

BAB IV

HASIL PENGUJIAN DAN ANALISA

Bab ini membahas cara pengujian aplikasi *video conference* berbasiskan web dengan menggunakan IntServ, dalam hal ini yang menjadi objek penelitian adalah Openmeetings, yaitu dengan melakukan pengujian terhadap kualitas layanan seperti *delay* dan *throughput* dengan menggunakan QoS IntServ. Tujuannya adalah untuk mengetahui kelebihan kekurangan sistem apakah sistem yang telah dibuat sesuai dengan apa yang direncanakan dan bekerja secara efektif sesuai dengan yang diharapkan.

4.1 Persiapan Pengujian

Sebelum pengujian dilakukan pemeriksaan pada aplikasi Openmeetings, tujuannya adalah agar dapat mengetahui apakah ada *error* diaplikasi dan memeriksa apakah aplikasi opemeetings sudah aktif. Cara mengaktifkan openmeetings dengan cara :

1. Mengaktifkan red5 terlebih dahulu.
2. Klik red5.bat pada aplikasi.
3. Pada browser ketik localhost:5080/openmeetings.

4.2 Tahap – Tahap Pengujian dan Pengambilan Data

Berikut pengujian dari sistem yang telah dirancang yaitu :

1. Dilakukan pengujian terhadap kualitas layanan pada aplikasi openmeetings dengan jumlah *client* yang sedang melakukan *video conference* sebanyak 2,3 dan 4 *client* dengan menggunakan QoS.
2. Dilakukan pengujian terhadap kualitas layanan pada aplikasi Openmeetings dengan jumlah *client* yang sedang melakukan *video conference* sebanyak 2,3 dan 4 *client* menggunakan QoS dan Tanpa QoS.

4.3 Pengujian dan Pengambilan Data

Setelah sistem berjalan dengan baik selanjutnya adalah pengujian dan pengambilan data berikut adalah pengujian yang dilakukan.

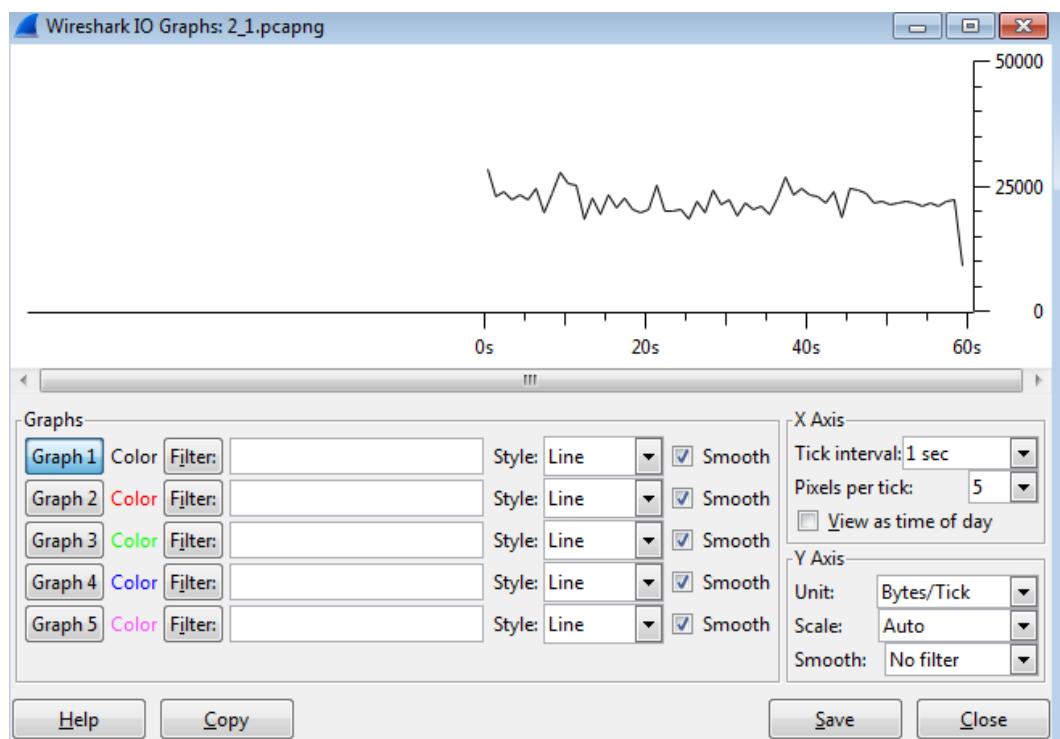
4.3.1 Pengujian yang Sedang Melakukan *Conference* Sebanyak 2 Client

Pada pengujian ini dilakukan dengan cara yang sedang melakukan *conference* berjumlah 2 *client* pada aplikasi Openmeetings.

