

BAB V

PENGUJIAN DAN ANALISIS

Pengujian yang dilakukan pada skripsi ini difokuskan pada pengujian performansi *Video On Demand* pada *Virtual Private Network* dengan menggunakan OpenVPN. Pengujian pada skripsi ini terdiri dari lima proses, yaitu : *Pengujian koneksi jaringan Virtual Private Network*, *pengujian client Virtual Private Network*, *pengujian koneksi jaringan Video On Demand*, *pengujian proses Streaming pada Video On Demand*, dan *pengujian streaming Video On Demand* melalui OpenVPN. Dari hasil pengujian diperoleh data-data primer yang diperlukan dalam analisis performansi jaringan, yaitu : Jumlah paket data, *throughput*, *jitter*, dan *packet loss*.

Analisis performansi *Video On Demand* pada *Virtual Private Network* dengan menggunakan OpenVPN yang dilakukan secara *analytical analysis* berdasarkan pada teori dan persamaan yang terdapat pada tinjauan pustaka. Analisis performansi *Video On Demand* pada *Virtual Private Network* dengan menggunakan OpenVPN, meliputi : *delay*, *throughput*, dan *packet loss*. Hasil analisis akan ditampilkan dalam bentuk grafik.

Performansi *Video On Demand* pada *Virtual Private Network* dengan menggunakan OpenVPN yang dibahas dalam skripsi ini memiliki variabel bebas jumlah *client* pengakses layanan *streaming video on demand* dan jenis resolusi *file video* yang dilakukan *streaming*.

5.1 Pengujian Koneksi Jaringan *Virtual Private Network*

Pengujian ini bertujuan untuk mengetahui perangkat komunikasi dalam jaringan *Virtual Private Network* menggunakan OpenVPN dapat berjalan dengan baik dengan cara mengirimkan paket dari *client* menuju komputer *server* kemudian dikirimkan kembali dalam jangka waktu yang telah ditentukan oleh komputer.

a. Peralatan Yang Digunakan Untuk Pengujian Koneksi Jaringan VPN

Peralatan yang digunakan dalam pengujian koneksi jaringan VPN antara lain:

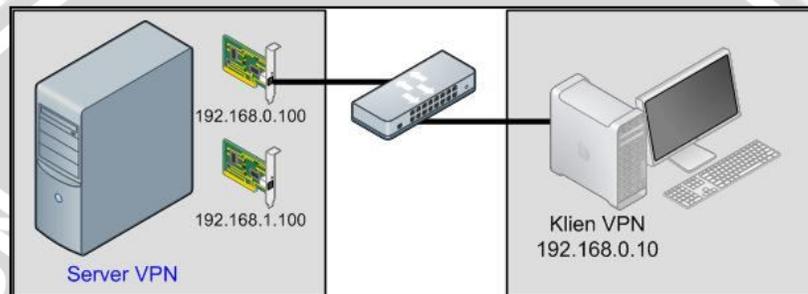
1. 1 unit komputer sebagai *client*
2. 1 unit komputer sebagai *server*
3. 1 buah *Switch*
4. 2 buah NIC
5. 2 buah kabel LAN sepanjang 5 m dengan konektor RJ-45

6. Sistem operasi Window Xp SP 3
7. Sistem Operasi Linux Slackware 12

b. Prosedur Pengujian Koneksi Jaringan *Virtual Private Network*.

Prosedur pengujian koneksi jaringan VPN adalah sebagai berikut :

1. Mengatur alamat IP *server Virtual Private Network* pada NIC pertama dengan alamat 192.168.0.100 dan pada NIC kedua dengan alamat 192.168.1.100.
2. Mengatur alamat IP *client Virtual Private Network* dengan alamat 192.168.0.10.



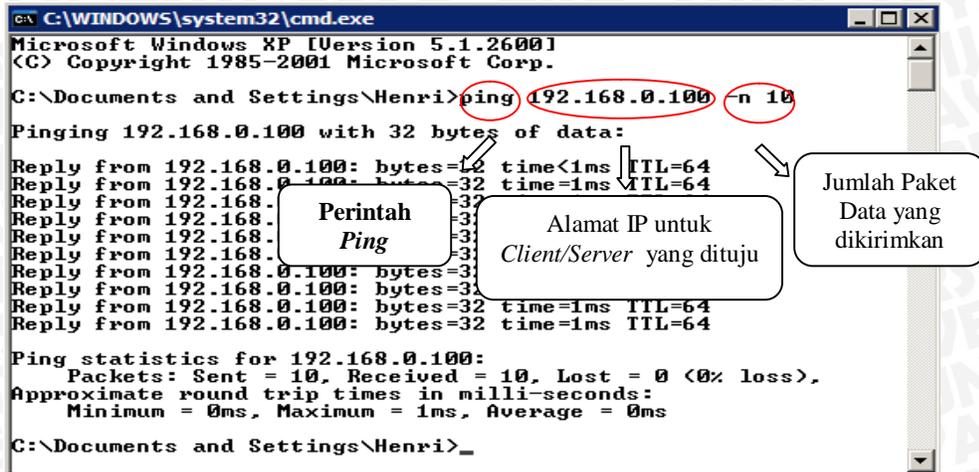
Gambar 5.1 Topologi Pengujian Koneksi Jaringan *Virtual Private Network*

Sumber : Pengujian

3. Menjalankan *command prompt* pada salah satu *client*.
4. Menjalankan perintah *ping* untuk menguji koneksi jaringan VPN berdasarkan respon waktu yang diperoleh.
5. Menjalankan perintah *ping* yang ditujukan pada *server* VPN.
6. Melakukan pengamatan terhadap respon waktu pada pengujian koneksi.

c. Hasil Pengujian Koneksi Jaringan *Virtual Private Network*

Data hasil pengujian koneksi jaringan VPN mengacu pada Gambar 5.2, antara *client* dan *server* adalah berupa *respons time* pengiriman paket data uji pada saat perintah dituliskan hingga sampai perintah dikirimkan kembali oleh *server* atau *client* yang dituju.



Gambar 5.2 Tampilan *Command Prompt* sisi *client* *Virtual Private Network*

Sumber : Pengujian

Dari hasil pengujian antara *client* dan *server* pada tampilan *Command Prompt*, analisis antara perintah-perintah dan respon yang dihasilkan dalam pengujian adalah :

1. PING

Secara umum, perintah ini dilakukan dengan cara menuliskan *IP address* yang dituju, pada pengujian kali ini ditambahkan perintah yang mengidentifikasi jumlah paket yang akan dikirimkan. Perintah ini dilakukan menguji instalasi *IP address* yang digunakan pada suatu perangkat *host* dapat terhubung dengan *host* lain dengan jalur yang benar dengan cara mengirimkan paket uji dan menerima kembali paket uji tersebut dengan jangka waktu tertentu ke *host* penguji.

2. *Request Time Out*

Respon ini berarti *host* dengan *IP address* yang dituju tidak merespon pada perangkat *host* sehingga paket yang dikirimkan oleh *host* penguji tidak diterima kembali.

3. *Reply*

Respon ini berarti *host* yang dituju sudah memiliki instalasi *IP address* ke perangkat jaringannya dengan benar. Perintah ini dinyatakan dengan “*Reply from 192.168.0.100: bytes 32 time =1ms TTL=64*” dengan identifikasi:

- *From*
Respon ini mengidentifikasi *IP address* dari *host* yang dituju.
- *Bytes 32*
Merupakan keterangan jumlah data yang dikirimkan pada saat proses pengujian.
- *Time=1ms*

Merupakan respon waktu yang didapatkan saat melakukan proses mengujian koneksi. Time menunjukkan nilai “*round trip delay*” (disebut juga sebagai *delay* atau *latency*) yang menunjukkan waktu yang diperlukan paket yang anda kirimkan untuk mencapai komputer yang dituju. Nilai ini dihitung dengan membagi dua selisih waktu PING *packet* mulai dikirimkan dengan waktu response dari PING *packet* diterima.

- TTL=64

Sedangkan TTL merupakan nilai “*Time-To-Live*” yang menunjukkan jumlah maksimal melewati IP *address* sebanyak 64 kali. Dengan mengurangi nilai TTL awal yaitu 64 dengan nilai TTL akhir maka bisa dihitung banyaknya *hop* yang dilalui dari komputer asal ke komputer tujuan. Setiap kali PING *packet* melalui sebuah IP *address* maka nilai TTL nya akan dikurangi satu. Sehingga jika TTL mencapai nilai nol, PING *packet* akan dibuang.

d. Kesimpulan Hasil Pengujian

Dari hasil analisis mengacu pada Gambar 5.2, kesimpulan pengujian yang didapat adalah :

1. Dari pengujian koneksi jaringan *Virtual Private Network* didapatkan bahwa antara *client* dengan *server* sudah terhubung sehingga dapat melakukan komunikasi satu dengan lain.
2. Hubungan antar *client* dengan *server* dapat berjalan dengan baik karena pada hasil pengujian didapatkan bahwa data yang dikirim sama dengan data yang diterima, jika respon waktu semakin rendah, maka kondisi jaringan semakin baik.

