

## APLIKASI TCP-IP UNTUK MENGENDALIKAN GERAK WEBCAM

Indar Sugiarto<sup>1</sup>, Petrus Santoso<sup>2</sup>, Andy Susanto<sup>3</sup>

<sup>1,2,3</sup>Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Kristen Petra  
Jl. Siwalankerto 121-131, Surabaya - 60236  
Telp. (031)2983442, Faks. (031) 8492562  
E-mail: indi@petra.ac.id, petrus@petra.ac.id

### ABSTRAK

Dalam paper ini dipaparkan sistem keamanan memanfaatkan teknologi kamera (dalam hal ini berupa webcam), dua buah motor sebagai penggerak kamera beserta rangkaian driver-nya, sebuah komputer server, dan beberapa komputer client. Webcam dihubungkan dengan komputer server dan kemudian mengirimkan data video tersebut ke komputer client melalui protokol TCP/IP. Salah satu komputer client terpilih (pemantau) dapat juga memerintahkan komputer server untuk menggerakkan kamera, sehingga hampir semua sudut ruangan dapat dipantau. Sistem ini, diimplementasikan dengan bahasa pemrograman C++, dan memanfaatkan library API bawaan Windows untuk mengakses webcam maupun Winsock untuk komunikasi lewat protokol TCP/IP. Sistem telah diuji untuk keadaan jaringan terisolasi maupun lewat koneksi global (internet). Dari hasil pengujian lokal di dapatkan bahwa komputer client dapat menangkap gambar bergerak yang dipancarkan oleh server melalui mekanisme video streaming dengan framerate hingga 30 fps dan hanya memanfaatkan sekitar 1/4 dari kapasitas maksimum bandwidth jaringan. Jika sistem dioperasikan secara online dan terhubung dengan internet, kualitas video streaming sangat tergantung infrastruktur jaringan yang ada. Namun penempatan komputer server dilokasi yang strategis seperti pada backbone jaringan utama akan dapat meningkatkan aksesibilitas dari sistem dan menurunkan penggunaan bandwidth jaringan hingga hanya 4% saja. Dari hasil pengujian pengendalian motor juga didapatkan bahwa motor dapat dikendalikan sempurna dengan tingkat kesalahan hanya 0,5°.

**Kata Kunci:** TCP-IP, winsock, client-server, webcam, security

### 1. PENDAHULUAN

Banyak gedung-gedung publik saat ini yang dilengkapi dengan sistem monitoring ruangan dengan kamera CCTV (*closed-circuit television*). Diantara kamera-kamera CCTV yang beredar di pasaran, masih banyak menggunakan teknologi analog dan konvensional, yaitu menggunakan kabel *coaxial*. Konsekuensinya, dibutuhkan juga biaya khusus untuk instalasi pengkabelan sistem CCTV. Selain itu, hanya sedikit teknologi CCTV yang dilengkapi dengan motor penggerak kamera. Selain berharga mahal, instalasinya juga membutuhkan penanganan lebih karena kabel data gambar dan kabel sinyal kendali tidak mungkin dijadikan satu.

Sementara itu, teknologi jaringan komputer, terutama LAN (*local area network*), sudah banyak diaplikasikan di hampir semua sarana gedung publik. Penggunaan teknologi jaringan komputer memungkinkan pemantauan suatu tempat dapat dilakukan dari tempat lain, bahkan dari tempat yang jauh sekalipun dimana hal ini sangat sulit dilakukan jika menggunakan sistem CCTV konvensional. Paper ini menjelaskan alternatif pengganti CCTV dengan memanfaatkan jaringan LAN yang sudah terpasang dalam suatu gedung/instansi publik dengan tetap memperhatikan keterbatasan resolusi webcam dan bandwidth LAN.

