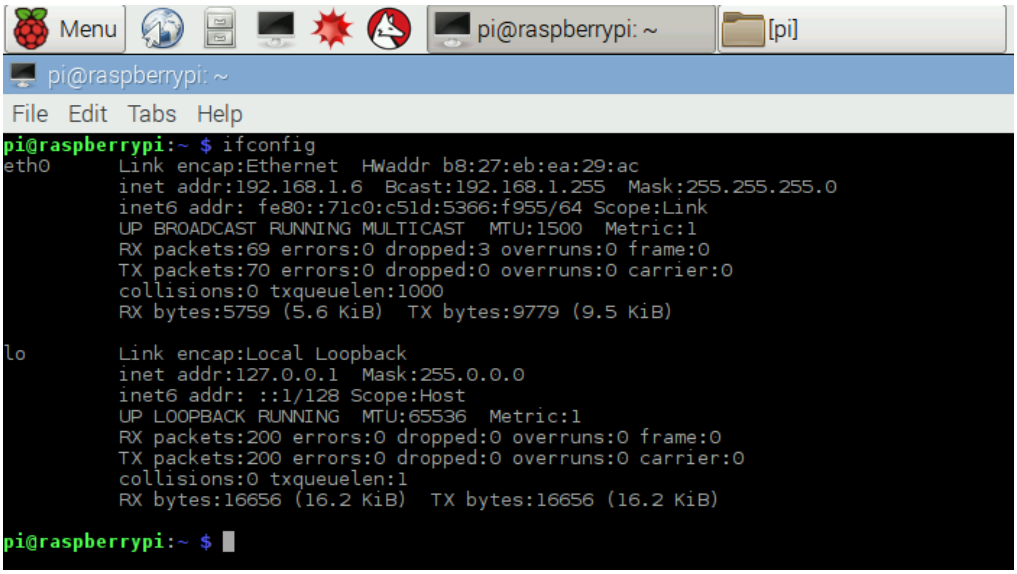
BAB IV
PENGUJIAN DAN ANALISA

Bab ini membahas tentang tahap-tahap pengujian dan analisa Mini PC Raspberry Pi sebagai *Server* video conference, dalam hal ini objek yang menjadi bahan peneletian ada pada dua sisi yaitu yang pertama dari sisi *Server* dan dari sisi *Client*. Dari sisi *Server* kita akan mengukur performa dari Raspberry Pi sebagai *Server* di lihat dari kemampuan penggunaan CPU dan RAM ketika *Client* sedang menjalakan aplikasi *video conference*, untuk sisi *Client* kita akan mengukur *delay* dan *througput* saat *video conference* di akses oleh *Client*.

4.1 Persiapan Pengujian

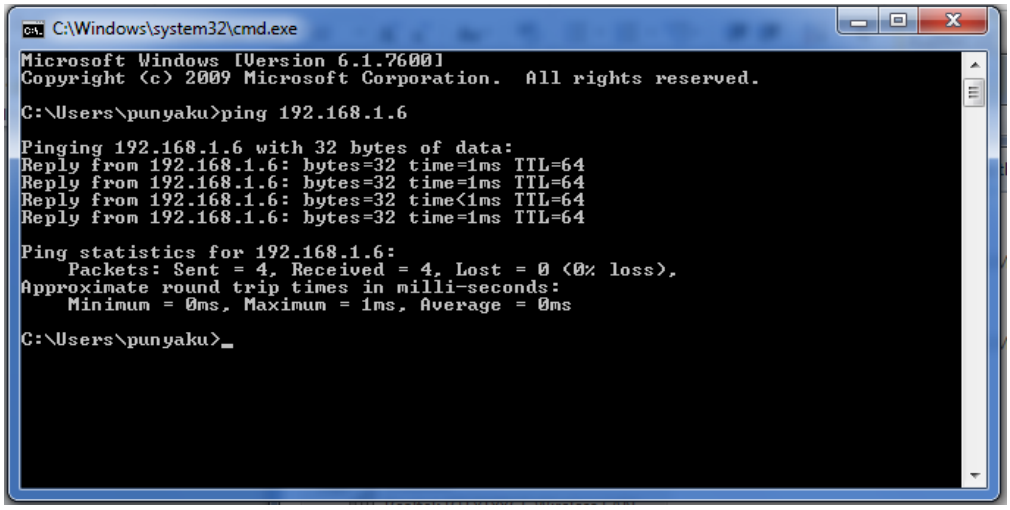
Sebelum pengujian dilakukan ada beberapa tahap yang perlu diperhatikan, uraiannya sebagai berikut:

1. Pastikan pengalamatan Raspberry Pi telah di *setting* DHCP kemudian periksa informasi Alamat IP di terminal dengan cara ketikan perintah `ifconfig` seperti terlihat pada gambar IV.1



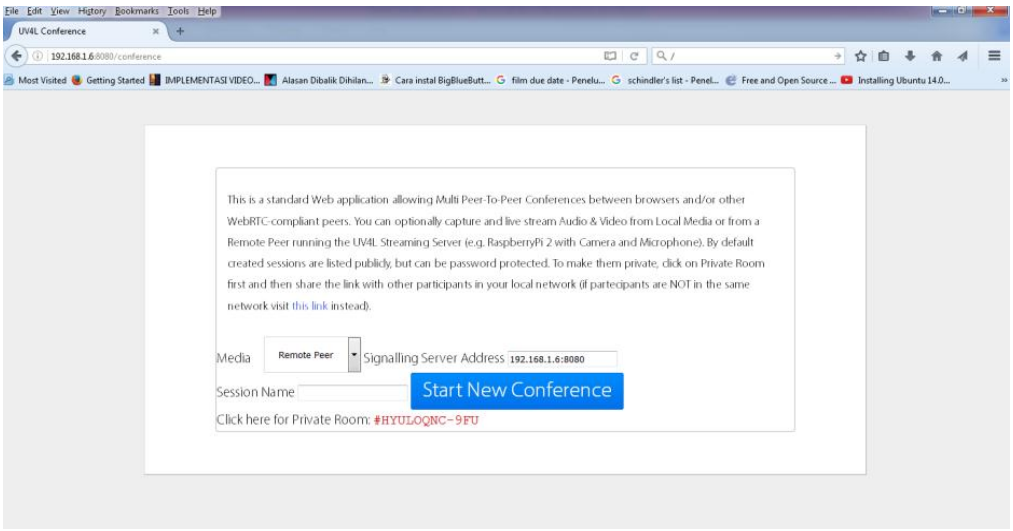
Gambar IV.1 Informasi Alamat IP *Server*

2. Setelah itu hubungkan *Client* 1 dengan jaringan *video conference* dengan cara mengaktifkan Wifi dan cek koneksi apakah *Client* 1 sudah tersambung pada *Server* seperti pada gambar IV.2 berikut.



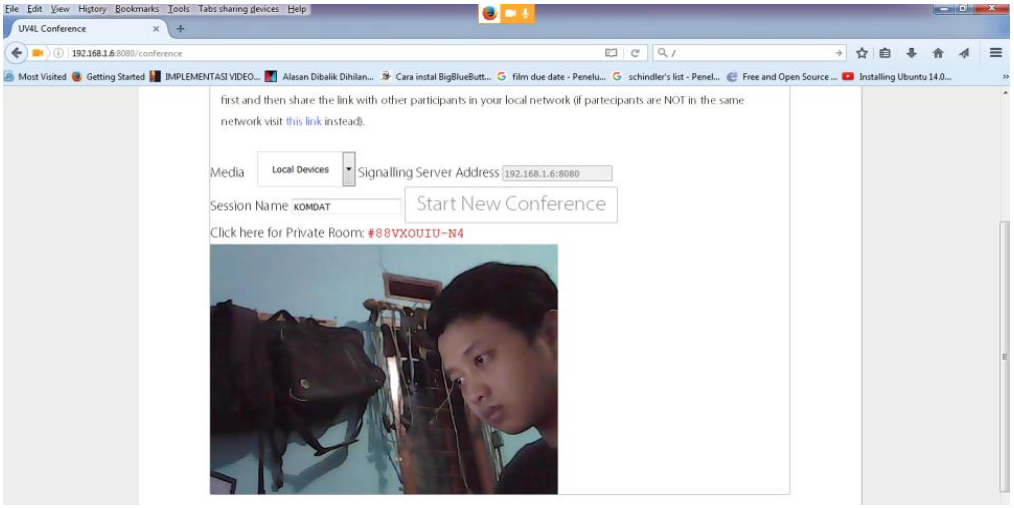
Gambar IV.2 Tes Koneksi ke *Server*

3. Lalu buka web browser dan ketikan perintah seperti ini (192.168.1.6:8080/conference) maka akan terbuka halaman untuk memulai *video conference* seperti di tunjukan pada gambar IV.3.



