PERANCANGAN DAN ANALISA PERBANDINGAN ANTARA *DELAY* DAN *THROUGHPUT* PADA *VIDEO STREAMING* MENGGUNAKAN IPv4 DAN IPv6 *TUNNELING*

Haruno Sajati, Dwi Nugraheny, Eko Cahyo Nugroho
Teknik Informatika STTA Yogyakarta
informatika@stta.ac.id

ABSTRACT

Along with the rapid growth of the Internet, causing problem in the allocation of IPv4 addressing. Therefore, IPv4 only have less than 4 billion addresses and will soon be exhausted. Therefore, IPv6 has been created and used hexadecimal system on it addressing and has very large address allocation that is 3,4 x 10^{38} addresses. But now many network infrastructures in terms of hardware and software that have not fully support IPv6. So, the devices that still using IPv4 can through IPv6-based network infrastructures, they can use tunneling technique. This makes the emergence of the idea to test the delay and throughput in streaming video system which has a heavy workload on the network. Video streaming system on Vision 1.0 application takes advantage motor stepper webcam in order to support the visibility of the image that can be captured. Then the client who connected to the server can control remotely the motor stepper webcam. On the implementation phase, Vision 1.0 is running on IPv4 and IPv6 network. Next on the testing phase, carried out delay and throughput data collections from those both network systems.From the obtained results in testing, showed no significant difference. So, in general, the performance difference is not too pronounced when the video streaming system is running on both network systems. However, from a comparison based on mean value of delay and throughput, IPv4 is still superior due to the header of the data packet does not as big asIPv6 header which of course affects the packet data frame size, then the traffic network more crowded.

Keywords: *Video Streaming, IPv6 Tunneling, Quality of Service, Motor Stepper*

1. PENDAHULUAN

Dengan semakin berkembangnya *internet* alokasi penggunaan IPv4 semakin menyempit sebab IPv4 hanya memiliki alokasi *address* tidak lebih dari 4 milyar dan akan segera habis. Untuk memecahkan masalah tersebut, maka dirancanglah sistem pengalamatan IPv6 yang saat ini statusnya sebagian besar masih dalam tahap percobaan.

IPv6 mempunyai sekitar 3,4 x 10^{38} IP *address*. Tetapi, infrastruktur jaringan masih banyak yang belum mendukung penuh IPv6. Sedangkan IPv6 sama sekali tidak kompatibel dengan IPv4. Agar kedua sistem pengalamatan tersebut dapat berkomunikasi, maka digunakan teknik

tunneling. Pada saatnya nanti seluruh *internet* akan mulai menggunakan IPv6. IPv6 memiliki *header* jauh lebih sederhana daripada IPv4, tetapi berukuran lebih besar yaitu 256 *bit*, maka mempengaruhi ukuran paket data yang harus dikirimkan. Tetapi IPv6 menawarkan QoS (*Quality of Service*) yang lebih baik daripada IPv4 sehingga IPv6 cocok untuk sistem komunikasi seperti *video streaming* karena membutuhkan *bandwidth* yang besar dan QoS yang baik.

Maka, dalam penelitian ini akan dirancang dan diimplementasikan aplikasi yang diberi nama ViSion 1.0. Citra *audio/video* yang ditangkap oleh *webcam* akan dilakukan proses *streaming*. Ketika terjadi proses *streaming* akan dilakukan langkah pengambilan data dan analisa *delay* dan *throughput* guna mengetahui pengaruh pengalamatan IPv4 dan IPv6 terhadap *delay* dan *throughput* dari sistem *video streaming* yang dibuat.

Pada penelitian yang berjudul "Sistem Aplikasi *Monitoring* Ruangan Berbasis *Webcam*", pemanfaatan pengambilan citra *audio/video* pada *webcam* dilengkapi *motor stepper* sehingga visibilitas *webcam* semakin luas. Tetapi pada sistem tersebut belum berbasis jaringan komputer. Oleh karena itu, pada penelitian yang dilaksanakan ini, *webcam* yang digunakan pada aplikasi ViSion 1.0 juga dilengkapi *motor stepper* yang sama dengan pengendalian berbasis jaringan komputer.

