

ANALISIS KINERJA TRAFIK *WEB BROWSER* DENGAN *WIRESHARK NETWORK PROTOCOL ANALYZER* PADA SISTEM *CLIENT-SERVER*

Roland Oktavianus Lukas Sihombing, Muhammad Zulfin

Konsentrasi Teknik Telekomunikasi, Departemen Teknik Elektro

Fakultas Teknik Universitas Sumatera Utara (USU)

Jl. Almamater, Kampus USU Medan 20155 INDONESIA

e-mail: rolandsihombing@students.usu.ac.id or rolandsihombing.lukas@yahoo.com

Abstrak

Teknologi pada zaman modern seperti sekarang telah banyak menghadirkan banyak fitur mulai dari *internet*, *game online*, media *social network*, dan lain-lain. Salah satunya adalah *Web Browser* yang berarti suatu program yang digunakan untuk menjelajahi dunia *internet* atau untuk mencari informasi tentang suatu halaman *web* yang tersimpan di komputer. Pada paper ini membahas tentang analisis kinerja trafik *web browser* melalui *software wireshark*, dengan tujuan untuk mengetahui kinerja trafik di dalam jaringan internet melalui *web browser*. *Web browser* atau disebut penjelajah *web*, adalah perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh *server web*. Adapun parameter dalam menganalisis kinerja trafik *web browser* ini adalah *delay*, *packet loss*, dan *throughput*. Seiring dengan perkembangan teknologi telah dikembangkan alat untuk telekomunikasi secara digital, diantaranya adalah memonitoring trafik *network* melalui PC atau laptop melalui *software* pendukung yaitu *Wireshark Network Protocol Analyzer* yang bekerja melalui media *interface* melalui PC. *Network Packet Analyzer* berguna untuk menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi di paket tersebut secara *detail*. Dari hasil pengujian, untuk *delay* terbesar adalah waktu mengakses *web youtube.com*, dan *delay* terkecil pada *web cisco.com*, *packet loss* 0%, dan *throughput* terbesar saat mengakses *web google.com*, dan *throughput* terkecil saat mengakses *web youtube.com*

Kata Kunci: *wireshark*, *web browser*, *website*

1. Pendahuluan

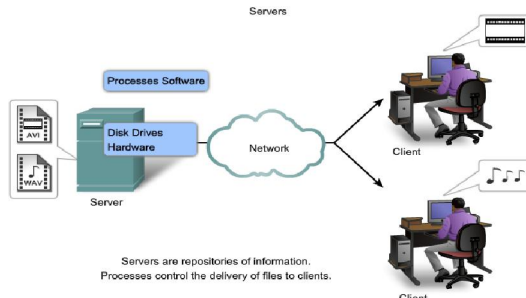
Perkembangan kemajuan teknologi pada dunia telekomunikasi sekarang ini semakin pesat, karena telah menghadirkan beberapa teknologi diantaranya adalah monitoring trafik *network* yang ada pada perusahaan telekomunikasi untuk mengetahui jaringan meraka apakah berjalan normal atau sedang mengalami masalah pada *network* di perusahaan telekomunikasi tersebut. *Website* adalah kumpulan dari sebuah *file* yang terletak pada sebuah komputer yang terhubung ke internet. Dimana ketika sebuah komputer telah terhubung ke internet, maka akan ditampilkan sebuah informasi yang akan dicari oleh *user/pengguna* itu sendiri. Oleh karena itu, kebutuhan *user/pengguna* di dalam jaringan internet mengakibatkan trafik di jaringan pun tinggi, sehingga penulis ingin memonitoring trafik di

jaringan dengan menggunakan *software wireshark network protocol analyzer*. *Software wireshark* dapat menangkap semua trafik selama menggunakan jaringan internet, baik *ip address*, *protocol*, lalu informasi di dalam paket data itu sendiri. Tujuan bagi seorang *user/pengguna* dalam menggunakan *wireshark* adalah agar memudahkan untuk melihat dan menganalisa paket data dalam lalu lintas jaringan internet pada *web browser*.