Pengujian dilakukan dengan cara *capture* data menggunakan *software* Wireshark.

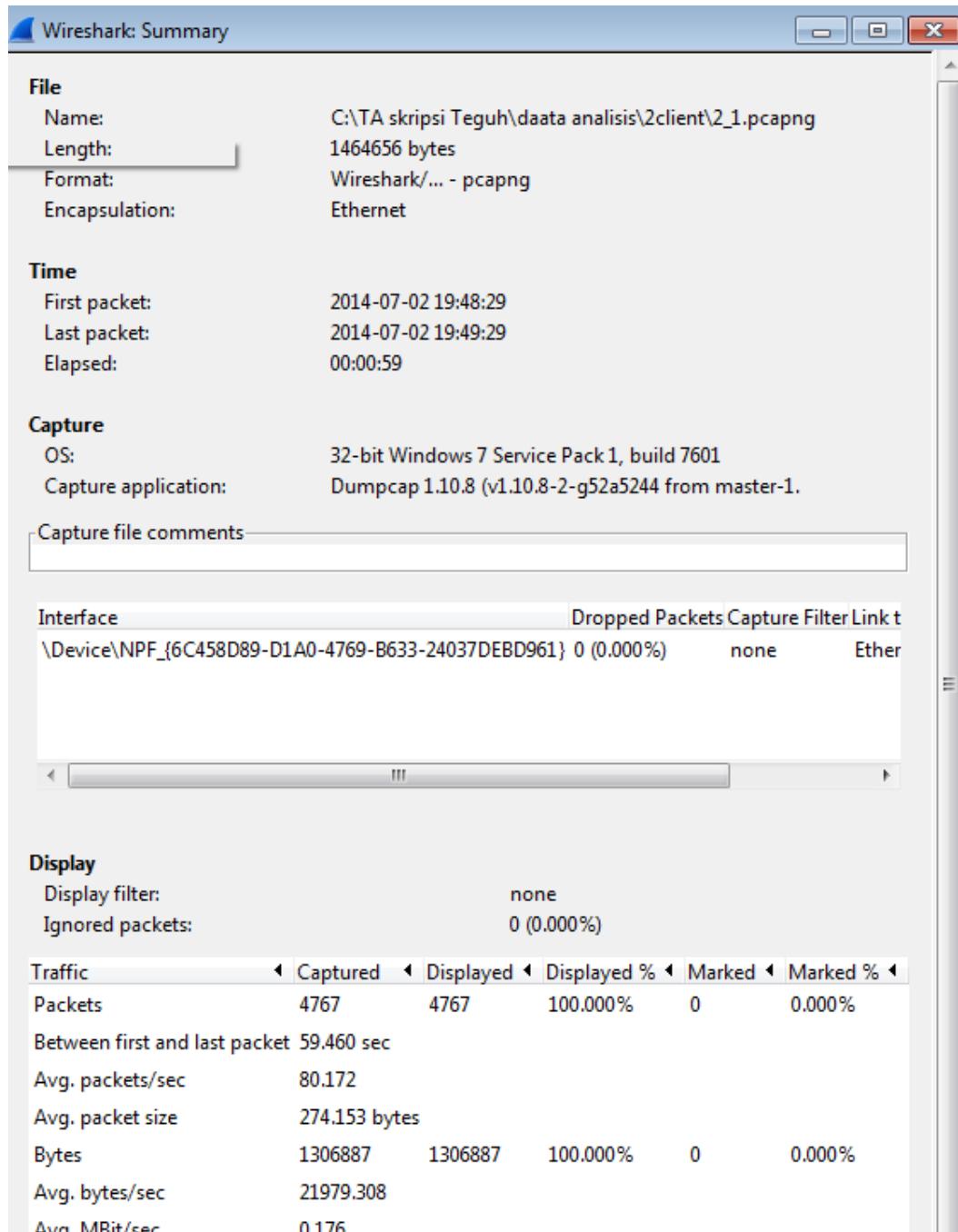
Berikut penjelasan pengujian :

1. Pengujian menggunakan 2 client



Gambar 4.1 Grafik TCP dan ICMP

Dari gambar 4.2 dapat dijelaskan, bahwa grafik pada pengujian menggunakan 2 *client* menunjukkan sumbu X (*second*) merupakan waktu pengiriman data, sedangkan sumbu Y (Bytes/Tick) merupakan besar paket data TCP dan ICMP. Pada grafik, waktu *capture* data berlangsung selama 60s. Selama waktu *capture* data, aktifitas *video conference* ditunjukkan oleh besarnya paket data TCP antara 10000 – 25000 bytes dan aktifitas yang terjadi pada paket data ICMP yaitu mendekati 0.



