

APLIKASI VIDEO CONFERENCE DENGAN KEMAMPUAN BEROPERASI PADA IPV4 DAN IPV6

Andreas Handoyo¹, Robin Chandra², Justinus Andjarwirawan¹

1Jurusan Teknik Informatika, 2 Alumni Jurusan Teknik Informatika

Fakultas Teknologi Industri Universitas Kristen Petra

Jalan Siwalankerto 121-131, Surabaya

E-mail: handoyo@petra.ac.id

ABSTRAK

Dengan semakin berkembangnya teknologi informasi dan komunikasi. Saat ini komunikasi dapat dilakukan secara elektronik misalnya melalui e-mail, instant messaging, sambungan telepon melalui internet, dan juga video conference. Kelebihan dari video conference adalah pengguna selain dapat berkomunikasi juga dapat secara langsung melihat lawan bicaranya sehingga seakan-akan bertatap muka secara langsung. Untuk melakukan video conference sangatlah mudah dimana pengguna cukup memiliki web camera, speaker dan microphone. Data dari web camera maupun microphone akan di-encode dan kemudian dikirimkan dengan menggunakan Realtime Transport Protocol. Penerima akan men-decode data dan menampilkan hasil yang berupa gambar yang akan tampak pada layar penerima dan suara yang akan terdengar melalui speaker. Pada penelitian ini pembuatan aplikasi video conference, akan menggunakan software Microsoft Visual C# 2005.

Kata Kunci: video conference, streaming, multicasting

1. LATAR BELAKANG

Saat ini, komunikasi adalah bagian yang tak dapat dipisahkan dalam kehidupan manusia. Seiring dengan perkembangan teknologi, bentuk dan media untuk berkomunikasi juga mengalami perubahan. Sebelum ditemukan komputer, komunikasi biasanya diadakan secara tatap muka ataupun melalui surat. Setelah ditemukannya komputer, terlebih teknologi internet, media komunikasi mengalami banyak perubahan. Pada awal perkembangan teknologi internet, user komputer mulai menggunakan surat elektronik (e-mail). Namun, hal ini tidak memfasilitasi user yang ingin berkomunikasi secara langsung.

Namun, semakin berkembangnya teknologi telekomunikasi yang semakin pesat saat ini, dengan tersedianya jaringan broadband yang mampu membawa data dalam jumlah besar, dikembangkan bentuk komunikasi yang baru, yaitu koneksi telepon melalui internet (Voice over IP atau VoIP) serta video conference, yang memungkinkan user komputer untuk dapat melakukan pertemuan dari tempat yang berbeda. VoIP memungkinkan user komputer untuk seolah-olah menelepon user yang lain, sementara dengan video conference, user komputer dapat melihat wajah serta mendengarkan suara dari user yang lain, sehingga seolah-olah komunikasi berlangsung secara tatap muka. Pada penelitian ini akan dibuat aplikasi video conference yang mampu beroperasi pada IPv4 dan IPv6, sehingga diharapkan dapat memberikan pelayanan pada user yang memiliki jaringan yang beroperasi pada IPv4 dan IPv6.

2. VIDEOCONFERENCING

Menurut Gough (2006), videoconferencing dapat dibagi menjadi 3 jenis, yaitu :

1. *Personal videoconferencing*, melibatkan dua orang yang berinteraksi satu sama lain. Dalam *personal videoconferencing*, terdapat komunikasi yang berupa video dan audio antara dua orang yang berinteraksi. Dapat ditambahkan fitur tambahan berupa pengiriman teks, seperti yang terdapat pada kebanyakan perangkat lunak *instant messaging* (IM), seperti Windows Live Messenger dan Yahoo! Messenger.
2. *Business videoconferencing*, memiliki fitur yang sama dengan *personal videoconferencing* ditambah dengan beberapa fitur seperti:
 - Kemampuan untuk berkomunikasi tidak hanya antara dua orang, namun bisa lebih
 - Fitur untuk berbagi file (*file sharing*)
 - Kemampuan untuk melakukan presentasi
 - Fasilitas *whiteboard* dan fitur-fitur lainnya*Business videoconferencing* ini membutuhkan biaya yang lebih besar dibandingkan dengan *personal videoconferencing*, sebagai akibat dari fasilitas-fasilitas yang disediakan.
3. *Web videoconferencing*, yaitu *video call* yang terdapat pada sebuah halaman web. Biasanya *web videoconferencing* ini digunakan pada seminar yang menggunakan web, dimana pemirsa dapat melihat video yang dikirimkan oleh pembicara seminar. Oleh karena itu, *web videoconferencing* merupakan komunikasi satu arah, karena pemirsa tidak dapat mengirimkan videonya kepada si pengirim.

