

**KAJIAN IMPLEMENTASI
IPTV (INTERNET PROTOCOL TELEVISION)
OVER JARINGAN METRO NG-SDH DI TELKOM RISTI BANDUNG**

**RESEARCH IMPLEMENTATION
IPTV (INTERNET PROTOCOL TELEVISION) OVER METRO NG-SDH NETWORK
IN TELKOM RISTI BANDUNG**

Maryono¹, Rendy Munadi², Lukman Hakim³

¹Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

Abstrak

IPTV (Internet Protocol Television) merupakan sistem pengiriman broadcast televisi digital dengan menggunakan metode IP Multicast pada infrastruktur jaringan dengan koneksi broadband. Namun pada aplikasinya, IPTV sering ditawarkan bersamaan dengan VoD (Video on Demand) dan Internet services. IP Multicast merupakan mekanisme pengiriman paket IP dengan cara mengirimkan satu stream paket tunggal ke multiple penerima yang terdaftar dalam suatu group multicast. Metode ini bertujuan untuk memberikan efisiensi bandwidth disisi uplink/WAN.

NG-SDH (next generation SDH) merupakan mekanisme transport yang mengakomodasi berbagai macam data interface ke dalam SDH secara efisien. Sebelumnya SDH (Synchronous Digital Hierarchy) hanya dioptimalkan untuk transport voice, namun dengan berkembangnya trafik data maka SDH juga diarahkan dapat dilalui trafik data. Dalam Tugas akhir ini dilakukan pembuatan sistem minimum agar IPTV bisa berjalan pada jaringan NG-SDH. Agar user bisa mengakses layanan IPTV, dalam sistem ini digunakan IP STB sebagai interface user. Dalam hal ini akan dikaji implementasinya di PT Telekomunikasi Indonesia Tbk divisi RISTI Bandung. Berdasarkan hasil analisa implementasi IPTV over jaringan metro NG-SDH, bisa disimpulkan bahwa aplikasi ini memiliki performansi yang baik.

Kata Kunci : IPTV, IP multicast, NG-SDH

Abstract

IPTV (Internet Protocol Television) is a system where a digital television broadcast service is delivered using the IP Multicast method over a network infrastructure, which may include delivery by a broadband connection. For residential users, IPTV is often provided in conjunction with VoD (Video on Demand) and may be bundled with Internet services. IP Multicast is an IP packet transport mechanism by sending one stream single packet to the multiple receiver that registered in a multicast group.

NG-SDH (next generation SDH) is a solution of SDH development. Initially, SDH (Synchronous Digital Hierarchy) just optimized to transport voices. Because of development of data traffic, so SDH also forced to transfer data traffic. NG SDH is a transport mechanism that accomodate various data interface into SDH efficiently. In this final project will be made a minimum system of IPTV over NG-SDH network. This application using IP STB as a user interface in order to access the services. This research held in PT Telekomunikasi Indonesia Tbk. R&D Center Bandung. Based on analysis implementation of IPTV over Metro NG-SDH, can be conclude that this application has good performance.

Keywords : IPTV, IP Multicast, NG-SDH

BAB I

PENDAHULUAN

I.1. Latar Belakang

Memahami akan perubahan dan percepatan arus informasi pada era globalisasi khususnya komunikasi berbasis IP, setiap individu dan organisasi harus mampu menemukan inspirasi baru untuk memasuki persaingan teknologi pada dekade awal abad 21. Untuk itu diperlukan suatu pola pikir baru yang mampu memberikan persiapan dalam memasuki dunia *Information Technology* (IT).

IT merupakan penggabungan antara teknologi komputer beserta *content* yang ada didalamnya dengan teknologi telekomunikasi dengan berbagai kemampuan dalam melakukan pengiriman atau penerimaan berbagai sumber informasi, baik berupa data, suara (*voice*) maupun gambar bergerak (*video*).

Jaringan multi service menjadi topik hangat dan banyak diminati pada saat sekarang ini karena mampu membawa berbagai macam layanan seperti voice, data dan video melalui jaringan IP. Namun layanan multiservice akan mengakibatkan kebutuhan *bandwidth* yang semakin meningkat. Dengan semakin meningkatnya kebutuhan bandwidth menimbulkan masalah baru dalam hal kualitas dari pelayan terutama pada layanan yang sangat sensitif terhadap delay seperti voice dan streaming.

Sebelumnya SDH (*Synchronous Digital Hierarchy*) hanya dioptimalkan untuk *transport voice*, namun dengan berkembangnya trafik data maka SDH juga diarahkan dapat dilalui trafik data. Sebagaimana telah diperkirakan sebelumnya bahwa perusahaan-perusahaan dalam beberapa tahun ke depan akan memindahkan layanan *voice* melalui data dimana akan menawarkan biaya yang lebih rendah. Ini berarti bahwa kedepan SDH tidak support terhadap pertumbuhan layanan data. *Next Generation* SDH memiliki mekanisme transport yang

memungkinkan layanan-layanan existing dan baru dapat dilewatkan dalam network yang sama tanpa saling mengganggu.