Gambar 4.2 *Throughput* untuk 2 client video conference

nilai delay didapat dengan cara menggunakan aplikasi wireshark pada waktu *video conference* jumlah total waktu transmisi sebanyak paket sebesar 4767 paket sehingga didapat nilai *delay* adalah sebagai berikut :

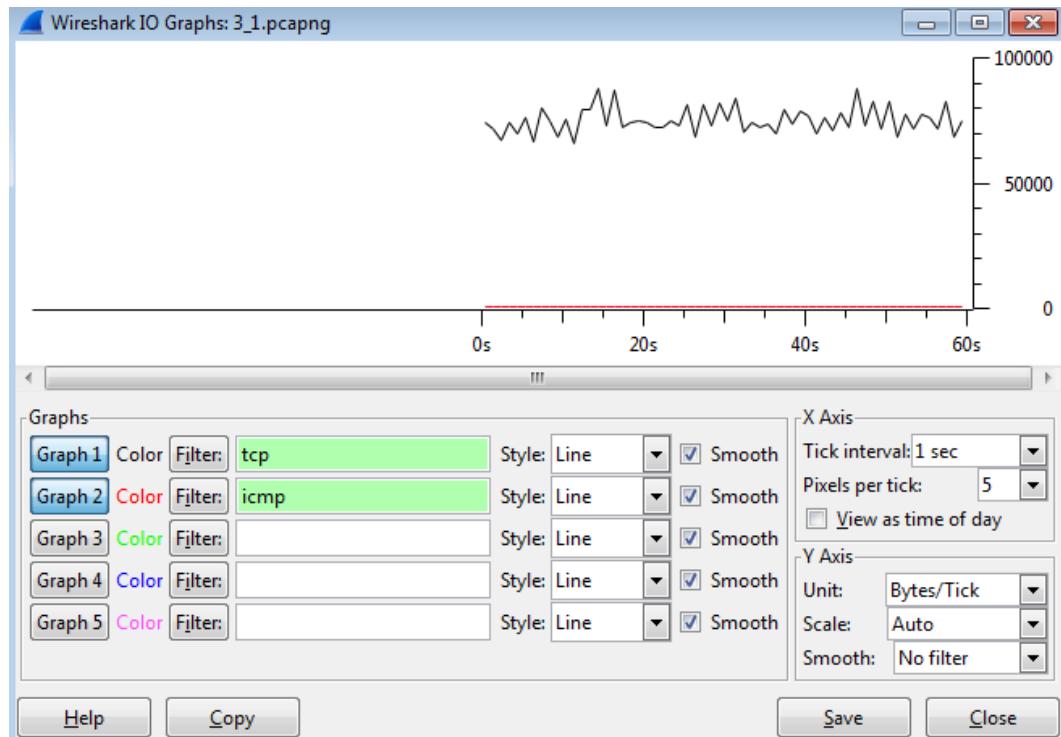
$$\text{Delay} = \frac{4767 \text{ paket}}{14205,79024} = 0,033557 \text{ s} = 33,557 \text{ ms}$$

Kemudian dari rata-rata hasil pada wireshark, didapat nilai *throughput* sebesar 0,176 Mbit/sec.