Dalam kaitannya dengan mekanisme pemantauan suatu tempat dari jarak jauh, dapat digunakan teknologi kamera *webcam* dan intranet. *Webcam* digunakan untuk mendapatkan gambar

untuk memantau kondisi tempat tersebut, sedangkan untuk melihat hasil pemantauannya dari jarak jauh secara *remote* digunakan teknologi intranet. Protokol yang akan dipakai sebagai perantara adalah TCP/IP. Kamera yang dipakai adalah jenis *webcam* dengan resolusi standar VGA yang dilengkapi dengan 2 buah motor sebagai penggerak arah horisontal dan vertikal. Lewat *parallel port*, alat penggerak kamera ini dihubungkan dengan komputer yang bertindak sebagai *server*. Dengan mekanisme ini, pengguna tidak hanya dapat melihat gambar yang ditangkap kamera dari jarak jauh, tetapi dapat juga menggerakkan kamera sesuai arah yang diinginkan, sehingga kamera dapat memiliki sudut pandang yang lebih luas.

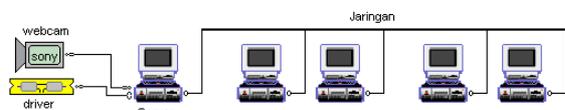
Paper ini disusun dengan sistematis sebagai berikut. Setelah bagian pendahuluan yang menjelaskan latar belakang penelitian, akan dijelaskan metode yang digunakan dalam disain sistem. Kemudian akan dijelaskan mekanisme pengujian dan hasil-hasilnya. Di bagian akhir, paper ini akan ditutup dengan kesimpulan.

### 2. MODEL YANG DIBANGUN

#### 2.1 Disain Sistem

Gambar di bawah ini menunjukkan skema secara garis besar dari sistem monitoring ruangan dengan menggunakan kamera yang dibuat dalam penelitian ini. Kamera untuk keamanan berada pada sisi *server* yang dapat dikendalikan dan dipantau oleh sisi *client* melalui jaringan LAN. *Webcam* terhubung dengan

komputer *server* melalui *USB port* sedangkan *driver* penggerak terhubung dengan komputer *server* melalui *parallel port*.

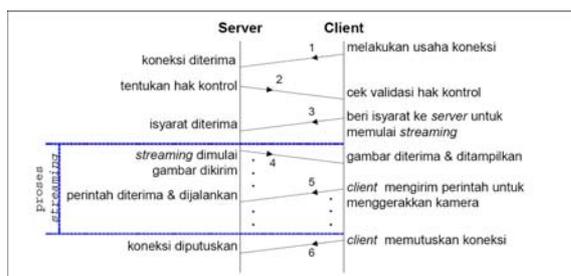


Gambar 1. Diagram blok sistem monitoring ruangan menggunakan jaringan LAN.

Komputer *server* yang terhubung dengan *webcam* bertugas mengambil data *video streaming* dari *webcam* dan mengirimkannya ke komputer *client* melalui kabel jaringan. Komputer *client* yang terhubung dengan komputer *server* dapat lebih dari satu unit.

## 2.2 Disain Perangkat Lunak

*Software* yang digunakan dalam sistem seperti ditunjukkan pada gambar 1 terdiri dari dua bagian, yaitu program aplikasi *server* dan aplikasi *client*. Aplikasi *server* akan digunakan oleh komputer *server* untuk mengambil data *streaming* video dari *webcam*, melakukan pengikatan *socket* dan menunggu koneksi dari komputer *client*, mengirimkan data *video streaming* yang diperoleh dari *webcam* dan menggerakkan motor ketika ada perintah dari komputer *client*. Aplikasi *client* akan digunakan oleh komputer *client* untuk melakukan koneksi ke komputer *server*, menerima data *video streaming* dari *server* dan menampilkannya di layar serta mengirimkan perintah ke *server* untuk menggerakkan kamera. Protokol koneksi antara *client* dan *server* ditunjukkan pada gambar berikut.



Gambar 2. Protokol koneksi antara *client* dan *server*.

