

**Jurnal Politeknik Caltex Riau**<https://jurnal.pcr.ac.id/index.php/elementer>

| ISSN : 2460 – 5263 (online) | ISSN : 2443 – 4167 (print)

Pengiriman Video Secara *Live Streaming* Menggunakan *Dynamic Adaptive Streaming over HTTP (DASH)*

Hamid Azwar¹Politeknik Caltex Riau, Teknik Telekomunikasi, email: hamid@pcr.ac.id

Abstrak

Keterbatasan bandwidth dan fluktuasi trafik pada jaringan menjadi kendala dalam proses pengiriman video. Pada saat ketersediaan bandwidth kecil dan traffic jaringan yang tinggi, proses pengiriman video akan memakan waktu yang lama dikarenakan proses transmisi video yang lambat atau lebih umum disebut dengan buffering. Penelitian ini melakukan evaluasi pengiriman video secara live streaming menggunakan metode Dynamic Adaptive Streaming over HTTP (DASH) pada jaringan LAN. DASH merupakan sebuah protokol streaming adaptive yang mampu menyajikan video dengan kualitas video sesuai kemampuan jaringan pada klien. Implementasi ini dilakukan menggunakan video kompresi H.265 dan proses streaming dilakukan pada jaringan LAN secara live streaming sebanyak 20 klien. Hasil penelitian ini diperoleh nilai throughput pengiriman video mencapai 1,4 Mbps dan nilai delay terbesar 12,5 ms. Persentase permintaan klien terhadap kualitas video tertinggi untuk 1 klien sebanyak 98%. Sedangkan penggunaan 20 klien, permintaan untuk video kualitas tinggi berkurang menjadi 46% yang membuktikan bahwa DASH berhasil diimplementasikan secara live streaming.

Kata kunci: Video, Live, Streaming, H.265, DASH

Abstract

Bandwidth limitations and fluctuations in traffic on the network become obstacles in the process of sending video. At the time of the availability of small bandwidth and high network traffic, the process of sending video will take a long time due to the operation of slow video transmission or more commonly called buffering. This study evaluates sending live streaming video using the Dynamic Adaptive Streaming over HTTP (DASH) method on the LAN network. DASH is an adaptive streaming protocol that can present the video with video quality according to network capabilities on the client. This implementation is carried out using H.265 video compression and the streaming process is carried out on a LAN network with 20 live streaming clients. The results of this study obtained the value of video delivery throughput reaches 1.4 Mbps and the most significant delay value is 12.5 ms. The highest percentage of client requests for video quality for one client is 98%. As for the use of 20 clients, the demand for high-quality video is reduced to 46% which proves that DASH was successfully implemented live streaming.

Keywords: Video, Live, Streaming, H.265, DASH

[1] Pendahuluan

Setiap orang pasti membutuhkan informasi. Ada banyak cara yang dapat dilakukan orang untuk mendapatkan informasi, salah satu contohnya adalah melalui media komputer yang terhubung dengan jaringan. Perkembangan teknologi komputer yang begitu pesat membuat informasi yang didapatkan semakin aktual dan cepat. Banyak informasi yang dapat diperoleh dan dikirimkan menggunakan jaringan komputer, antara lain berupa gambar, suara, ataupun video. Untuk mengirimkan video dalam jaringan tidak semudah mengirimkan gambar saja atau mengirim suara saja. Salah satu cara dalam mengirimkan video dalam jaringan adalah menggunakan streaming atau yang lebih dikenal dengan video streaming.

Masalah dasar dalam video streaming, khususnya untuk implementasi pada jaringan internet yang bersifat global adalah bandwidth, delay, jitter dan packet loss[1]. Pada saat ketersediaan bandwidth kecil akibat padatnya traffic data pada jaringan maka proses streaming akan memakan waktu yang lama dikarenakan proses transmisi paket video yang lambat atau biasa lebih umum disebut dengan proses buffer. Tentu proses menunggu yang kadang lama ini akan menjadi penghalang bagi klien yang memiliki jaringan internet yang tidak stabil. Pada penelitian ini, penulis akan mengevaluasi live streaming video menggunakan Dynamic Adaptive Streaming over HTTP (DASH) pada jaringan LAN.