4.3.2 Pengujian yang Sedang Melakukan *Conference* Sebanyak 3 Client

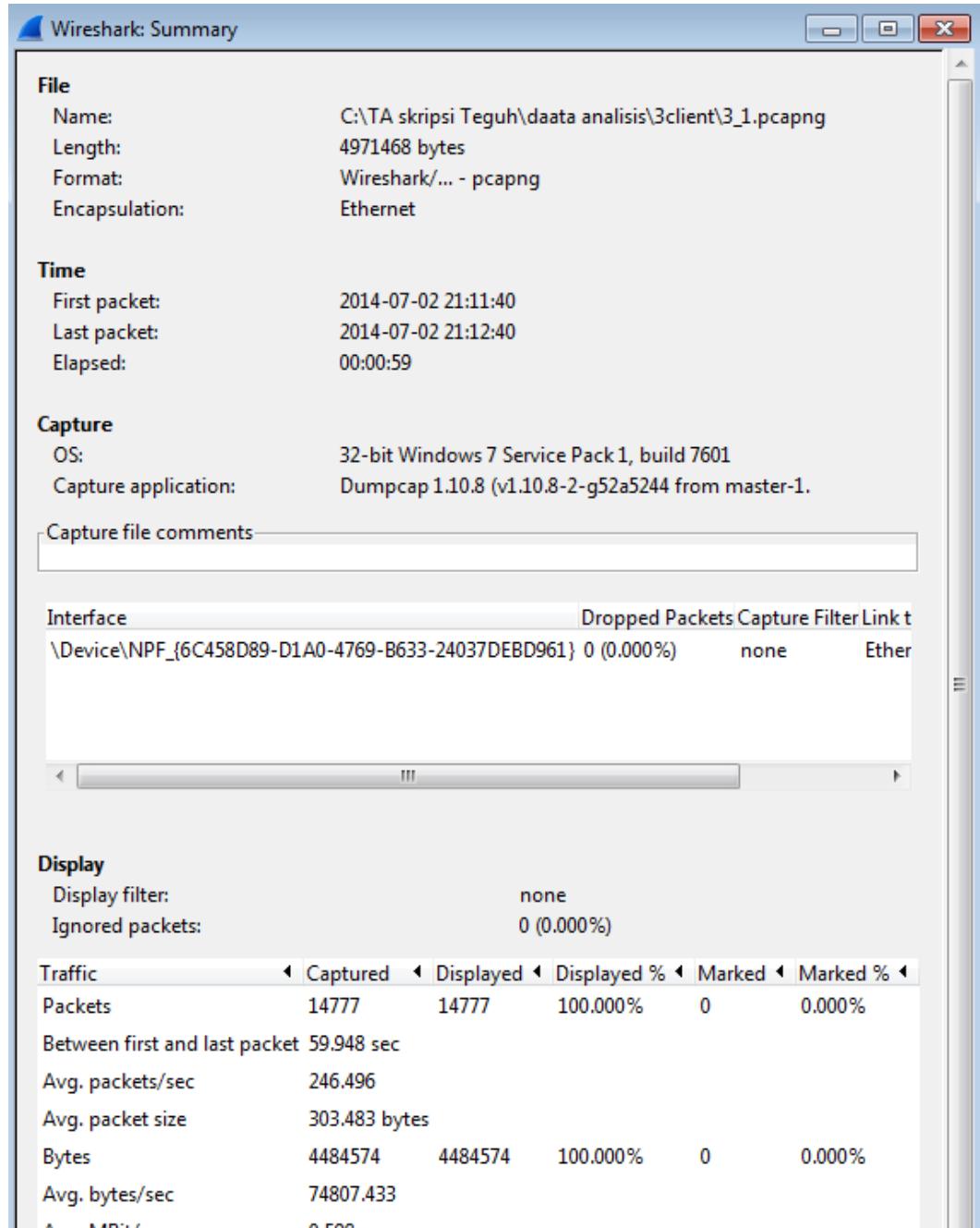
Pada pengujian ini dilakukan dengan cara yang sedang melakukan *conference* berjumlah 3 *client* pada aplikasi Openmeetings :

1. Pengujian menggunakan 3 Client



Gambar 4.3 Grafik TCP dan ICMP pada 3 client.

Dari gambar 4.5 dapat dijelaskan, bahwa grafik pada *video conference dengan 3 client* menunjukkan hal yang sama pada saat 2 *client* , yaitu sumbu X (*second*) merupakan waktu pengiriman data, sedangkan sumbu Y (Bytes/Tick) merupakan besar paket data TCP dan aktifitas *video conference* yang ditunjukan oleh paket data TCP besarnya antara 70000 – 800000 Bytes dan untuk paket data ICMP yang diperoleh, yaitu mendekati 0 dan pada *video conference* menggunakan 3 *client* terlihat data semakin bertambah banyak datanya.



Gambar 4.4 *Throughput* pada 3 client.

nilai delay didapat dengan cara menggunakan aplikasi wireshark pada waktu *video conference* jumlah total waktu transmisi sebanyak paket sebesar 14777 paket sehingga didapat nilai *delay* adalah sebagai berikut :