Mekanisme komunikasi *client-server* dari sistem ini adalah sebagai berikut. Setelah *server* dijalankan dan siap menerima koneksi, *client* berusaha menjalin koneksi dengan *server*, yang pada gambar 2 ditunjukkan dengan panah nomor 1. Pada saat koneksi telah terbentuk, *server* memeriksa apakah pada saat itu ada *client* lain yang memiliki hak kontrol. Bila sudah ada *client* lain yang memiliki hak kontrol maka *client* yang baru terhubung tidak berhak menerima hak kontrol tersebut. Hanya *client* yang memiliki hak kontrol yang dapat memberikan perintah kepada *server* untuk menggerakkan kamera

dan pada suatu saat tertentu sedangkan *client* yang lain hanya dapat menikmati hasil *capture* kamera.

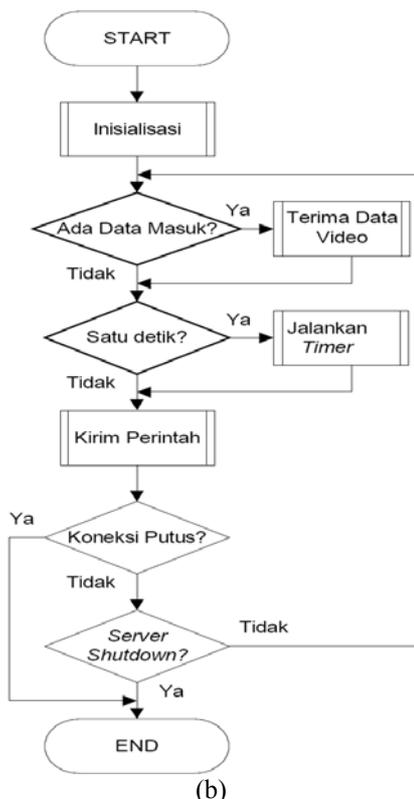
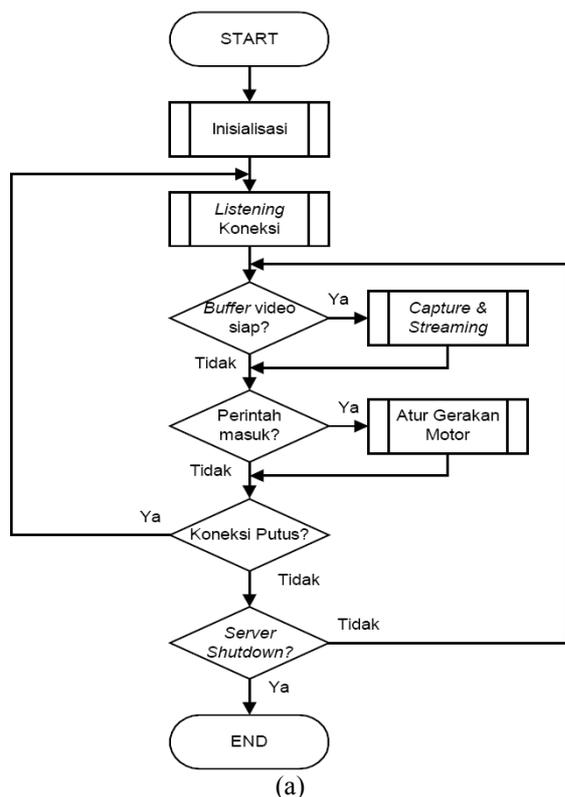
Panah nomor 2 menyatakan pesan *server* kepada *client* yang akan memberitahukan berhak atau tidaknya *client* menerima hak kontrol. Setelah *client* menerima pesan tersebut, aplikasi *client* memeriksa apakah hak kontrol diberikan kepada *client* atau tidak. Pada aplikasi *client* yang memiliki hak kontrol, tombol-tombol arah (kanan, kiri, atas dan bawah) dihidupkan, sedangkan pada aplikasi *client* yang tidak memiliki hak kontrol, tombol-tombol arah tersebut dimatikan.

Memiliki atau tidak hak kontrol, *client* harus tetap memberitahukan kepada *server* untuk memulai proses *streaming* (panah nomor 3). Setelah aplikasi *server* menerima pemberitahuan ini, maka proses *streaming* dijalankan oleh *server* dengan mengirimkan data video ke *client*. Data video akan terus dikirimkan ke *client* selama koneksi antara *client* dan *server* belum terputus.