2. Model Client / Server

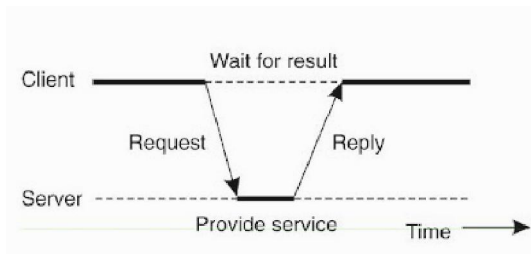
Dalam model *client / server*, perangkat yang meminta informasi disebut *client* dan perangkat menanggapi permintaan disebut *server*. *Client* dan *server* proses dianggap berada di lapisan aplikasi. *Client* memulai pertukaran dengan meminta data dari *server*, yang merespon dengan mengirimkan satu atau lebih aliran data

ke *client*. *Server* sebuah perangkat komputer yang biasanya mengandung informasi yang dapat dibagikan ke banyak sistem komputer atau *client*. Konsep jaringan *server* menuju *client* ditunjukkan pada Gambar 1 [1].



Gambar 1. Konsep jaringan *server* menuju *client*

Interaksi antara *client* dengan *server* adalah hal yang sangat berhubungan di dalam sistem jaringan internet. Salah satunya adalah proses *request* dan *reply*. *Request* berarti permintaan yang dikirim dari *client* menuju *server*. *Reply* berarti proses pengiriman atau *response* dari *server* kepada *client* yang menunjukkan bahwa *server* telah menerima permintaan *web page* dari *client*. Proses dari interaksi antara *client-server* ditunjukkan pada Gambar 2.



Gambar 2. Interaksi antara *client-server*

Total waktu tunggu yang dialami transmitter sebelum mentransmisikan *frame-I* berikutnya, meskipun *frame* sebelumnya telah diterima oleh *receiver* dengan benar dapat menggunakan Persamaan 1 [2] :

$$T_t = T_{ix} + T_{ip} + T_{ax} + T_{ap} + 2 T_p \quad (1)$$

Pada prakteknya, waktu untuk memproses sebuah *frame-I* (T_{ip}) dan *frame ACK*-nya (T_{ap}) sangat singkat dibandingkan dengan waktu pentransmisiannya. Disamping itu karena sebuah *frame ACK* sangat pendek dibandingkan dengan *frame-I*, maka T_{ax} dapat diabaikan bila dibandingkan dengan T_{ix} . Oleh karena itu, total waktu sebelum *frame* berikutnya ditransmisikan menggunakan Persamaan 2 [2] :

$$T_t = T_{ix} + 2 T_p \quad (2)$$

Dari persamaan 2, untuk mendapatkan waktu pentransmisi *frame*-nya (T_{ix}) dengan menggunakan Persamaan 3 [2] :

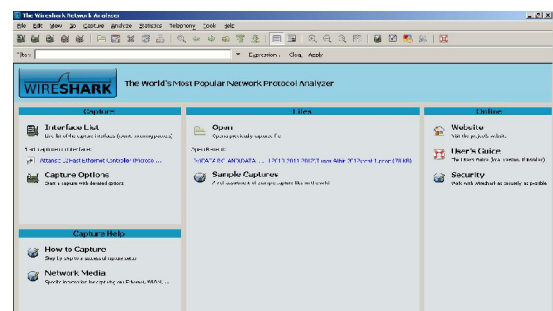
$$T_{ix} = N / R \quad (3)$$

Dimana :

N : jumlah bit di dalam *frame* = *Avg. Packet size (byte)*
 R : bit rate (bps)

3. Pengenalan Software Wireshark

Wireshark Network Protocol Analyzer adalah sebuah aplikasi perangkat lunak (*software*) yang digunakan untuk dapat melihat dan mencoba menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi di paket tersebut sedetail mungkin. *Open Source* dari *Wireshark* menggunakan *Graphical User Interface (GUI)* seperti ditunjukkan pada Gambar 3.