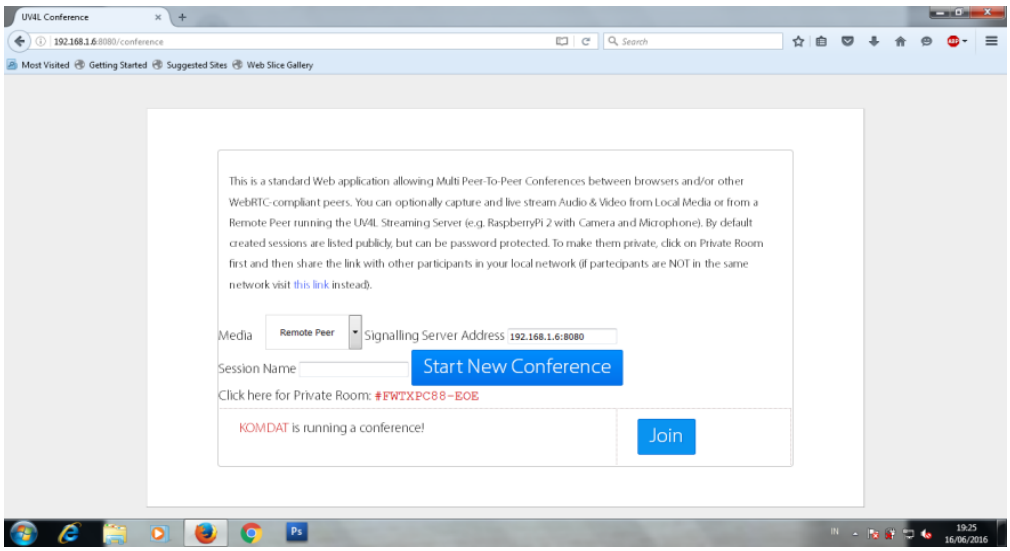
Gambar IV.3 Halaman Utama *Video Conference*

4. Isikan *session name* dan pilih *local devices* pada tab media kemudian klik *start new conference* maka akan muncul halaman *video conference* dengan satu *Client* yang sudah terhubung, seperti pada contoh gambar IV.4.



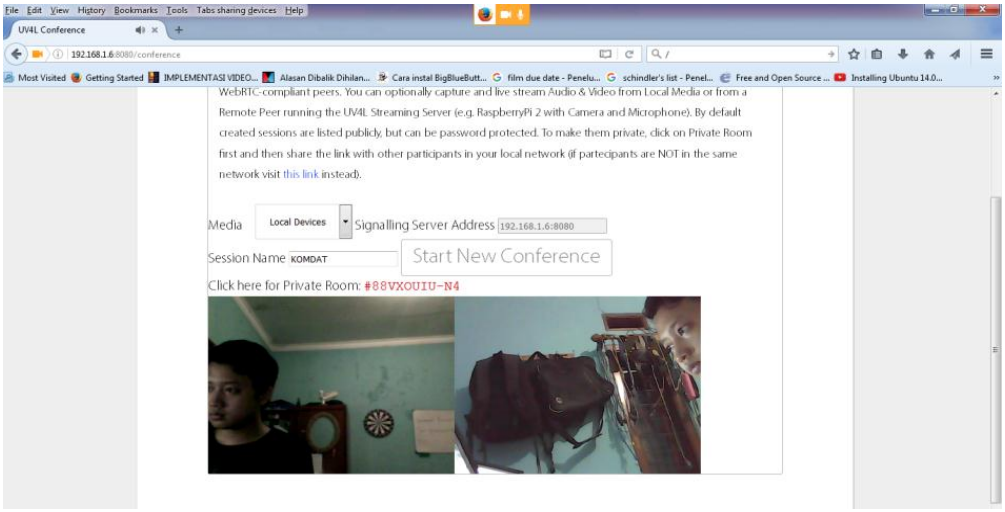
Gambar IV.4 Satu *Client* Aktif di *Video Conference*

5. Setelah *Client* 1 terhubung tambahkan dengan *Client* 2 caranya sama seperti pada *Client* 1, pada web browser *Client* 2 ketikan (192.168.1.6:8080/conference) setelah muncul halaman *video conference* tunggu beberapa saat hingga tombol *join* muncul seperti terlihat pada gambar IV.5



Gambar IV.5 *Client* 2 *Join Video Conference*

6. Terakhir akan muncul dua *frame video conference* pada masing masing *Client*, *Client* 1 dan *Client* 2 siap untuk berkomunikasi lewat *video conference*.