5.2 Pengujian Client Virtual Private Network

Pengujian ini bertujuan untuk mengetahui aplikasi OpenVPN pada sisi *client* sudah terdaftar kedalam *server* OpenVPN dengan benar sehingga dapat melakukan koneksi yang aman melalui *tunneling* yang telah dibentuk oleh *server Virtual Private Network*. Pada pengujian ini dilakukan pengamatan terhadap proses autentifikasi sertifikat keamanan yang telah dihasilkan oleh *server Virtual Private Network*.

a. Peralatan Pengujian Client

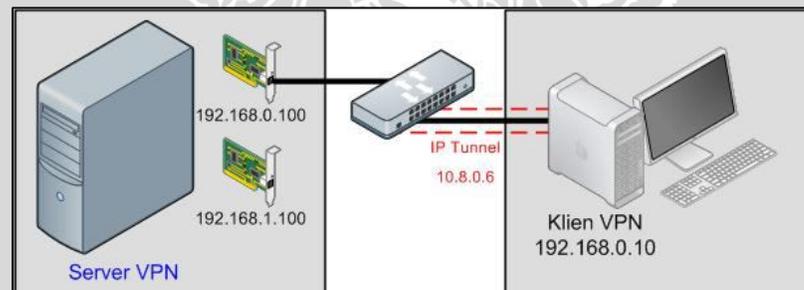
Peralatan yang digunakan dalam pengujian *Client* antara lain:

1. 1 unit komputer sebagai *client*
2. 1 unit komputer sebagai *server*
3. 1 *Switch* 5 port
4. 2 buah *NIC*
5. 2 buah kabel LAN sepanjang 5 m dengan konektor RJ-45
6. Sistem operasi Window Xp SP3
7. Wireshark v.1.8

b. Prosedur Pengujian Client

Prosedur pengujian *Client* adalah sebagai berikut :

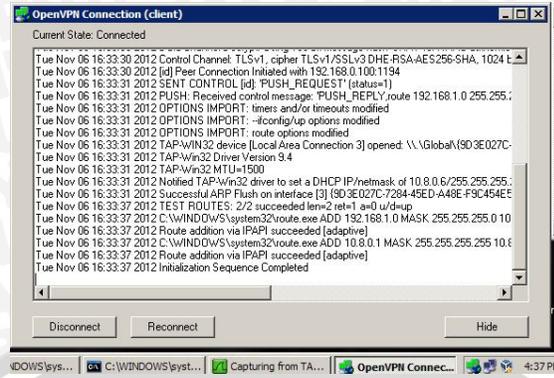
1. Mengatur alamat IP *server Virtual Private Network* pada *NIC* pertama dengan alamat 192.168.0.100 dan pada *NIC* kedua dengan alamat 192.168.1.100.
2. Mengatur alamat IP *client Virtual Private Network* dengan alamat 192.168.0.10.



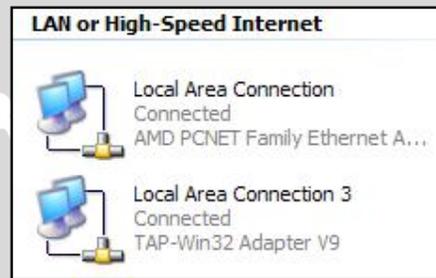
Gambar 5.3 Topologi Pengujian Aplikasi *Client Virtual Private Network*

Sumber : Pengujian

3. Memasukkan sertifikat keamanan pada *client* yang telah terpasang aplikasi OpenVPN GUI 1.0.3.
4. Menjalankan *OpenVPN GUI* sebagai aplikasi *Virtual Private Network* di sisi *client*.
5. Menjalankan *server Virtual Private Network* dengan mengetikkan *command* `/usr/local/sbin# ./openvpn server.conf &`
6. Melakukan koneksi terhadap *server Virtual Private Network* dari aplikasi OpenVPN GUI di sisi *client*.



(a)



(b)

Gambar 5.4 Tampilan *client Virtual Private Network* yang telah terhubung ke *server*

(a) Tampilan koneksi pada OpenVPN GUI dan

(b) Tampilan pada *Network Properties*

Sumber : Pengujian

7. Pada sisi *client* akan terbentuk IP *tunneling* dari *server Virtual Private Network* secara otomatis. Alamat IP *tunneling* yang terbentuk dari pengujian ini adalah 10.8.0.6.
8. Pada *Network Properties* sisi *client*, akan terbentuk sebuah koneksi baru berupa IP *tunneling* yang telah dialokasikan otomatis oleh *server Virtual Private Network*.
9. Menjalan *command prompt* pada salah satu *client*.
10. Menjalankan perintah *ping* untuk menguji koneksi jaringan VPN berdasarkan respon waktu yang diperoleh.
11. Menjalankan perintah *ping* yang ditujukan pada *server* VPN.
12. Melakukan analisis terhadap respon waktu pada alamat IP *tunneling* antara *client* dengan *server Virtual Private Network*.

c. Hasil Pengujian *Client*

Data hasil pengujian koneksi jaringan VPN antara *client* dan *server* adalah berupa respon waktu pada saat perintah dituliskan hingga sampai perintah dikirimkan kembali oleh *server* atau *client* yang dituju.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Henri>ping 10.8.0.6 -n 10

Pinging 10.8.0.6 with 32 bytes of data:

Reply from 10.8.0.6: bytes=32 time<1ms TTL=128

Ping statistics for 10.8.0.6:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
  
```

Diagram annotations in the image:

- Perintah Ping**: Points to the command `ping`.
- Alamat IP untuk Client/Server yang dituju**: Points to the IP address `10.8.0.6`.
- Jumlah Paket Data yang dikirimkan**: Points to the parameter `-n 10`.

Gambar 5.5 Tampilan *Command Prompt* sisi *client Virtual Private Network*

Sumber : Pengujian

d. Kesimpulan Hasil Pengujian

Dari hasil analisis mengacu pada Gambar 5.5, kesimpulan pengujian yang didapat adalah :

1. Dari pengujian koneksi jaringan *Virtual Private Network* didapatkan antar *client* dengan *server* terbentuk hubungan/jalur melalui *tunneling* yang telah dibentuk oleh *server*, sehingga dapat melakukan komunikasi dengan aman satu dengan lain.
2. Hubungan antar *client* dengan server dapat berjalan dengan baik karena pada hasil pengujian didapatkan bahwa bila respon waktu semakin rendah, maka kondisi jaringan semakin baik.

5.3 Pengujian Koneksi Jaringan *Video On Demand*

Pengujian ini bertujuan untuk mengetahui perangkat komunikasi dalam jaringan *Video On Demand* dapat berjalan dengan baik dengan cara mengirimkan paket dari *client* menuju komputer *server* kemudian dikirimkan kembali dalam jangka waktu yang telah ditentukan oleh komputer.

a. Peralatan Yang Digunakan Untuk Pengujian Koneksi Jaringan VOD

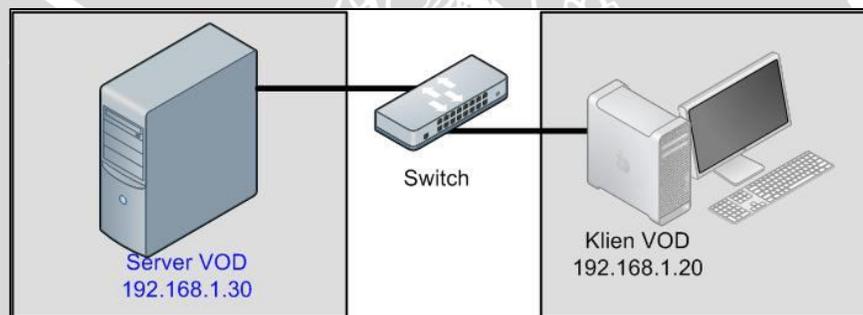
Peralatan yang digunakan dalam pengujian koneksi jaringan VOD antara lain:

1. 1 unit komputer sebagai *client*
2. 1 unit komputer sebagai *server*
3. 1 buah *Switch*
4. 2 buah NIC
5. 2 buah kabel LAN sepanjang 5 m dengan konektor RJ-45
6. Sistem operasi Window Xp SP 3

b. Prosedur Pengujian Koneksi Jaringan Video On Demand.

Prosedur pengujian koneksi jaringan VOD adalah sebagai berikut :

1. Mengatur alamat IP *server Video On Demand* dengan alamat 192.168.1.30.
2. Mengatur alamat IP *client Video On Demand* dengan alamat 192.168.1.20.



Gambar 5.6 Topologi Pengujian Koneksi Jaringan *Video On Demand*

Sumber : Pengujian

3. Menjalankan *command prompt* pada salah satu *client*.
4. Menjalankan perintah *ping* untuk menguji koneksi jaringan VOD berdasarkan respon waktu yang diperoleh.
5. Menjalankan perintah *ping* yang ditujukan pada *server VOD*.
6. Melakukan pengamatan terhadap respon waktu pada pengujian koneksi.

c. Hasil Pengujian Koneksi Jaringan Video On Demand

Data hasil pengujian koneksi jaringan VOD antara *client* dan *server* adalah respon waktu data uji pada saat perintah dituliskan hingga sampai perintah dikirimkan kembali oleh *server* atau *client* yang dituju. Berikut ini merupakan hasil pengujian koneksi jaringan *Video On Demand*.