Sebelumnya telah dilakukan penelitian tentang Analisis Kualitas Layanan Video Live Streaming pada Jaringan lokal Universitas Telkom[2]. Pada penelitian ini menggunakan video yang direkam dari webcam dan di kirim langsung kepada client melalui jaringan lokal LAN Universitas Telkom. Dari penelitian ini dapat disimpulkan bahwa penambahan background traffic yang makin besar dapat membuat QoS menjadi memburuk. Penelitian tentang Implementasi Video on Demand Secara Adaptive Menggunakan Mobile Application[5]. Penelitian ini melakukan streaming video berdasarkan permintaan client dimana video yang akan di stream sebelumnya telah tersimpan di database server. Hasil dari penelitian ini video streaming mengalami delay sebesar 8,5 ms untuk video dengan resolusi 640x360 throughput sebesar 1,35 Mbit/sec.

Evaluasi live streaming menggunakan Dynamic Adaptive Streaming over HTTP (DASH) pada jaringan LAN ini menggunakan video asli dengan format yuv yang dikompresi di laptop server, kemudian dilakukan streaming menggunakan Dynamic Adaptive Streaming over HTTP pada jaringan LAN. Dimana DASH merupakan sebuah protokol streaming adaptive yang memungkinkan streaming konten kualitas tinggi melalui internet dari HTTP web server, dan mampu menyajikan video dengan frame rate yang konsisten dengan menyesuaikan format dan kualitas video sesuai kemampuan client. MPEG-DASH bekerja dengan memecah konten ke dalam beberapa segmen file yang berbasis HTTP. Video tersedia dengan tingkatan bitrate berbeda. Kemudian klien secara otomatis memilih dari alternatif segmen yang tersedia untuk diunduh dan diputar ulang berdasarkan kondisi jaringan klien pada saat itu.

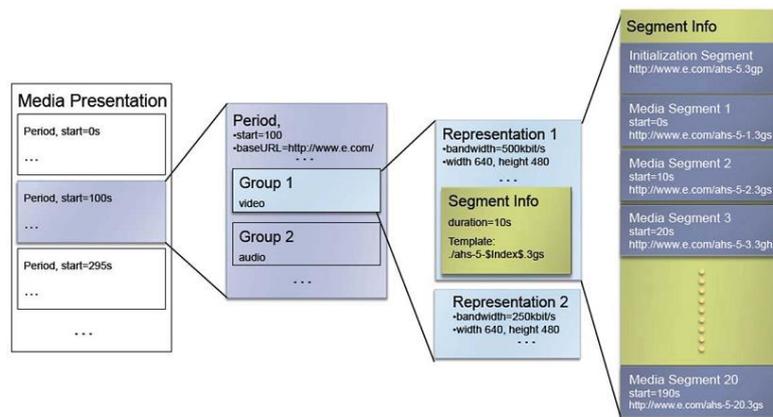
[2] Dynamic Adaptive Streaming over HTTP (DASH)

Moving Picture Expert Group (MPEG) yang mendapat partisipasi dan dukungan yang luas dari banyak industri mengembangkan teknologi MPEG-DASH dalam rangka standarisasi adaptive video streaming melalui HTTP dengan membuatnya lebih sederhana dan fleksibel sehingga memungkinkan klien dan server dari vendor yang berbeda untuk bekerja sama[1]. MPEG-DASH singkatan dari MPEG Dynamic Adaptive Streaming over HTTP. Ini merupakan standar ISO internasional yang mencoba untuk menyatukan HTTP streaming untuk satu standar. MPEG DASH iniditerbitkan sebagai ISO/IEC 23009-1:2012 pada bulan April 2012[2]. MPEG-DASH menyediakan format yang universal untuk pengiriman konten dengan menggunakan konten perangkat dan infrastruktur yang ada. Hal ini meningkatkan kualitas pada media streaming pada Struktur MPEG-DASH sebagian besar terdiri dari tiga jenis file yaitu "Manifest" (.mpd) yang merupakan file XML yang menggambarkan segmen, "Inisialisasi File" yang berisi header yang dibutuhkan untuk memecahkan kode byte pada segmen dan "Segmen Files" yang berisi media