Gambar IV.6 Dua *Client* Aktif di *Video Conference*

4.2 Tahap Pengujian

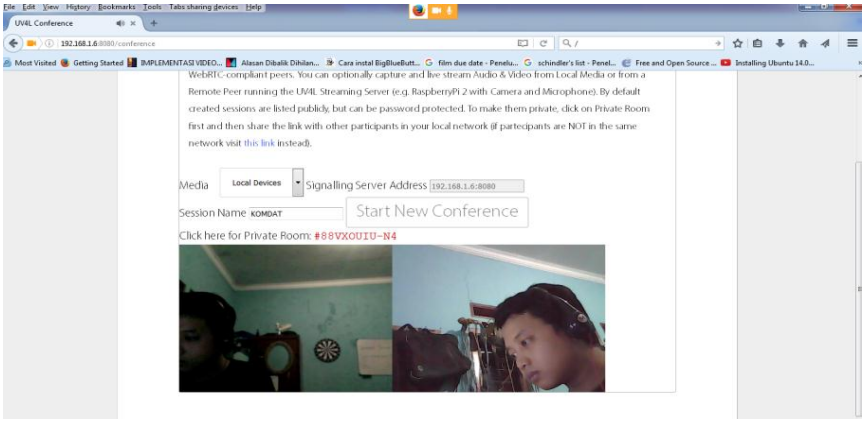
Berikut ini adalah rancangan pengujian pada sistem *video conference* yang akan dilakukan:

1. Dilakukan pengujian dari sisi *Client*, pada 2 *Client* yang sedang melakukan akses *video conference* lalu di ukur kualitas layanannya kemudian *Client* akan ditambah menjadi 3 *user* lalu mengukur kembali kualitas layanan.
2. Dilakukan pengujian dari sisi *Server*, pada 2 *Client* yang sedang melakukan akses *video conference* lalu di ukur penggunaan CPU dan RAM kemudian *Client* akan di tambah menjadi beberapa *user* lalu mengukur kembali penggunaan CPU dan RAM sampai beban *Server* maksimal.

Setelah sistem berjalan dengan baik selanjutnya masuk kedalam tahap pengujian dan pengambilan data, berikut ini adalah pengujian yang dilakukan.

4.2.1 Pengujian dari Sisi *Client*

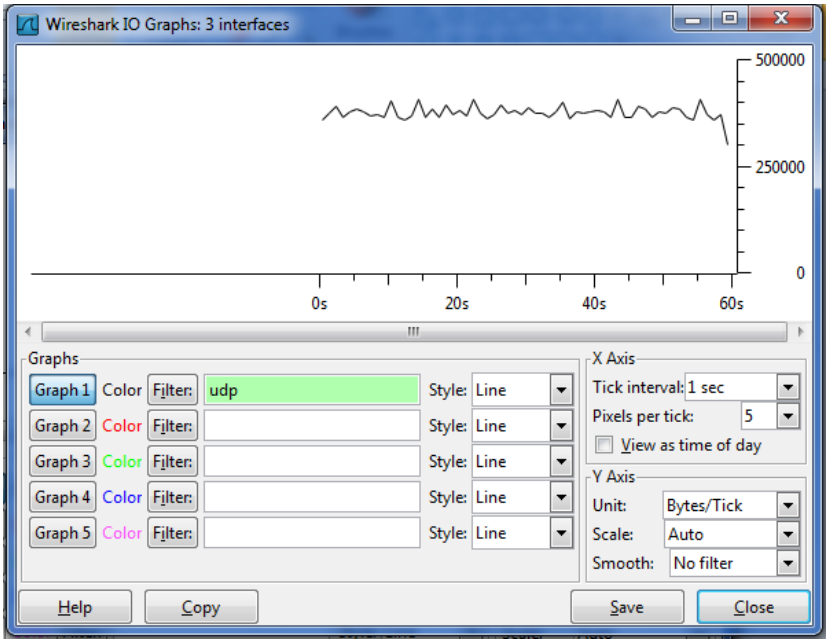
Pengujian kali ini dilakukan pada saat *Client* 1 dan *Client* 2 sedang berkomunikasi melalui *video conference* seperti terlihat pada gambar dibawah:



Gambar IV.7 Dua *Client* Saat Komunikasi

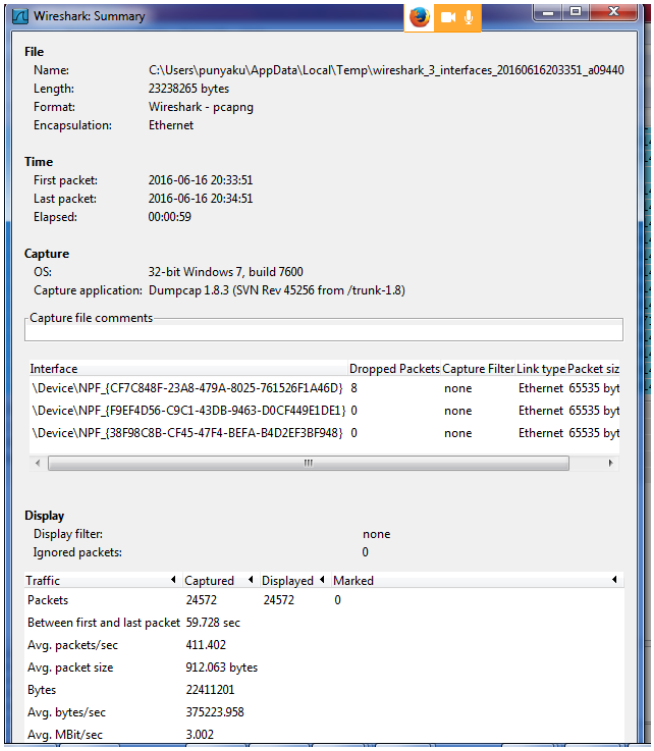
Data hasil pengujian diambil dengan cara capture menggunakan aplikasi wireshark, berikut adalah penjelasan hasil pengujian:

1. Pengujian 2 Buah *Client*



Gambar IV.8 Grafik UDP Saat Menguji *Video Conference*

Dari gambar IV.8 dapat di jelaskan bahwa grafik pada aplikasi wireshark menunjukan sumbu X merupakan waktu pengiriman data, sedangkan sumbu Y merupakan besar paket data UDP, waktu capture data berlangsung selama 1 menit. Selama waktu capture data aktifitas video dan suara ditunjukan oleh besarnya paket data UDP antara 300.000 – 400.000 Bytes.



Gambar IV.9 Hasil *Delay* dan *Througput*

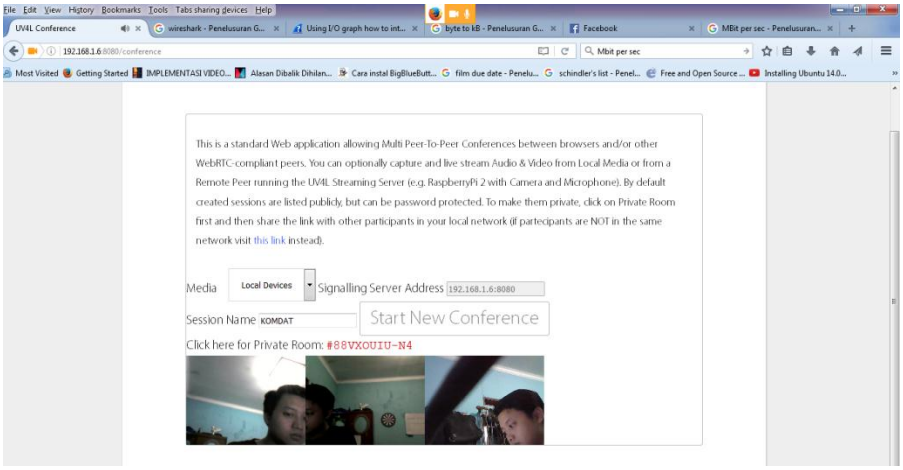
Dari gambar IV.9 didapat nilai waktu transmisi sebesar 59,728 sec dan total paket data sebesar 24572, sehingga didapat nilai *delay* dengan menggunakan rumus 2.1.

$$Delay = \frac{59{,}728 \text{ sec}}{24572 \text{ paket}} = 0{,}0024 s/paket = 2{,}4 ms/paket$$

Kemudian dari hasil Summary Wireshark didapat nilai throughput sebesar 3,002 Mbit/sec.