```

C:\WINDOWS\system32\cmd.exe
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Henri>ping 192.168.1.30 -n 10
Pinging 192.168.1.30 with 32 bytes of data:
Reply from 192.168.1.30: bytes=32 time=2ms TTL=128
Reply from 192.168.1.30: bytes=32 time=1ms TTL=128
Ping statistics for 192.168.1.30:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
C:\Documents and Settings\Henri>
  
```

Diagram annotations in the image:

- Perintah Ping**: Points to the command `ping`.
- Alamat IP untuk Client/Server yang dituju**: Points to the IP address `192.168.1.30`.
- Jumlah Paket Data yang dikirimkan**: Points to the parameter `-n 10`.

Gambar 5.7 Tampilan *Command Prompt* Koneksi Jaringan VOD

Sumber : Pengujian

d. Kesimpulan Hasil Pengujian

Dari hasil analisis mengacu pada Gambar 5.7, kesimpulan pengujian yang didapat adalah

1. Dari pengujian koneksi jaringan *Video On Demand* didapatkan bahwa antar *client* dengan *server* sudah terbentuk hubungan/jalur sehingga dapat melakukan komunikasi satu dengan lain.
2. Hubungan antar *client* dengan *server* dapat berjalan dengan baik karena pada hasil pengujian didapatkan bahwa data yang dikirim sama dengan data yang diterima, jika respon waktu semakin rendah, maka kondisi jaringan semakin baik.

5.4 Pengujian Proses *Streaming* pada *Video On Demand*.

Pengujian ini bertujuan untuk mengetahui proses *streaming* pada *server Video On Demand* dapat berjalan dengan baik dengan cara melakukan *streaming* dengan menggunakan aplikasi Unreal Media Server sebagai *server Video On Demand* dan aplikasi Unreal Media Player sebagai aplikasi *streaming* pada sisi *client*.

a. **Peralatan Yang Digunakan Untuk Pengujian *Streaming VOD***

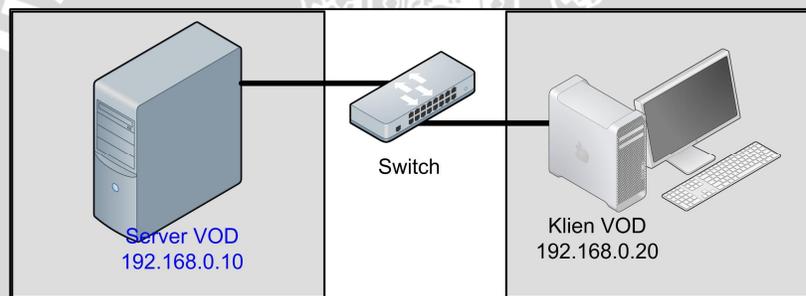
Peralatan yang digunakan dalam pengujian koneksi jaringan VOD antara lain:

1. 1 unit komputer sebagai *client*
2. 1 unit komputer sebagai *server*
3. 1 buah *Switch*
4. 2 buah NIC
5. 2 buah kabel LAN sepanjang 5 m dengan konektor RJ-45
6. Sistem operasi Window Xp SP 3

a. **Prosedur Pengujian *Streaming Video On Demand*.**

Prosedur pengujian koneksi jaringan VOD adalah sebagai berikut :

1. Mengatur alamat IP *server Video On Demand* dengan alamat 192.168.0.10.
2. Mengatur alamat IP *client Video On Demand* dengan alamat 192.168.0.20.



Gambar 5.8 Topologi Pengujian *Streaming Video On Demand*

Sumber : Pengujian

3. Menjalankan aplikasi Unreal Media Player pada *client*.



Gambar 5.9 Tampilan Unreal Media Player

Sumber : Pengujian

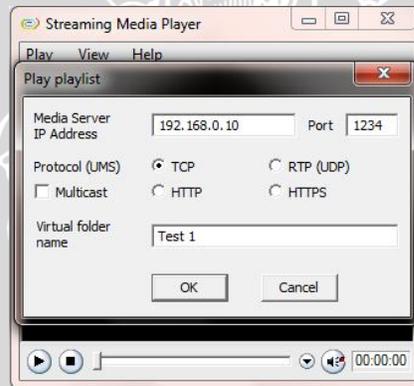
4. Kemudian klik “Play - Play Playlist..”



Gambar 5.10 *Setting Playlist*

Sumber : Pengujian

5. Pada dialog “Play Playlist”, isikan alamat IP *server video streaming* dan nomor *port* yang digunakan oleh *server*. Kemudian isikan nama pada *Virtual Folder* dengan “Test 1”



Gambar 5.11 *Setting IP Address*

Sumber : Pengujian

6. Kemudian klik Ok untuk memulai proses *streaming Video On Demand*.

c. Hasil Pengujian Proses Streaming Video On Demand

Hasil pengujian proses *streaming Video On Demand* yang dilakukan dari *Client* ke *server*, proses *streaming* berjalan tanpa adanya *buffering*. Sehingga pada sisi *client* didapatkan kualitas *video* sama seperti pada *server Video On Demand*.



Gambar 5.12 Hasil Pengujian Proses *streaming*

Sumber : Pengujian

d. Kesimpulan Hasil Pengujian

Dari hasil pengujian *streaming*, kesimpulan yang didapat adalah :

1. Dari pengujian koneksi jaringan *Video On Demand* didapatkan *client* dapat melakukan proses *streaming* dan dapat mengakses daftar putar dari *file video* yang telah disediakan oleh *server*.
2. Proses *streaming* pada sisi *client* dapat berjalan dengan baik tanpa adanya proses *buffering* dengan kualitas *file video* yang sama seperti pada *server*.

5.5 Pengujian Streaming Video On Demand Melalui OpenVPN

Pengujian ini bertujuan untuk mengetahui proses *streaming* pada sistem *Video On Demand* dapat berjalan pada sistem *Virtual Private Network* dengan menggunakan OpenVPN.

Pengujian ini bertujuan untuk mengetahui *client Virtual Private Network* sudah teregister kedalam *server Virtual Private Network* dengan benar sehingga dapat melakukan koneksi yang aman melalui *tunneling* yang telah dibentuk oleh *server Virtual Private Network*. Pada pengujian ini dilakukan pengamatan terhadap proses *streaming Video On Demand* pada OpenVPN.

a. Peralatan Pengujian

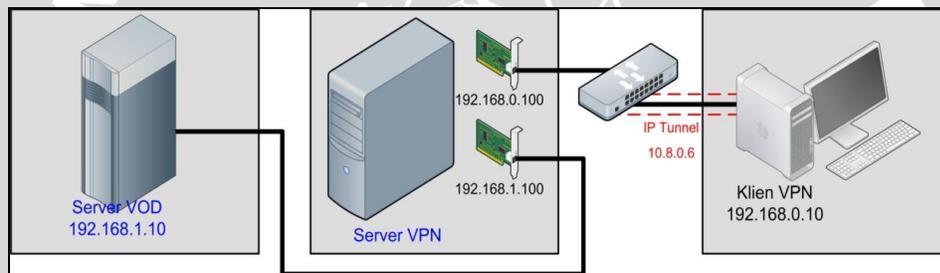
Peralatan yang digunakan dalam pengujian *Client* antara lain:

1. 1 unit komputer sebagai *client Video On Demand*
2. 1 unit komputer sebagai *server Video On Demand*

3. 1 unit komputer sebagai *server Virtual Private Network*
4. 1 *Switch* 5 port
5. 4 buah NIC
6. 3 buah kabel LAN sepanjang 5 m dengan konektor RJ-45
7. Sistem Operasi Linux Slackware 12
8. Sistem operasi Window XP SP3
9. Wireshark v.1.8

b. Prosedur Pengujian Streaming Video On Demand Melalui OpenVPN

1. Mengatur alamat IP *server Video On Demand* dengan alamat 192.168.1.10, IP server *Virtual Private Network* pada NIC pertama 192.168.1.100, dan IP server *Virtual Private Network* pada NIC kedua 192.168.0.100.
2. Mengatur alamat IP *client Virtual Private Network* dengan alamat 192.168.0.10.



Gambar 5.13 Topologi Pengujian Proses *Streaming* Melalui OpenVPN

Sumber : Pengujian

3. Menjalankan server *Virtual Private Network* dengan mengetikkan *command* `/usr/local/sbin# ./openvpn server.conf &`