2.1 Streaming

Streaming adalah sebuah teknik yang digunakan untuk melakukan transfer data sehingga dapat diproses secara tetap dan kontinyu (Austerberry, 2005). Teknologi *streaming* berkembang sesuai dengan perkembangan *internet*, dimana kebanyakan *user internet* masih belum memiliki koneksi *broadband* untuk mengunduh file multimedia yang berukuran besar dengan cepat. Namun, sumber yang digunakan untuk *streaming* tidak selalu dalam bentuk file multimedia, bisa juga dari peralatan multimedia seperti *webcam*, kamera televisi, dan lain sebagainya. *Streaming* biasanya diidentikkan dengan *realtime*. Namun, tidak dapat dipungkiri bahwa setiap media yang digunakan untuk melakukan *streaming* memiliki *latency*. Televisi memiliki *latency* dalam hitungan milidetik. Dengan menggunakan *high compression codec*, *latency* yang ditimbulkan bisa dalam hitungan detik. Faktor utama yang menyebabkan *streaming* bersifat *realtime* adalah tidak adanya media penyimpanan yang digunakan untuk menyimpan paket data. Paket data akan disimpan pada sebuah *buffer* dan kemudian ditampilkan ke layar. Setelah selesai, data pada *buffer* akan dibuang dan *buffer* digunakan untuk menyimpan data yang baru. Walaupun demikian, masih tetap dimungkinkan jika data *streaming* ingin disimpan pada media penyimpanan seperti *harddisk*. Selain itu, data *streaming* juga dapat dikirimkan melalui jaringan, seperti pada aplikasi VoIP dan *video conference*.

Proses *streaming* melalui jaringan dapat digambarkan seperti terlihat pada gambar 1 :



Gambar 1. Proses *Streaming* Melalui Jaringan

Pada awalnya, data dari *source* (bisa berupa audio maupun video) akan di-*capture* dan disimpan pada sebuah *buffer* yang berada pada memori komputer (bukan media penyimpanan seperti *harddisk*) dan kemudian di-*encode* sesuai dengan format yang diinginkan. Dalam proses *encode* ini, *user* dapat mengompresi data sehingga ukurannya tidak terlalu besar (bersifat *optional*). Namun pada aplikasi *streaming* menggunakan jaringan, biasanya data akan dikompresi terlebih dahulu sebelum dilakukan *streaming*, karena keterbatasan *bandwidth* jaringan. Setelah di-*encode*, data akan di-*stream* ke *user* yang lain. *User* akan melakukan *decode* data dan menampilkan hasilnya ke layar *user*. Waktu yang dibutuhkan agar sebuah data sampai mulai dari pemancar sampai penerima disebut dengan *latency*.

2.2 Video

Untuk *video conference*, digunakan *webcam* sebagai data sumber yang akan dikirimkan.

Webcam memiliki resolusi pengambilan gambar, dan resolusi antar satu *webcam* dengan *webcam* yang lain dapat bervariasi. Dahulu, *webcam* masih memiliki resolusi yang kecil, misalnya 160x120. Namun sekarang sudah ada *webcam* yang memiliki resolusi beberapa *megapixel*. Semakin besar ukuran resolusi, semakin besar pula jumlah data yang dikirimkan, sehingga *bandwidth* yang diperlukan juga semakin besar. Oleh karena itu, jarang sekali dilakukan *conference* dengan ukuran resolusi yang besar. Umumnya ukuran resolusi yang digunakan untuk *video conference* adalah 320x240.