I.2. Rumusan Masalah

Permasalahan yang dijadikan objek penelitian dan pengembangan dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana membuat implementasi layanan IPTV pada jaringan metro NG-SDH khususnya di lingkungan Telkom Risti Bandung?
2. Bagaimana cara kerja IPTV pada jaringan metro NG-SDH khususnya di lingkungan Telkom Risti Bandung?
3. Bagaimana mendapatkan parameter nilai yang dapat digunakan sebagai acuan untuk menentukan parameter kualitas *IPTV over Metro NG-SDH* di Telkom Risti Bandung

I.3. Batasan Masalah

Dalam tugas akhir ini, penelitian dilakukan dengan beberapa pembatasan, yaitu :

1. Lebih menitikberatkan pada pembuatan dan implementasi jaringan untuk menyediakan aplikasi IPTV yang dilakukan di Telkom Risti Bandung.
2. Tidak membahas aspek keamanan yang dipakai
3. Analisa kualitas jaringan dari layanan tersebut berdasarkan parameter-parameter *Packet loss, Throughput, Delay, dan Jitter*
4. Tidak membahas mengenai pembuatan *server headend* untuk layanan IPTV.
5. Tidak membahas mengenai sistem modulasi dan coding

I.4. Tujuan Penelitian

Tujuan tugas akhir didasarkan dari paparan masalah diatas adalah sebagai berikut:

1. Membuat topologi jaringan sistem minimum yang dapat didesain, diimplementasikan, dan dapat mewakili untuk layanan *IPTV over Metro NG-SDH* di Telkom Risti Bandung

2. Menganalisa performansi kualitas IPTV pada jaringan NG-SDH Telkom RISTI Bandung.
3. Diharapkan hasil analisis dapat dijadikan referensi untuk pengembangan lebih lanjut atau untuk implementasi lebih luas.

I.5. Metode Penelitian

Sistem informasi dirancang dengan pendekatan terstruktur sebagai berikut:

1. Perumusan masalah
2. Penetapan tujuan
3. Studi literature dan studi lapangan
4. Pengumpulan data
5. Analisa spesifikasi kebutuhan sistem
6. Desain sistem dan perancangan implementasi jaringan
7. Uji coba perangkat dan pengukuran
8. Analisa dan pengambilan kesimpulan

I.6. Sistematika Penulisan

Secara umum keseluruhan Tugas Akhir ini akan dibagi menjadi lima bab bahasan, ditambah dengan lampiran dan daftar istilah yang diperlukan.

BAB I PENDAHULUAN

Berisi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, metode penelitian, dan sistematika penulisan.

BAB II DASAR TEORI

Berisikan teori-teori yang mendukung dan melandasi penulisan tugas akhir ini, antara lain yaitu tentang konsep dasar IPTV dan konsep dasar metro NG-SDH.

BAB III PERANCANGAN SISTEM DAN REALISASI

Berisi tentang implementasi IPTV over Metro NG-SDH, meliputi perancangan sistem jaringan, penyediaan perangkat pendukung sistem, dan pendimensian jaringan yang telah dilakukan di

Telkom RisTI Bandung.

BAB IV ANALISA SISTEM

Berisi tentang analisa terhadap sistem hasil perancangan. Pada bab ini akan dijelaskan analisa data–data yang diperoleh dari hasil implemntasi tentang parameter kualitas yaitu packet loss, delay, throughput, dan jitter.

BAB V KESIMPULAN DAN SARAN

Berisikan kesimpulan dari analisa yang telah dilakukan, serta rekomendasi atau saran untuk perbaikan dan pengembangan lebih lanjut.

ST **Telkom**
University

BAB IV

PENGUJIAN DAN ANALISA

IV.1. Gambaran Analisa

Pada bab ini akan dibahas hasil implementasi yang telah dilakukan. Dari hasil implementasi diperoleh parameter-parameter yang mempengaruhi performansi dari aplikasi layanan IPTV pada jaringan metro ONS SDH di Telkom RisTI Bandung.

Kajian implementasi dijalankan pada area Telkom RisTI Bandung yang meliputi tiga gedung, yaitu Laboratorium Wireline Akses, Menara RisTI, dan Gedung RisTI. Pada masing-masing gedung terdapat metro ONS SDH yang saling terhubung menggunakan fiber optik.

Pengukuran dilakukan dengan melakukan akses aplikasi IPTV secara langsung menggunakan perangkat STB dan PC yang terdapat pada *client* yang telah dibangun dan telah terhubung pada server *headend*. Selama proses komunikasi dilakukan perekaman *trafik* data dengan *network analyzer*, kemudian setelah proses pengukuran berakhir akan dilakukan proses *filterisasi* objek yang diukur, karena tidak hanya *trafik video broadcast* saja yang terekam dalam jaringan tersebut. Proses hubungan panggilan atau perekaman rata-rata dilakukan selama 60 detik. Dalam pengukuran ini dilakukan dalam dua skenario. Pengujian masing masing skenario dilakukan secara berulang sampai 10 kali percobaan.