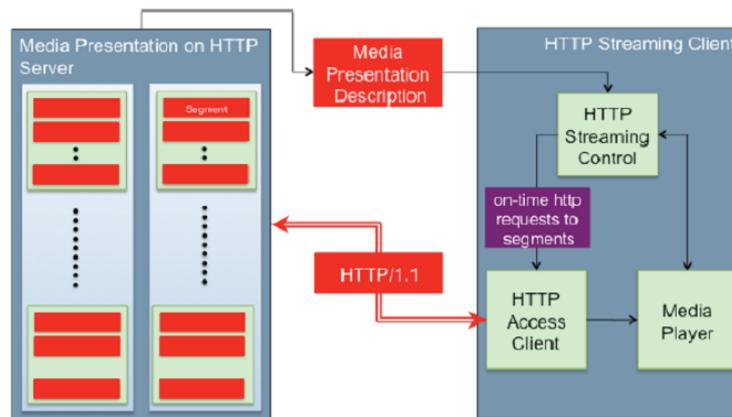
yang dimainkan. Media Presentation Description (MPD) menyediakan meta data untuk media meminta segmen melalui URL dalam rangka untuk mencari dan melakukan download segmen. Hal ini juga memberikan informasi mengenai jumlah representasi serta karakteristik setiap representasi yang digunakan untuk keperluan adaptasi level seperti minimum waktu penyangga, representasi bandwidth dan durasi segmen. Contoh bentuk MPD dapat dilihat pada gambar 1. MPEG-DASH mendefinisikan format MPD dan format pengiriman video menggunakan ISO BMFF dan MPEG2-TS. MPEG-DASH yang menentukan penyediaan konten seperti ukuran dan durasi segmen, jumlah dan bit rate representasi. Selain itu juga MPEG-DASH menentukan perilaku klien secara normatif seperti peralihan representasi. Gambar 3 menunjukkan arsitektur yang menyusun MPEG-DASH.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD minBufferTime="PT10S">
  <Period start="PT0S">
    <Representation mimeType="video/3gpp; codecs=263, samr" bandwidth="256000" id="256">
      <SegmentInfo duration="PT10S" baseUrl="rep1/">
        <InitialisationSegmentURL sourceURL="seg-init.3gp"/>
        <Url sourceURL="seg-1.3gp"/>
        <Url sourceURL="seg-2.3gp"/>
        <Url sourceURL="seg-3.3gp"/>
      </SegmentInfo>
    </Representation>
    <Representation mimeType="video/3gpp; codecs=mp4v.20.9" bandwidth="128000" id="128">
      <SegmentInfo duration="PT10S" baseUrl="rep2/">
        <InitialisationSegmentURL sourceURL="seg-init.3gp"/>
        <Url sourceURL="seg-1.3gp"/>
        <Url sourceURL="seg-2.3gp"/>
        <Url sourceURL="seg-3.3gp"/>
      </SegmentInfo>
    </Representation>
  </Period>
</MPD>
```

Gambar 1 Contoh MPD [2]

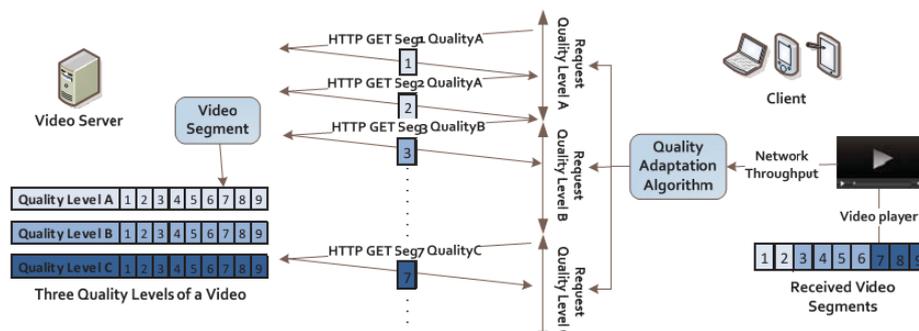


Gambar 2 Struktur MPD [3]



Gambar 3 Arsitektur MPEG-DASH [4]

Pada bagian kiri gambar 3 merupakan presentasi media dalam bentuk segmen video yang sudah dikodekan dengan kualitas yang bervariasi dan diletakkan pada server HTTP. Variasi media ini dikirimkan kepada konsumen menggunakan Media Presentation Description (MPD). Pertukaran data tersebut dapat dilakukan dengan menggunakan protokol HTTP. Proses MPEG-DASH pada klien saat terjadinya pengiriman video dapat ditunjukkan pada gambar 4 berikut ini.



Gambar 4 Proses MPEG-DASH pada klien [5]

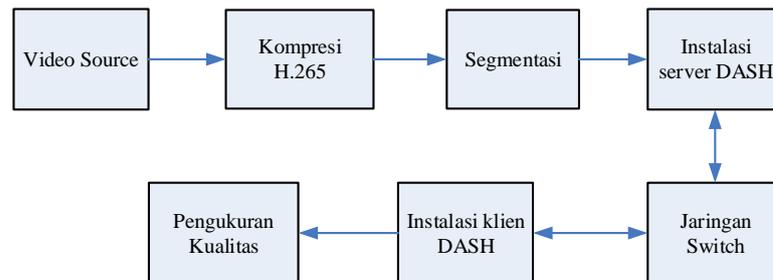