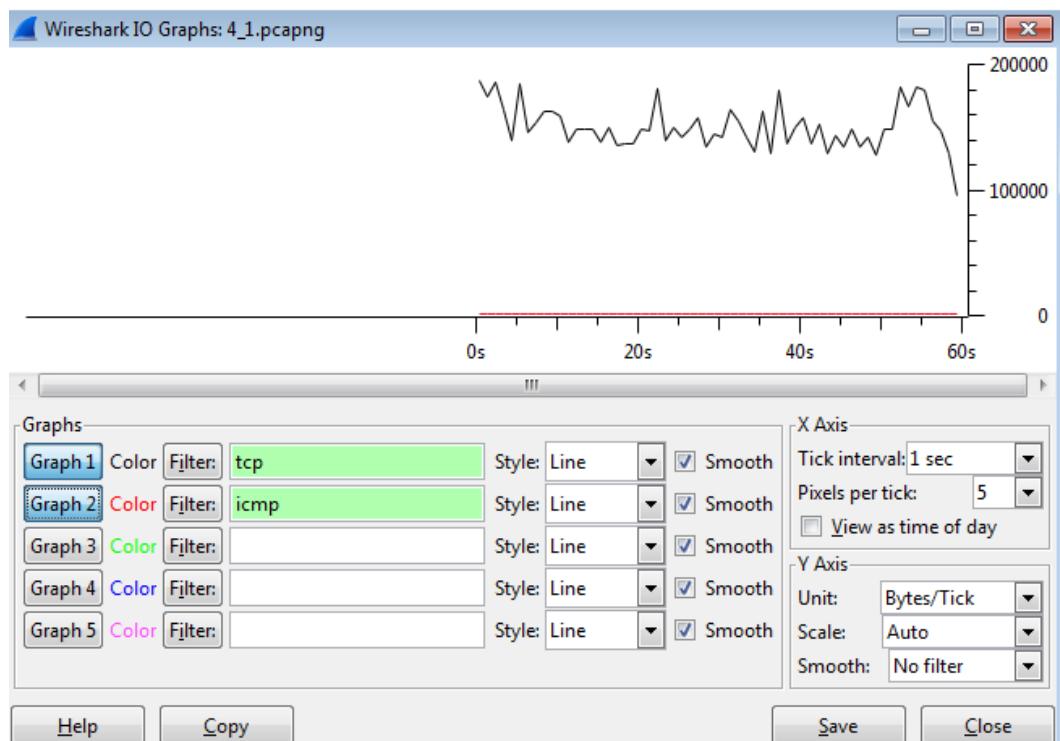
$$Delay = \frac{14777 \text{ paket}}{446648,0627} = 0,03308 \text{ s} = 33,08 \text{ ms}$$

Kemudian dari rata-rata hasil pada wireshark, didapat nilai *throughput* sebesar =0,598 Mbit/sec.

4.3.3 Pengujian yang Sedang Melakukan *Conference* Sebanyak 4 *Client*

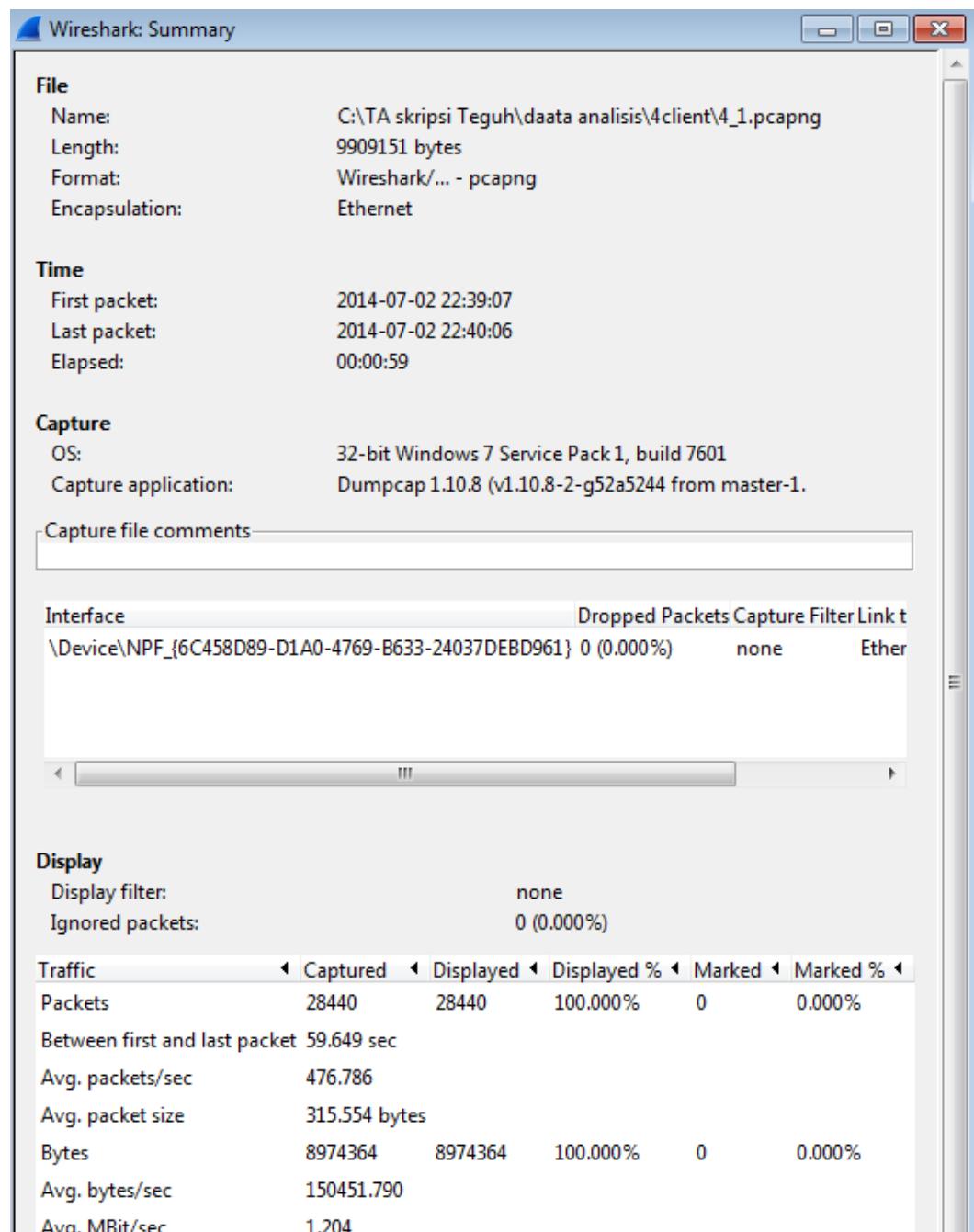
Pada pengujian ini dilakukan dengan cara yang sedang melakukan *conference* berjumlah 4 *client* pada aplikasi Openmeetings seperti terlihat pada gambar berikut :