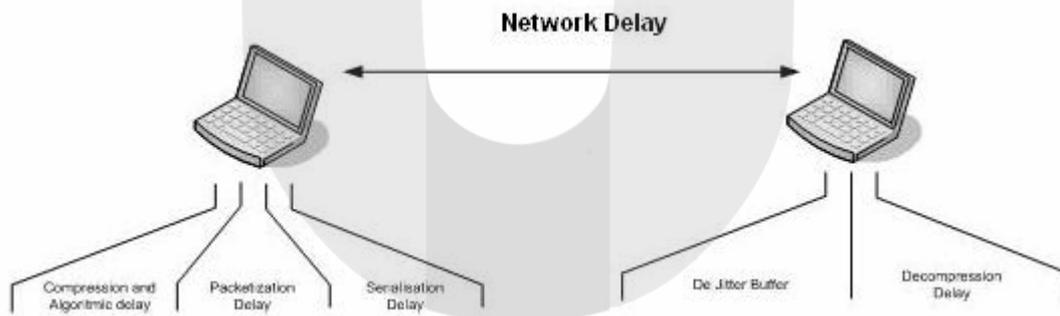
Analisa menggunakan alat ukur *Wireshark* yaitu software *network protokol analyzer*. *Wireshark* merupakan varian terbaru dari *Ethereal* yang berbasis windows. *Wireshark* digunakan untuk meng-*capture* paket-paket data yang terdapat pada jaringan. Dari data-data hasil pengcapturan dan dengan menggunakan rumus yang telah diformulasikan maka akan diperoleh *delay network*, *jitter*, *delay coder*, *delay paketization*, *delay serialization*, *packet loss*, *throughput*, dan *actual rate video/frame rate*.

IV.2. Pengukuran dan Analisa Performasi

IV.2.1 Pengukuran Delay

Pengukuran ini bertujuan untuk mengevaluasi *delay* satu arah pada sistem broadcasting dari satu *entitas* ke *entitas* lainnya (*one way delay*). *Delay* atau *latency* adalah waktu yang diperlukan oleh suatu paket data dari *source node* hingga mencapai *destination*. Pengujian ini juga dilakukan untuk membandingkan *delay* dari penggunaan *codec* yang berbeda pada server, yaitu H.264 dan MPEG 2.

Dalam teknologi *video broadcast* untuk mengurangi beban *server* maka digunakan sistem *multicast* yang menggunakan UDP sebagai protocol jaringan. Sedangkan parameter *delay* disebabkan oleh beberapa komponen *delay* yang secara garis besar yaitu *delay coder* (*processing*), *delay serialization*, *delay packetization*, *delay dejitter buffer* dan *delay network*.



Gambar IV.1 Topologi perhitungan *one way delay budget*

- *Coder (Processing) Delay*

Delay ini disebabkan oleh standar *codec* yang digunakan. Adapun besarnya *codec* adalah sebagai berikut :

$$\text{Coder (Processing)} = (\text{Waktu kompresi}) + (\text{Waktu dekompresi})$$

Dalam perhitungan delay coder h.264 dan mpeg 2 dilakukan dengan cara pendekatan dengan *codec* yang bersifat *proprietary*. Dengan *codec* yang bersifat *proprietary* hanya didapatkan data delay kompresi dekompresi berkisar antara 8 – 11 ms. Untuk analisa maka diambil *delay* maksimal yaitu 11 ms.

- *Packetization Delay*

Delay ini disebabkan oleh pengakumulasian bit *voice* sample ke frame. Mengacu pada hasil pengukuran *network analyzer* diketahui panjang paket data, besar data informasi paket dapat diperoleh dengan cara sebagai berikut :

Data *payload* = Panjang paket IP – (Ethernet header + IP header + UDP header)

$$\begin{aligned} \text{Payload} &= 1358 \text{ bytes} - (14 + 20 + 8) \\ &= 1316 \text{ bytes} \end{aligned}$$

Data untuk delay standard pakettisasi dan *depaketisasi* *codec* berkisar antara 4 - 6 ms. Untuk analisa maka diambil *delay* maksimal yaitu 6 ms.

- *Serialization Delay*

Delay ini terjadi karena adanya waktu yang dibutuhkan untuk pentransmisian paket IP dari sisi *originating* (pengirim). Mengacu pada hasil pengukuran *network analyzer* saat *multicast* diketahui rata-rata ukuran paket data untuk masing-masing *setting*. Besar *delay* serialisasi dapat diperoleh dengan rumus berikut :

$$\text{Serialization Delay (ms)} = \frac{\text{Packet Size (bytes)} \times 8}{\text{Link speed (kbps)}}$$

Pada pengujian kecepatan taransfer data yang digunakan pada modem ADSL2 sebesar 16 Mbps.

$$\begin{aligned} \text{Serialization Delay (ms)} &= \frac{1358 \text{ (bytes)} \times 8}{16000 \text{ (Kbps)}} \\ &= 0.658 \end{aligned}$$

- *Network Delay*

Delay dapat diukur sebagai waktu interval antara ketika paket mulai dikirimkan dan keluar setelah melalui proses antrian di titik sumber awal sampai diterima di titik yang dituju. Dalam simulasi yang diukur adalah *one way delay*. Berdasarkan standar ITU-T yaitu *delay* yang masih dapat diterima adalah sebesar kurang dari 150 ms.

Berikut ini hasil pengukuran *network delay* yang telah disajikan dalam bentuk tabel:

Tabel IV.1 *Network Delay* hasil pengukuran

Pengukuran	Delay Network [ms]	
	H 264	MPEG 2
1	14.186934	13.809545
2	14.060136	13.568594
3	16.826147	13.850804
4	15.938248	13.679298
5	14.315500	14.130308
6	13.725500	13.499754
7	13.558158	13.742701
8	14.615053	14.499977
9	15.676449	14.186934
10	14.823194	14.060136

- *Delay Jitter Buffer*

Delay jitter buffer merupakan lama paket bisa ditampung untuk proses antrian. Untuk nilai *delay jitter buffer* telah dikonfigurasi dengan besar 50 ms untuk masing-masing *endpoint*. Besarnya nilai *buffer*, bisa dikonfigurasi di *client*.

