boer, M. (2008). *Performance Analysis of OpenFlow Hardware*. USA: Stanford University.
- [12] Stallings, W. (2013). Software-defined Networks and Openflow. *The Internet Protocol Journal*, Vol. 16, No. 1, pp. 2-14.
- [13] Limoncelli, T.A. (2012). Openflow: a radical new idea in networking. *Queue*, Vol. 10, No. 6, pp. 40.
- [14] Hamidi, E.A.Z. (2015) The Simulation of VLAN Using OpenFlow at Laboratory of Electrical Engineering. *The 2015 International Conference on Electrical, Telecommunication and Computer Engineering*. Medan.
- [15] Open Networking Fundation. (2012). Software-defined Networking: The New Norm for Networks. *ONF White Paper*, Vol. 2, pp. 2-6.
- [16] Wijaksa, D.S., Mardiaty, R., Ismail, N. & Juhana, T. (2016). TestBed Open vSwitch Raspberry Pi Pada Skala Kecil. *Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung (SENTER 2016)*. Bandung.
- [17] Hakim, A. (2014). *Implementasi Dan Analisis Kinerja Switch OpenFlow dan Switch Konvensional Pada Jaringan Komputer*. Bogor: Institut Pertanian Bogor.
- [18] Effendi, M.R., Hamidi, E.A.Z & Saepulloh, A. (2017). Implementasi GRE Tunneling Menggunakan Open vSwitch Pada Jaringan Kampus. *TELKA - Jurnal Telekomunikasi, Elektronika, Komputasi dan Kontrol*, Vol. 3, No. 2, pp. 103-111.
- [19] ITU-T. (2014). *Recommendation ITU-T G.107: The E-model: a Computational Model for Use in Transmission Planning*. ITU-T.
- [20] ITU-T. (2013). *Recommendation ITU-T P.800.2: Mean opinion score interpretation and reporting*. ITU-T.