1. Pengujian menggunakan 4 *Client*



Gambar 4.5 Grafik TCP dan ICMP pada 4 *client*.

Dari gambar 4.6 dapat dijelaskan, bahwa grafik pada *video conference* dengan 4 *client* menunjukkan hal yang sama pada saat 3 *client* , yaitu sumbu X (*second*) merupakan waktu pengiriman data, sedangkan sumbu Y (Bytes/Tick) merupakan besar paket data TCP dan aktifitas *video conference* yang ditunjukan oleh paket data TCP besarnya antara 100000 - 190000 Bytes dan untuk paket data ICMP yang diperoleh, yaitu mendekati 0 dan pada *video conference* menggunakan 4 *client* terlihat data semakin bertambah banyak datanya.



Gambar 4.6 *Throughput* pada 4 client.

nilai delay didapat dengan cara menggunakan aplikasi wireshark pada waktu *video conference* .jumlah total waktu transmisi sebanyak paket sebesar 28440

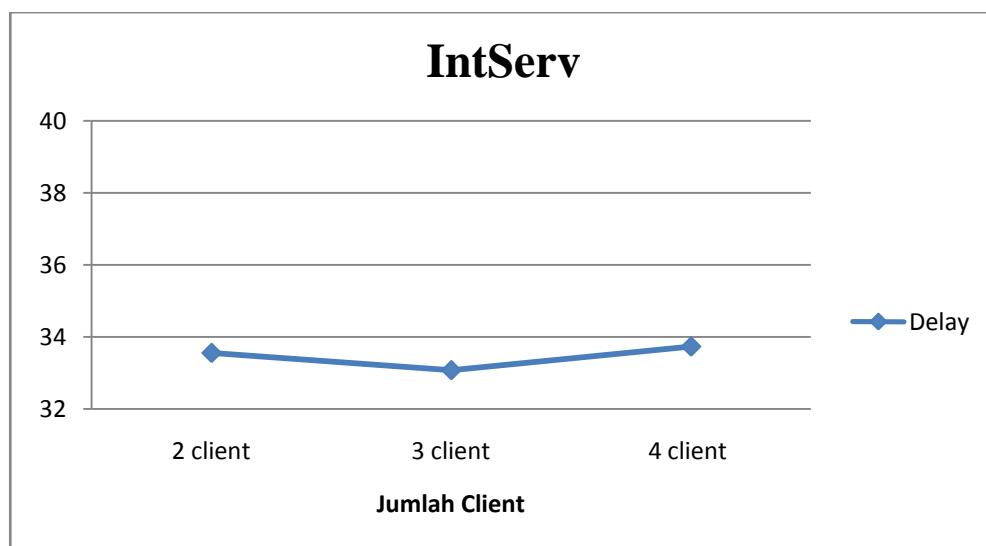
paket sehingga, sehingga didapat nilai *delay* adalah sebagai berikut $Delay =$

$$\frac{28440 \text{ paket}}{842993,385} = 0,033737 \text{ s} = 33,737 \text{ ms}$$