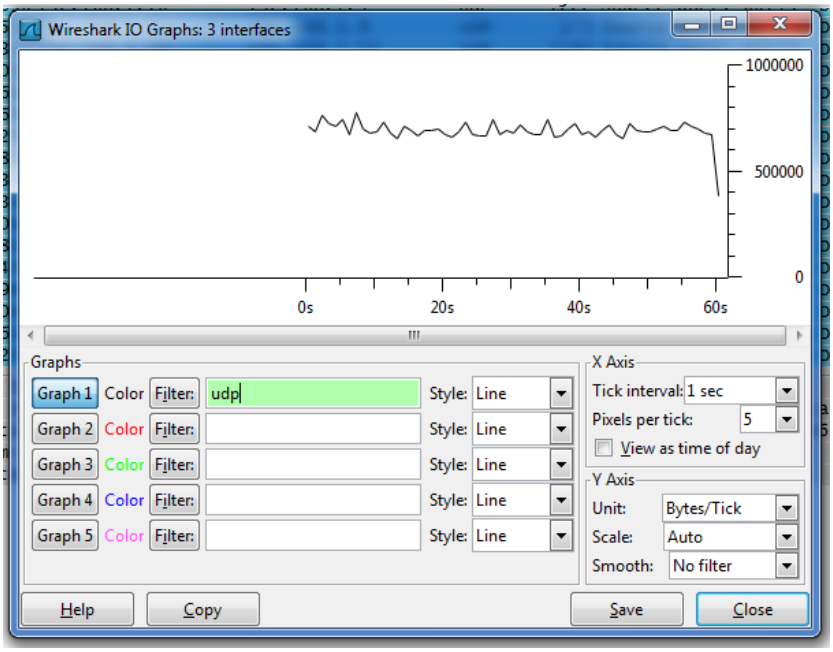
2. Pengujian 3 Buah *Client*

Setelah 2 buah *Client* diuji langkah selanjutnya adalah menambah jumlah *Client* menjadi 3 buah, caranya sama seperti menambah *Client* 2 dengan mengetik (192.168.1.6:8080/conference) pada web browser di *Client* 3 tunggu beberapa saat hingga muncul tombol join lalu klik join. Untuk penambahan menjadi 3 *Client* dibutuhkan proses tunggu cukup lama sampai 3 *frame video conference* muncul di setiap *Client*. Gambar IV.10 menampikan hasil *video conference* dengan 3 buah *Client* yang aktif.



Gambar IV.10 3 *Client* Aktif di *Video Conference*

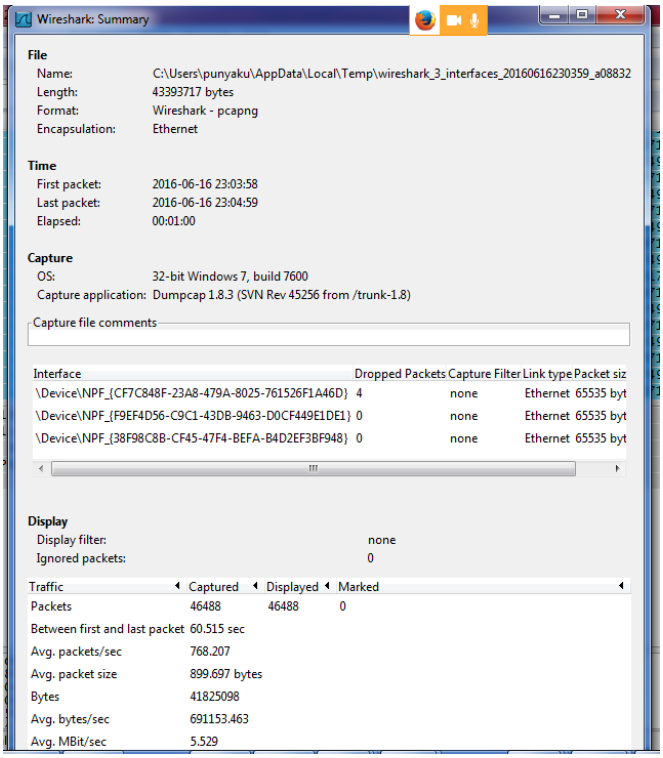
Setelah 3 *Client* join *video conference* langkah selanjutnya adalah menguji kembali kualitas layanan, hasil dari pengujian yang di capture oleh aplikasi wireshark adalah sebagai berikut:



Gambar IV.11 Grafik UDP Saat Menguji 3 *Client*

Dari gambar IV.11 dapat di jelaskan bahwa grafik pada aplikasi wireshark menunjukan sumbu X merupakan waktu pengiriman data, sedangkan sumbu Y merupakan besar paket data UDP, waktu capture data sama seperti ketika menguji 2 *Client* yaitu berlangsung selama 1 menit. Selama waktu capture data aktifitas

video dan suara ditunjukan oleh besarnya paket data UDP antara 650.000 – 700.000 Bytes.



Gambar IV.12 Hasil Ke-2 Delay dan Througput

Dari gambar IV.12 didapat nilai waktu transmisi sebesar 60,515 sec dan total paket data sebesar 46848, sehingga didapat nilai delay sebesar:

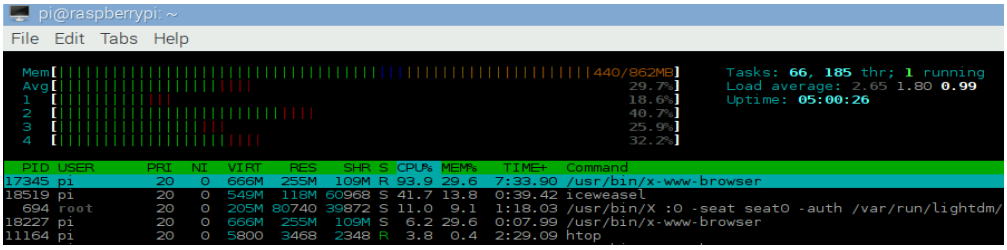
$$Delay = \frac{60{,}515 \text{ sec}}{46848 \text{ paket}} = 0{,}0012 s/paket = 1{,}2 ms/paket$$

Kemudian dari hasil Summary Wireshark didapat nilai throughput sebesar 5,529 Mbit/sec.