Gambar 3. Tampilan *Graphical User Interface*

3.1 Delay

Delay merupakan penundaan waktu suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik yang lain yang menjadi tujuannya. *Delay* merupakan penundaan waktu paket tiba ke dalam sistem komputer *client* atau *host* sampai selesai ditransmisikan. Untuk menghitung *delay* transmisi dapat dicari dengan menggunakan Persamaan 4 [3].

$$\text{Delay (sec) } T_x = \frac{\text{Time between first and last packet}}{\text{Jumlah paket}} \quad (4)$$

Dimana :

$\text{Delay (sec) } T_x$ = *delay* transmisi yang dikirim oleh *client* menuju *server*

Time between first and last packet = waktu paket yang dikirim oleh *server* menuju *client*

Jumlah paket = jumlah paket data yang di filter

Dari persamaan 3 diatas, untuk mencari *delay* paket yang diterima dari *server* menuju *client* dapat menggunakan Persamaan 5[4].

$$\text{Delay paket} = \frac{\text{Total delay}}{\text{Jumlah total paket}} \quad (5)$$

Dimana :

Total delay = jumlah keseluruhan dari *delay* 1, 2, 3.....n

Jumlah total paket = jumlah paket data yang di filter

3.2 Packet Loss

Packet Loss dapat didefinisikan sebagai kegagalan mentransmisikan paket pada alamat tujuannya sehingga menyebabkan beberapa paket dalam waktu pengiriman hilang atau *lost*. Untuk menghitung *Packet Loss* dapat menggunakan Persamaan 6 [5].

$$\text{Paket Loss} = \frac{\text{Packet_transmitted} - \text{Packet_received}}{\text{Packet_transmitted}} \times 100\% \quad (6)$$

Dimana :

Packet_transmitted = jumlah paket yang dikirim dari *client* menuju *web server*

Packet_recieved = jumlah paket yang diterima oleh *client* dari *web server*

3.3 Troughput

Troughput merupakan suatu kinerja jaringan yang terukur. *Troughput* juga diartikan sebagai kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data per satuan waktu. Untuk menghitung *Troughput* dapat menggunakan Persamaan 7 [5].

$$\text{Troughput} = \frac{\text{Jumlah data yang dikirim}}{\text{Waktu pengiriman data}} = \frac{\text{Average Bytes/sec}}{\text{Time between first \& last packet (sec)}} \quad (7)$$

Dimana :

Jumlah data yang dikirm oleh *server* menuju *client* = *Average Bytes/sec (bytes)*

Waktu pengiriman data dari *server* menuju *client* = *time between first \& last packet (sec)*

4. Analisis Perhitungan Delay

Delay merupakan penundaan waktu paket tiba ke dalam sistem komputer *client* atau *host* sampai selesai ditransmisikan. Salah satu jenis *delay* adalah *delay* transmisi (T_{ix}), yaitu waktu

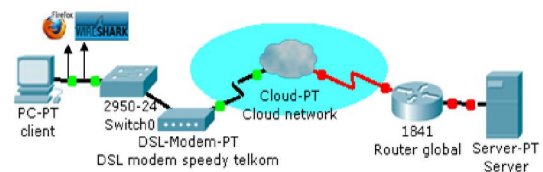
yang dibutuhkan pengirim mengirimkan sebuah paket atau *frame*. Untuk pengujian parameter yang dibahas adalah dengan menggunakan *web browser mozilla firefox 12.0*.

Media perangkat yang digunakan dalam menganalisis jaringan di internet menggunakan modem *router ADSL speedy* dari Telkom, seperti yang ditunjukkan pada Gambar 4.



Gambar 4. Modem Router ADSL Speedy

Selain perangkat modem *Router ADSL Speedy*, juga terdapat model pengujian untuk mendapatkan hasil data dari kinerja suatu parameter yang dibahas seperti yang ditunjukkan pada Gambar 5.



