

# Perancangan Sistem Pemantau Ruangan Dengan Webcam Menggunakan Jaringan WLAN

*Design of Room Monitoring System Using Webcam and WLAN Network*

Hadi Bangun Santosa \* dan Heru Abrianto \*\*

\* Field Monitoring Engineer, PT. Telekomunikasi Indonesia, Jakarta

\*\* Dosen Program Studi Teknik Elektro, Fakultas Teknologi Industri, Institut Sains Dan Teknologi Nasional Jakarta

**Abstrak---***Makalah ini membahas mengenai pengukuran dan analisa performansi layanan dari implementasi sistem pemantau ruangan menggunakan webcam berbasis web. Parameter-parameter yang akan diukur dan dianalisa yaitu delay, jitter, packet loss, dan throughput. Sistem pemantau ruangan ini menggunakan interface website untuk mempermudah dalam mengakses dan memantau video yang tertangkap oleh webcam. Untuk menyediakan interface website diperlukan web server dan database server dengan menggunakan perangkat lunak XAMPP yang sudah mendukung Apache sebagai web server dan MySQL sebagai database server. Perangkat yang digunakan untuk menangkap gambar yaitu menggunakan webcam. Sedangkan perangkat lunak yang digunakan untuk menyiarkan video hasil tangkapan webcam ke jaringan yaitu VLC. Untuk mengambil data pengukuran menggunakan perangkat lunak Wireshark. Pada tahap pengukuran, pengambilan data dilakukan dengan simulasi pada jaringan local area network. Data yang diambil dan diukur merupakan data trafik streaming video antara video server dengan client. Tiap-tiap pengukuran dilakukan pengambilan data pada lima kali percobaan. Dalam setiap percobaan, pengambilan data pengukuran menggunakan perangkat lunak Wireshark dalam durasi pengamatan selama 5, 10, dan 15 menit. Dari data hasil pengukuran tersebut, selanjutnya data pengukuran diolah untuk mendapatkan nilai-nilai delay, jitter, packet loss, dan throughput.*

**Kata kunci---***RTSP, Real Time Video Stream, Webcam*

**Abstract---***This paper discusses about the measurement and analysis of service performance implementation of web-based room monitoring system using a webcam. The parameters to be measured and analyzed are delay, jitter, packet loss, and throughput. This room monitoring system use the website as an interface to make it easier to access and monitor the video captured by the webcam. To provide the interface website, it need web server and database server using XAMPP software that supports Apache as web server and the MySQL as database server. The device that use to capture images is the webcam. While the software used for broadcasting the captured video of the webcam to the network is VLC. To retrieve measurement data using Wireshark software. In the measurement phase, the data collection is done by simulating on a local area network. The data which is taken and measured is the video streaming traffic data between video server and the client. Each measurements were performed on five time experiments. In each experiment, the collecting of measurement data is using Wireshark software with 5, 10, and 15 minutes observation duration. From the measurement data, then the data proceed to obtain the values of delay, jitter, packet loss, and throughput.*

**Keywords---***RTSP, Real Time Video Stream, Webcam*

## 1. PENDAHULUAN

Industri informasi dan telekomunikasi selalu mengalami perkembangan seiring dengan meningkatnya kebutuhan dari masyarakat dan kemajuan teknologi yang juga semakin berkembang pesat. Salah satu kebutuhan masyarakat saat ini yaitu layanan untuk memantau suatu ruangan atau suatu lokasi.

Layanan pemantau ruangan digunakan untuk memantau aktifitas yang terjadi dari ruangan yang dipantau. Dengan sistem pemantau ruangan masyarakat dapat menggunakannya sebagai kamera pengawas kamar bayi, kamera pemantau warnet, kamera pemantau parkir, dan lain-lain.

Pada makalah ini, sistem pemantau ruangan ini diimplementasikan pada ruang kerja Divisi Infratel Assurance Center Datin Care Center.

Pemasangan sistem pemantau ruangan ini bertujuan untuk memantau dan mengawasi aktifitas pegawai *outsourse*, serta menyimpan video hasil *webcam* sebagai data video digital.