4. Memasukkan sertifikat keamanan pada *client* yang telah terpasang aplikasi OpenVPN GUI 1.0.3.
5. Menjalankan aplikasi Open VPN GUI 1.0.3 sebagai aplikasi *Virtual Private Network* pada *client*.
6. Melakukan setting alamat IP *server Virtual private Network* pada aplikasi.
7. Melakukan koneksi ke *server Virtual Private Network* melalui aplikasi OpenVPN GUI 1.0.3.
8. Menjalan aplikasi Unreal Media Server pada *server Video On Demand* pada *client*.

9. *Setting* alamat IP *server Video On Demand* dan nomor *port* yang akan digunakan *server* dalam proses *streaming*.
10. Menentukan lokasi *file* dan nama *folder* yang akan diakses *client*.
11. Menjalankan aplikasi Unreal Media Player pada sisi *client*.
12. *Setting* alamat IP *server*, jenis protokol yang digunakan, nomor *port* dan nama *folder* yang akan diakses oleh *client*.
13. Melakukan proses *streaming*.

c. Hasil Pengujian Proses Streaming Video On Demand Pada OpenVPN

Hasil pengujian proses *streaming Video On Demand* yang dilakukan dari *client* ke *server*, proses *streaming* berjalan tanpa adanya *buffering*. Sehingga pada sisi *client* didapatkan kualitas *video* sama seperti pada *server Video On Demand*.



Gambar 5.14 Hasil Pengujian Proses *streaming* melalui OpenVPN

Sumber : Pengujian

d. Kesimpulan Hasil Pengujian

Dari hasil pengujian *streaming* melalui OpenVPN, kesimpulan yang didapat adalah :

1. Dari pengujian koneksi jaringan *Video On Demand* didapatkan *client* dapat melakukan proses *streaming* melalui OpenVPN dan dapat mengakses daftar putar dari *file video* yang telah disediakan oleh *server Video On Demand*.
2. Proses *streaming* pada sisi *client* dapat berjalan dengan baik tanpa proses *buffering* dan dengan kualitas *file video* yang sama seperti pada *server*.

5.6 Pengambilan Data

Pada skripsi ini, dipilih komunikasi searah, yaitu dari *server Video On Demand* menuju *client Video On Demand* melalui *server Virtual Private Network* menggunakan OpenVPN. Pada *server Video On Demand* juga disediakan *file video .mp4* dengan durasi 60 detik dan mempunyai resolusi yang bervariasi, mulai dari 320x200p, 352x288p dan 480x320p. Proses pengambilan data dimulai dengan jumlah *client* pengakses *video streaming* satu sampai dengan lima puluh buah *client* pada masing-masing resolusi *file video* yang telah disediakan *server*.

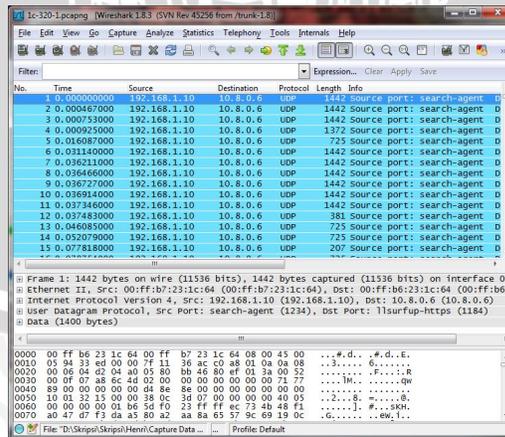
Wireshark mampu membaca paket-paket data yang lewat pada jaringan dan menganalisisnya. Aplikasi ini diinstall pada salah satu *client* yang akan melakukan proses *video streaming* dan melakukan analisis terhadap parameter performansi jaringan seperti *throughput*, *packet loss* dan *delay*. *User Datagram Protocol (UDP)* merupakan salah satu protokol yang didukung oleh aplikasi Wireshark. Untuk menggunakan Wireshark, pilih menu *Capture* lalu *Interface* hingga muncul daftar *Network Interface* seperti pada Gambar 5.15.



Gambar 5.15 Hasil Pengujian Proses *streaming* melalui OpenVPN

Sumber : Pengujian

Untuk menjalankan aplikasi Wireshark, pilih "Start" sesuai dengan *interface* yang dikehendaki, kemudian Wireshark akan membaca data seperti pada Gambar 5.16.



Gambar 5.16 Hasil Pembacaan Data pada Wireshark

Sumber : Pengujian

Untuk menghentikan pilih *Stop*. Kemudian simpan file dengan ekstensi *.pcap.

5.6.1 Jumlah Packet Data

Wireshark mampu untuk melakukan *capture* jumlah packet pada saat proses *streaming* berlangsung. Aplikasi wireshark di-*install* pada salah satu *client* untuk meng-*capture* paket data *video streaming*. Proses *capture packet* dilakukan pada masing-masing konfigurasi. Hasil *capture* data dapat dilihat pada Tabel-Tabel berikut :

Tabel 5.1 Data Jumlah Packet (Packet)

Jumlah Client	Resolusi 320x240p	Resolusi 352x288p	Resolusi 480x320p
1	4958	5194	5923
2	4949	5136	5946
3	4920	5198	5983
4	4931	5192	5922
5	4978	5235	5902
6	4967	5192	5974
7	4982	5186	5912
8	4971	5211	5972
9	4943	5196	5961
10	4958	5192	5924
11	4949	5136	5946
12	4921	5192	5983
13	4934	5202	5922
14	4971	5211	5972
15	4943	5196	5961
16	4958	5192	5924
17	4949	5179	5946
18	4921	5191	5983
19	4934	5192	5922
20	4978	5235	5902
21	4923	5195	5983
22	4931	5192	5922
23	4978	5235	5902
24	4967	5192	5974
25	4982	5186	5912
26	4971	5211	5972
27	4943	5196	5961
28	4958	5194	5923
29	4949	5220	5946
30	4982	5186	5912
31	4958	5194	5923
32	4949	5142	5946
33	4958	5194	5923
34	4921	5179	5983
35	4934	5192	5922
36	4943	5196	5961
37	4958	5194	5923
38	4949	5197	5946

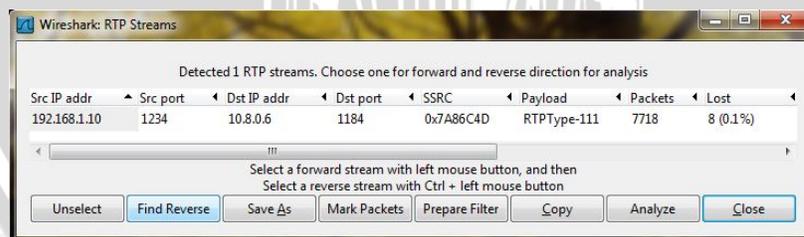
39	4958	5194	5923
40	4921	5182	5983
41	4934	5192	5922
42	4978	5235	5902
43	4923	5190	5983
44	4940	5192	5922
45	4943	5196	5961
46	4958	5194	5923
47	4949	5231	5946
48	4920	5196	5983
49	4955	5192	5922
50	4981	5278	5902
Rata-Rata	4950.58	5196.5	5941.72
Rata-Rata Total	5362.93		

Sumber : Pengujian

Pada MPEG-TS, setiap *frame* penyusun *video* akan dipecah menjadi paket-paket data dengan panjang maksimum 188 *byte*. Setiap paket data terdiri 4 *byte* TS *header* dan 184 *byte* *payload multimedia*. Semakin besar resolusi *file video*, maka semakin banyak paket data yang ditransmisikan. Sehingga probabilitas *packet loss* dan *jitter* semakin besar. Jumlah paket terbesar selama 60 detik diperoleh pada proses *streaming video* resolusi *file video* 480 x 320p dengan jumlah 5941.72 paket.

5.6.2 Throughput