Server video menyimpan file MPD dan semua Encoding yang berbeda levelnya dari sebuah aliran video. Dalam gambar tersebut, video dikodekan menjadi tiga tingkatan kualitas yaitu kualitas A, B dan C. Klien tersebut dapat berupa perangkat streaming seperti desktop, laptop, smartphone, notepad dan perangkat lainnya. Pemutar video pada klien bertanggung jawab untuk streaming video secara adaptif tergantung pada kualitas yang direkomendasikan oleh algoritma adaptasi kualitas pada klien. Pada awalnya klien mulai meminta file MPD dari server. Setelah diterima, klien melakukan parsing file dan membuat daftar segmen yang diakses untuk setiap representasi. Selanjutnya klien mulai meminta Encoding terendah yang tersedia pada server. Setelah itu, algoritma adaptasi kualitas pada sisi klien mulai memperbarui kualitas yang direkomendasikan berdasarkan beberapa kriteria seperti throughput jaringan. Berikutnya klien memilih segmen yang disarankan dari representasi yang dipilih dan kemudian mengirimkan permintaan (HTTP GET request) pada server. Misalnya pada contoh ini tingkat kualitas yang dipilih dari segmen ketiga adalah kualitas B. Ketika klien melakukan pemutaran video, proses pemeriksaan tingkat kualitas yang tepat dan permintaan video melalui segmen berlangsung secara kontinyu pada setiap terjadinya permintaan. Kemudian pemutar video akan memasang kembali fragmen video tersebut. Akhir dari proses ini akan diperoleh video dengan segmen yang bervariasi tingkat kualitasnya.

[3] Perancangan Sistem

Pada bagian ini akan dijelaskan perancangan sistem yang terkait pada penelitian ini.

a. Blok Diagram Sistem

Adapun blok diagram yang digunakan pada sistem ini terlihat pada gambar 5 dibawah ini.



Gambar 5 Blok Diagram Sistem

Video source yang digunakan pada sistem ini merupakan sebuah video YUV yang belum terkompresi. Adapun spesifikasi video YUV tersebut dapat dilihat pada Tabel 1. Spesifikasi video yang ditunjukkan pada Tabel I memiliki jumlah frame sebanyak 1431. Namun pada sistem ini hanya menggunakan jumlah frame sebanyak 5760 dari video tersebut.

Tabel 1. Spesifikasi Video Source

Nama video	Resolusi (Piksel)	Jumlah Frame	Deskripsi
Big Buck Bunny	1280 x 720	5760	Animasi

A. Kompresi Video H.265

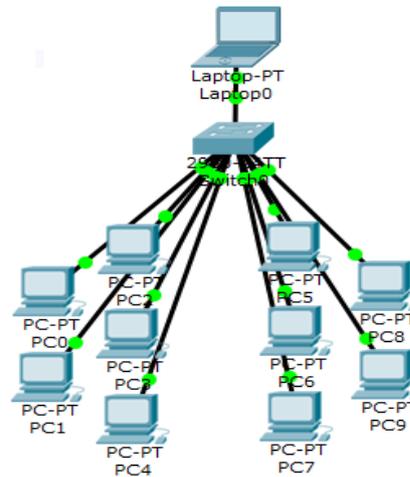
Encoding video H.265 dapat dilakukan dengan menggunakan DivX265. Dalam hal ini DivX265 dijalankan pada sistem operasi Windows. Pada saat ini DivX265 dapat mendukung proses *Encoding* video yang berasal dari *lossless* video (YUV). *Encoding* video dapat dilakukan berdasarkan *bit rate* dan resolusi video yang diinginkan oleh pengguna. Oleh karena itu untuk mendukung pengiriman video melalui DASH tersebut, maka *Encoding* video dilakukan dengan menghasilkan sejumlah *bit rate* video yang bervariasi. Tabel 2 ini menunjukkan spesifikasi *Encoding* video yang ingin dilakukan.

Tabel 2. Spesifikasi Encoding Video

No	<i>Bit rate</i> rata-rata	Resolusi	<i>Frame rate (fps)</i>
1	100	360p	24
2	300	360p	24