Gambar 5. Model Pengujian dalam Pengambilan Data

Berdasarkan model pengujian pada Gambar 5, untuk menghitung *delay* menurut teorinya adalah dengan pengujian 2 sampel *web* terhadap 5 *server* alamat *web* yang diuji, maka didapat :

- Untuk *server* ke-1 (www.kompas.com) banyak paket adalah 20 *frame*, dengan nilai *average packet size*-nya 107.750 *byte*, dimana bit rate adalah 1668.832 bit/detik. Maka berdasarkan Persamaan 3 nilai T_{ix} untuk 20 paket adalah :

$$\begin{aligned} T_{ix} &= \text{average packet size (byte)} / \text{bit rate (R)} \\ &\text{bps} \\ T_{ix} &= 107.750 \times 8 / 1668.832 \\ T_{ix} &= 0.5165 \text{ detik} \end{aligned}$$

- Untuk *server* ke-2 (www.google.com) banyak paket yang di filter adalah 21 frame, dengan nilai *average packet size* = 125.143 *byte*, bit rate = 1295.48 bit/detik. Maka berdasarkan Persamaan 3 nilai T_{IX} untuk 21 paket adalah :

$$T_{IX} = \text{average packet size (byte)} / \text{bit rate (R)} \text{ bps}$$

$$T_{IX} = 125.143 \times 8 \text{ bit} / 1295.48$$

$$T_{IX} = 0.7727 \text{ detik}$$

Untuk menghitung *delay* menurut *software wireshark*, pada *server* ke-1 (www.kompas.com) jumlah paket yang di filter = 20, waktu penerimaan paket (*time between first and last packet*) = 10.332 detik, rata-rata jumlah paket yang diterima (*Avg.packet/sec*) = $20 / 10.332 = 1.936$ paket/detik dan Rata-rata ukuran per paket = 107.750 *byte*. Berdasarkan jumlah paket dan ukuran per paket data terakhir diperoleh bit rate transfer data sebagai berikut : Bit Rate = $1.936 \times 107.750 \times 8 \text{ bit/detik} = 1668.832 \text{ bit/detik}$. Selanjutnya dengan menggunakan Persamaan 4, maka *delay* pengiriman per paket data adalah :

$$\text{Delay (sec)} = 10.332 / 20 = 0.5166 \text{ detik}$$

Untuk *server* ke-2 (www.google.com), bila jumlah paket yang di filter = 21, waktu penerimaan paket (*time between first and last packet*) = 16.225 detik, rata-rata jumlah paket yang diterima (*Avg.packet/sec*) = $21 / 16.225 = 1.294$ paket/detik dan Rata-rata ukuran per paket = 125.143 *byte*. Berdasarkan jumlah paket dan ukuran per paket data terakhir diperoleh bit rate transfer rate data sebagai berikut: Bit Rate = $1.294 \times 125.143 \times 8 \text{ bit/detik} = 1295.48 \text{ bit/detik}$. Selanjutnya dengan menggunakan Persamaan 4, maka *delay* pengiriman per paket data adalah :

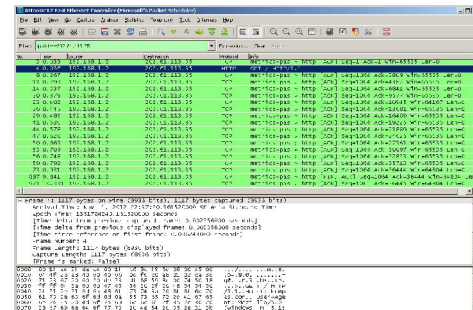
$$\text{Delay (sec)} = 16.225 / 21 = 0.7726 \text{ detik}$$

Berdasarkan perhitungan dari Persamaan 3 dan 4, maka hasil perhitungan keseluruhan sampai dengan *server* yang ke-5 dapat diimplementasikan ke dalam bentuk Tabel 1.