Dengan sistem pemantau ruangan ini, staf organik dapat memantau ruangan kerja Care Center dari ruang kerja mereka masing-masing. Pemantauan ruangan juga dapat diakses melalui perangkat-perangkat seperti komputer, laptop, *smartphone*, dan PC tablet karena menggunakan *website* sebagai *interface* yang memiliki dapat bekerja diberbagai perangkat dengan kemampuan yang hampir sama ditiap perangkat. Sitem menggunakan XAMPP 1.7.7 sebagai perangkat lunak yang mendukung *webserver* Apache 2.2.21 dan *database* MySQL 5.5.16, serta VLC 2.1.2 untuk *streaming* video *webcam*. Kinerja yang diukur berupa *delay*, *jitter*, *packetloss*, dan *throughput*, diambil dari trafik *streaming* video hasil

simulasi antara video *server* dengan *client* pada jaringan lokal.

## 2. TINJAUAN PUSTAKA

### 2.1 Halaman Web

Halaman *web* adalah dokumen yang membentuk *World Wide Web*. Dokumen-dokumen ini ditulis dalam Hypertext Markup Language (HTML) dan diterjemahkan oleh *webbrowser*. Halaman *web* dapat berupa halaman statis maupun dinamis. Halaman statis menampilkan konten yang sama setiap kali halaman tersebut diakses. Halaman dinamis memiliki konten yang bisa berubah setiap kali diakses. Halaman *web* dapat ditulis dengan bahasa pemrograman yang bersifat *server-side* ataupun *client-side*. Bahasa pemrograman *server-sides* seperti PHP, Perl, ASP, atau JSP diolah disisi *server* yang kemudian hasil olahan dari server berupa halaman HTML. Sedangkan bahasa pemrograman *client-side* seperti HTML, CSS, dan Javascript diolah disisi *client*, yaitu pada *webbrowser*. Halaman *web* diakses dengan menggunakan protokol *Hypertext Transfer Protocol* (HTTP) atau *Hypertext Transfer Protocol Secure* (HTTPS).

### 2.2 Local Area Network (LAN)

*Local Area Network* adalah jaringan komputer yang area jaringannya hanya mencakup wilayah kecil seperti jaringan komputer kampus, gedung, kantor, dalam rumah, sekolah atau yang lebih kecil. Saat ini, kebanyakan LAN berbasis pada teknologi IEEE 802.3 *Ethernet* menggunakan perangkat *switch*, yang mempunyai kecepatan transfer data 10, 100, atau 1000 Mbit/s. Selain teknologi *Ethernet*, saat ini teknologi 802.11b atau biasa disebut Wi-Fi (*Wireless Fidelity*) juga sering digunakan untuk membentuk jaringan LAN.

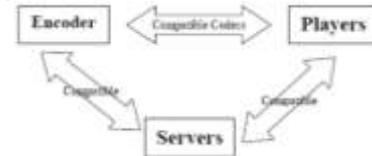
Pada sebuah LAN, setiap komputer juga dapat berkomunikasi mengakses sumber daya yang ada di LAN sesuai dengan hak akses yang telah diatur. Sumber daya tersebut dapat berupa data atau perangkat seperti *printer*. Pada jaringan LAN, seorang pengguna juga dapat berkomunikasi dengan pengguna yang lain dengan menggunakan aplikasi yang sesuai.

### 2.3 Webcam

*Webcam* atau *webcamera* adalah sebuah kamera video digital kecil yang dihubungkan ke komputer, yang biasanya dihubungkan melalui port USB atau port COM. Beberapa contoh pemakaian *webcam* yaitu untuk video *chat*, video *call*, video *conference*, dan *surveillancecam*. Bagian utama dari *webcam* yaitu lensa dan sensor gambar. Ada dua macam sensor gambar yang bisa digunakan pada *webcam* yaitu *Charge-Coupled Devices* (CCD) dan *Complementary Metal-Oxide-Semiconductor* (CMOS), namun saat ini sensor gambar yang umum digunakan adalah sensor CMOS.

### 2.4 Konsep Streaming

Sistem *streaming* tersusun dari kombinasi *server*, *player*, transmisi dan metode *encoding* yang digunakan. Gambar di bawah merupakan bagan hubungan setiap komponen penyusun sistem *streaming*.



Gambar 1. Komponen sistem penyusun *streaming*

### 2.5 Protokol Streaming

Pada sistem pemantau ruangan ini, untuk melakukan *streaming* video diperlukan beberapa protokol. Protokol yang digunakan yaitu RTSP, RTP, dan RTCP.

### 2.6 H.264 Video Codec

*Video codec* adalah suatu alat atau perangkat lunak yang berfungsi untuk kompresi dan dekompresi video digital. H.264/MPEG-4 Part 10 atau AVC (*Advanced Video Coding*) adalah standar kompresi video yang saat ini umum digunakan untuk rekaman, kompresi, dan distribusi *High Definition video*.