3	500	360p	24
4	1000	360p	24

B. Topologi Jaringan

Pada bagian ini akan ditunjukkan topologi jaringan yang digunakan pada penelitian ini. Adapun topologi yang digunakan ditunjukkan pada gambar 6 berikut ini.



Gambar 6 Topologi Jaringan

Pada penelitian ini menggunakan 1 unit kamera untuk perekaman video dan 1 unit laptop yang berfungsi sebagai HTTP Server. Jumlah klien yang digunakan pada penelitian ini sebanyak 10 klien berupa Komputer yang terhubung dengan jaringan switch untuk mengakses video pada HTTP Server.

C. Klien DASH

Adapun spesifikasi klien berupa perangkat bergerak yang digunakan pada penelitian ini ditunjukkan pada Tabel 3 berikut ini.

Tabel 3. Spesifikasi Klien

No	Komponen	Spesifikasi
1	Processor	Intel Octa-core 1,5 GHz
2	RAM	DDR2 2 GB
3	Sistem Operasi	Windows
4	Playback	Osmo4

Untuk mendukung proses DASH pada klien, maka pada klien harus dilakukan instalasi Osmo4 untuk melakukan playout video H.265 pada klien.

D. Pengukuran Kualitas

Pengukuran pada penelitian ini meliputi parameter-parameter *quality of service* yaitu *throughput*, *delay* dan *segment-segment video*, untuk kebutuhan analisis. Untuk pengukuran ini digunakan aplikasi *wireshark*.

[1] Hasil dan Analisa

Pada bagian ini peneliti akan memaparkan dan membahas hasil dari penelitian yang telah di kerjakan. Pembahasan ini dilakukan dengan mengambil data, menguji dan memberikan analisis terhadap sistem yang dibuat. Pengujian dan analisa bertujuan untuk membuktikan bahwa hasil dari penelitian ini sesuai dengan yang diharapkan dan telah melalui prosedur-prosedur yang seharusnya.

a. Kompresi Video H.265

Kompresi video H.265 menggunakan aplikasi divX265. Video source dengan format YUV dikompresi menjadi empat tingkatan bitrate yang berbeda. Hasil kompresi video tersebut dapat dilihat pada tabel 4 berikut ini.

Tabel 4. Hasil Kompresi Video

Nama Video	Durasi	Bitrate	Size
Bunny360_100.mp4	00:04:00	132 kbps	3951 KB
Bunny360_300.mp4	00:04:00	386 kbps	11388 KB
Bunny360_500.mp4	00:04:00	618 kbps	18186 KB
Bunny360_1000.mp4	00:04:00	1160 kbps	34063 KB

Dari hasil kompresi terlihat semakin tinggi bitrate video semakin besar ukuran yang dihasilkan. Bitrate ini merupakan banyaknya jumlah bit video dalam satuan waktu, pada bunny360_100.mp4 memiliki nilai bitrate 132 kbps yang berarti video ini dalam 1 detiknya memiliki jumlah bit sebanyak 132 ribu bit. Begitu juga dengan tiga video lainnya yang memiliki jumlah bit tiap detiknya lebih banyak. Nilai bitrate ini akan berpengaruh kepada kualitas dan ukuran video, semakin besar nilai bitrate nya kualitas video semakin bagus dan ukuran file videonya juga akan semakin besar.

b. Implementasi Live Streaming