4.2.2 Pengujian dari Sisi *Server*

Setelah pengujian dari sisi *Client* selesai maka selanjutnya kita masuk ke dalam pengujian di sisi *Server*, kita lihat kemampuan penggunaan CPU dan RAM pada *Server* raspberry pi dalam menangani *Client* yang sedang mengakses *video conference*. Berikut adalah penjelasan hasil pengujian performa *Server* menggunakan aplikasi htop:

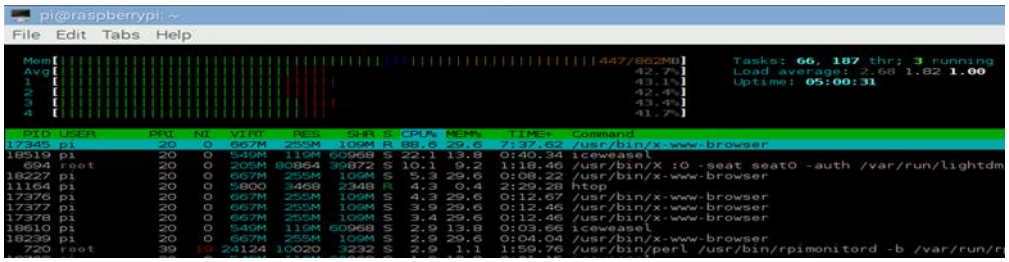
1. Pengujian *Server* dengan 2 *Client* Aktif



Gambar IV.13 Hasil Capture Penggunaan CPU dan RAM

Dari gambar IV.13 dapat dijelaskan bahwa hasil capture data menggunakan aplikasi htop menunjukan pemakaian memory sebesar 440MB dan beban terhadap CPU sebesar 29.7% saat 2 *Client* aktif dalam *Server video conference*.

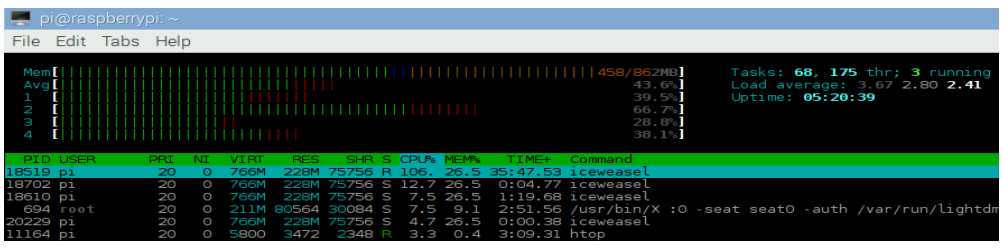
2. Pengujian *Server* dengan 3 *Client* Aktif



Gambar IV.14 Hasil Capture Penggunaan CPU dan RAM Ke-2

Dari gambar IV.14 dapat dijelaskan bahwa hasil capture data menggunakan aplikasi htop menunjukan pemakaian memory sebesar 447MB dan beban terhadap CPU sebesar 42.7% saat 3 *Client* aktif dalam *Server video conference*.

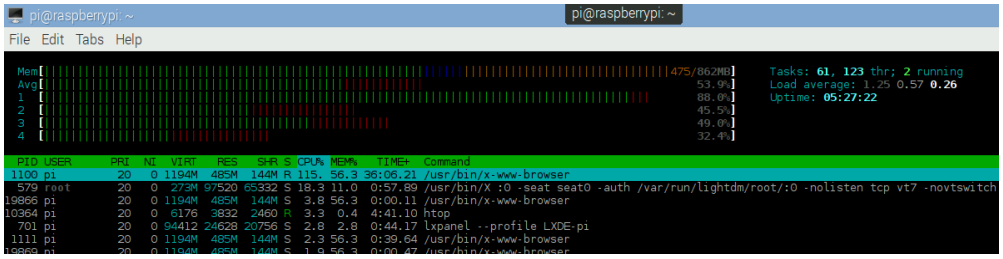
3. Pengujian *Server* dengan 4 *Client* Aktif



Gambar IV.15 Hasil Capture Penggunaan CPU dan RAM Ke-3

Dari gambar IV.15 dapat dijelaskan bahwa hasil capture data menggunakan aplikasi htop menunjukan pemakaian memory sebesar 450MB dan beban terhadap CPU sebesar 43.6% saat 4 *Client* aktif dalam *Server video conference*.