Berdasarkan data yang didapat dari hasil pengukuran tersebut maka *one way delay* dapat dihitung dengan menjumlahkan *coder processing delay*,

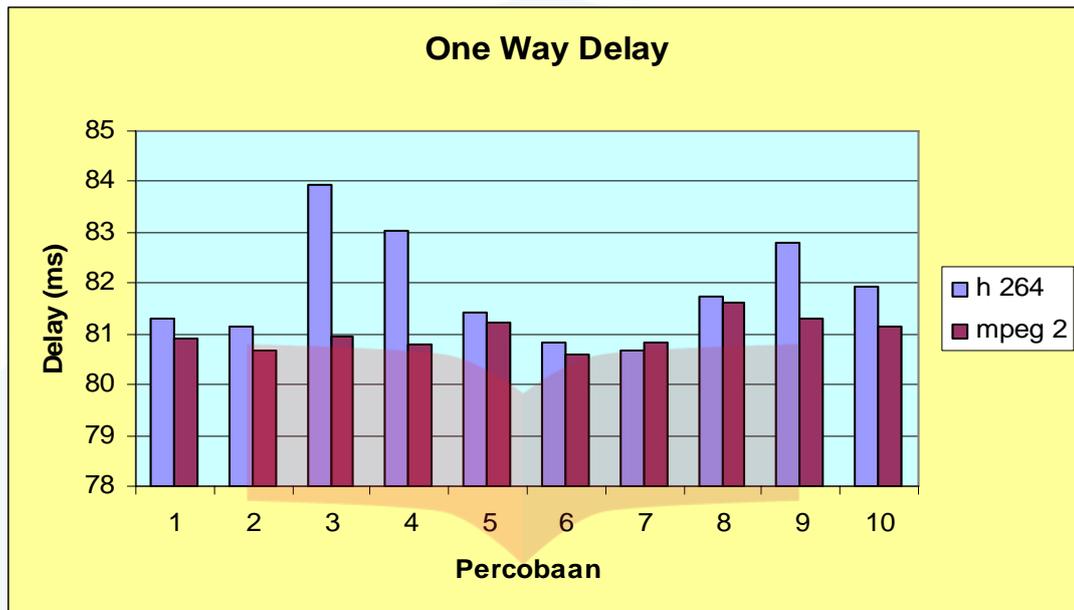
packetization delay, serialization delay, dan network delay, dan delay jitter buffer. Tetapi karena delay pengkodean dan paketisasi hanya berupa range, maka untuk analisa diambil nilai yang paling besar. Hasil perhitungan tersebut diperlihatkan pada tabel berikut ini :

Tabel IV.2 Rata-rata hasil perhitungan one way delay H.264

No	Delay Network [ms]	Delay Paketisasi [ms]	Delay Coder [ms]	Delay Serialisasi [ms]	Delay Buffer [ms]	Delay Total [ms]
1	14.186934	6.0000	11.0000	0.6580	50.0000	81.2914
2	14.060136	6.0000	11.0000	0.6580	50.0000	81.1646
3	16.826147	6.0000	11.0000	0.6580	50.0000	83.9306
4	15.938248	6.0000	11.0000	0.6580	50.0000	83.0427
5	14.315500	6.0000	11.0000	0.6580	50.0000	81.4200
6	13.725500	6.0000	11.0000	0.6580	50.0000	80.8300
7	13.558158	6.0000	11.0000	0.6580	50.0000	80.6627
8	14.615053	6.0000	11.0000	0.6580	50.0000	81.7196
9	15.676449	6.0000	11.0000	0.6580	50.0000	82.7809
10	14.823194	6.0000	11.0000	0.6580	50.0000	81.9277

Tabel IV.3 Rata-rata hasil perhitungan one way delay MPEG 2

No	Delay Network [ms]	Delay Paketisasi [ms]	Delay Coder [ms]	Delay Serialisasi [ms]	Delay Buffer [ms]	Delay Total [ms]
1	13.809545	6.0000	11.0000	0.6580	50.0000	80.9140
2	13.568594	6.0000	11.0000	0.6580	50.0000	80.6731
3	13.850804	6.0000	11.0000	0.6580	50.0000	80.9553
4	13.679298	6.0000	11.0000	0.6580	50.0000	80.7838
5	14.130308	6.0000	11.0000	0.6580	50.0000	81.2348
6	13.499754	6.0000	11.0000	0.6580	50.0000	80.6043
7	13.742701	6.0000	11.0000	0.6580	50.0000	80.8472
8	14.499977	6.0000	11.0000	0.6580	50.0000	81.6045
9	14.186934	6.0000	11.0000	0.6580	50.0000	81.2914
10	14.060136	6.0000	11.0000	0.6580	50.0000	81.1646



Grafik 1 Perbandingan one way delay H 264 dan MPEG 2

- Delay Access

Selain one way delay, ketika user mengakses aplikasi akan ada *delay akses*, sebagai akibat dari *one way delay* tersebut. Berikut hasil pengukuran *delay akses* :