Panah nomor 4 menunjukkan pengiriman data video dari *server* ke *client*. Pada sisi *client*, data video yang diterima dari *server* ditampilkan pada layar dalam bentuk *movie* (gambar bergerak). Selain dapat menerima gambar yang berupa data video dari *server* dan menampilkannya di layar, aplikasi *client* juga dapat memberikan perintah kepada *server* untuk menggerakkan motor yang terpasang pada kamera *server*. Namun hanya *client* yang memiliki hak kontrol yang dapat memberikan perintah tersebut (panah nomor 5).

Ketika perintah dari *client* untuk menggerakkan kamera diterima *server*, *server* mengeluarkan sinyal ke *parallel port* untuk mengatur arah pergerakan motor serta fungsi yang mengatur mati dan hidupnya motor. Pada kamera *server* dipasang 2 buah motor untuk mengatur gerakan kamera secara horisontal (motor horisontal) dan vertikal (motor vertikal). Panah nomor 6 menunjukkan isyarat yang digunakan oleh *client* untuk memberitahukan *server* bahwa koneksi yang telah menghubungkan keduanya akan segera diputuskan. Aplikasi *server* menerima pesan *client* dan memutuskan koneksi dengan *client* yang bersangkutan. Jika ternyata *client* yang memutuskan koneksi tersebut memiliki hak kontrol maka hak kontrol tersebut akan diberikan kepada *client* lain yang terhubung dengan *server* setelah pemutusan koneksi tersebut.

Jenis protokol yang dipakai adalah TCP karena mekanisme kontrol yang digunakan mengharuskan jenis koneksi yang *connection-oriented* (Stallings, 2000). Dalam hal koneksi dengan *webcam driver*, digunakan *library DrawDib* yang didalamnya telah tersedia fungsi-fungsi *capture* milik Windows dalam sebuah kelas yang disebut *AVICap* (Gorbatenko, 1999). Diagram alur aplikasi *server* dan *client* adalah sebagai berikut.



Gambar 3. Diagram alir dari program *server* (a) dan *client* (b).

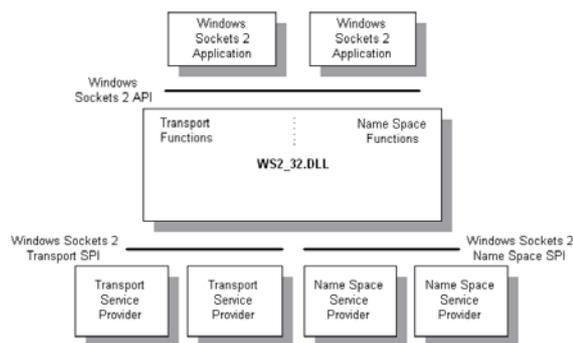
Untuk mengakses *parallel port* komputer, *driver* I/O port khusus diperlukan bila komputer yang kita pakai sebagai *server* menggunakan sistem operasi

Windows NT/2000/XP/. Windows NT/2000/XP tidak mendukung pengalamatan secara langsung terhadap *parallel port* (Eldad, 2005). *Driver* yang digunakan dalam paper ini adalah UserPort.sys yang berfungsi untuk mendukung pengalamatan secara langsung *parallel port* pada Windows NT/2000/XP. *Driver* ini tidak akan berpengaruh bila komputer *server* yang dipakai menggunakan sistem operasi Windows 98. Supaya *driver* I/O port tersebut bisa berjalan dengan baik maka file UserPort.sys di-copykan ke direktori %windir%\system32\drivers sebelum menjalankan kode-kode program berikut:

- Initial\_DriverNT();
- ReadRegistry\_DriverNT();
- StartDriver\_DriverNT();

Fungsi Initial\_DriverNT() menuliskan alamat I/O port yang digunakan oleh aplikasi yaitu alamat 378h dan 379h pada *Windows registry*. Hal ini dilakukan supaya ketika fungsi ReadRegistry\_DriverNT() dijalankan, data pada HKEY\_LOCAL\_MACHINE\Software\UserPort sudah terisi alamat I/O port yang dibutuhkan. Setelah alamat I/O port dimuat, maka untuk menjalankan *driver*, aplikasi harus memanggil fungsi StartDriver\_DriverNT().