Tabel 1. Hasil Pengujian *Delay* menurut *software wireshark*

No	Waktu	Alamat Web	Jumlah paket	Bit rate (bps)	Avg. Packet size (byte)	Delay T_x (sec)	Delay T_s (detik)
1	22:37:20 s/d 22:37:44 Wtb	www.kompas.com	20	1668.832	107.750	0.5166	0.5165
2	23:00:27 s/d 23:00:43 Wtb	www.google.com	21	1295.48	125.143	0.7726	0.7727
3	23:07:55 s/d 23:08:42 Wtb	www.youtube.com	43	676.3629	92.907	1.0993	1.0989
4	23:17:23 s/d 23:17:46 Wtb	www.cisco.com	192	10291.74	157.193	0.1221	0.1221
5	23:25:46 s/d 23:26:15 Wtb	www.detik.com	209	3383.881	129.913	0.1107	0.1107
Total delay						2.6513	2.6509
delay rata-rata						0.53026	0.53018

Dari hasil pengujian pada *software wireshark*, maka didapat parameter *delay* dalam pengujian untuk setiap *server* seperti ditunjukkan pada Gambar 6.



Gambar 6. Captured data *delay* pada *software wireshark*

Dari hasil pengujian pada Gambar 6, maka diperoleh *delay* paket yang diterima dari *server* menuju *client* dengan menggunakan Persamaan 5.

- Untuk *delay* alamat web www.kompas.com

$$\text{Delay paket} = \frac{\text{Total delay}}{\text{Jumlah total paket}} = 2.6513 / 20 = 0.1325 \text{ sec}$$

- Untuk *delay* alamat web www.google.com

$$\text{Delay paket} = \frac{\text{Total delay}}{\text{Jumlah total paket}} = 2.6513 / 21 = 0.1262 \text{ sec}$$

- Untuk *delay* alamat web www.youtube.com

$$\text{Delay paket} = \frac{\text{Total delay}}{\text{Jumlah total paket}} = 2.6513 / 43 = 0.0616 \text{ sec}$$

- Untuk *delay* alamat web www.cisco.com

$$\text{Delay paket} = \frac{\text{Total delay}}{\text{Jumlah total paket}} = 2.6513 / 192 = 0.0138 \text{ sec}$$

- Untuk *delay* alamat web www.detik.com

$$\text{Delay paket} = \frac{\text{Total delay}}{\text{Jumlah total paket}} = 2.6513 / 209 = 0.0126 \text{ sec}$$

5. Analisis Perhitungan *Packet Loss*

Packet Loss dapat didefinisikan sebagai kegagalan mentransmisikan paket pada alamat

tujuannya sehingga menyebabkan beberapa paket dalam waktu pengiriman hilang atau *lost*. Hal ini disebabkan oleh sejumlah faktor, mencakup penurunan signal dalam media jaringan, melebihi batas saturasi jaringan, paket yang *corrupt* yang menolak untuk transit, kesalahan *hardware* jaringan. Beberapa *network transport* protokol seperti TCP menyediakan pengiriman paket yang dapat dipercaya. Dalam hal kerugian paket, penerima akan meminta *retarnsmission* atau pengiriman secara otomatis *resends* walaupun *segment* telah tidak diakui. Oleh karena itu, untuk mendapatkan *packet loss* dilakukan pengujian terhadap alamat *website* dengan menggunakan *software wireshark* dan untuk menghitung *packet loss* terhadap alamat *website* dapat menggunakan Persamaan 6.

- a. Untuk *packet loss* alamat web *www.kompas.com*

$$\text{Paket Loss} = \frac{\text{Packet_transmitted} - \text{Packet_received}}{\text{Packet_transmitted}} \times 100\%$$

$$= (20 - 20 / 20) \times 100\%$$

$$= 0 \%$$

- b. Untuk *packet loss* alamat web *www.google.com*

$$\text{Paket Loss} = \frac{\text{Packet_transmitted} - \text{Packet_received}}{\text{Packet_transmitted}} \times 100\%$$

$$= (16 - 16 / 16) \times 100\%$$

$$= 0 \%$$

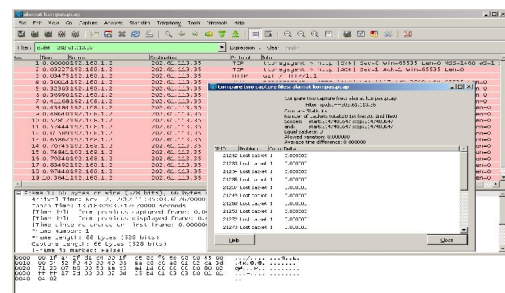
Berdasarkan perhitungan dari Persamaan 6, maka hasil perhitungan keseluruhan sampai dengan *server* yang ke-5 dapat diimplementasikan ke dalam bentuk Tabel 2.