### 2.7 Delay

*Delay* adalah waktu tunda yang disebabkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. Dalam makalah ini, *delay* paket didapat dari data trafik paket RTP yang ditangkap oleh *wireshark*, yang selanjutnya dihitung dengan rumus (1) di bawah ini.

$$\text{Delayrata - rata} = \frac{\sum_{i=1}^n (t_{i+1} - t_i)}{n} \quad (1)$$

Dimana  $t$  merupakan waktu pengiriman paket dan  $n$  adalah jumlah paket.

### 2.8 Jitter

*Jitter* merupakan variasi dari *delay*. *Jitter* dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan. Pengaruh *jitter* pada performansi layanan harus dilihat bersama *delay*. Ketika *jitter* besar namun memiliki *delay* kecil maka performansi layanan tidak bisa dikatakan buruk karena besarnya *jitter* dapat dikompensasi dengan nilai *delay* yang kecil. Performansi layanan menurun jika nilai *jitter* dan *delay* besar. Berikut ini rumus (2) untuk menghitung *jitter*.

$$\text{Jitter} = \frac{\sum_{i=1}^n (D_{i+1} - D_i)}{n-1} \quad (2)$$

Dimana  $D$  adalah *delay* dan  $n$  adalah jumlah paket.

### 2.9 Packet Loss

*Packet loss* didefinisikan sebagai kegagalan transmisi paket mencapai tujuannya. Untuk

mendapatkan nilai *packet loss* dapat menggunakan rumus (3) seperti di bawah ini.

$$Packet Loss = \frac{Jumlah\ paket\ dikirim - Jumlah\ paket\ diterima}{Jumlah\ paket\ dikirim} \times 100\% \quad (3)$$

**2.10 Throughput**

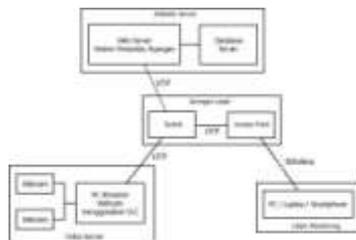
*Throughput* adalah kecepatan rata-rata data yang diterima oleh suatu *node* dalam selang waktu pengamatan tertentu. *Throughput* merupakan *bandwidth* aktual saat sedang dilakukan koneksi. Satuan yang dimilikinya sama dengan *bandwidth* yaitu bps (*bit per second*). Berikut ini rumus (4) yang dapat digunakan untuk mendapatkan nilai *throughput*.

$$Throughput = \frac{Total\ bit}{Waktu\ antara\ paket\ pertama\ dan\ terakhir} \quad (4)$$

**3. METODA**

**3.1 Blok Diagram Sistem**

Pada makalah ini, sistem pemantau ruangan ini terdiri dari tiga bagian blok yang saling berhubungan dan memiliki fungsinya masing-masing.



**Gambar 2.** Diagram Blok Sistem

**3.1.1 Blok Website Server**

Dalam sistem pemantau ruangan ini digunakan halaman *web* sebagai *interface* untuk mengakses video dari tiap *webcam*. Oleh karena itu dibutuhkan *web site server* yang berfungsi untuk mendukung layanan *web* tersebut. Blok *web site server* terdiri dari *web server* dan *data base server*.

**3.1.2 Blok Video Server**

Untuk dapat mengolah dan menyiarkan video dari *webcam* ke jaringan diperlukan *videosever*. Blok *video server* ini berada pada ruangan yang akan dipantau oleh *webcam*. Blok ini terdiri dari sebuah PC yang terhubung ke jaringan dan *webcam-webcam* yang dihubungkan kepada PC tersebut.

**3.1.3 Blok Client Monitoring**

Blok *client monitoring* ini merupakan blok dimana *user* mengakses video yang diputar secara *real time* sesuai dengan *webcam* yang diakses. *User* dapat mengakses video yang

tertangkap melalui *webcam* yang dimilikinya, melalui halaman *web* sebagai *interface* untuk mengakses video tersebut.