Dalam paper ini digunakan protokol Winsock 2. Winsock 2 memiliki arsitektur yang fleksibel sehingga mampu mendukung *multiple protocol stack*, antarmuka dan *service provider* secara simultan (Burns, 2003). Pada bagian *layer* atas winsock 2 memang masih menggunakan satu DLL (winsock 2 API), namun terdapat *layer* lagi di bawahnya dan sebuah antar muka *service provider* standar, yang mana masing-masing juga bersifat fleksibel. Gambar berikut menunjukkan arsitektur dari Winsock 2 (Microsoft, 2004).



Gambar 4. Arsitektur Winsock 2.

Tidak semua fungsi-fungsi Winsock 2 digunakan dalam program, tetapi hanya fungsi-fungsi berikut: accept(), bind(), closesocket(), connect(), listen(), recv(), send(), shutdown(), dan socket().

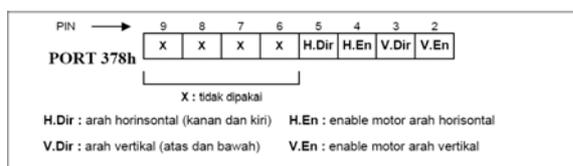
Sebuah *socket* membutuhkan sebuah struktur sockaddr\_in untuk menentukan alamat *endpoint* baik lokal maupun *remote* dimana *socket* akan terhubung. Karena ada 2 macam *socket* yang dibuat,

maka terdapat juga 2 buah struktur `sockaddr_in` yaitu, `sockaddr_in` dengan nama `saListen` untuk socket `sListen` dan `sockaddr_in` dengan nama `saClient` untuk socket `sockClient`. Sebuah kelas dengan nama `CServerDlg` dibuat dengan memuat deklarasi 2 buah `socket` beserta 2 buah struktur `sockaddr_in` tersebut.

Secara khusus, fungsi `recv()` milik `winsock` berfungsi untuk menerima pesan yang masuk ke `socket server`. Pesan yang masuk ke `server` dapat digolongkan menjadi dua macam, yaitu pesan yang berisi perintah untuk menggerakkan kamera dan pesan yang berisi perintah untuk status `server`. Berikut ini potongan kode pemanggilan terhadap fungsi `recv()` yang terdapat dalam fungsi `OnReceiverData()` milik kelas `CServerDlg`:

```
char buff[256];
. . .
for (int i = 0; i < 10; i++)
{
    if ( sockClient[i] != INVALID_SOCKET )
    {
        int nbyte =
            recv(sockClient[i],buff,255,0);
    }
}
```

Pesan yang berisi perintah untuk menggerakkan kamera tersebut terdiri dari empat macam, yaitu untuk arah kiri, kanan, atas dan bawah. Gambar berikut menunjukkan disain keluaran pada pin-pin `parallel port` (alamat 378h) untuk masing-masing arah tersebut.



Gambar 5. Konfigurasi keluaran `port 378h`.

Dengan berpatokan pada konfigurasi seperti ditunjukkan pada gambar 5, algoritma untuk menggerakkan kamera adalah sebagai berikut.