2. METODE PENELITIAN

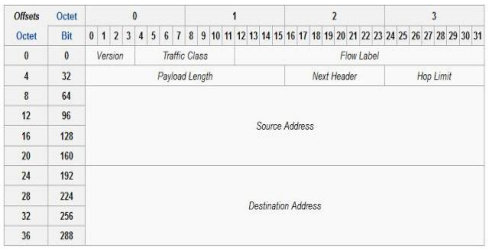
2.1. Pengacuan Pustaka

2.1.1. Perbedaan *Packet Header* pada IPv4 dan IPv6

Header pada IPv6 memiliki format baru yang didesain untuk menjaga agar *overhead header* minimum, dengan menghilangkan *field-field* yang tidak diperlukan serta beberapa *field* opsional yang ditempatkan setelah *header* IPv6.*Header* IPv6 sendiri besarnya adalah dua kali dari besar *header* dari IPv4. Namun pada *header* IPv6 terdapat trafik yang diidentifikasi menggunakan *field Flow Label*, sehingga dukungan QoS dapat tetap diimplementasikan meskipun *payload* paket terenkripsi melalui *tunneling* (Sofana S., 2009). Gambar *header* IPv4 dan IPv6 dapat dilihat pada Gambar 1 dan Gambar 2.

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															

Gambar 1 Format *Header* IPv4



Gambar 2 Format *Header* IPv6

2.1.2. Parameter *Delay dan Throughput* pada *Video Streaming*

Delay merupakan waktu yang dibutuhkan untuk sebuah paket untuk mencapai tujuan, karena adanya antrian yang panjang, atau mengambil rute yang lain untuk menghindari kemacetan. *Delay* dapat di cari dengan membagi antara panjang paket di bagi dengan link *bandwidth*. *Delay* dapat dibagi menjadi:

1. *Packetization Delay*, *delay* yang disebabkan oleh waktu yang diperlukan untuk proses pembentukan paket IP dari informasi *user*. *Delay* ini hanya terjadi sekali, yaitu di *source* informasi.
2. *Queuing Delay*, *delay* ini disebabkan oleh waktu proses yang diperlukan oleh *router* didalam menangani antrian transmisi paket di sepanjang jaringan. Umumnya *delay* ini sangat kecil , kurang lebih 100*ms.*
3. *Delay Propagasi*, merupakan proses perjalanan informasi selama didalam media transmisi, misalnya SDH, *coax* atau tembaga, menyebabkan *delay* yang disebut dengan *delay* propagasi.
4. *Transmission Delay*,adalah waktu yang diperlukan sebuah paket data untuk melintasi suatu media. *Transmission delay* ditentukan oleh kecepatan media dan besar paket data.
5. *Processing Delay,* adalah waktu yang diperlukan oleh suatu perangkat jaringan untuk melihat rute, mengubah *header* dan tugas *switching* lainnya.

Throughput adalah *bandwidth* yang sebenarnya (aktual) yang diukur dengan satuan Mbit/detik dan pada kondisi jaringan tertentu yang digunakan untuk melakukan *transfer file* dengan ukuran tertentu. Berikut hal-hal yang mempengaruhi *throughput* pada jaringan komputer:

1. Peralatan jaringan yang digunakan.
2. Tipe data yang ditransmisikan.
3. Topologi jaringan yang digunakan.
4. Banyaknya pengguna jaringan.
5. Spesifikasi komputer *client* atau *server*.
6. Jenis media transfer.

2.2. PERANCANGAN SISTEM

2.2.1. Spesifikasi *Hardware*

Kebutuhan perangkat keras dalam sistem *video streaming* ini adalah sebagai berikut:

1. PC *Server* dengan spesifikasi CPU AMD Phenom II 555 BE 3,2 GHz, RAM 4 Gb, Hard disk 250 Gb.
2. Laptop/PC Client
3. USB *Webcam* Genius 2Mpx
4. Modul lengkap *motor stepper* dengan kabel USB-to-RS232 sebagai media koneksi ke PC Server.
5. *Switch*
6. *Router* Mikrotik RB-750 dengan RouterOS v5.25 x86