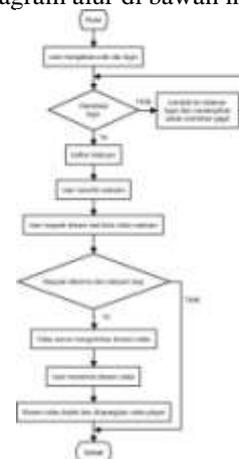
**3.2 Konfigurasi Sistem**

Pada sistem pemantau ruangan ini konfigurasi yang dibutuhkan yaitu pada sisi *videosever* yaitu pada perangkat lunak VLC. Konfigurasi ini digunakan untuk proses pengolahan video hingga proses menyiarkan video ke jaringan. Konfigurasi-konfigurasi tersebut yaitu:

1. *StreamingMethod*
2. *Transcoding*

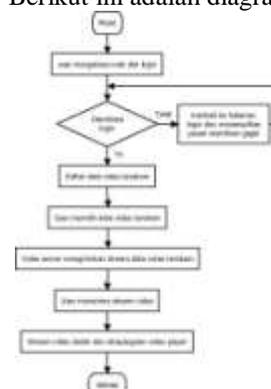
**3.3 Alur Penggunaan Sistem**

Dalam sistem pemantau ruangan ini video *streaming* dari *webcam* dapat diakses dan dimainkan secara *realtime* melalui halaman *web* yang ada pada sistem ini. Alur proses *user* mengakses *real time* video *webcam* dalam menggunakan sistem ini dapat dilihat pada diagram alur di bawah ini.



**Gambar 3.** Diagram alur *user* mengakses *real time* video *webcam*

Selain dapat diakses secara *real time*, dapat juga memainkan video yang sudah tersimpan dalam *video server*. Berikut ini adalah diagram alurnya.



**Gambar 4.** Diagram alur *user* mengakses video terekam

**3.4 Proses Kerja Stream Video**

Dalam sistem pemantau ruangan ini, video yang telah ditangkap oleh *webcam* yang kemudian dikirimkan melalui jaringan hingga video tersebut

dapat ditayangkan di sisi *client*, merupakan hasil dari beberapa serangkaian proses. Secara umum proses-proses tersebut dapat digambarkan melalui diagram alur di bawah ini.



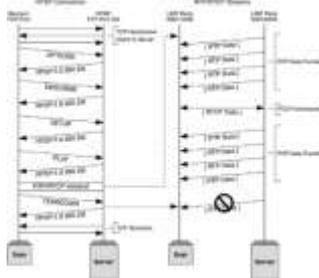
Gambar 5. Diagram alur proses kerjastreamvideo

### 3.5 Proses Transmisi Data Video

Metode transmisi data video ke jaringan pada sistem pemantau ruangan ini menggunakan metode transmisi data *unicast*. Transmisi data *unicast* merupakan metode transmisi data hubungan *one-to-one*, dimana dalam sistem ini merupakan komunikasi data hubungan antara *videosever* dengan *client*. Transmisi data yang digunakan menggunakan metode *unicast* karena dalam sistem pemantau ruangan ini menggunakan protokol RTSP untuk mengontrol *videostream* dengan memberi perintah-perintah ke pada *videosever* seperti perintah *pause* dan *stop*.

#### 3.5.1 Interaksi Protokol Pada Proses Transmisi Video

Pada sistem pemantau ruangan ini RTSP digunakan untuk membangun, mengontrol, dan mengakhiri sesi antara *video server* dan *client* yang mengakses *webcam* pada *videosever* tersebut secara *real time*. Protokol RTP dengan menggunakan protokol *transport UDP*. RTP didukung dengan protokol RTCP untuk mengontrol kualitas *stream video*. Seperti prokol RTP, protokol RTCP menggunakan protokol *transport UDP*. Dalam sistem pemantau ruangan ini, paket RTSP, RTP, dan RTCP dikirim menggunakan metode transmisi data *unicast*. Di bawah ini merupakan gambar interaksi protokol yang terjadi antara *videoplayer* dengan *videosever*.

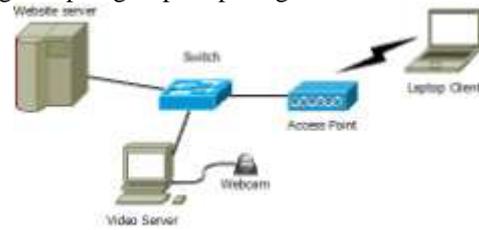


Gambar 6. Interaksi protokol antara *videoplayer* dan *videosever*

## 4. HASIL DAN PEMBAHASAN

### 4.1 Skenario Pengukuran

Pengukuran dilakukan pada jaringan lokal dengan topologi seperti pada gambar di bawah ini.



Gambar 7. Topologi jaringan untuk pengukuran

### 4.2 Hasil Pengukuran

Setelah dilakukan pengambilan data pengukuran dari lima kali percobaan dengan variasi durasi pemantauan tiap percobaan sebesar 5, 10, dan 15 menit, maka didapatkan data-data yang menjadi parameter kualitas layanan.