Tabel IV.4 Delay akses

Pengukuran	Delay [second]
1	1.65
2	1.50
3	1.60
4	1.70
5	1.55
6	2.15
7	2.10
8	1.95
9	2.10
10	2.00

IV.2.1.1 Analisa Hasil Pengukuran *Delay*

Dari hasil pengukuran dan perhitungan dapat dilihat bahwa *delay* untuk *buffering* paling besar. Melihat hasil pengukuran dan perhitungan pada Tabel 2 dan Tabel 3, diketahui bahwa *one way delay* untuk *codec* H 264 lebih besar dibanding *delay* MPEG 2. Hal ini disebabkan karena besar *bandwidth* pada *codec* H 264 lebih kecil daripada paket untuk *codec* MPEG 2. Meskipun demikian *delay* untuk aplikasi dengan *codec* H 264 masih bisa ditolerir karena masih lebih kecil dari 150 ms sesuai standart ITU-T. Dan H 264 memiliki efisiensi *bandwidth* yang lebih baik yaitu 2 Mb.

Pada aplikasi live TV menggunakan skenario *multicast* untuk mendapatkan efisiensi *bandwidth* jaringan dan untuk mengurangi kinerja sistem *headend* atau server. Dan untuk mengurangi *delay* digunakan protocol UDP yang di disain untuk transmisi data gambar atau video yang mana tidak menggunakan sistem *recovery error packet*.

IV.2.2 Pengukuran *Jitter*

Jitter merupakan variasi dari *delay* diantara satu kedatangan paket dengan kedatangan paket yang lain. *Jitter* merupakan masalah yang masih ada dan terus ada dalam jaringan data berbasis paket. Jika frame di transmisikan lewat jaringan IP, tiap frame akan mengalami *delay* yang berbeda. Harga dari *jitter* yang tinggi dapat mengurangi kualitas layanan. Berdasarkan standar Cisco yaitu *jitter* yang masih dapat ditoleransi adalah sebesar kurang dari 30 ms.

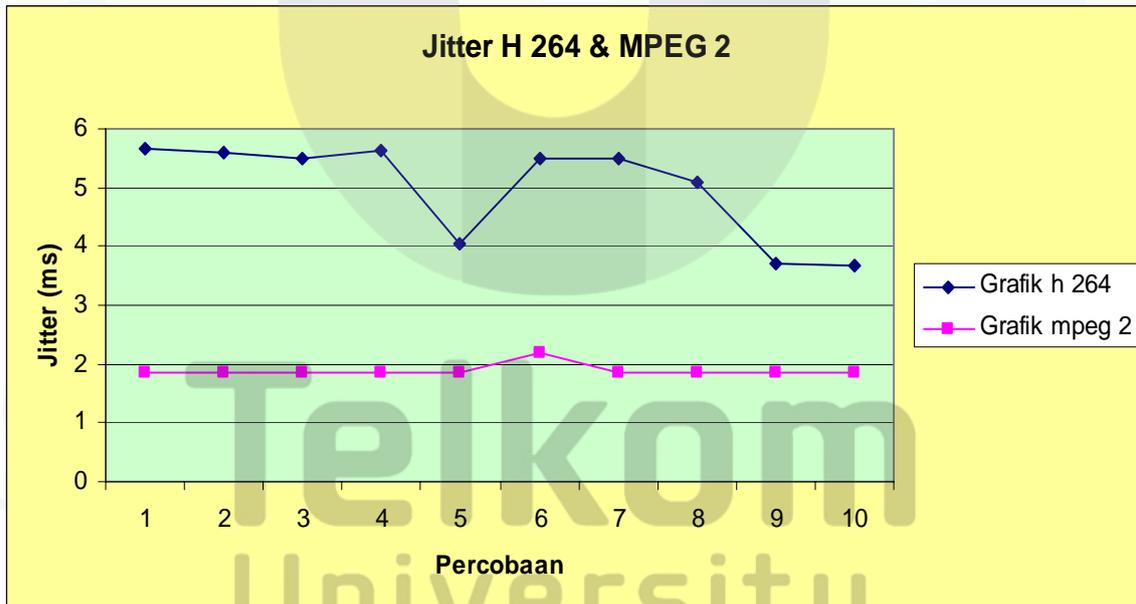
Jitter dapat disebabkan oleh lintasan tempuh dari paket yang berbeda-beda atau bisa juga disebabkan karena *collison* pada jaringan, sehingga menyebabkan paket memiliki waktu tempuh yang berbeda. Pengukuran ini dilakukan untuk mengetahui besarnya interval waktu antar paket yang dikirimkan oleh *entitas originating* ke *destination terminal*.