2.2.2. Spesifikasi *Software*

Dalam perancangan sistem *video streaming* ini, spesifikasi kebutuhan *software* adalah sebagai berikut:

1. Windows 7 Ultimate Edition
2. Borland Delphi 7
3. DSPack 2.3.4, ComPort 4.1.1, dan ZeosDBO 6.6.6 stable.
4. XAMPP MySql Server
5. USB WebcamGenius driver
6. USB-to-RS232 driver
7. Wireshark

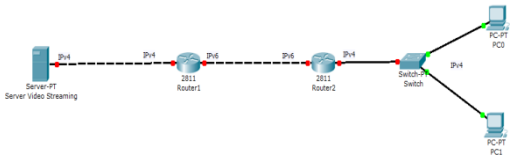
2.2.3. Konsep Sistem *Video Streaming*

Konsep sistem yang akan dibangun untuk penerapan sistem *video streaming* adalah sebagai berikut:

1. *Server database* MySql, sebagai *host* penampung data *user* (*username/password*) pengguna aplikasi ViSion 1.0. Sehingga komputer (*host*) akan otomatis berstatus *client* atau *server* sesuai hak aksesnya ketika melakukan proses *login*. Tetapi, pada sistem ini hanya ada satu akun yang berstatus admin (*server*) dan yang lainnya berstatus *client*. Begitu juga *user* yang baru melakukan proses registrasi akan berstatus *client*.
2. *Server Streaming*, berada dalam satu *host*/PC dengan *server database* MySql, yang bertugas merekam citra yang ditangkap oleh USB *Webcam* yang tersambung langsung melalui *port* USB. Jika belum ada *client* yang terhubung, *server* akan terus merekam dan berstatus *listening* pada *port* yang telah ditentukan. Jika ada *client* yang berhasil terhubung, maka s*erver* akan otomatis memberikan data *stream* yang berupa data *video/audio*. Untuk *remote server* penggerak *motor stepper* memiliki nomor *port* yang

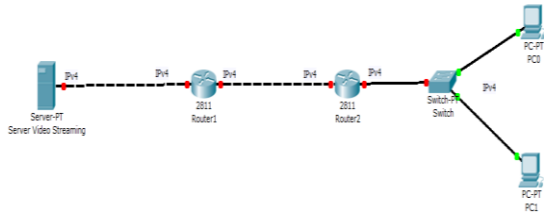
berbeda dengan *port streaming server*.

3. Pada sisi *client* diharuskan menggunakan minimal OS Windows XP atau lebih baru. *Client* juga menggunakan aplikasi yang sama seperti pada sisi *server*. Tetapi agar aplikasi berjalan dalam *mode client*, *user* harus memasukkan data akun yang memiliki hak akses *user*. Ketika berhasil *login*, *client* harus memasukkan alamat IP dan nomor *port server* secara benar, baik untuk *streaming* maupun *remote motor stepper*. Ketika berhasil terhubung, akan muncul keterangan bahwa telah berhasil terhubung dengan *server* dan akan muncul citra *video/audio* yang ditangkap oleh *server*. Begitu juga ketika *client* akan me-*remote motor stepper*, *client* hanya mengklik tombol *left*, *right*, *auto scan*, atau *stop* dan pada saat itu juga, *motor stepper* pada *server* akan bergerak sesuai keinginan *client*.
4. Pada tahap implementasi IPv6 *tunneling*, menggunakan rancangan jaringan seperti yang digambarkan pada Gambar 3.



Gambar 3.Implementasi *Video Streaming* dengan IPv6

5. *Pada* tahap implementasi pengalamatan IPv4, menggunakan rancangan seperti yang digambarkan pada Gambar 4.