#### 4.2.1 Delay

*Delay* yang diukur merupakan *delay stream video* yang dibawa dalam *payload* paket RTP. Data pengukuran *delay* didapat dari data paket RTP yang ditangkap menggunakan perangkat lunak *wireshark*. Pengukuran dilakukan disisi *client* sebanyak lima kali percobaan dengan durasi pengamatan tiap pengukuran yaitu selama 5, 10, dan 15 menit.

Tabel 1. Hasil pengambilan data paket RTP

No.	Paket No.	Waktu (s)	IP Sumber	IP Tujuan	Protokol
1	24	2.064317	10.11.29.35	10.11.29.83	RTP
2	25	2.064454	10.11.29.35	10.11.29.83	RTP
3	27	2.233858	10.11.29.35	10.11.29.83	RTP
4	28	2.508494	10.11.29.35	10.11.29.83	RTP
5	29	2.508735	10.11.29.35	10.11.29.83	RTP
6	30	2.769045	10.11.29.35	10.11.29.83	RTP
7	31	2.782608	10.11.29.35	10.11.29.83	RTP
8	32	2.782959	10.11.29.35	10.11.29.83	RTP
9	33	2.943686	10.11.29.35	10.11.29.83	RTP
10	34	3.185895	10.11.29.35	10.11.29.83	RTP

Dari data tabel 1. di atas, dapat dihitung nilai rata-rata *delay* dari tiap paket berikut ini.

$$\begin{aligned}
 & (2.064454 - 2.064317) \\
 & + (2.233858 - 2.064454) + \dots \\
 \text{Delayrata - rata} &= \frac{(2.064454 - 2.064317) + (2.233858 - 2.064454) + \dots + (3.185895 - 2.943686)}{10} = 0.112s
 \end{aligned}$$

#### 4.2.1.1 Hasil Pengukuran Delay Durasi 5 Menit

Di bawah ini Tabel 2. hasil pengukuran *delay stream video* dari 5 kali percobaan dengan durasi 5 menit.

Tabel 2 Hasil pengukuran *delay* durasi 5 menit

Percobaan Ke-	Rata-rata Delay (s)
1	0.107
2	0.107
3	0.107
4	0.099

5	0.110
Rata-rata	0.106

Berdasarkan tabel dan grafik pengukuran *delay* durasi 5 menit, nilai *delay* terendah sebesar 0.099s pada percobaan ke-4, tertinggi sebesar 0.110s pada percobaan ke-5, dan rata-rata *delay* dari kelima percobaan sebesar 0.106s. Dari hasil percobaan ini nilai *delay* masih sesuai standar rekomendasi ITU-T G.1010 untuk layanan video satu arah (*one-way*) yaitu kurang dari 10s.

#### 4.2.1.2 Hasil Pengukuran *Delay* Durasi 10 Menit

Di bawah ini hasil pengukuran *delay stream* video dari 5 kali percobaan dengan durasi 10 menit.

**Tabel 3.** Hasil pengukurandelaydurasi 10 menit

Percobaan Ke-	Rata-rata Delay (s)
1	0.110
2	0.118
3	0.106
4	0.106
5	0.106
Rata-rata	0.109

Berdasarkan tabel dan grafik pengukuran *delay* durasi 10 menit, nilai *delay* terendah sebesar 0.106s pada percobaan ke-3, 4, dan 5, tertinggi sebesar 0.118s pada percobaan ke-2, dan rata-rata *delay* dari kelima percobaan sebesar 0.109s. Dari hasil percobaan ini nilai *delay* masih sesuai standar rekomendasi ITU-T G.1010 untuk layanan video satu arah (*one-way*) yaitu kurang dari 10s.

#### 4.2.1.3 Hasil Pengukuran *Delay* Durasi 15 Menit

Di bawah ini hasil pengukuran *delay stream* video dari 15 kali percobaan dengan durasi 5 menit.

**Tabel 4.** Hasil pengukurandelaydurasi 15 menit

Percobaan Ke-	Rata-rata Delay (s)
1	0.107
2	0.108
3	0.109
4	0.106
5	0.107
Rata-rata	0.107

Berdasarkan tabel dan grafik pengukuran *delay* durasi 5 menit, nilai *delay* terendah sebesar 0.106s pada percobaan ke-4, tertinggi sebesar 0.109s pada percobaan ke-3, dan rata-rata *delay* dari kelima percobaan sebesar 0.107s. Dari hasil percobaan ini nilai *delay* masih sesuai standar rekomendasi ITU-T G.1010 untuk layanan video satu arah (*one-way*) yaitu kurang dari 10s.