Setelah melakukan *capture* paket data pada masing-masing konfigurasi kemudian hasil *capture* dilakukan *decode* menjadi paket RTP. Kemudian, pilih *Statistic - RTP - Show RTP Stream* sehingga diperoleh tampilan seperti Gambar 5.18.



Gambar 5.17 Hasil Pembacaan Data pada Wireshark

Sumber : Pengujian

Kemudian pilih trafiknya dan pilih *Analyze*, hingga muncul data *throughput* yang diinginkan. Dari data yang telah didapatkan, pilih "Save As CSV", kemudian simpan sebagai file CSV dan dihitung rata-ratanya menggunakan aplikasi *Spreadsheet* seperti MS Excel, lalu semua data *throughput* tersebut dihitung rata-ratanya.

Tabel 5.2 Data Throughput (kbps)

Jumlah Client	Resolusi 320x240p	Resolusi 352x288p	Resolusi 480x320p
1	392.32	444.26	585.19
2	392.96	444.42	586.61
3	392.58	443.87	586.53
4	392.89	444.58	586.74
5	392.84	444.32	587.14
6	392.78	444.53	586.57
7	393.03	444.43	586.53
8	392.85	444.85	586.71
9	392.38	444.60	586.35
10	393.61	444.47	586.38
11	392.81	444.69	586.54
12	392.92	444.81	586.30
13	392.74	445.32	586.23
14	393.49	444.98	586.53
15	392.88	445.18	586.82
16	392.79	444.61	587.38
17	393.18	444.35	586.51
18	393.18	444.55	587.26
19	392.85	444.93	586.85
20	392.97	444.97	586.29
21	392.84	444.61	587.29
22	394.26	444.65	586.82
23	392.89	445.35	586.58
24	392.92	444.81	586.89
25	393.47	445.27	586.83
26	393.45	444.73	587.69
27	392.70	444.95	586.32
28	392.98	445.39	586.65
29	393.72	444.23	587.72
30	392.38	444.52	586.88
31	392.84	444.61	587.29
32	394.26	444.65	586.82
33	392.89	445.35	586.58
34	392.92	444.81	586.89
35	393.47	445.27	586.83
36	393.45	444.73	587.69
37	392.70	444.95	586.32
38	392.98	445.39	586.65
39	393.72	444.23	587.72
40	392.38	444.52	586.88
41	393.77	445.83	587.43
42	392.87	445.18	586.82
43	393.33	444.24	586.93
44	393.58	444.67	586.88
45	392.97	444.86	587.32
46	392.91	444.89	587.23

47	392.83	445.32	586.78
48	393.48	444.67	586.88
49	393.59	445.32	586.93
50	392.98	445.41	587.38
Rata-Rata	393.072	444.803	586.828
Rata-Rata Total	474.9		

Sumber : Pengujian

Throughput dihitung berdasarkan banyaknya paket data yang diterima di sisi *client* dibagi dengan durasi pengiriman paket data. Semakin besar resolusi *file video*, semakin banyak paket data yang dikirimkan dan semakin besar pula nilai *throughput* yang dihasilkan. Pada hasil pengujian mengacu pada Tabel 5.2, diketahui *throughput* terbesar pada proses *streaming video* resolusi *file video* 460 x 640p dengan nilai rata-rata 586.83 kbps.

5.6.3 Packet Loss

Nilai *packet loss* diperoleh dari pembacaan *capture* data pada saat proses *streaming*. Dari hasil *capture* Wireshark, nilai *packet loss* dapat diperoleh dari masing-masing konfigurasi dan ditampilkan dalam Tabel-Tabel berikut ini :

Tabel 5.3 Data Packet Loss (packet)

Jumlah Client	Resolusi 320x240p	Resolusi 352x288p	Resolusi 480x320p
1	0	0	3
2	0	0	3
3	0	2	4
4	0	0	3
5	0	0	5
6	0	0	2
7	0	0	2
8	0	0	5
9	0	0	2
10	0	0	4
11	0	0	6
12	0	0	1
13	0	0	3
14	0	0	4
15	0	0	4
16	0	0	3
17	0	0	7
18	0	0	2
19	0	0	1
20	0	4	3
21	0	1	5
22	0	0	6

23	0	0	3
24	0	0	11
25	0	0	16
26	0	0	6
27	0	0	8
28	0	0	1
29	0	0	3
30	0	0	2
31	0	0	7
32	0	0	10
33	0	0	3
34	0	0	8
35	0	0	2
36	0	0	3
37	0	0	7
38	0	3	2
39	0	0	5
40	0	0	9
41	0	0	4
42	0	0	8
43	0	0	6
44	0	1	3
45	0	0	5
46	0	0	1
47	0	0	9
48	0	0	2
49	0	0	6
50	0	2	5
Rata-Rata	0	0.26	4.66
Rata-Rata Total	1.64		

Sumber : Pengujian

Dari hasil pengujian, dapat diketahui nilai *packet loss* rata-rata resolusi *file video* 320 x 240p sebesar 0 paket, *file video* 352 x 288p sebesar 0,26 paket dan *file video* 480 x 320p sebesar 4.66 paket.

5.7 Analisis

Analisis sistem dilakukan secara matematis dengan menggunakan persamaan-persamaan yang ada pada tinjauan pustaka. Analisis dilakukan terhadap parameter-parameter sistem yang digunakan dan hasil yang didapat dari proses pengujian. Pada skripsi ini analisa sistem terdiri dari delapan analisa, yaitu : Analisis *payload* data *multimedia*, analisis *throughput*, analisis *packet loss*, analisis *delay* enkapsulasi dan dekapsulasi, analisis *delay* transmisi, analisis *delay* propagasi, analisis *delay* antrian, analisis *delay jitter*, dan analisis *delay end-to-end*.

5.7.1 Analisis Payload data Multimedia

Pada skripsi ini kompresi yang digunakan adalah MPEG-4 dengan bitrate 256 kbps, sedangkan untuk kompresi *audio* menggunakan teknologi MP3 dengan bitrate 192 kbps. Packet data *audio* dan *video* dienkapsulasi menggunakan teknologi MPEG transport stream (MPEG TS).

Pada metode enkapsulasi MPEG TS, data *audio* dan *video* dienkapsulasi menjadi satu data *Multimedia* kemudian di pecah menjadi beberapa *transport stream* (TS) dengan nilai maksimum 188 *byte* yang terdiri dari *header* sebesar 4 *byte* dan *optional header* serta *payload* dengan panjang maksimum 184 *byte*. Dalam satu MTU, MPEG TS membatasi jumlah TS yang boleh dikirim adalah sebesar 7 *packet* TS. Dari sini kita bisa menghitung berapa besar MTU dan berapa *payload* data yang bisa ditampung dalam satu MTU. Nilai MTU didapat dari perhitungan sebagai berikut :

$$\begin{aligned} \text{MTU} &= (7 \times \text{paket}_{\text{MPEG TS}}) + \text{Header}_{(\text{Ethernet} + \text{IPv4} + \text{UDP})} \\ &= (7 \times 188 \text{ byte}) + (14 + 20 + 8) \text{ byte} \\ &= 1358 \text{ byte} \end{aligned}$$