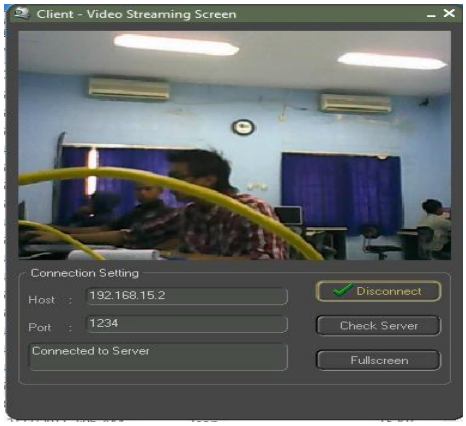
Gambar 4. Implementasi *Video Streaming* dengan IPv4

3. HASIL DAN PEMBAHASAN

3.1. Pengujian Sistem *Video Streaming* pada Jaringan IPv4

3.1.1. Uji *Streaming*

Pada tahap ini dilakukan proses pengujian *video streaming* sehingga diketahui berhasil atau tidaknya sistem yang telah dibuat pada jaringan IPv4. Proses *video streaming* ini menggunakan Format *Video*: RGB 1600x1200 24 bit yang merupakan resolusi tertinggi dan Format *Audio*: Wave Format 44100 Hz 16 bit serta *Streaming Bit rate*: 768 Kbps dengan jumlah *client* adalah 10 komputer. Tampilan ketika *client* menerima citra dari *server* dapat dilihat pada Gambar 5.



Gambar 5.Citra Visual Pada *Client*

Pada Gambar 5 terlihat bahwa sistem *video streaming* pada jaringan IPv4 berjalan normal sebab *client* berhasil menerima *streaming video* yang berasal dari *server* yang beralamatkan 192.168.15.2.

3.1.2. Analisa *Delay* dan *Throughput*

Pada tahap ini dilakukan proses *capture* data yang dikirimkan oleh *server* menuju *client*. Program *Wireshark* dijalankan pada sisi *client* yang menerima *streaming video*, sehingga ketika *client* menerima alur data dari *server*, pada saat itu juga paket data akan dianalisa oleh *Wireshark* sehingga diperoleh nilai *delay* dan *throughput*. Proses *capture* data dilakukan sebanyak 5 kali selama 30 detik.dan akan dilihat nilai pada baris *packet* sebagai jumlah *packet* yang tertangkap, baris *between first and last packet* sebagai jumlah nilai *delay*, dan baris *Avg. Mbit/sec* sebagai jumlah nilai *throughput*. Hasil *capture* dan analisa paket data yang tertangkap menggunakan *Wireshark* dapat dilihat pada Tabel 1.

Tabel 1 Hasil Uji IPv4

No.	Jumlah *Packet*	*Delay*	*Throughput*
1	2623	30,556 s	0,518 Mbit/s
2	2638	31,345 s	0,530 Mbit/s
3	2341	30,938 s	0,512 Mbit/s
4	2382	31,314 s	0,491 Mbit/s
5	2866	32,162 s	0,545 Mbit/s

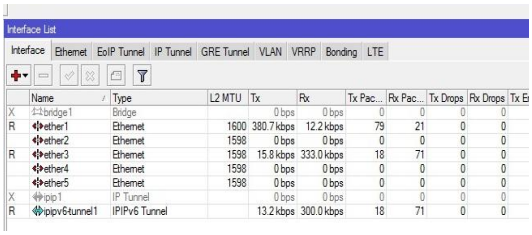
Pada Tabel 1 dapat dilihat nilai *delay* tertinggi pada angka 31,345 detik dan *throughput* tertinggi pada angka 0,545 Mbit/detik. Jika *delay* dan *throughput* dirata-rata, nilai *delay* menunjukkan angka 31,263 detik dan *throughput* pada angka 0,519 Mbit/detik.

3.2. Pengujian Sistem *Video Streaming* pada Jaringan IPv6 *Tunneling*

3.2.1. Uji *Streaming*

Pada tahap ini dilakukan proses pengujian *video streaming* pada jaringan IPv6 *tunneling* sehingga dapat diketahui berhasil atau tidaknya proses *streaming* pada *host* yang beralamatkan IPv4 melewati infrastruktur jaringan beralamatkan IPv6. Proses *video streaming* ini menggunakan Format *Video*: RGB 1600x1200 24 bit yang merupakan resolusi tertinggi dan Format *Audio*: Wave Format 44100 Hz 16 bit serta *Streaming Bit rate*: 768 Kbps

dengan jumlah *client* adalah 10 komputer.