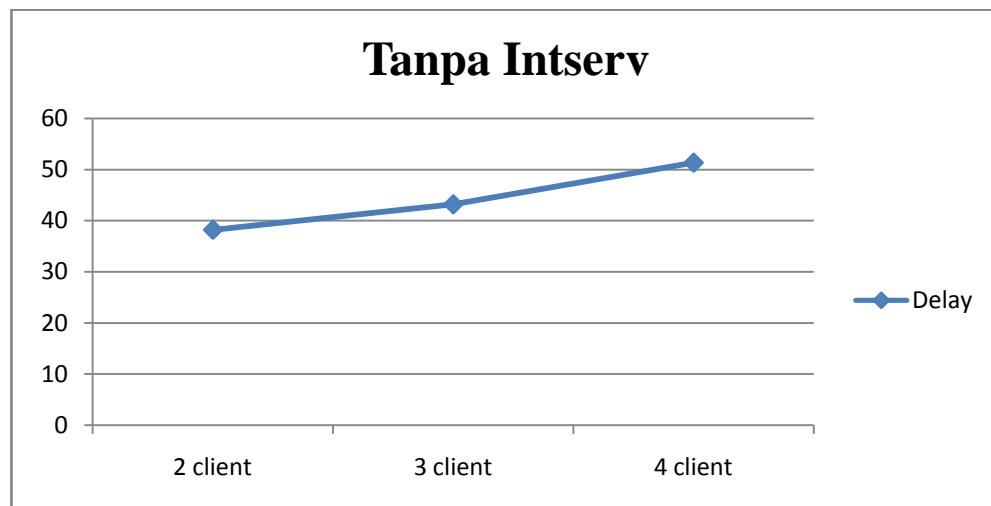
Kemudian dari rata-rata hasil pada wireshark, didapat nilai *throughput* sebesar =1,204 Mbit/sec.

Tabel 4.1 Hasil Pengujian

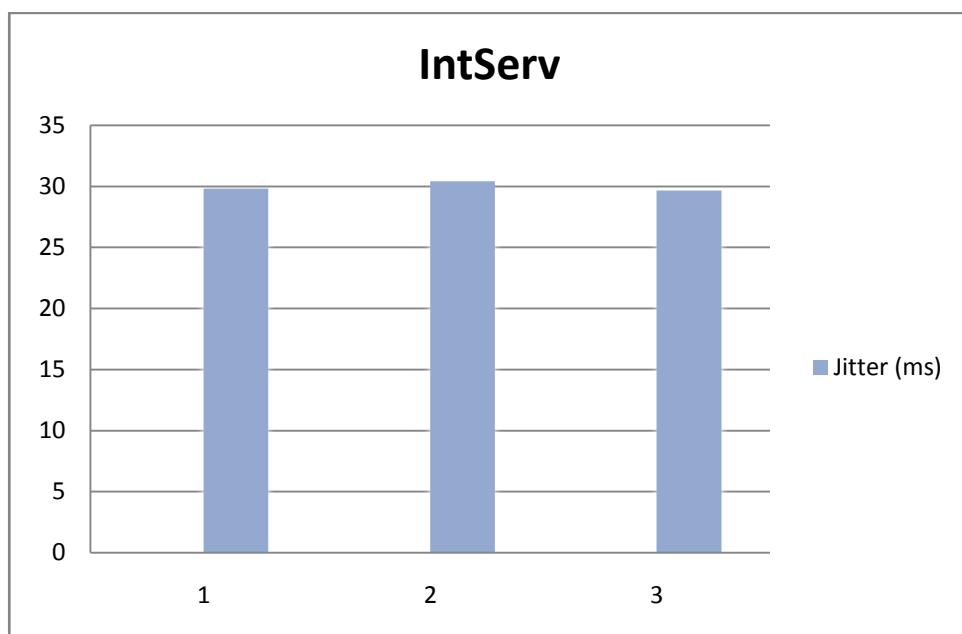
IntServ				Tanpa Intserv		
Jumlah client	Delay (ms)	Throughput (Mbps)	Jitter (ms)	Delay (ms)	Throughput (Mbps)	Jitter (ms)
2	33,557	0,176	29,79	38,23	0,150	35,14
3	33,08	0,598	30,4	43,201	0,430	42,2
4	33,737	1,204	29,64	51,344	1,103	49,5