Selain itu, hal yang berpengaruh pada ukuran data adalah *frame rate*. *Frame rate* adalah jumlah gambar yang dikirimkan tiap detik. Misalkan ukuran gambar 320x240 dengan 30 *frame per second* (fps), jumlah piksel yang dikirimkan tiap detiknya adalah $320 \times 240 \times 30 = 2.304.000$ piksel. Jika *frame rate* 15 fps, jumlah piksel yang dikirimkan tiap detiknya berkurang drastis menjadi 1.152.000 piksel, dengan demikian dapat menghemat *bandwidth*. Namun jika *frame rate* diturunkan, video yang dihasilkan tidak akan lancar seperti video dengan *frame rate* yang tinggi.

2.3 Audio

Untuk video conference, digunakan sebuah microphone untuk input audio. Sama halnya dengan data video, terdapat faktor yang dapat mempengaruhi ukuran data yang dikirimkan, misalnya *sampling rate* (dalam satuan kHz) dan jumlah channel. Pada umumnya, ukuran data audio yang dikirimkan melalui streaming ini lebih kecil dibandingkan dengan data video. Sebuah data audio yang tidak dikompres menghasilkan data sebesar 5 *megabyte* per *channel* per menit. Tetapi, masih dimungkinkan jika *input* dari *device* ingin dikompres sehingga lebih menghemat *bandwidth* yang ada.

2.4 Codec

Codec memiliki beberapa definisi, antara lain (Wikipedia, 2009) :

1. *Coder-decoder*, sebuah perangkat keras yang berguna untuk melakukan konversi data analog ke digital atau digital ke analog.
2. *Compressor-decompressor*, yaitu perangkat keras atau perangkat lunak yang digunakan untuk melakukan kompresi dan dekompresi terhadap data video dan audio.

Ada 2 jenis *codec*, yaitu:

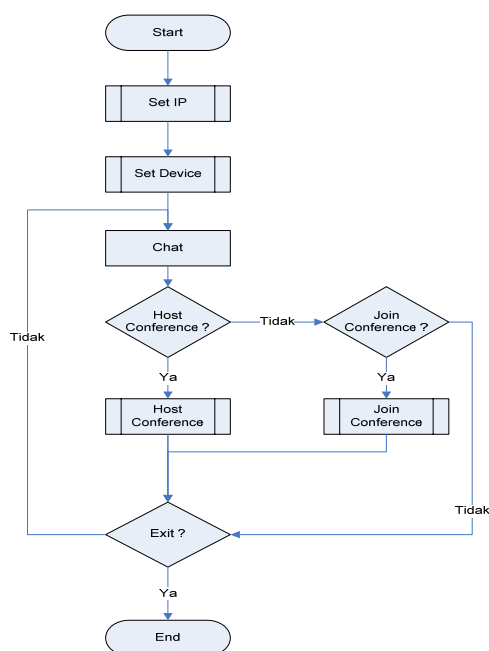
- *Lossy codec*: Banyak *codec* yang populer termasuk ke dalam kategori ini, dimana *codec* ini akan mengurangi kualitas data untuk mengurangi ukuran data (kompresi). Biasanya *codec* ini digunakan untuk menyimpan data pada media penyimpanan yang berukuran terbatas seperti CD-ROM dan DVD. *Lossy codec* ini juga biasanya

digunakan untuk *streaming*, karena *bandwith* jaringan yang terbatas. Contoh: Windows Media Video, H.264.

- *Lossless codec*: Selain *lossy codec*, terdapat juga beberapa *lossless codec*, dimana pada *lossless codec* ini, kualitas data yang dihasilkan tidak akan berkurang. Konsekuensinya adalah ukuran data yang dihasilkan oleh *lossless codec* ini akan lebih besar dibandingkan dengan *lossy codec*. Biasanya *lossless codec* ini digunakan pada video yang masih memerlukan *editing*, karena dalam proses *editing* dilakukan *encode-decode* berulang kali, sehingga jika menggunakan *lossy codec*, kualitas video akan jauh menurun dibandingkan dengan video aslinya. Contoh: CorePNG, huffyuv, Apple Lossless Audio Codec.