Dari hasil perhitungan dapat kita lihat bahwa MPEG TS membatasi jumlah *packet* TS yang dikirim dalam satu MTU sebanyak 7 *packet* TS agar nilai MTU tidak melebihi nilai yang diperbolehkan, yaitu 1500 *byte*. Dengan panjang satu *packet* TS sebesar 188 *byte*, *header* TS sebesar 4 *byte* dan dalam satu MTU terdapat 7 *packet* TS maka jika diasumsikan *optional header* tidak ada, *payload* maksimum data *multimedia* dalam satu MTU adalah sebagai berikut :

$$\begin{aligned} \text{Payload} &= 7 \times (\text{Panjang paket}_{\text{MPEG TS}} - \text{Header}_{\text{MPEG TS}}) \\ &= 7 \times (188 - 4) \text{ byte} \\ &= 1288 \text{ byte} \end{aligned}$$

Sehingga *payload* maksimum data *multimedia* dalam satu MTU adalah sebesar 1288 *byte*.

5.7.2 Analisis Throughput

Throughput yang dimaksud adalah *throughput* yang membawa paket-paket *payload multimedia*. Dari analisa mengacu pada Tabel 5.2 didapatkan nilai rata-rata *throughput* :

- *File video* resolusi 320x240 mempunyai *throughput* 393.07 kbps
- *File video* resolusi 352x288 mempunyai *throughput* 444.8 kbps

- *File video* resolusi 480x320 mempunyai *throughput* 586.83 kbps sehingga dapat dilihat nilai *throughput* berbanding lurus dengan jumlah *client* dan resolusi *file video* yang dilakukan *streaming*.

5.7.3 Analisis Packet Loss

Packet loss adalah banyaknya paket data yang hilang pada saat proses transmisi. Semakin besar prosentase *packet loss*, maka semakin buruk performansi sebuah saluran transmisi sesuai persamaan (2.17).

Dari hasil pengujian mengacu pada Tabel 5.20 nilai rata-rata *packet loss* dalam 60 detik pada konfigurasi satu sampai empat *client* dengan *file video* resolusi 320x240 adalah 0 *packet*, pada resolusi 352x288 adalah 0.25 *packet*, dan pada resolusi 480x320 adalah 5.2 *packet*. dari data tersebut dapat diketahui besarnya *packet loss* sebagai berikut:

Rata-rata *packet loss* pada *file video* dengan resolusi 320 x 240p :

$$\begin{aligned} \text{Packet Loss} &= \frac{N_{\text{packet loss}}}{N_{\text{packet}} - N_{\text{packet loss}}} \times 100\% \\ &= \frac{0}{4950.58 - 0} \times 100\% = 0\% \end{aligned}$$

Rata-rata *packet loss* pada *file video* dengan resolusi 352 x 288p :

$$\begin{aligned} \text{Packet Loss} &= \frac{N_{\text{packet loss}}}{N_{\text{packet}} - N_{\text{packet loss}}} \times 100\% \\ &= \frac{0.26}{5196.5 - 0.26} \times 100\% = 0.005\% \end{aligned}$$

Rata-rata *packet loss* pada *file video* dengan resolusi 480 x 320p :

$$\begin{aligned} \text{Packet Loss} &= \frac{N_{\text{packet loss}}}{N_{\text{packet}} - N_{\text{packet loss}}} \times 100\% \\ &= \frac{4.66}{5941.7 - 4.66} \times 100\% = 0.078\% \end{aligned}$$

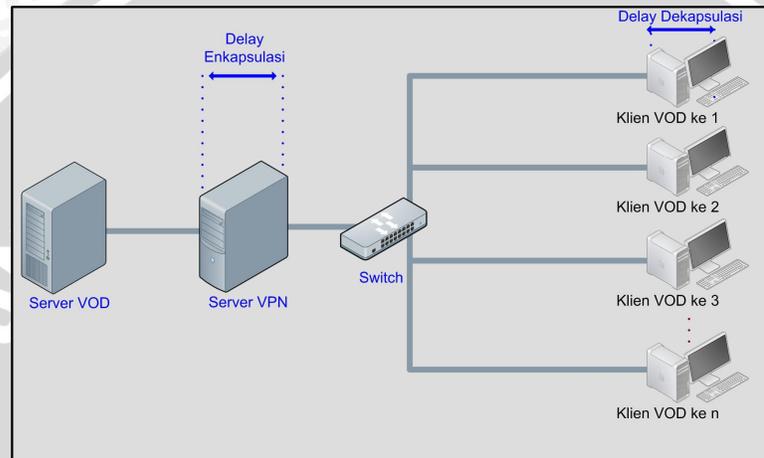
Rata-rata *packet loss* keseluruhan :

$$\begin{aligned} \text{Packet Loss} &= \frac{N_{\text{packet loss}}}{N_{\text{packet}} - N_{\text{packet loss}}} \times 100\% \\ &= \frac{1.64}{5362.93 - 1.64} \times 100\% = 0.031\% \end{aligned}$$

Dari analisa didapat prosentase *packet loss* untuk *file video* resolusi 320x240 adalah 0 %, pada resolusi 352x288 adalah 0.005 %, pada resolusi 480x320 adalah 0.078 % dan rata-rata *packet loss* total sebesar 0.031%. Mengacu kriteria *packet loss* sesuai Tabel 2.7, sistem ini dapat diaplikasikan karena mempunyai *packet loss* kurang dari 1 %.

5.7.4 Analisis Delay Enkapsulasi dan Dekapsulasi

Delay enkapsulasi merupakan waktu yang diperlukan untuk meletakkan header pada paket data yang akan dikirimkan.



Gambar 5.18 Delay Enkapsulasi dan Delay Dekapsulasi dalam sistem

Sumber : Pengujian

Secara matematis dapat ditulis berdasarkan persamaan (2.5) berikut ini :

$$t_{enc} = \frac{L_{Header\ UDP} + L_{Header\ IP} + L_{Header\ Eth} + L_{Header\ RTP} + L_{Header\ VPN}}{C_{pros}} \times 8$$

Dengan C_{pros} ,

$$C_{pros} = \frac{N_k}{T_k}$$

Keterangan:

- t_{enc} = delay enkapsulasi (s)
- $L_{Header\ UDP}$ = panjang header UDP (byte/paket)
- $L_{Header\ IP}$ = panjang header IP (byte/paket)
- $L_{Header\ Eth}$ = panjang header Ethernet (byte/paket)
- $L_{Header\ RTP}$ = panjang header RTP (byte/paket)
- $L_{Header\ VPN}$ = panjang header VPN (byte/paket)

- C_{pros} = kecepatan pemrosesan pada terminal pengirim (bps)
 N_k = jumlah total data yang dikirimkan (bit)
 T_k = waktu pengiriman total data (s)

Dari tinjauan pustaka didapatkan data sekunder diantaranya adalah :

- panjang *header* UDP ($L_{Header\ UDP}$) = 8 (*byte/paket*)
- panjang *header* IP ($L_{Header\ IP}$) = 20 (*byte/paket*)
- panjang *header Ethernet* ($L_{Header\ Eth}$) = 14 (*byte/paket*)
- panjang *header RTP* ($L_{Header\ RTP}$) = 12 (*byte/paket*)
- panjang *header VPN* ($L_{Header\ VPN}$) = 1 (*byte/paket*)
- jumlah total data yang dikirimkan = 5362 (paket) dengan setiap paket terdiri dari 188 byte maka total data = 1008056 byte = 8064448 bit
- waktu pengiriman total data = 60 (s)

sehingga perhitungan *delay* enkapsulasi diawali dengan perhitungan kecepatan pemrosesan pada terminal pengirim, sebagai berikut :

$$C_{pros} = \frac{N_k}{T_k}$$

$$C_{pros} = \frac{8064448}{60} = 134407.47 \text{ bps}$$

Jadi, *delay* enkapsulasi adalah

$$t_{enc} = \frac{L_{Header\ UDP} + L_{Header\ IP} + L_{Header\ Eth} + L_{Header\ RTP} + L_{Header\ VPN}}{C_{pros}} \times 8$$

$$t_{enc} = \frac{8 + 20 + 14 + 12 + 1}{134407.47} \times 8 = 3.273 \cdot 10^{-3} \text{ s} = 3.273 \text{ ms}$$

Nilai *delay* dekapsulasi ditentukan dengan persamaan (2.7) berikut ini:

$$t_{dec} = \frac{L_{Header\ UDP} + L_{Header\ IP} + L_{Header\ Eth} + L_{Header\ RTP} + L_{Header\ VPN}}{C_{pros}} \times 8$$

Dengan C_{pros2} ,

$$C_{pros2} = \frac{N_t}{T_t}$$