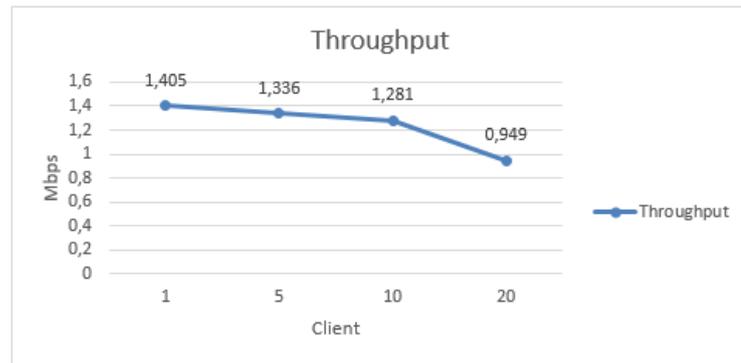
Implementasi dilakukan di lab 234 Politeknik Caltex Riau menggunakan satu buah laptop server dengan jumlah client yang ditentukan 1 client, 5 client, 10 client, dan 20 client. PC client dapat mengakses video yang di-streaming server menggunakan aplikasi Osmo4 (GPAC) dengan mengakses IP server.

c. Implementasi Live Streaming

Perbandingan parameter throughput pada sistem ini dapat dilihat pada gambar 7 dan gambar 8.

Display					
Display filter:		ip.src==172.16.30.16&&tcp			
Ignored packets:		0 (0,000%)			
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	28204	23762	84,250%	0	0,000%
Between first and last packet	203,769 sec	203,768 sec			
Avg. packets/sec	138,411	116,613			
Avg. packet size	1280 bytes	1506 bytes			
Bytes	36092295	35787164	99,155%	0	0,000%
Avg. bytes/sec	177123,362	175627,081			
Avg. MBit/sec	1,417	1,405			

Gambar 7 Data Throughput



Gambar 8 Grafik Data Throughput

Dari gambar 8 terlihat bahwa jumlah client berpengaruh terhadap nilai throughput. Semakin banyak jumlah *client* yang mengakses *video* nilai throughput semakin kecil yang menunjukkan jumlah *client* berbanding terbalik dengan nilai *throughput*.

a. Perbandingan Delay

Nilai parameter delay pada saat pengiriman video dapat dilihat pada gambar 9 dan gambar 10 berikut ini

Display					
Display filter:	ip.src==172.16.30.16&&tcp				
Ignored packets:	0 (0,000%)				
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	28204	23762	84,250%	0	0,000%
Between first and last packet	203,769 sec	203,768 sec			
Avg. packets/sec	138,411	116,613			
Avg. packet size	1280 bytes	1506 bytes			
Bytes	36092295	35787164	99,155%	0	0,000%
Avg. bytes/sec	177123,362	175627,081			
Avg. MBit/sec	1,417	1,405			

Gambar 9 Tampilan Data Delay



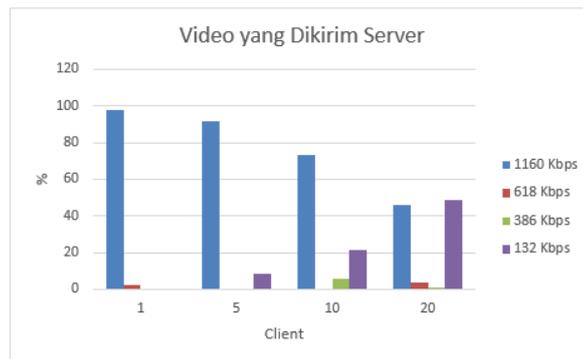
Gambar 10 Diagram Data Delay

Delay biasa dikaitkan dengan kecepatan *throughput*, dimana pada saat nilai *throughput* besar *delay* akan bernilai kecil. Pada percobaan dengan jumlah *client* 1 PC, nilai *throughput* yang terukur yaitu 1,405 Mbps, *delay* akses video 8,58 ms. Setelah dilakukan penambahan jumlah

client yang mengakses video menyebabkan nilai *throughput* berkurang dan menyebabkan waktu delay semakin besar.

b. Perbandingan Bit-rate Pengiriman Video

Pada saat streaming berlangsung, *client* melakukan request terhadap server untuk meminta video yang akan dimainkan di sisi *client*. Permintaan *client* akan ditanggapi server dengan mengirimkan segment-segment video dengan kualitas tertentu yang disesuaikan dengan waktu antara permintaan pertama dengan permintaan selanjutnya secara terus menerus selama proses streaming berlangsung.