3. FLOWCHART SISTEM

Cara kerja dari sistem secara garis besar dapat dilihat pada Gambar 2, dimana pada awalnya, sistem akan membuat daftar dari semua *network interface* yang terdapat pada komputer *user* dan menanyakan kepada *user* mengenai jenis IP yang ingin digunakan, apakah IPv4 atau IPv6. Sistem akan menggunakan jenis IP sesuai dengan pilihan *user*, dan jenis IP ini tidak dapat diubah selama program berjalan. Setelah itu, sistem akan mendeteksi semua *input device* yang berupa *webcam* dan *microphone* yang terdapat pada komputer *user*, serta membuat daftar semua *compressor* yang ada, baik *compressor* video maupun *compressor* audio.



Gambar 2. Flowchart sistem

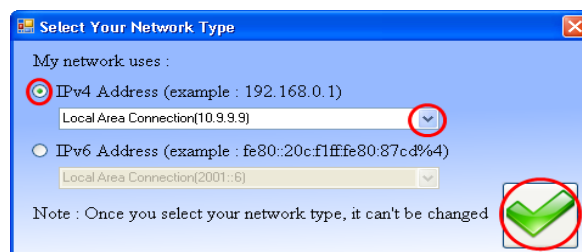
Kemudian, *user* akan bergabung pada sebuah *chat room* dimana semua *user* aplikasi akan muncul pada daftar *online user*. Di sini *user* dapat berinteraksi satu sama lain, baik hanya untuk sekedar berkomunikasi maupun mengajak *user* lain untuk melakukan *video conference*. *User* yang membuat sebuah konferensi disebut juga dengan *moderator*. *Moderator* berhak untuk menerima permintaan *user* lain untuk bergabung ke dalam konferensi maupun menolaknya. Selain itu, *moderator* juga berhak untuk menentukan siapa saja yang boleh berbicara pada sebuah konferensi serta mengeluarkan *user* dari konferensi tersebut.

4. IMPELEMENTASI SISTEM

Impelementasi dilakukan dengan menggunakan software Microsoft Visual C# 2005 merupakan *Integrated Development Environment (IDE)* yang berbasiskan .NET Framework 2.0 SP1. Microsoft Visual C# menyediakan *Graphical User Interface (GUI)* yang mudah didesain dan memiliki banyak *library* yang sangat membantu dalam pembuatan sistem. Selain itu, digunakan beberapa *library* tambahan seperti:

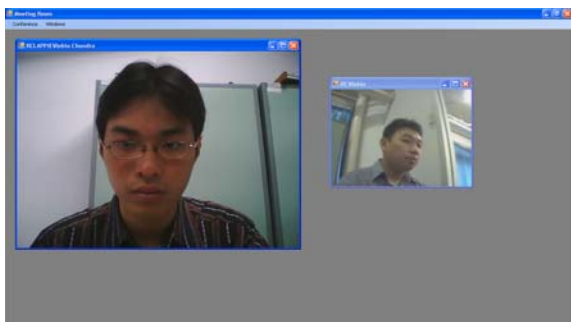
- Microsoft Managed DirectShow
- ConferenceXP RTP Filter
- Microsoft Visual C++ 2005 Redistributable
- Windows Media Player 11 Codec

Pertama kali, *user* harus memilih jenis IP yang akan digunakan (IPv4 atau IPv6) dengan memilih *radio button*. Untuk memilih IP *address*, *user* dapat mengklik *combo box* dan memilih IP yang terdapat pada komputer. Setelah itu, *user* menekan tombol centang yang ada di kanan bawah (Gambar 3).