Gambar 6.*Traffic Data* pada *Mikrotik Router*

Pada Gambar 6 dapat dilihat adanya lalu lintas data yang melewati tiap-tiap *interface router* terutama pada *interface* ipipv6-tunnel1. Maka, pada tahap uji sistem *video streaming* ini dipastikan berjalan normal pada jaringan berbasis IPv6 *tunnel*.

3.2.2. Analisa *Delay* dan *Throughput*

Pada tahap ini dilakukan proses *capture data* menggunakan *Wireshark* untuk mendapatkan nilai *delay* dan *throughput* pada jaringan IPv6 *tunneling* ini. Proses *capture packet data* dilakukan sebanyak 5 kali dalam durasi waktu 30 detik dengan jumlah pengguna jaringan sebanyak 1 *server* dan 10 *client.* Hasil *capture* dapat dilihat pada Tabel 2.

Tabel 2 Hasil Uji IPv6 *Tunneling*

No.	Jumlah *Packet*	*Delay*	*Throughput*
1	5435	29,988 s	0,590 Mbit/s
2	5221	31,751 s	0,556 Mbit/s
3	4261	29,188 s	0,488 Mbit/s
4	5319	38,903 s	0,485 Mbit/s
5	4948	33,611 s	0,516 Mbit/s

Berdasarkan hasil pada Tabel 2 dapat dilihat nilai *delay* tertinggi pada angka 38,903 detik dan *throughput* pada angka 0,590 Mbit/detik. Apabila nilai *delay* dan *throughput* dirata-rata maka menghasilkan angka *delay* sebesar 32,688 detik dan *throughput* sebesar 0,527 Mbit/detik.

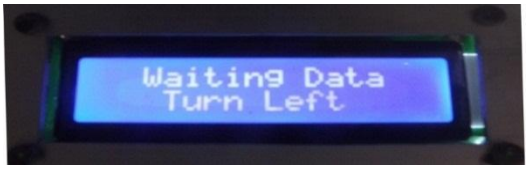
3.3. Pengendalian Jarak Jauh *Motor Stepper* dengan ViSion 1.0

3.3.1. Pengujian Pengendalian *Motor Stepper*

Dalam pengujian ini, dilakukan pengendalian jarak jauh *motor stepper* oleh *client* agar dapat diketahui apakah sistem pengendalian menggunakan komunikasi *socket* ini berhasil atau tidak. Sebelum *client* melakukan koneksi dengan *server*, *client* harus memastikan apakah *server comport* sudah berjalan.

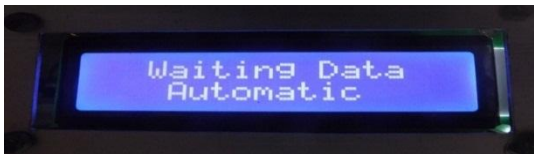
Setelah *client* berhasil terhubung ke *server* maka *client* dapat mengendalikan *motor stepper* sepenuhnya. Apabila *client* memberikan perintah kepada *motor stepper*, tampilan informasi pada LCD *display* akan berubah sesuai arah putar *motor stepper*. Tampilan informasi arah putar pada

LCD *display* ditunjukkan pada Gambar 7, Gambar 8, Gambar 9, dan Gambar 10.

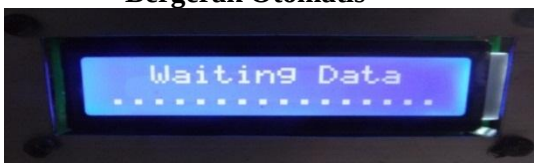


Gambar 7. Informasi *Motor Stepper* Bergerak ke Kiri

Gambar 8. Informasi *Motor Stepper* Bergerak ke Kanan



Gambar 9. Informasi *Motor Stepper* Bergerak Otomatis



Gambar 10. Informasi Gerak *Motor Stepper* Dihentikan

3.3.2. Analisa *Pseucode* Pengendalian Jarak Jauh *Motor Stepper*

Pada implementasi pengendalian *motor stepper* ini, *client* mengirimkan perintah kepada *server* dengan isi data tipe *string* yang telah ditentukan pada kode program di tiap tombol kendali.