#### 4.2.2 Jitter

*Jitter* merupakan variasi dari *delay*. Hasil nilai *jitter* didapat melalui perhitungan dengan data dari hasil perhitungan *delay*. Seperti pada pengukuran *delay*, *jitter* yang diamati merupakan *jitter* dari transmisi *stream* video pada paket RTP. Untuk mendapatkan nilai *jitter* yaitu dengan melakukan perhitungan seperti di bawah ini.

Delay 1 = 0 ms  
 Delay 2 = 0.137 ms  
 Delay 3 = 169.404 ms  
 ...  
 Delay 9 = 160.727 ms  
 Delay 10 = 242.209 ms

$$\text{Maka,} \\ \text{Jitter} = \frac{(|0.137 - 0|) + (|169.40 - 0.137|) + \dots + (|242.209 - 160.727|)}{9} \\ = 145.655\text{ms} = 0.146\text{s}$$

#### 4.2.2.1 Hasil Pengukuran *Jitter* Durasi 5 Menit

Berikut ini hasil pengukuran *jitter stream* video dari 5 kali percobaan dengan durasi 5 menit.

**Tabel 5.** Hasil pengukuranjitterdurasi 5 menit

Percobaan Ke-	Jiter (s)
1	0.064
2	0.065
3	0.065
4	0.067
5	0.061
Rata-rata	0.064

Dapat dilihat dari tabel dan grafik pengukuran *jitter* durasi 5 menit, bahwa nilai *jitter* terendah sebesar 0.061s pada percobaan ke-5, tertinggi sebesar 0.067s pada percobaan ke-4, dan rata-rata *jitter* dari kelima percobaan sebesar 0.064s.

#### 4.2.2.2 Hasil Pengukuran *Jitter* Durasi 10 Menit

Berikut ini hasil pengukuran *jitter stream* video dari 5 kali percobaan dengan durasi 10 menit.

**Tabel 6.** Hasil pengukuranjitterdurasi 10 menit

Percobaan Ke-	Jiter (s)
1	0.063
2	0.070
3	0.068
4	0.067
5	0.070
Rata-rata	0.068

Dapat dilihat dari tabel dan grafik pengukuran *jitter* durasi 10 menit, bahwa nilai *jitter* terendah sebesar 0.063s pada percobaan pertama, tertinggi sebesar 0.070s pada percobaan ke-2 dan ke-5, dan rata-rata *jitter* dari kelima percobaan sebesar 0.068s.

#### 4.2.2.3 Hasil Pengukuran *Jitter* Durasi 15 Menit

Berikut ini hasil pengukuran *jitter stream* video dari 5 kali percobaan dengan durasi 15 menit.

**Tabel 7.** Hasil pengukuranjitter durasi 15 menit

Percobaan Ke-	Jiter (s)
1	0.066
2	0.070
3	0.067
4	0.071
5	0.070
Rata-rata	0.069

Dapat dilihat dari tabel dan grafik pengukuran *jitter* durasi 15 menit, bahwa nilai *jitter* terendah sebesar 0.066s pada percobaan pertama, tertinggi sebesar 0.071s pada percobaan ke-4, dan rata-rata *jitter* dari kelima percobaan sebesar 0.069s.

**4.2.3 PacketLoss**

Pengukuran *packet loss* ini memiliki tujuan untuk melihat besar persentase jumlah paket yang hilang yang diterima disisi *client*. Paket yang diukur adalah paket RTP yang merupakan paket yang membawa *payload stream* video. Di bawah ini merupakan penghitungan untuk mendapatkan nilai *packet loss*.

Jumlah paket dikirim = 2822

Jumlah paket diterima = 2822

Maka,

$$Packet\ Loss = \frac{2822 - 2822}{2822} \times 100\% = 0\%$$

**4.2.3.1 Hasil Pengukuran Packet Loss Durasi 5 Menit**

Berikut ini hasil pengukuran *packet lost stream* video dari 5 kali percobaan dengan durasi 5 menit.