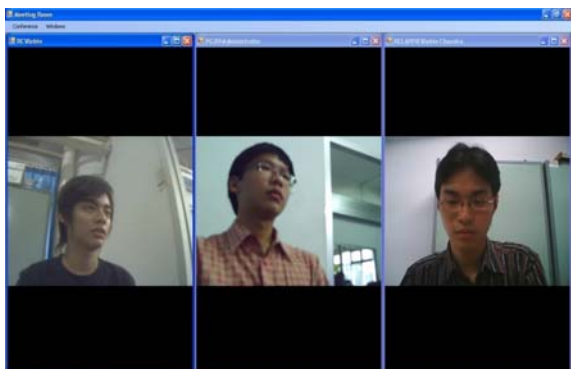
Gambar 3. Pemilihan IP address

Jika *user* diperbolehkan untuk bergabung ke dalam konferensi, akan muncul *form set destination IP and port*. Setelah *user* mengisi alamat IP dan nomor *port*, *user* akan bergabung ke dalam konferensi, seperti terlihat pada gambar 4.



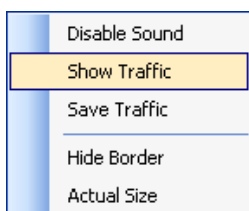
Gambar 4. Ruang konferensi

Ruang konferensi juga dilengkapi dengan fasilitas pengaturan window peserta secara otomatis dalam bentuk *tile horizontal*, *tile vertical*, *cascade*, serta *custom* untuk sesuai selera dari *user* masing-masing (Gambar 5).



Gambar 5. Tampilan konferensi tile horisontal dengan 3 user

Setiap *window user* memiliki menu yang dapat diakses melalui penekanan tombol kanan pada *mouse*. Ada 5 buah menu, yaitu “Disable/Enable Sound”, “Show/Hide Traffic”, “Save Traffic”, “Show/Hide Border” dan “Actual Size”. “Show Traffic” menampilkan jumlah data yang diterima dari *user* tersebut (Gambar 6 dan 7)



Gambar 6. Menu show traffic

Traffic for RC\Robin			
	Description	Video Data	Audio Data
▶	Bytes Received	2532224	189818
	Bytes / sec	43189	3210
	Frames	1702	N/A
	Frames / sec	31	N/A

Hide Traffic

Gambar 7. Traffic untuk user yang bersangkutan

Salah satu aspek yang penting dalam aplikasi *video conference* adalah penggunaan *bandwidth*, karena aplikasi ini mengirimkan data berupa gambar dan suara yang memiliki ukuran yang cukup besar, sehingga untuk dapat berjalan dengan lancar, *bandwidth* yang ada harus mencukupi. Pada pengujian ini dijelaskan mengenai konsumsi *bandwidth* tiap detiknya, baik untuk data gambar maupun data suara. Total dari data gambar dan data suara adalah kapasitas jaringan yang diperlukan untuk melakukan *video conference*.

Jenis alamat yang digunakan dalam pengujian adalah alamat IPv4 dengan kelas D (*multicast address*) dan jaringan yang diujicobakan adalah jaringan komputer dengan kecepatan sebesar 100 Mbps pada satu *Local Area Network (LAN)*. Hasil pengujian dapat dilihat pada tabel mengenai tipe *webcam* yang digunakan beserta dengan resolusi, *frame per second* yang mampu ditangkap oleh *webcam* dan juga *bitrate* tiap detiknya baik data tanpa dikompresi terlebih dahulu (Tabel 1) dan data dengan kompresi terlebih dahulu (Tabel 2)

Tabel 1. Pengiriman data video dan audio tanpa dikompres

Tipe webcam	Video Data		Audio Data
	Avg B/s	Avg fps	Avg B/s
Intel(r) PC Camera CS110	707.844,0	24,4	200.982
Vimicro PC USB Camera	3.266.103,0	28,3	201.096
Logitech QuickCam Pro USB	775.475,4	29,7	201.096
Logitech QuickCam Pro 4000	3.464.020,8	30,0	201.096
Bluelover T629pro	6.681.610,8	10,9	201.096
Acer Crystal Eye Webcam	8.997.769,8	14,1	201.438