Kode program untuk menggerakkan *motor stepper* ke **kiri**

1. Procedure TFormRemote.btnLeftClick(Sender: TObject);
2. begin
3. ClientSocket1.Socket.SendText('a');
4. end;

Kode program untuk menggerakkan *motor stepper* ke **kanan**:

1. procedure TFormRemote.btnRightClick(Sender: TObject);
2. begin
3. ClientSocket1.Socket.SendText('b');
4. end;

Kode program untuk menggerakkan *motor stepper* **otomatis**:

1. procedure TFormRemote.btnAutoClick(Sender: TObject);
2. begin
3. ClientSocket1.Socket.SendText('c');
4. end;

Kode program untuk **menghentikan** gerak *motor stepper*:

1. procedure TFormRemote.btnStopClick(Sender: TObject);
2. begin
3. ClientSocket1.Socket.SendText('d');
4. end;

Ketika *server* menerima data *string* yang berasal dari *client*, *server*

akan mengecek isi pesan *client* dengan pengkondisian *if* pada kode program. Apabila isi pesan *client* cocok dengan isi variabel yang telah ditentukan pada pengkondisian *if* tersebut, maka *server* akan mengirimkan perintah yang telah ditentukan kepada modul *motor stepper* melalui koneksi *serial com port* sehingga *motor stepper* akan bergerak sesuai perintah *client*. Berikut kode program pada *server*:

```
1. procedure TFormCPServer.ServerSocket1ClientRead(Sender: TObject;
2. Socket: TCustomWinSocket);
3. var
4. baca: string;
5. Str: string;
6. begin
7. baca:= Socket.ReceiveText;
8. Edit1.Text:= baca;
9. if baca = 'a' then
10. begin
11. Memo.Lines.Add('Turning Left');
12. Str:= baca + #13#10;
13. ComPort.WriteStr(Str);
14. end
15. else if baca = 'b' then
16. begin
17. Memo.Lines.Add('Turning Right');
18. Str:= baca + #13#10;
19. ComPort.WriteStr(Str);
20. end
21. else if baca = 'c' then
22. begin
23. Memo.Lines.Add('Webcam automove');
24. Str:= baca + #13#10;
25. ComPort.WriteStr(Str);
26. end
27. else if baca = 'd' then
28. begin
29. Memo.Lines.Add('Webcam Stopped');
30. Str:= baca + #13#10;
31. ComPort.WriteStr(Str);end;end;
```

Pada kode program di atas dapat dilihat apabila isi pesan *client* sesuai dengan isi data variabel ‘baca’, maka *server* akan menyimpan isi data variabel ‘baca’ ke dalam variabel ‘Str’. Kemudian *server* mengirimkan data dari variabel ‘Str’ kepada modul *motor stepper* melalui koneksi *com port* menggunakan kode program ComPort.WriteStr(Str). Selanjutnya *motor stepper* akan menerjemahkan isi pesan tersebut sesuai dengan data yang tersimpan pada memori modul *motor stepper* dan bergerak sesuai dengan apa yang diperintahkan. Berdasarkan hasil pengamatan perubahan informasi yang muncul pada LCD *display motor stepper* dan arah putar *motor stepper* sesuai dengan apa yang diperintahkan oleh *client* maka dapat diketahui bahwa sistem pengendalian jarak jauh *motor stepper* berjalan dengan normal.

3.4. Analisa Perbandingan *Delay* dan *Throughput* pada Jaringan IPv4 dan IPv6 *Tunneling*

Berdasarkan hasil analisa *delay* dan *throughput* pada Tabel 1 dan Tabel 2 dilakukan uji *dependent sample t-test* dengan rumus seperti yang ditunjukkan pada Rumus 1. Apabila nilai t yang diperoleh < 0.05, maka dapat disimpulkan kedua sampel tersebut memiliki perbedaan signifikan.

$$t = \frac{x1 - x2}{\sqrt{\frac{s1^2}{n1} + \frac{s2^2}{n2} - 2r\left(\frac{s1}{\sqrt{n1}}\right)\left(\frac{s2}{\sqrt{n2}}\right)}} \quad (1)$$

Keterangan:

X_1 = Rata-rata sampel 1

X_2 = Rata-rata sampel 2

S_1 = Simpangan baku sampel 1

S_2 = Simpangan baku sampel 2

S_1^2 = Varian sampel 1

S_2^2 = Varian sampel 2

r = Korelasi antara dua sampel

Supaya proses perhitungan dapat dilakukan lebih mudah dan cepat, maka perhitungan dilakukan dengan memanfaatkan *Microsoft Excel*. Berdasarkan pengujian t-tes *delay* dan *throughput video streaming* antara jaringan berbasis IPv4 dan IPv6 *tunneling* diperoleh hasil yang ditunjukkan pada Gambar 11 dan Gambar 12.