Konfigurasi pengukuran sama dengan konfigurasi pengukuran saat mengambil *one way delay*. Akan tetapi *filterisasi* yang dilakukan pada *network analyzer* hanyalah *jitter* yang terjadi pada saat terjadi pengiriman paket *video*. Oleh karena itu *filter* yang digunakan adalah *filter* untuk paket protokol UDP

untuk sistem *multicast*. Pengukuran dilakukan secara berulang 10 kali percobaan. Berikut adalah hasil pengukuran *jitter* :

Tabel IV.5 Hasil Pengukuran Jitter

Pengukuran	Rata-rata Jitter (ms)	
	H 264	MPEG 2
1	5.658	1.849
2	5.607	1.846
3	5.51	1.845
4	5.643	1.847
5	4.03	1.849
6	5.486	2.202
7	5.486	1.849
8	5.101	1.843
9	3.702	1.848
10	3.674	1.847



Grafik 2 Perbandingan jitter H 264 dan MPEG 2

IV.2.2.1 Analisa Hasil Pengukuran *Jitter*

Jitter sangat erat kaitannya dengan delay. *Jitter* dapat disebabkan lintasan tempuh paket yang berbeda-beda. *Jitter* akan terpengaruh oleh *buffer*. Semakin besar nilai *buffer*, maka akan semakin besar kemungkinan untuk dapat mengakomodasi *jitter* untuk dapat disinkronisasi. Dari Grafik 2 terlihat bahwa *jitter* untuk H 264 lebih besar dari *jitter* MPEG 2. Hal ini disebabkan karena delay untuk H 264 juga lebih besar dibanding MPEG 2.

Merujuk pada rekomendasi yang dikeluarkan oleh CISCO, bahwa *jitter* yang masih dapat ditoleransi adalah kurang dari 30 ms. Dari hasil percobaan terlihat rata-rata *jitter* masih termasuk dalam rekomendasi. Sehingga *jitter* masih dapat diterima.

IV.2.3 Pengukuran *Packet Loss* dan *Throughput*

Packet loss merupakan jumlah paket yang hilang dalam perjalanan atau di jaringan terhadap paket yang dikirim. *Packet Lost* dalam jaringan data merupakan hal yang biasa terjadi.

Topologi pengukuran masih sama dengan yang sebelumnya. Namun untuk mengetahui *packet loss* dan *packet receive* digunakan software VLC yang mampu menunjukkan jumlah *loss frame* dan *display frame* dalam setiap pengukuran. Pengukuran dilakukan secara berulang 10 kali percobaan. Besar *packet loss* dan *packet receive* dapat diperoleh dengan cara berikut :

$$\text{Packet loss} = \text{Loss frame} \times 1358 \text{ bytes}$$

$$\text{Packet receive} = \text{Display frame} \times 1358 \text{ bytes}$$

Nilai 1358 bytes diperoleh dari hasil capture menggunakan *wireshark*, dimana 1 frame sama dengan 1358 bytes.

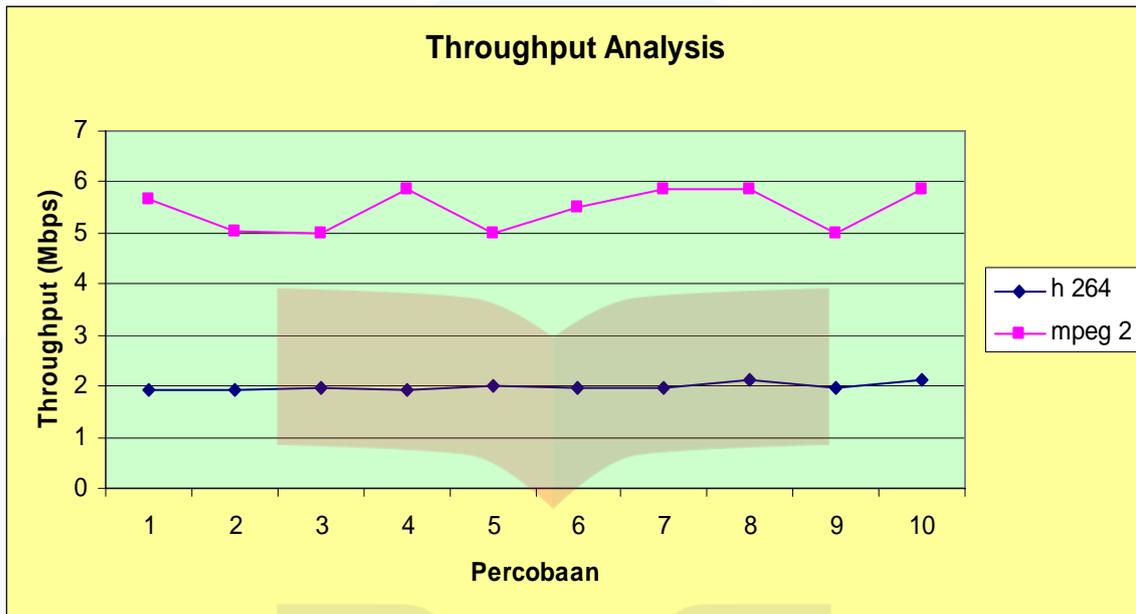
Tabel IV.6 Hasil perhitungan *Packet Loss*

Pengukuran	Loss Frame		Packet Loss (bytes)	
	H 264	MPEG 2	H 264	MPEG 2
1	4	5	5432	6790
2	14	11	19012	14938
3	12	5	16296	6790
4	8	5	10864	6790
5	4	5	5432	6790
6	10	7	13580	9506
7	4	6	5432	8148
8	10	6	13580	8148
9	6	8	8148	10864
10	4	5	5432	6790

Sedangkan nilai throughput diperoleh dari summary hasil capturing pada wireshark. Berikut adalah nilai throughput untuk H 264 dan MPEG 2 yang disajikan dalam bentuk tabel.