Gambar 11 Perbandingan *bitrate* Video yang Dikirim Server ke Client

Gambar 11 menunjukkan persentase *bitrate video* yang *direquest client*. Terlihat dari gambar tersebut pada saat *client* 1 PC menggambarkan kondisi jaringan yang stabil, permintaan hanya *video* dengan *bitrate* 1160 dan 618 kbps. Saat ditambahkan jumlah *client* pada jaringan tersebut, persentase permintaan *video* dengan kualitas tinggi menurun. Sehingga pada saat *client* 20 PC, persentase permintaan *video* dengan kualitas rendah lebih besar yang menyatakan *traffic* jaringan padat.

Tabel 5. Hasil Kompresi Video

Bitrate video	Rata-rata waktu request
1160	0,51 detik
618	0,57 detik
386	0,61 detik
132	1,47 detik

Pengiriman segment video didasarkan pada lamanya waktu permintaan *client* terhadap server. Pada saat delay permintaan segment dari *client* cepat, maka server akan mengirimkan video dengan *bitrate* tinggi, dan apabila delay waktu permintaan *client* lama, maka server akan mengirimkan segment video dengan *bitrate* yang lebih rendah.

[2] Kesimpulan

Kompresi video dengan *bitrate* 132 kbps mengubah ukuran video menjadi 3951 KB, *bitrate* 386 kbps ukuran video menjadi 11388 KB, *bitrate* 618 kbps ukuran video menjadi 18186 KB, *bitrate* 1160 kbps ukuran video menjadi 34063 KB yang menunjukkan semakin tinggi *bitrate* video, ukuran file video juga akan semakin besar. Jumlah *client* mempengaruhi nilai *throughput* dan delay, semakin banyak jumlah *client* maka nilai *throughput* semakin kecil yang mengakibatkan delay menjadi semakin besar. Nilai *throughput* tertinggi yaitu 1,4 Mbps pada saat 1 *client*,

dan nilai *delay* terbesar 12,5 ms pada saat 20 *client*. Dynamic Adaptive Streaming over HTTP (DASH) berhasil di implementasikan, terbukti pada saat traffic jaringan yang padat persentase permintaan client untuk video dengan kualitas rendah lebih besar dibandingkan permintaan video dengan kualitas tinggi.

Daftar Pustaka

- [1] “HEVC: What it Is, What You Need to Know.” [Online]. Available: <http://www.onlinevideo.net/2013/04/hevc-what-it-is-what-you-need-to-know/>. [Accessed: 06-Aug-2019].
- [2] R. Pantos and W. May, “HTTP Live Streaming,” *IETF draft*. pp. 1–33, 2012.
- [3] R. Y. Rahmanda, E. S. Pramukantoro, and W. Yahya, “Perancangan dan Implementasi Kelas Virtual FILKOM Universitas Brawijaya dengan Memanfaatkan Teknologi WebRTC (Web Real-Time Communication),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, 2018.
- [4] “H.265: High efficiency video coding.” [Online]. Available: <http://www.itu.int/rec/T-REC-H.265-201304-S>. [Accessed: 30-Apr-2017].
- [5] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” 1996.