**Tabel 8.** Hasil pengukuran *packet loss* durasi 5 menit

Percobaan Ke-	Jumlah Paket Dikirim	Jumlah Paket Diterima	Packet Loss (%)
1	2822	2822	0.000
2	2837	2837	0.000
3	2838	2838	0.000
4	3047	3047	0.000
5	2769	2769	0.000
Rata-rataPacketLoss			0.000

Dari tabel dan grafik pengukuran *packet loss* durasi 5 menit, dapat dilihat bahwa pada kelima percobaan didapat nilai persentasi *packet loss* sebesar 0%. Sehingga dari hasil percobaan ini nilai *packet loss* masih sesuai standar rekomendasi ITU-T G.1010 yaitu kurang dari 1%.

**4.2.3.2 Hasil Pengukuran Packet Loss Durasi 10 Menit**

Berikut ini hasil pengukuran *packet lost stream* video dari 5 kali percobaan dengan durasi 10 menit.

**Tabel 9.** Hasil pengukuran *packet loss* durasi 10 menit

Percobaan Ke-	Jumlah Paket Dikirim	Jumlah Paket Diterima	Packet Loss (%)
1	5639	5639	0.000
2	5118	5118	0.000
3	6135	6126	0.147
4	5686	5682	0.070
5	5666	5662	0.071
Rata-rataPacketLoss			0.058

Dari tabel dan grafik pengukuran *packet loss* durasi 10 menit, dapat dilihat bahwa nilai *packet loss* tertinggi terdapat pada percobaan ke-3 yaitu

sebesar 0.147% dan nilai rata-rata *packet loss* dari kelima percobaan ini sebesar 0.058%. Sehingga dari hasil percobaan ini nilai *packet loss* masih sesuai standar rekomendasi ITU-T G.1010 yaitu kurang dari 1%.

**4.2.3.3 Hasil Pengukuran Packet Loss Durasi 15 Menit**

Berikut ini hasil pengukuran *packet lost stream* video dari 5 kali percobaan dengan durasi 15 menit.

**Tabel 10.** Hasil pengukuran *packet loss* durasi 15 menit

Percobaan Ke-	Jumlah Paket Dikirim	Jumlah Paket Diterima	Packet Loss (%)
1	8482	8479	0.035
2	8428	8348	0.949
3	8353	8331	0.263
4	8597	8597	0.000
5	8491	8471	0.236
Rata-rataPacketLoss			0.297

Dari tabel dan grafik pengukuran *packet loss* durasi 15 menit, dapat dilihat bahwa nilai *packet loss* tertinggi terdapat pada percobaan ke-2 yaitu sebesar 0.949% dan nilai rata-rata *packet loss* dari kelima percobaan ini sebesar 0.297%. Sehingga dari hasil percobaan ini nilai *packet loss* masih sesuai standar rekomendasi ITU-T G.1010 yaitu kurang dari 1%.

**4.2.4 Throughput**

Pada pengukuran *throughput* disini yaitu untuk melihat *bandwidth* aktual pada sisi *client*, saat sedang melakukan *stream* paket RTP dari video *server*. Di bawah ini penghitungan mendapatkan nilai *throughput*.

Total bit paket data = 5069648bit

Waktu antara paket pertama dan terakhir = 58.337s

Maka,

$$Throughput = \frac{5069648}{58.337} = 86902.789\text{bps} = 86.903\text{kbps}$$

**4.2.4.1 Hasil Pengukuran Throughput Durasi 5 Menit**

Hasil pengukuran *throughput* trafik video *streaming* untuk durasi 5 menit dengan perangkat lunak *wireshark*, didapatkan seperti tabel di bawah ini.

**Tabel 11.** Hasil pengukuran *throughput* durasi 5 menit

Percobaan Ke-	Jumlah Paket (bit)	Selisih Waktu Paket (s)	Throughput (bps)	Throughput (kbps)
1	12512568	301.926	41442.488	41.442
2	12591200	304.178	41394.217	41.394
3	12650392	305.020	41474.043	41.474
4	14523624	302.618	47993.229	47.993
5	10204248	304.043	33561.864	33.562
Rata-rataThroughput			41173.168	41.173

Dapat dilihat dari tabel dan grafik pengukuran *throughput* durasi 5 menit, bahwa nilai *throughput* terendah sebesar 33.562kbps pada percobaan ke-5, tertinggi sebesar 47.993kbps pada percobaan ke-4, dan rata-rata *throughput* dari kelima percobaan sebesar 41.173kbps.

#### 4.2.4.2 Hasil Pengukuran *Throughput* Durasi 10 Menit

Hasil pengukuran *throughput* trafik video *streaming* untuk durasi 10 menit dengan perangkat lunak *wireshark*, didapatkan seperti tabel di bawah ini.