	A	B	C
1	t-Test: Paired Two Sample for Means		
2			
3		*Variable 1*	*Variable 2*
4	Mean	31.263	32.6882
5	Variance	0.35575	14.9740307
6	Observations	5	5
7	Pearson Correlation	0.445867747	
8	Hypothesized Mean Difference	0	
9	df	4	
10	t Stat	-0.874779782	
11	P(T<=t) one-tail	0.215537642	
12	t Critical one-tail	2.131846786	
13	P(T<=t) two-tail	0.431075284	
14	t Critical two-tail	2.776445105	

Gambar 11. Hasil *Data Analysis Delay*

	A	B	C	D
1	t-Test: Paired Two Sample for Means			
2				
3		*Variable 1*	*Variable 2*	
4	Mean	0.527	0.5192	
5	Variance	0.002054	0.000408	
6	Observations	5	5	
7	Pearson Correlation	0.387661		
8	Hypothesized Mean Difference	0		
9	df	4		
10	t Stat	0.416666		
11	P(T<=t) one-tail	0.349156		
12	t Critical one-tail	2.131847		
13	P(T<=t) two-tail	0.698311		
14	t Critical two-tail	2.776445		

Gambar 12. Hasil *Data Analysis Throughput*

Berdasarkan uji t-tes pada *delay* dan *throughput* menunjukkan tidak adanya perbedaan signifikan. Sebab dari hasil yang diperoleh, nilai P(T<=t) *two tail* pada *delay* sebesar 0,43 dan *throughput* sebesar 0,698 yang bernilai > 0,05. Kemungkinan besar hal ini disebabkan pada sistem pengalamatan IPv6 masih melibatkan pengalamatan IPv4, yaitu dengan teknik IPv4 *over* IPv6 *tunneling.*

IPv4 memiliki rata – rata *delay* sebesar 31,263 detik dan rata – rata *throughput* sebesar 0,519 Mbit/detik.Sedangkan pada skenario IPv6 *tunneling* memiliki rata – rata *delay* sebesar 32,688 detik dan rata – rata *throughput* sebesar 0,527

Mbit/detik.Maka dapat diketahui bahwa *delay* pada IPv6 *tunneling* yang bernilai 32,688 detik lebih besar daripada IPv4 yang bernilai 31,263 detik.Hal ini kemungkinan besar disebabkan karena bertambahnya ukuran *header* pada paket data yang dikirimkan. Sebab pada proses enkapsulasi IPv6 *tunneling*, *header* IPv6 berukuran sangat besar yaitu 320 bit yang kemudian ditambahkan pada *header* milik IPv4 yang berukuran 32 bit. Apabila dijumlahkan *header* paket data akan menjadi 352 bit. Dengan bertambahnya ukuran *header* paket data, maka data aktual yang dapat ditampung menjadi berkurang. Sehingga data harus difragmentasi semakin banyak yang mengakibatkan arus data semakin padat dan mempengaruhi nilai *delay* karena antrian *packet data* semakin banyak.

Throughput pada skenario IPv4 yang bernilai 0,519 Mbit/detik lebih kecil daripada IPv6 *tunneling* yang bernilai 0,527 Mbit/detik.Hal ini kemungkinan besar disebabkan karena pada sistem pengalamatan IPv4 tidak terjadi kepadatan trafik jaringan dibandingkan IPv6. Sebab *header* milik IPv4 berukuran kecil dan setiap fragmen paket data dapat menampung data aktual yang cukup banyak yang mengakibatkan data tidak perlu dipecah – pecah sebanyak mungkin seperti apa yang terjadi pada IPv6 *tunneling*. Hal ini dibuktikan pada hasil jumlah *packet data* yang tertangkap pada IPv4 berada di kisaran 2000 – 3000 *packet data*, sedangkan *packet data* pada IPv6 *tunneling* berada di kisaran 4000 – 6000 *packet data*. Maka dapat dilihat bahwa terjadi peningkatan jumlah paket data ketika menggunakan IPv6 *tunneling.*