Tabel 2. Hasil Pengujian *Packet Loss* menurut *software wireshark*

No	Waktu Pengujian	Alamat Web	Jumlah Paket	Packet Loss (%)
1	11:35:03 s/d 11:35:13 Wib	www.kompas.com	20	0
2	11:44:04 s/d 11:44:35 Wib	www.google.com	16	0
3	11:51:36 s/d 11:51:59 Wib	www.youtube.com	21	0
4	11:58:46 s/d 11:59:31 Wib	www.cisco.com	10	0
5	12:04:49 s/d 12:04:56 Wib	www.detik.com	91	0

Untuk alamat web *google.com*, *youtube.com*, *cisco.com* dan *detik.com* juga menghasilkan *packet loss* sebesar 0 %, yang artinya paket yang dikirim dan diterima dari *client* menuju *server* atau sebaliknya dari *server* menuju *client*, paket

tidak ada mengalami *broken* (rusak) ataupun hilang (*lost*) pada saat pengiriman ataupun penerimaan data. Biasanya hal yang menjadi penyebab adanya *packet loss*, pada saat *request* ataupun *receive* data ke / dari *server* adalah kegagalan pada jaringan, kepadatan trafik di jaringan, karena kesalahan *hardware*, dan keterbatasan *bandwidth* pada jaringan internet saat melakukan transmisi data ke *server* ataupun menerima data dari *server* menuju *client*. Dan *protocol TCP* akan melakukan *retransmission packet* menuju alamat yang dituju pada saat *request* ataupun *receive* data sampai paket tersebut benar-benar utuh (kompleks) dikirim ke alamat yang dituju. Dari hasil pengujian pada *software wireshark*, maka didapat parameter *packet loss* dalam pengambilan data untuk setiap *server* seperti ditunjukkan pada Gambar 7.



Gambar 7. Captured data *Packet Loss* pada *software wireshark*

6. Analisis Perhitungan *Troughput*

Troughput merupakan suatu kinerja jaringan yang terukur. *Troughput* juga diartikan sebagai kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data per satuan waktu. Untuk menghitung *troughput* terhadap alamat *website* dapat menggunakan Persamaan 7.

- a. Untuk *troughput* pada alamat web *www.kompas.com*

$$\text{Troughput} = \frac{\text{Jumlah data yang dikirim}}{\text{Waktu pengiriman data}} = \frac{\text{Average Bytes / sec}}{\text{Time between first \& last packet (sec)}}$$

$$= 502.746 / 7.813$$

$$= 64.34 \text{ kbps}$$

- b. Untuk *troughput* pada alamat web *www.google.com*

$$\text{Troughput} = \frac{\text{Jumlah data yang dikirim}}{\text{Waktu pengiriman data}} = \frac{\text{Average Bytes / sec}}{\text{Time between first \& last packet (sec)}}$$

$$= 2484.364 / 0.324$$

$$= 7667.79 \text{ kbps}$$

- c. Untuk *troughput* pada alamat web *www.youtube.com*

$$\begin{aligned} \text{Troughput} &= \frac{\text{Jumlah data yang dikirim}}{\text{Waktu pengiriman data}} = \frac{\text{Average Bytes/sec}}{\text{Time between first \& last packet (sec)}} \\ &= 31.446 / 38.002 \\ &= 0.827 \text{ kbps} \end{aligned}$$

d. Untuk *troughput* pada alamat web *www.cisco.com*

$$\begin{aligned} \text{Troughput} &= \frac{\text{Jumlah data yang dikirim}}{\text{Waktu pengiriman data}} = \frac{\text{Average Bytes/sec}}{\text{Time between first \& last packet (sec)}} \\ &= 443.187 / 35.603 \\ &= 12.44 \text{ kbps} \end{aligned}$$

e. Untuk *troughput* pada alamat web *www.detik.com*

$$\begin{aligned} \text{Troughput} &= \frac{\text{Jumlah data yang dikirim}}{\text{Waktu pengiriman data}} = \frac{\text{Average Bytes/sec}}{\text{Time between first \& last packet (sec)}} \\ &= 687.431 / 13.793 \\ &= 49.83 \text{ kbps} \end{aligned}$$

Berdasarkan perhitungan dari Persamaan 7, maka hasil perhitungan keseluruhan sampai dengan *server* yang ke-5 dapat diimplementasikan ke dalam bentuk Tabel 3.