**Tabel 12.** Hasil pengukuran *throughput* durasi 10 menit

Percobaan Ke-	Jumlah Paket (bit)	Selisih Waktu Paket (s)	<i>Throughput</i> (bps)	<i>Throughput</i> (kbps)
1	25685048	619.586	41455.202	41.455
2	22126008	602.063	36750.290	36.750
3	26916016	652.102	41275.804	41.276
4	25133648	601.337	41796.251	41.796
5	24868352	601.468	41346.091	41.346
Rata-rata <i>Throughput</i>			40524.728	40.525

Dapat dilihat dari tabel dan grafik pengukuran *throughput* durasi 10 menit, bahwa nilai *throughput* terendah sebesar 36.750kbps pada percobaan ke-2, tertinggi sebesar 41.796kbps pada percobaan ke-4, dan rata-rata *throughput* dari kelima percobaan sebesar 40.525kbps.

#### 4.2.4.3 Hasil Pengukuran *Throughput* Durasi 15 Menit

Hasil pengukuran *throughput* trafik video *streaming* untuk durasi 15 menit dengan perangkat lunak *wireshark*, didapitkans eperti tabel 13.

**Tabel 13.** Hasil pengukuran *throughput* durasi 15 menit

Percobaan Ke-	Jumlah Paket (bit)	Selisih Waktu Paket (s)	<i>Throughput</i> (bps)	<i>Throughput</i> (kbps)
1	38664640	910.992	42442.358	42.442
2	37136872	904.086	41076.721	41.077
3	37668304	906.375	41559.292	41.559
4	37683056	913.527	41250.056	41.250
5	37180448	903.004	41174.186	41.174
Rata-rata <i>Throughput</i>			41500.522	41.501

Dapat dilihat dari tabel dan grafik pengukuran *throughput* durasi 15 menit, bahwa nilai *throughput* terendah sebesar 41.077kbps pada percobaan ke-2, tertinggi sebesar 42.442kbps pada percobaan pertama, dan rata-rata *throughput* dari kelima percobaan sebesar 41.501kbps.

## 5. SIMPULAN

Berdasarkan standar rekomendasi ITU-T G.1010 untuk layanan video satu arah (*one-way*), standar rekomendasinya yaitu nilai *delay* kurang dari 10s dan nilai *packetloss* kurang dari 1%. Sehingga dari hasil pengukurandelaydan*packetloss* yang telah dilakukan

dapat disimpulkan bahwa *videostream* pada sistem pemantau ruangan ini layak dan sesuai dengan standar rekomendasi ITU-T G.1010.

Berdasarkan hasil pengukuran, diperoleh bahwa rata-rata *delay* dari lima kali percobaan dengan variasi durasi pemantauan 5, 10, dan 15 adalah 0.106s, 0.109s, dan 0.107s. Untuk nilai *jitter* dengan variasi durasi pemantauan 5, 10, dan 15 adalah 0.064s, 0.068s, dan 0.069s. Besar *packet loss* dengan variasi durasi pemantauan 5, 10, dan 15 adalah 0%, 0.058%, dan 0.297%. *Throughput* dengan variasi durasi pemantauan 5, 10, dan 15 adalah 41.173kbps, 40.525kbps, dan 41.501kbps. Variasi nilai *delay*, *jitter*, *packetloss*, dan *throughput* dapat disebabkan oleh tingkat kepadatan trafik serta beban yang terjadi dalam jaringan.

## DAFTAR PUSTAKA

- Damjanovski, Vlado 2005. *CCTV: Networking and Digital Technology*, Heinemann, Butterworth.
- Detlev M, Thomas W, Gary J. Sullivan. 2006. *The H.264/MPEG4 Advanced Video Coding Standard and its Applications*, Heinrich Hertz Institute (HHI).
- F.Kurose, James dan W. Ross, Keith 1999-2000. *Computer Networking*, Addison-Wesley.
- Farid, Azis M., 2001. *Pemograman PHP 4 bagi Web Programmer*. Elex Media Komputindo, Jakarta.
- Hananto E W, Indra S W 2001. *Kompresi Video Menggunakan Discrete Cosine Transform*, Laboratorium Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika, Institut Teknologi Bandung.
- Richardson Ian E.G, 2003. *H.264 and MPEG-4 Video Compression*, Wiley, Aberdeen.
- Sutarman 2003. *Membangun Aplikasi Web dengan PHP dan MySQL*, Graha Ilmu, Jakarta.