4. KESIMPULAN

Berdasarkan hasil implementasi dan analisa pengujian, dapat diambil kesimpulan:

1. Aplikasi ViSion 1.0 dapat mengirimkan citra *audio* dan *video* kepada *client* secara *live streaming* melalui jaringan berbasis pengalamatan IPv4 maupun IPv6 *tunneling* karena *client* dapat menerima dan menampilkan citra *audio* dan *video* yang ditangkap oleh *server*.
2. Aplikasi ViSion 1.0 dapat digunakan oleh *client* untuk mengendalikan *motor stepper* secara jarak jauh melalui jaringan komputer karena arah gerak *motor stepper* dan informasi pada LCD *display* pada modul sesuai dengan apa yang diperintahkan oleh *client.*
3. Berdasarkan hasil uji *dependent samplet-test* nilai t pada *delay* sebesar 0,43 dan *throughput* sebesar 0,698 dimana keduanya

bernilai > 0,05. Maka dapat dinyatakan bahwa *delay* dan *throughput* sistem *video streaming* pada jaringan IPv4 dan IPv6 *tunneling* tidak menunjukkan perbedaan yang signifikan. Namun dari hasil rata -rata nilai *delay* dan *throughput*, IPv4 memiliki *delay* sebesar 31,263 detik dan *throughput* sebesar 0,519 Mbit/detik, sedangkan IPv6 *tunneling* memiliki *delay* sebesar 32,688 detik dan *throughput* sebesar 0,527 Mbit/detik. Maka dapat terlihat dalam sistem *video streaming* pada aplikasi ViSion 1.0 memiliki nilai *delay* dan *throughput* yang lebih baik ketika menggunakan jaringan dengan pengalamatan IPv4.

DAFTAR PUSTAKA

Aditiya Prasetya, Bayu. 2008. *Pengaruh Video Bit-Rate dan Background Traffic Terhadap Kinerja Video Streaming Pada Jaringan Wireless LAN*. Bogor: Institut Pertanian Bogor.

Artondo, Reko. 2011. *Analisa dan Implementasi IPv6 Tunnel Broker Untuk Interkoneksi Antara IPv6 dan IPv4*. Semarang: Universitas Diponegoro.

Balza, Achmad. 2011. *Pemrograman Delphi untuk Aplikasi Mesin Visi Menggunakan Webcam.*Yogyakarta: Gava Media.

Fadlisyah, Rizal. 2011. *Pemrograman Computer Vision Pada Video Menggunakan Delphi + Vision Lab VCL 4.0.1*. Jakarta: Graha Ilmu.

Fadlisyah, Rizal. Kurniawan, Dayat. 2010. *Pemrograman Kamera PC Dengan DELPHI*. Jakarta: Graha Ilmu.

Kristanto, Andri. 2004. *Rekayasa Perangkat Lunak (Konsep Dasar).*Yogyakarta: Gava Media.

Linto Herlambang, Moch. Catur, Azis L. 2008. *Panduan Lengkap Menguasai Router Masa Depan Menggunakan Mikrotik RouterOS*. Yogyakarta: Penerbit Andi.

Masya, Fajar. Flado, Andrew. 2011. *Socket Programming*. Jakarta: Graha Ilmu.

Rafiudin, Rahmat. 2008. *IPv6 Addressing*. Jakarta: Elex Media Komputindo.

Sudaryono. 2012. *Statistika Probabilitas.* Yogyakarta: Penerbit Andi.

Wahana Komputer. 2003. *Pengembangan Aplikasi Client/Server dengan Borland Delphi*. Jakarta: PT.Elex Media Komputindo.

Wahyu Sulistiyanto, Debby. 2012. *Sistem Aplikasi Monitoring Ruangan Berbasis Webcam.*Yogyakarta: Sekolah Tinggi Teknologi Adisutjipto.