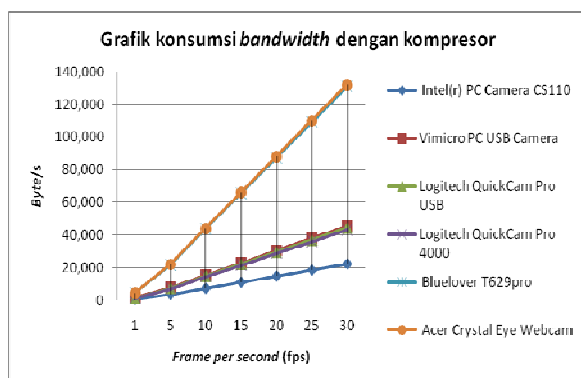
Walaupun data yang dikirimkan berjumlah besar, namun terlihat bahwa data audio berukuran jauh lebih kecil dibandingkan dengan data video, terlebih dengan data video untuk *webcam* yang memiliki resolusi tinggi seperti Bluelover T629pro dan Acer Crystal Eye Webcam. Dari hasil pengujian ini, terlihat bahwa dengan kompresor jumlah data yang dikirimkan sangat kecil walaupun fps mencapai 30. Sehingga dengan kompresor, kapasitas jaringan 100 Mbps jauh lebih dari cukup untuk melakukan *video conference*.

Tabel 2. Pengiriman data video dengan kompresor wmvideo8 encoder DMO dan data audio dengan kompresor windows media audio V2

Tipe webcam	Video Data		Audio Data
	Avg B/s	Avg fps	Avg B/s
Intel(r) PC Camera CS110	17.183,2	24,3	5.767,2
Vimicro PC USB Camera	43.026,7	28,2	5.740,5
Logitech QuickCam Pro USB	44.024,6	29,7	5.687,1
Logitech QuickCam Pro 4000	43.260,3	30,0	5.740,5
Bluelover T629pro	112.639,8	24,9	5.740,5
Acer Crystal Eye Webcam	124.833,5	28,5	5.740,5

Pada pengujian kali ini baik data video maupun audio dikompres dengan menggunakan kompresor. Terlihat bahwa jumlah data video dan audio menurun jauh dibandingkan tanpa kompresor. Sebaiknya untuk melakukan *video conference* digunakan kompresor baik untuk data video maupun audio, mengingat jumlah transfer data yang sangat besar jika tidak dikompres.

Pada gambar 8 tampak grafik perbandingan konsumsi *bandwidth* berdasarkan *frame per second* dari *webcam* dimana konsumsi *bandwidth* dihitung dengan membagi *average B/s* dengan *average fps*. Kompresor yang digunakan adalah kompresor WMVideo8 Encoder DMO.



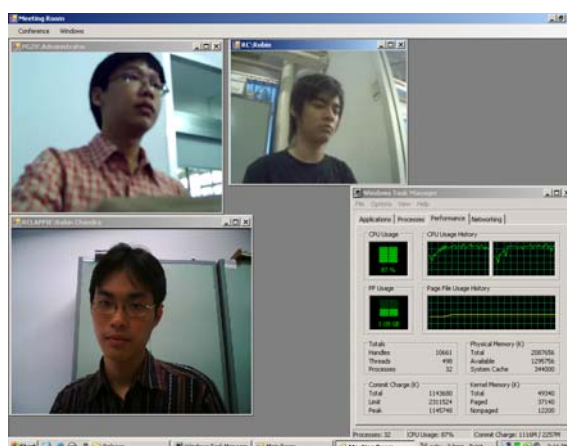
Gambar 8. Grafik konsumsi *bandwidth* dengan kompresor

Dari pengujian hasil pengiriman video dilakukan pengujian pada komputer dengan prosesor *single core*, dengan prosesor Pentium 4 3.0E GHz. Didapatkan CPU *usage* mencapai 100%. Serta hasil gambar juga sedikit kelihatan pecah untuk user “RCLAPPIE\Robin Chandra” dan user “RC\Robin” (Gambar 9)