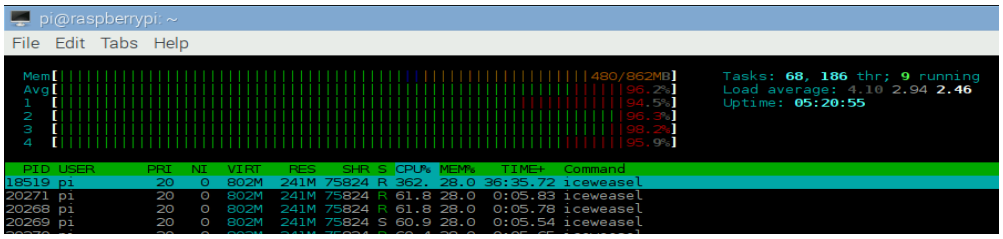
4. Pengujian *Server* dengan 5 *Client* Aktif



Gambar IV.16 Hasil Capture Penggunaan CPU dan RAM Ke-4

Dari gambar IV.17 dapat dijelaskan bahwa hasil capture data menggunakan aplikasi htop menunjukan pemakaian memory sebesar 475MB dan beban terhadap CPU sebesar 53.9% saat 5 *Client* aktif dalam *Server video conference*.

5. Pengujian *Server* dengan 6 *Client* Aktif



Gambar IV.17 Hasil Capture Penggunaan CPU dan RAM Ke-5

Dari gambar IV.18 dapat dijelaskan bahwa hasil capture data menggunakan aplikasi htop menunjukan pemakaian memory sebesar 490MB dan beban terhadap CPU sebesar 96.2% saat 6 *Client* aktif dalam *Server video conference*.

4.3 Analisa

Dari dua hasil pengujian baik sisi *Client* atau sisi *Server* dibuat analisa sebagai berikut:

1. Analisis dari Sisi *Client*

   Dari pengujian di sisi *Client* terdapat dua hasil data, yaitu grafik dan ringkasan (summary), didapatkan ketika 2 *Client* sedang melakukan *video conference* dan selanjutnya ditambah menjadi 3 *Client* lalu diambil datanya menggunakan aplikasi Wireshark. Pengambilan data dilakukan dalam waktu 1 menit, hal tersebut dilakukan untuk membatasi skala pengujian agar tidak memakan waktu yang lama. Hasil data grafik yang didapat berupa paket data UDP. Paket data UDP yang dikirim dari *Server video conference* ke setiap *Client* menunjukan penggunaan *video conference* berfungsi dengan baik. Selanjutnya hasil data pada ringkasan (summary) berupa nilai delay dan throughput, untuk nilai delay pada saat pengujian 2 *Client* aktif dan 3 *Client* aktif pada *video conference*, tidak ada perbedaan delay yang terlalu jauh sedangkan untuk nilai throughput selisih antara 2 *Client* dan 3 *Client* sebesar 2,572 Mbit/sec.Dari hasil data yang di ambil menggunakan aplikasi wireshark tidak ada perbedaan hasil data yang terlalu jauh antara 2 *Client* yang aktif dan 3 *Client* yang aktif saat mengakses *video conference*, tetapi pada kenyataannya saat *Client* 1 dan 2 aktif dan *Client* 3 join ke dalam *video conference* terjadi *loading* yang cukup lama di halaman *join video conference* lalu kualitas gambar di setiap *frame video conference* mengalami penurunan kualitas.

2. Analisis dari Sisi *Server*

   Untuk pengujian di sisi *Server* kita mengukur kinerja dari Penggunaan CPU dan RAM menggunakan aplikasi htop. Raspberry pi sebagai *Server video conference* diukur kemampuan penggunaan CPU dan RAM pada saat 2 *Client* mengakses *video conference* dan selanjutnya ditambah sampai 6 *Client* mengakses *video conference*. Tabel berikut adalah hasil data Penggunaan CPU dan RAM dalam bentuk tabel.

Tabel IV.1 Hasil Pengujian Kinerja penggunaan CPU dan RAM

Pengujian *Server*		
Jumlah *Client*	Penggunaan CPU	RAM
2	29.7%	440MB
3	42.7%	447MB
4	43.6%	450MB
5	53.9%	475MB
6	96.2%	490MB

Dari hasil tabel IV.1 dapat di analisa bahwa penambahan 1 *Client* pada *video conference* akan membebani kinerja *Server*, jika *Client* di tambah terus untuk join di *video conference* maka kinerja *Server* akan terus terbebani sampai akhirnya *Server* down.