Tabel 3. Hasil Pengujian *Troughput* menurut *software wireshark*

No	Waktu pengujian	Alamat Web	Average Bytes / sec	Time between first & last packet (s)	Troughput (kbps)
1	21:45:47 s/d 21:46:35 Wib	www.kompas.com	502.746	7.813	64.34
2	21:51:19 s/d 21:52:00 Wib	www.google.com	2484.364	0.324	7667.79
3	22:05:05 s/d 22:05:57 Wib	www.youtube.com	31.446	38.002	0.827
4	22:09:23 s/d 22:10:05 Wib	www.cisco.com	443.187	35.603	12.44
5	22:13:35 s/d 22:14:13 Wib	www.detik.com	687.431	13.793	49.83

7. Kesimpulan

Adapun kesimpulan yang dapat diambil dari Paper ini adalah sebagai berikut :

1. Untuk hasil *delay* yang terbesar terdapat pada saat mengakses alamat web *www.youtube.com*, dimana menghasilkan *delay* sebesar 1.09939 sec. Sedangkan untuk hasil *delay* yang terkecil terdapat pada saat mengakses alamat web *www.cisco.com*, dimana menghasilkan *delay* sebesar 0.1221 sec dengan menggunakan web browser mozilla firefox 12.0
2. Untuk hasil *packet loss* yang dihasilkan pada saat mengakses alamat web adalah 0%, yang artinya paket yang dikirim dan diterima dari *client* menuju *server* atau sebaliknya dari *server* menuju *client*, paket tidak ada mengalami

broken (rusak) ataupun hilang (*lost*) pada saat pengiriman ataupun penerimaan data.

3. Untuk hasil *troughput* yang terbesar terdapat pada saat mengakses alamat web *www.google.com* yaitu 7667.79 kbps, sedangkan untuk hasil nilai *troughput* yang terkecil pada saat mengakses alamat web *www.youtube.com* yaitu 0.827 kbps, dimana semakin besar nilai *troughput* yang dihasilkan saat mengakses alamat web atau situs web, maka penundaan waktu / *delay* yang dihasilkan semakin kecil, dan sebaliknya.

8. Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Sutrisno Sihombing, SH dan Tumiar Sibuea selaku orang tua penulis, Ir. M. Zulfin, MT selaku dosen pembimbing, juga Ali Hanafiah Rambe ST, MT, Maksun Pinem ST, MT, selaku dosen penguji penulis yang sudah membantu penulis dalam menyelesaikan paper ini, dan semua pihak yang tidak sempat penulis sebutkan satu persatu.

9. Daftar Pustaka

- [1]. Boger, Paul. 2010. "CCNA Exploration Course Booklet Network Fundamentals Version 4.0". Cisco Press, United States of America. <http://www.cisco.com/edu>
- [2]. Zulfin, Muhammad. 2012. "Komunikasi Data", Medan.
- [3]. Wirawan, Widhi. 2012. "Analisis dan Implementasi IPTV dengan menggunakan Media Webcam". Universitas Udayana, Bali.
- [4]. Ilma, Zidni Urida. 2011. "Rancang Bangun dan analisa *Quality of Service (QoS)* pada Sistem *Voice Over IP (VoIP)* menggunakan *Open Source Elastix*". <http://uridadotzidni.blogdetik.com/>
- [5]. Faruq. 2011. "Praktikum 14 Analisa QoS Jaringan". <http://lecturer.eepis-its.edu/~zenhadi/kuliah/Jarkom1/Praktik%20Modul%2014%20Analisa%20QoS.pdf>