Tabel IV.7 Hasil *Throughput* dari *wireshark*

Pengukuran	Throughput (Mbps)	
	H 264	MPEG 2
1	1.911	5.646
2	1.932	5.029
3	1.975	5.013
4	1.939	5.848
5	2.015	5.011
6	1.978	5.503
7	1.978	5.85
8	2.131	5.85
9	1.981	4.976
10	2.123	5.849



Grafik 3 Perbandingan *Throughput* H 264 dan MPEG 2

IV.2.3.1 Analisa Hasil Pengukuran Packet Loss dan Throughput

Throughput dipengaruhi dari besar *jitter* yang terjadi. Semakin besar *jitter*, maka *throughput* akan semakin menurun. Merujuk pada hasil perhitungan *jitter* di awal terlihat bahwa besar nilai *throughput* sangat bergantung pada nilai *jitter*.

IV.2.4 Pengukuran *Frame Rate*

Frame rate merupakan banyaknya *frame* yang bisa terlihat di end user dalam satuan waktu tertentu. Percobaan ini mengukur *frame* video yang dilihat end user. Dimana terdapat minimal *frame rate* yang harus diterima oleh user.

Untuk mengetahui *frame rate* digunakan software VLC yang mampu menunjukkan jumlah *display frame* dalam setiap pengukuran. Pengukuran dilakukan secara berulang 10 kali percobaan. Besar *frame rate* dapat diperoleh dengan cara berikut :

$$\text{Frame rate} = \frac{\text{Display frame}}{60 \text{ sekon}}$$

Nilai 60 second merupakan lamanya waktu pengamata rata-rata tiap percobaan. Berikut adalah hasil perhitungan *frame rate* yang disajikan dalam bentuk tabel.

Tabel IV.8 Hasil Perhitungan Frame Rate

Pengukuran	Display frame		Frame Rate (fps)	
	H 264	MPEG 2	H 264	MPEG 2
1	1499	1498	24.9833	24.9667
2	1519	1757	25.3167	25.2833
3	1473	1765	24.5500	25.4167
4	1466	1484	24.4333	24.7333
5	1492	1741	24.8667	25.0167
6	1465	1503	24.4167	25.0500
7	1508	1492	25.1333	24.8667
8	1480	1479	24.6667	24.6500
9	1509	1532	25.1500	25.5333
10	1482	1493	24.7000	24.8833

IV.2.4.1 Analisa Hasil Perhitungan *Frame Rate*

Frame rate dipengaruhi dari besar *packet lost* yang terjadi. Semakin besar *packet lost*, maka *frame rate* akan semakin menurun. Standar *frame rate* yang baik adalah lebih besar 17 *fps*. Dari hasil perhitungan terlihat bahwa keduanya memiliki *frame rate* lebih besar dari 17 *fps*. Sehingga *frame rate* bisa diterima atau masih memenuhi syarat.

IV.3 Estimasi Parameter Hasil Pengukuran ke MOS

Untuk menentukan nilai kualitas layanan dalam tugas akhir ini dilakukan pendekatan dengan menggunakan MOS (*Mean Opinion Score*). Untuk itu harus dilakukan konversi parameter kedalam *E-Model*. *E-Model* didefinisikan dalam ITU recommendation G.170. Nilai akhir dari esimasi *E-Model* disebut dengan "*R factor*". Untuk mencari nilai *R factor* digunakan rumus berikut:

$$R = 94.2 - I_d - I_{ef}$$

Maka terlebih dahulu dicari besar nilai I_d dan I_{ef} . Nilai I_d berkaitan dengan besar *one way delay* jaringan. Sedangkan nilai I_{ef} berkaitan dengan besarnya *packet loss* yang terjadi. Berikut adalah rumus untuk menghitung I_d dan I_{ef} :

$$I_d = 0.024 d + 0.11 (d-177.3) H(d-177.3); \text{ dan}$$

$$I_{ef} = 7 + 30 \ln (1 + 15e)$$

Konfersi Nilai R-Factor menjadi MOS

Nilai MOS yang digunakan dalam perhitungan tugas akhir ini menggunakan standarisasi dari ITU- P800. Dengan nilai R Factor yang dihasilkan berada pada range $0 < R > 100$, maka nilai MOS diperoleh dengan rumus berikut:

$$MOS = 1 + 0.035 R + 7 \times 10^{-6} R(R-60)(100-R)$$

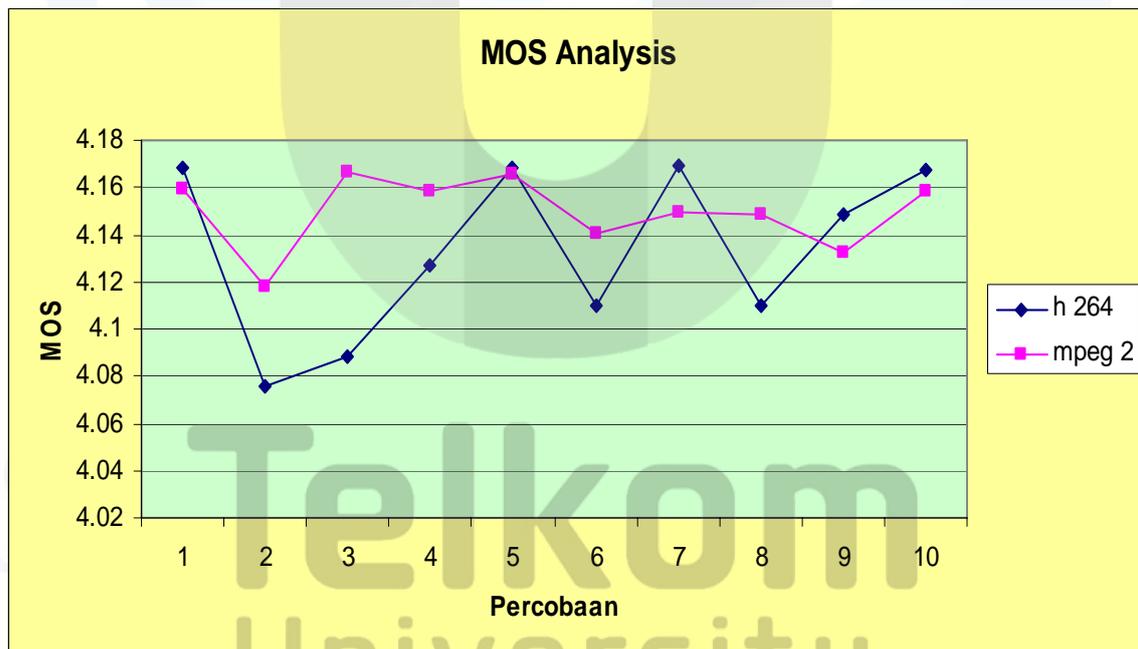
Berikut adalah hasil perhitungan MOS:

Tabel IV.9 Hasil Perhitungan H 264 dengan MOS

Pengukuran	Delay Total (ms)	Id	Packet Loss (%)	Packet Loss (desimal)	Ief	R Factor	MOS
1	81.29143	1.95099	0.26613	0.00266	8.17432	84.07469	4.16825
2	81.16464	1.94795	0.91324	0.00913	10.85143	81.40061	4.07583
3	83.93065	2.01434	0.80808	0.00808	10.43231	81.75335	4.08852
4	83.04275	1.99303	0.54274	0.00543	9.34800	82.85897	4.12733
5	14.315500	1.94823	0.26810	0.0027	8.18281	84.06896	4.16787
6	13.725500	1.89613	0.67797	0.00678	9.90549	82.39838	4.10981
7	80.66266	1.93590	0.26455	0.00265	8.16746	84.09663	4.16897
8	81.71955	1.96127	0.67114	0.00671	9.87760	82.36113	4.11004
9	82.78095	1.98674	0.39604	0.00396	8.73125	83.48201	4.14854
10	81.92769	1.96626	0.26918	0.00269	8.18749	84.04625	4.16732

Tabel IV.10 Hasil Perhitungan MPEG 2 dengan MOS

Pengukuran	Delay Total (ms)	Id	Packet Loss (%)	Packet Loss (desimal)	Ief	R Factor	MOS
1	80.91405	1.94194	0.33378	0.00334	8.17432	83.79245	4.15892
2	80.67309	1.93615	0.62217	0.00622	9.67673	82.58712	4.11792
3	80.95530	1.94293	0.28249	0.00282	8.24499	84.01208	4.1662
4	80.78380	1.93881	0.33580	0.00336	8.47426	83.78693	4.15874
5	81.23481	1.94964	0.28637	0.00286	8.26175	83.98861	4.16542
6	80.60425	1.93450	0.46358	0.00464	9.01676	83.24874	4.14065
7	80.84720	1.94033	0.40053	0.00401	8.75033	83.50933	4.14945
8	81.60448	1.95851	0.40404	0.00404	8.76522	83.47628	4.14834
9	81.29143	1.95099	0.51948	0.00519	9.25106	82.99795	4.13210
10	81.16464	1.94795	0.33378	0.00334	8.46561	83.78644	4.15872



Grafik 4 Perbandingan MOS H 264 dan MPEG 4

IV.3.1 Analisa Hasil Perhitungan MOS

Dari Grafik 4 bisa diketahui bahwa MOS masih dalam range baik, sehingga hasil perhitungan masih bisa ditolerir. Bisa dilihat juga bahwa perbedaan kedua codec tidak berpengaruh signifikan terhadap kualitas video.



ST **Telkom** *UNIVERSITY*
University

DAFTAR PUSTAKA

- [1].Azikin, Askari, *Video/TVStreaming dengan Video Lan Project*, Yogyakarta: Andi Offset, 2005.
- [2].Hasan, Taufik, *Implementasi Layanan IPTV*, Bandung: PT Telekomunikasi Tbk. R&D Center, 2005.
- [3].Britt, Roger. 2000. *IP Telephony*. FCC Industry Meeting : Nortel Networks.
- [4].Cisco System. 2004. *Understanding Delay in Packet Voice Networks*. USA : Cosco Press.
- [5].NetPredict.Inc. 2003. *Performance Analysis for Video Stream across Network*.USA Menlo Park
- [6].Minoli, Daniel. 1998. *Delivering Voice over IP Networks*. New York : John Wiley Press.
- [7]. www.cisco.com
- [8]. www.ietf.org
- [9]. www.google.co.id
- [10]. www.wikipedia.org

ST
Telkom
University