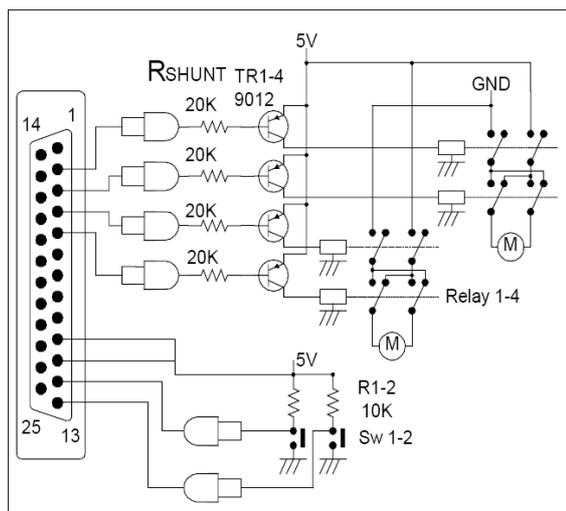
```
char buff[256];
char aLeft[256] = "KIRI";
char aRight[256] = "KANAN";
. . .
if ( i == whoControl )
{
    . . .
    if ( strcmp(buff,aLeft) == 0 )
    {
        __asm mov edx,0x378
        __asm mov al,3
        __asm out dx,al
        Sleep(40);
        __asm mov edx,0x378
        __asm mov al,255
        __asm out dx,al
    }
    if ( strcmp(buff,aRight) == 0 )
```

```
{
    __asm mov edx,0x378
    __asm mov al,11
    __asm out dx,al
    Sleep(40);
    __asm mov edx,0x378
    __asm mov al,255
    __asm out dx,al
}
. . .
}
```

Gambar 6. Algoritma untuk menggerakkan kamera.

### 2.3 Disain Perangkat Keras

Perangkat keras sistem terdiri dari rangkaian `driver` dan kerangka penyangga. Rangkaian `driver` terdiri 4 buah `relay 5V`, sebuah IC TTL `74LS08`, 4 buah transistor `TR9012`, 4 buah resistor `10KΩ`, 2 buah `limit switch`, 2 buah motor DC, sebuah kabel data dengan konektor `DB 25 male` dan sebuah kabel `USB` untuk `power supply`. Gambar berikut adalah skema rangkaian `driver`.



Gambar 7. Rangkaian `driver` untuk menggerakkan `webcam`.

### 3. PENGUJIAN

Secara umum telah diketahui bahwa dalam protokol `TCP`, reliabilitas data sangat tinggi tetapi memiliki `delay` yang besar dibandingkan `UDP`. Dengan demikian, jika sistem diimplementasikan secara `online` dalam jaringan internet, maka dibutuhkan `bandwidth` yang juga besar untuk menghasilkan `framerate` yang tinggi.

Pengujian dilakukan dengan menggunakan jaringan `LAN` di Universitas Kristen Petra pada beberapa gedung dengan konfigurasi jaringan yang berbeda. Sistem telah diujicoba pada jaringan yang bersifat `adhoc` maupun terstruktur yang memiliki alamat `10.0.0.0/8`, `192.168.0.0/16`, dan `202.43.254.0/24`. Berikut ini adalah tampilan dari program `server` dan `client`.



(a)



(b)



(c)

Gambar 8. Tampilan program *server* (a) dan program *client* (b) dan (c). Program *client* pada gambar (b) melakukan koneksi lebih dulu dengan *server* daripada *client* gambar (c) sehingga mendapat prioritas untuk mengendalikan gerak motor kamera *server*.

Komputer yang digunakan sebagai *server* memiliki konfigurasi sebagai berikut: prosesor Intel Pentium 4 - 2,4GHz, memory DDR 1536MB, *gigabit ethernet card* Intel Pro/100VE, dan Microsoft Windows XP Professional Service Pack 2. Sedangkan komputer *client* yang digunakan memiliki konfigurasi standar sebagai berikut: prosesor Intel 4 - 1,4GHz, memory DDR 128MB, *ethernet card* dan Microsoft Windows XP.

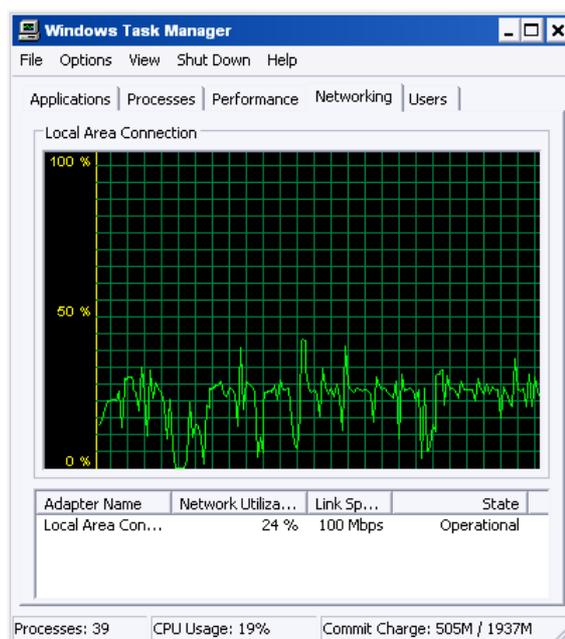
### 3.1 Pengujian Pada Jaringan yang Terisolasi