Keterangan:

- T_{dec} = *delay* dekapsulasi (s)
 $L_{Header\ UDP}$ = panjang *header* UDP (*byte/paket*)
 $L_{Header\ IP}$ = panjang *header* IP (*byte/paket*)

- $L_{\text{Header Eth}}$ = panjang *header Ethernet* (byte/paket)
 $L_{\text{Header RTP}}$ = panjang *header RTP* (byte/paket)
 $L_{\text{Header VPN}}$ = panjang *header VPN* (byte/paket)
 C_{pros} = kecepatan pemrosesan pada terminal penerima (bps)
 N_t = jumlah total data yang diterima (bit)
 T_t = waktu penerimaan total data (s)

Dari tinjauan pustaka didapatkan data sekunder diantaranya adalah :

- panjang *header UDP* ($L_{\text{Header UDP}}$) = 8 (byte/paket)
- panjang *header IP* ($L_{\text{Header IP}}$) = 20 (byte/paket)
- panjang *header Ethernet* ($L_{\text{Header Eth}}$) = 14 (byte/paket)
- panjang *header RTP* ($L_{\text{Header RTP}}$) = 12 (byte/paket)
- panjang *header VPN* ($L_{\text{Header VPN}}$) = 1 (byte/paket)
- jumlah total data yang dikirimkan = 5362 (paket) dengan setiap paket terdiri dari 188 byte maka total data = 1008056 byte = 8064448 bit
- waktu pengiriman total data = 60 (s)

sehingga perhitungan *delay* dekapsulasi diawali dengan perhitungan kecepatan pemrosesan pada terminal pengirim, sebagai berikut :

$$C_{\text{pros2}} = \frac{N_k}{T_k}$$

$$C_{\text{pros}} = \frac{8064448}{60} = 134407.47 \text{ bps}$$

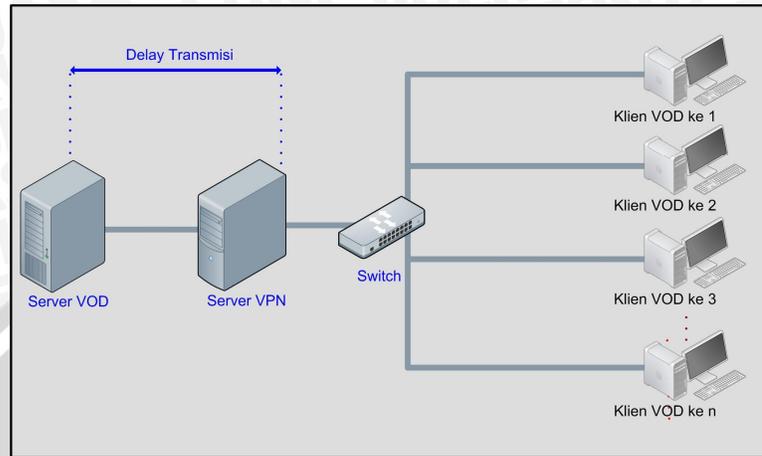
Jadi, *delay* dekapsulasi adalah

$$t_{\text{dec}} = \frac{L_{\text{Header UDP}} + L_{\text{Header IP}} + L_{\text{Header Eth}} + L_{\text{Header RTP}} + L_{\text{Header VPN}}}{C_{\text{pros}}} \times 8$$

$$t_{\text{dec}} = \frac{8 + 20 + 14 + 12 + 1}{134407.47} \times 8 = 3.273 \cdot 10^{-3} \text{ s} = 3.273 \text{ ms}$$

5.7.5 Analisis Delay Transmisi

Delay transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket data ke media transmisi



Gambar 5.19 Delay Transmisi dalam sistem

Sumber : Pengujian

delay transmisi dapat dihitung dengan persamaan (2.9),

$$t_t = \frac{(L + L')}{C} \times 8 = \frac{W_{frame}}{C} \times 8$$

Keterangan:

- t_t = delay transmisi (s)
- L = panjang paket data (byte/paket)
- L' = panjang header (byte/paket)
- C = kecepatan media transmisi (bps)
- W_{frame} = panjang frame Ethernet (byte/paket)

Dari proses pengukuran didapatkan data primer diantaranya adalah :

- panjang paket data (L) = 188 bytes = 1504 bit
- panjang header = 55 bytes = 440 bit
- kecepatan media transmisi = 100 (Mbps) = 104857600 bps

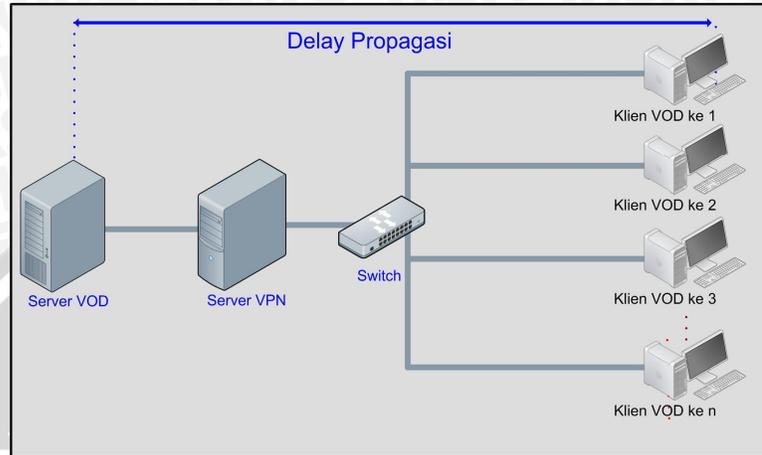
Jadi, delay transmisi dapat dihitung sebagai berikut :

$$t_t = \frac{(L + L')}{C} \times 8 = \frac{W_{frame}}{C} \times 8$$

$$t_t = \frac{1944}{104857600} \times 8 = 0.1483 \cdot 10^{-3} \text{ s} = 0.1483 \text{ ms}$$

5.7.6 Analisis Delay Propagasi

Delay propagasi (*propagation delay*) adalah waktu yang dibutuhkan untuk merambatkan paket data melalui media transmisi UTP (*unshield twisted pair*) dari *host* sumber ke *host* tujuan,



Gambar 5.20 Delay Propagasi dalam sistem

Sumber : Pengujian

delay propagasi dihitung dengan persamaan (2.10).

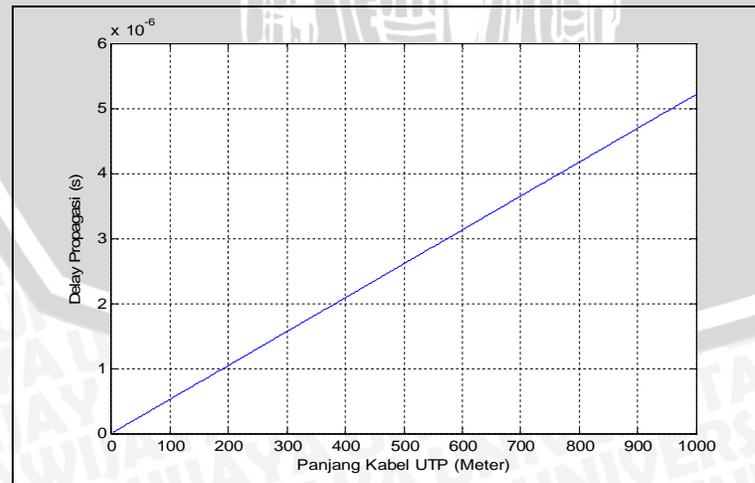
$$t_p = \frac{L_k}{V_{prop}}$$

Keterangan:

t_p = Delay propagasi (s)

L_k = Panjang Kabel (m)

V_{prop} = Kecepatan sinyal melalui kabel UTP (m/s)



Gambar 5.21 Pengaruh Panjang kabel UTP terhadap Delay Propagasi

Sumber : Simulasi

Dari proses pengukuran didapatkan data primer diantaranya adalah :

- Panjang kabel UTP (L_k) = 300 m
- Data sekunder dari *literature* diantaranya adalah :
- Kecepatan sinyal melalui kabel UTP = $0.64 \times c$
- c adalah kecepatan rambat cahaya = $3 \cdot 10^8$ m/s

Nilai *delay* propagasi pada jarak kabel 100 m dapat dihitung sebagai berikut :

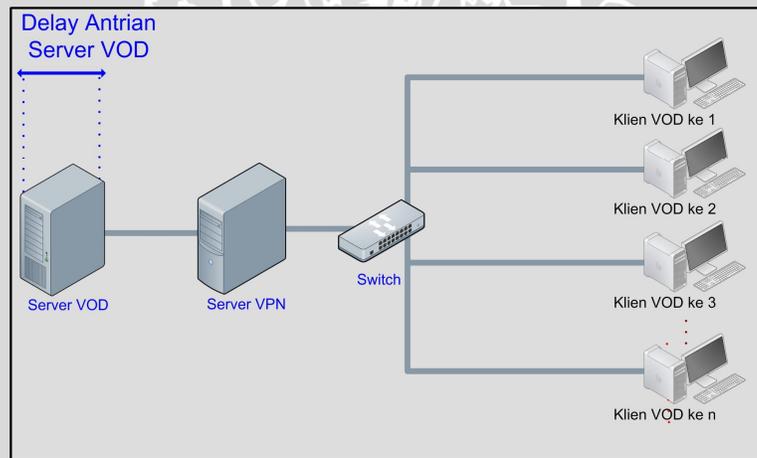
$$t_p = \frac{L_k}{V_{prop}}$$

$$t_p = \frac{300}{1.92 \cdot 10^8} = 1.5625 \cdot 10^{-6} \text{ ms}$$

5.7.7 Analisis Delay Antrian

a. Analisis Delay Antrian Server Video On Demand

Delay antrian adalah waktu di mana paket data berada dalam antrian untuk diproses oleh *server*.



Gambar 5.22 Delay Antrian Server VOD dalam sistem

Sumber : Pengujian

Delay antrian pada *server Video On Demand* dihitung dengan menggunakan model antrian M/M/1. Sesuai dengan persamaan (2.15), secara matematis *delay* antrian adalah :

$$E(T) = \frac{1}{\mu - \lambda_p}$$

Dimana,

$$\lambda_p = \frac{N}{T}$$

$$\mu = \frac{C}{L_t}$$

Keterangan :

$E(T)$ = delay antrian pada server Video On Demand (s)

λ_p = kecepatan kedatangan paket pada server (paket/s)

μ = kecepatan pelayanan server (paket/s)

N = total paket yang dikirim (paket)

T = waktu pengiriman paket total (s)

C = kapasitas kanal (bps)

L_t = panjang paket data (bit/paket)

Dari proses pengukuran didapatkan data primer diantaranya adalah :

- Total paket data yang dikirimkan (N) = 5362.93 (paket)
- Waktu pengiriman paket total (T) = 60 (s)
- Panjang paket data (L_t) =

$$L_{Header\ UDP} + L_{Header\ IP} + L_{Header\ Eth} + L_{Header\ RTP} + L_{Header\ TS} + Payload =$$

$$8 + 20 + 14 + 12 + 4 + 184 = 242 \text{ (byte/paket)} = 1936 \text{ (bit/paket)}$$

- Kapasitas kanal (C) = 104857600 (bps)

Sehingga perhitungan delay antrian diawali dengan perhitungan :

1. Kecepatan kedatangan paket pada server (λ_p) sebagai berikut :

$$\lambda_p = \frac{N}{T}$$

$$\lambda_p = \frac{5362.93}{60} = 89.38 \text{ paket / s}$$

2. Kecepatan pelayanan server (μ) sebagai berikut :

$$\mu = \frac{C}{L_t}$$