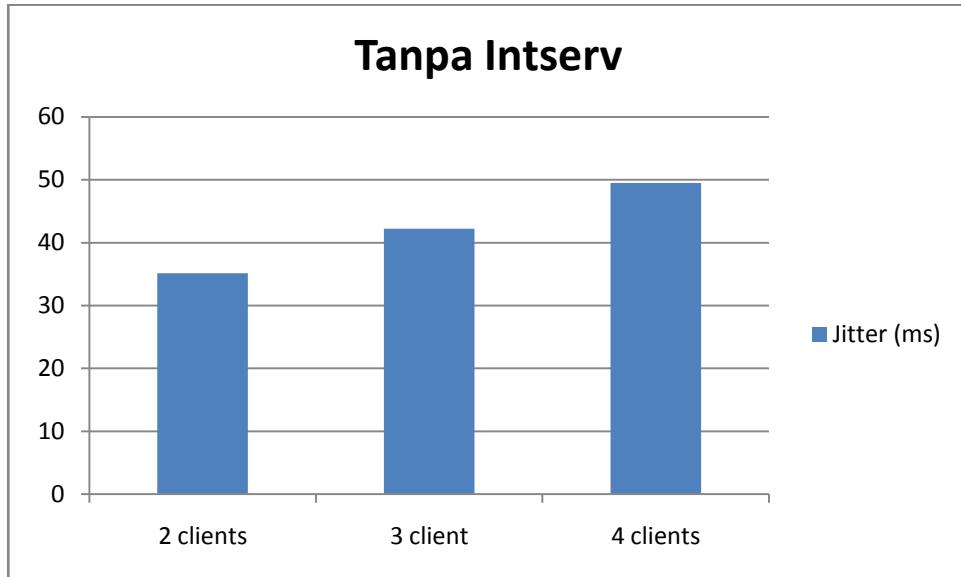
Gambar 4.7 Grafik *Delay* Hasil Pengujian dengan IntServ



Gambar 4.8 Grafik *Delay* Hasil Pengujian Tanpa IntServ



Gambar 4.9 Grafik *Jitter* Hasil Pengujian dengan IntServ



Gambar 4.10 Grafik *Jitter* Hasil Pengujian Tanpa IntServ

4.4 Analisa

Dari pengujian yang dilakukan terdapat dua hasil, yaitu hasil dengan menggunakan grafik dan ringkasan (*summary*). Hasil dengan grafik didapat ketika *client* sedang melakukan *conference* dan diambil datanya menggunakan *software* Wireshark. Pada masing-masing pengujiannya, data yang diambil melalui *software* Wireshark dilakukan dalam waktu 60s, hal ini dilakukan untuk membatasi skala pengujian agar tidak terlalu luas. Kemudian dari hasil grafik yang didapat, paket data TCP yang dikirim dalam *video conference* pada setiap *client* menunjukkan penggunaan *video conference* pada aplikasi OpenMeetings berjalan dengan baik. Sedangkan untuk paket data ICMP yang didapat dari hasil setiap pengujian hampir mendekati 0, ini disebabkan bahwa paket data ICMP berfungsi untuk melihat *respon time* pada aplikasi OpenMeetings. Hasil selanjutnya yaitu dengan menggunakan ringkasan (*summary*). Ringkasan (*summary*) didapat dari *software* Wireshark. Ringkasan digunakan untuk melihat kualitas layanan *video conference*. Parameter yang digunakan untuk melihat kualitas layanan pada *software* Wireshark, yaitu *delay* dan *throughput*. *Delay* yang dihasilkan *video conference* menggunakan IntServ berjalan dengan baik ketika 2 client maupun dengan 4 *client* dengan nilai jitter di antara 33 ms – 33,7 ms ini terbukti bahwa RSVP berjalan dengan baik dengan memberikan *source*

yang di butuhkan setiap *client* sehingga memperoleh nilai *delay* yang stabil , yang membuat kualitas *video conference* berjalan dengan baik tanpa ada *delay* yang bernilai besar.semakin banyak jumlah *client* data yang dihasilkan akan semakin besar , disinilah peran penting teknik IntServ untuk mengatur *source* yang di minta *client* karena *video conference* harus mempunyai nilai *delay* yang kecil agar kenyamanan disaat melakukan *video conference* berjalan baik tidak tergaggu oleh *delay*. Semakin banyak *client* yang melakukan *video conference* memerlukan throughput yang besar ini terbukti dari uji yang dilakukan ketika *video conference* dengan 2 *client* ,3 *client* dan 4 *client*.