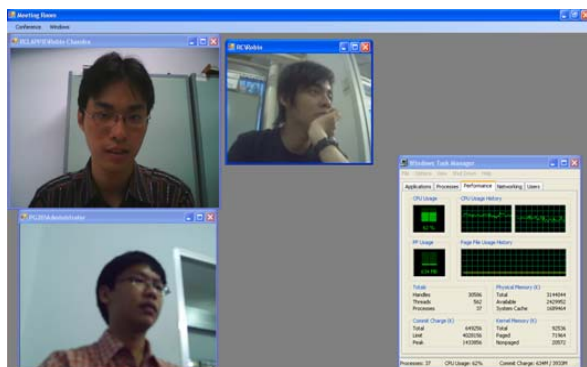
Gambar 9. Pengujian pada prosesor *single core* dengan *clock* 3 ghz tanpa teknologi *hyper-threading*

Pengujian juga dilakukan pada prosesor *single core* yang sama dengan mengaktifkan fitur *hyper-threading*. Hasil pengujian menunjukkan penurunan pada CPU *usage* menjadi 87% dan gambar masih terlihat baik untuk semua *user* (Gambar 10).



Gambar 10. Pengujian pada prosesor dengan teknologi *hyper-threading*

Untuk pengujian pada prosesor *dual core* (Intel Core2Duo T7500 2.2 GHz). Didapatkan hasil CPU *usage* sebesar 62%, lebih rendah dibandingkan dengan prosesor *single core* dengan teknologi *hyper-threading* dan dengan kualitas gambar sama bagus (Gambar 11).



Gambar 11. Pengujian pada prosesor dual core

5. KESIMPULAN

Kesimpulan-kesimpulan yang dapat diambil adalah sebagai berikut :

1. Penggunaan kompresor WMVideo8 Encoder DMO untuk data video dan Windows Media Audio V2 untuk data audio terbukti dapat menurunkan jumlah data yang dikirim sampai lebih dari 70% kapasitas jaringan dengan kecepatan 100 Mbps.
2. Jumlah fps tiap *webcam* yang dapat ditangkap bergantung pada fitur *webcam*. Fitur seperti pengaturan warna secara otomatis mengurangi jumlah fps.
3. Jika *bandwidth* yang ada tidak mencukupi, jumlah data yang dikirim akan berkurang sehingga fps-nya pun juga menurun.
4. Resolusi yang digunakan untuk menampilkan gambar berpengaruh pada penggunaan CPU *usage*, semakin besar resolusi, semakin tinggi CPU *usage* dan sebaliknya.
5. Jumlah fps juga berpengaruh terhadap CPU *usage*, semakin tinggi fps semakin tinggi CPU *usage* dan sebaliknya.
6. Penggunaan *bitrate* yang tepat akan menghasilkan *output* yang baik. *Bitrate* yang terlalu kecil ataupun terlalu besar akan mengakibatkan penurunan kualitas *output*.

PUSTAKA

- Austerberry, David. (2005). *The technology of video and audio streaming* (2nd ed.). Burlington: Focal Press.
- Codec Definition*. (2009)
<http://encyclopedia2.thefreedictionary.com/code+c>
diakses 20 February 2009
- Forouzan, Behrouz A. (2007). *Data communications and networking* (4th ed.). New York : McGraw-Hill.
- Gough, Michael. (2006). *Video conferencing over IP*. Canada: Syngress Publishing
- Hagen, Silvia. (2002). *IPv6 essentials*. Sebastopol: O'Reilly Media Inc.

Internet Protocol, Version 6 (IPv6) Specification.
<http://www.faqs.org/rfcs/rfc2460.html>

IPv4 Address Report.

<http://www.potaroo.net/tools/ipv4/index.html>

Introduction to DirectShow.

[http://msdn.microsoft.com/en-us/library/ms786508\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms786508(VS.85).aspx)

Jones, Wendy. (2004). *Beginning DirectX 9*. Boston: Course PTR.

Minoli, Daniel. (2006). *Voice over IPv6*. Burlington: Elsevier Inc.

Pesce, Mark D. (2003). *Programming Microsoft DirectShow for Digital Video and Television*. Microsoft Corporation.

RTP: A Transport Protocol for Real-Time Applications. <http://www.ietf.org/rfc/rfc1889.txt>

Silberschatz, A., Galvin, P.B., & Gagne, G. (2005). *Operating System Concepts* (7th ed.). New Jersey : John Wiley & Sons, Inc.