$$\mu = \frac{104857600}{1936} = 54161.98 \text{ paket / s}$$

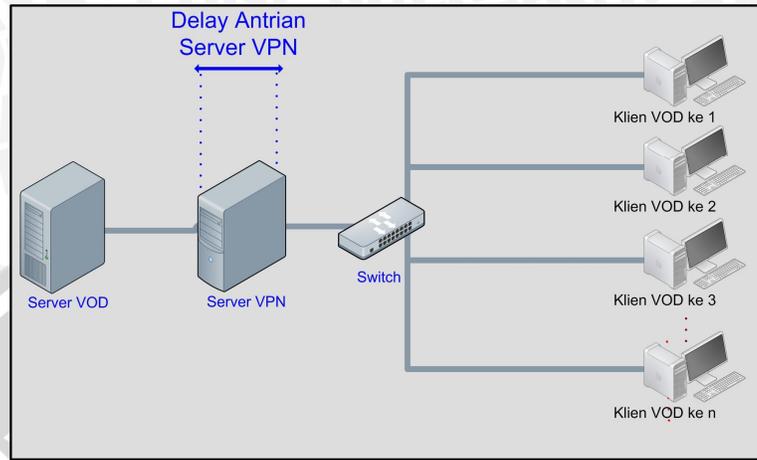
Jadi, delay antrian dapat dihitung sebagai berikut :

$$E(T) = \frac{1}{\mu - \lambda_p}$$

$$E(T) = \frac{1}{54161.98 - 89.38} = \frac{1}{54072.59} = 0.01849 \cdot 10^{-3} \text{ s} = 0.01849 \text{ ms}$$

b. Analisis Delay Antrian Server Virtual Private Network

Delay antrian adalah waktu di mana paket data berada dalam antrian untuk diproses oleh server.



Gambar 5.23 Delay Antrian Server VPN dalam sistem

Sumber : Pengujian

Delay antrian pada server Virtual Private Network dihitung dengan menggunakan model antrian M/M/1. Sesuai dengan persamaan (2.15), secara matematis delay antrian adalah :

$$E(T) = \frac{1}{\mu - \lambda_p}$$

Dimana,

$$\lambda_p = \frac{N}{T}$$

$$\mu = \frac{C}{L_t}$$

Keterangan :

$E(T)$ = delay antrian pada server Virtual Private Network (s)

λ_p = kecepatan kedatangan paket pada server (paket/s)

μ = kecepatan pelayanan server (paket/s)

N = total paket yang dikirim (paket)

T = waktu pengiriman paket total (s)

C = kapasitas kanal (bps)

L_t = panjang paket data (bit/paket)

Dari proses pengukuran didapatkan data primer diantaranya adalah :

- Total paket data yang dikirimkan (N) = 5362.93 (paket)
- Waktu pengiriman paket total (T) = 60 (s)

- Panjang paket data (L_t) =
 $L_{Header\ UDP} + L_{Header\ IP} + L_{Header\ Eth} + L_{Header\ RTP} + L_{Header\ VPN} + L_{Header\ TS} + Payload$
 $8 + 20 + 14 + 12 + 1 + 4 + 184 = 243 \text{ (byte/paket)} = 1944 \text{ (bit/paket)}$
- Kapasitas kanal (C) = 104857600 (bps)

Sehingga perhitungan *delay* antrian diawali dengan perhitungan :

- Kecepatan kedatangan paket pada *server* (λ_p) sebagai berikut :

$$\lambda_p = \frac{N}{T}$$

$$\lambda_p = \frac{5362.93}{60} = 89.38 \text{ paket / s}$$

- Kecepatan pelayanan *server* (μ) sebagai berikut :

$$\mu = \frac{C}{L_t}$$

$$\mu = \frac{104857600}{1944} = 53939.094 \text{ paket / s}$$

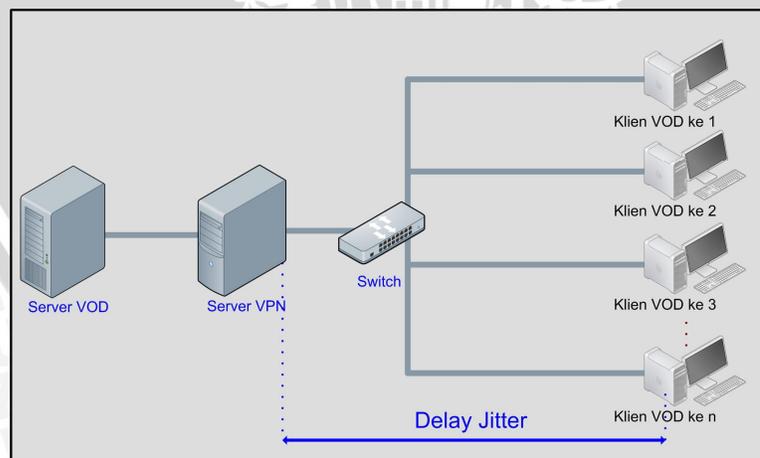
Jadi, *delay* antrian dapat dihitung sebagai berikut :

$$E(T) = \frac{1}{\mu - \lambda_p}$$

$$E(T) = \frac{1}{53939.094 - 89.38} = \frac{1}{53849.71} = 0.01857 \cdot 10^{-3} \text{ s} = 0.01857 \text{ ms}$$

5.7.8 Analisis Delay Jitter

Delay jitter disebabkan oleh kedatangan paket yang acak karena setiap paket melewati jalur yang berbeda-beda pada jaringan (Mischa Schwartz, 1987).



Gambar 5.24 *Delay Jitter* dalam sistem
Sumber : Pengujian

delay jitter rata-rata ditentukan dengan persamaan (2.16).

$$\theta = \frac{1}{N_{packet} - 1} \sum_{k=0}^{N_{packet}-1} t_{end-to-end(n+1)} - t_{end-to-end(n)}$$

$$= \frac{t_v}{N_{packet} - 1} \quad (2.16)$$

Dengan :

- θ = delay jitter rata-rata (s)
- t_v = waktu transmisi (s)
- $t_{end-to-end}$ = delay end to end pada urutan ke-n (s)
- $t_{end-to-end(n+1)}$ = delay end to end pada urutan ke-(n+1) (s)
- N_{packet} = jumlah paket multimedia yang diterima

Dari hasil pengambilan data primer mengacu pada Tabel 5.1, dapat diketahui nilai jitter rata-rata :

- Jitter pada resolusi file video 320 x 240p :

$$\theta = \frac{t_v}{N_{packet} - 1} = \frac{60}{4950.58 - 1} = 12.1ms$$

- Jitter pada resolusi file video 352 x 288p :

$$\theta = \frac{t_v}{N_{packet} - 1} = \frac{60}{5196.5 - 1} = 11.5ms$$

- Jitter pada resolusi file video 480 x 320p :

$$\theta = \frac{t_v}{N_{packet} - 1} = \frac{60}{5941.72 - 1} = 10ms$$

- Rata-rata Jitter total :

$$\theta = \frac{t_v}{N_{packet} - 1} = \frac{60}{5941.72 - 1} = 11.1ms$$

mengacu pada Tabel (2.6), sistem video on demand melalui OpenVPN dapat diterapkan karena mempunyai jitter kurang dari 100ms.

5.7.9 Analisis Delay End-to-end

Delay end-to-end merupakan nilai *delay* total dari *delay* yang ada dihitung dari *host* sumber sampai *host* tujuan. Sesuai dengan persamaan (2.4) besar *delay end-to-end* adalah :

$$t_{end-to-end} = t_{codec} + t_{enc} + t_i + t_p + t_w + t_{dec} + t_{jitter} + t_{codec}$$

1. t_{codec} adalah waktu yang dibutuhkan untuk memproses paket data *Multimedia* digital menjadi *Payload* data MPEG. Menurut referensi untuk metode *encoding* MPEG, *delay codec* sebesar 100 ms (Vigato,2006:52).
2. t_{enc} dan t_{dec} didapat dari analisis hasil pengujian sebesar 6.546 ms
3. t_i *delay* transmisi didapat dari analisis hasil pengujian sebesar 0.1483 ms
4. t_p *delay* propagasi didapat dari simulasi mulai dari panjang kabel 100m sampai dengan 1000m sebesar $5.21 \cdot 10^{-7}$ ms sampai dengan $5.21 \cdot 10^{-6}$ ms
5. t_w *delay* antrian server VOD dan *delay* antrian server VPN didapat dari analisis hasil pengujian sebesar 0.0185 ms dan 0.0186 ms.
6. t_{jitter} *delay jitter* didapat dari rata-rata hasil pengujian sebesar 11.1 ms

Tabel 5.4 Perbandingan Panjang Kabel Dengan Nilai *Delay End To End*

Jarak Kabel UTP (Meter)	<i>Delay End To End</i> (ms)
100	117.8314005
200	117.831401
300	117.8314016
400	117.8314021
500	117.8314026
600	117.8314031
700	117.8314036
800	117.8314042
900	117.8314047
1000	117.8314052

Sumber : Analisis

nilai *delay end-to-end* pada kabel UTP dengan panjang 1000 meter adalah $100\text{ms} + 6.546\text{ms} + 0.1483\text{ms} + 0.0000521\text{ms} + 0.0371\text{ms} + 11.1\text{ms} = 117.8314052$ ms.

Berdasarkan kriteria *delay* rekomendasi ITU-T G.1010 sesuai Tabel 2.2, sistem *Video On Demand* pada *Virtual Private Network* dengan menggunakan OpenVPN layak untuk diterapkan karena nilai *delay end-to-end* masih kurang dari 10 s.