Yang dimaksud dengan jaringan terisolasi adalah jaringan dalam satu *subnet* yang tidak terhubung dengan jaringan lain atau dengan internet. Pengujian dilakukan dengan mengukur utilitas *bandwidth* pada komputer *server* serta *framerate* yang dapat

ditampilkan oleh komputer *client*. Program *client* dilengkapi utilitas bawaan dari *library* DrawDib dari kelas AVICap milik Windows API (*Application Program Interface*) untuk mengukur berapa *framerate* dari *video streaming* dalam satuan *frame per second* (fps). Dalam pengujian ini dibuat skenario sebagai berikut. Komputer *server* dan beberapa komputer *client* dimasukkan dalam jaringan dengan alamat 192.160.0/16 dengan menggunakan sebuah LAN-*switch* dan terpisah dari jaringan lain di dalam kampus. Berikut ini adalah hasil pengujian dengan menggunakan 1 hingga 5 komputer *client*.

Tabel 1. Hasil pengujian menggunakan jaringan LAN yang terisolasi

Jumlah <i>client</i>	<i>Framerate</i> maksimum (fps)	<i>Bandwidth utilization</i> rata-rata (%)
1	30	10
2	30	14
3	27	19
4	25	23
5	25	24



Gambar 9. *Network utilization* pada pengujian dengan 5 *client* pada jaringan dengan *subnet* yang sama.

Dari pengujian ini terlihat bahwa *server* dapat bekerja dengan baik dan mampu melayani permintaan beberapa *client* sekaligus dengan kualitas video yang cukup baik (>25 fps). Terlihat bahwa *bandwidth* yang digunakan naik cukup signifikan meskipun tidak linier (10 – 24%) yang disebabkan karena *streaming* video dari *server* kepada beberapa *client* sekaligus.

### 3.2 Pengujian Pada Jaringan yang Terhubung dengan Internet

Inti dari pengujian ini sama dengan metode pengujian sebelumnya, tetapi komputer *server* diletakkan dalam *backbone* jaringan kampus yang memiliki *real IP* (202.43.254.73/24) dan bisa diakses secara *online* via internet, sedangkan 3 buah komputer *client* di pasang pada tiga jaringan yang berbeda, yaitu: 10.0.0.0/8, 192.168.0.0/16, dan 202.43.254.0/24. Hasil pengujiannya ditunjukkan pada tabel berikut.

Tabel 2. Hasil pengujian *online* menggunakan infrastruktur jaringan yang berbeda

Alamat <i>client</i>	<i>Framerate</i> maksimum (fps)	<i>Bandwidth utilization server</i> rata-rata (%)
202.43.254.101	28	4
192.168.11.100	22	4
10.3.22.77	21	4

Dari pengujian ini terlihat bahwa dengan jumlah *client* yang tidak terlalu banyak, sistem mampu bekerja dengan baik pada kondisi *online* (terhubung dengan internet). Khusus untuk komputer *client* yang memiliki *subnet* yang sama dengan *server*, performa *video streaming* masih sangat baik. Hal ini bisa disebabkan karena *router* tidak mengirimkan paket TCP keluar dari jaringan. Sedangkan untuk jaringan yang *subnet*-nya berbeda, ada kemungkinan paket TCP di-*routing* melalui *gateway* di luar jaringan kampus sehingga performa *video streaming* sedikit turun. *Bandwidth* yang digunakan juga sangat rendah karena *server* diletakkan di *backbone* kampus yang memiliki *bandwidth* lebih besar dibanding pada saat skenario pengujian sebelumnya.

### 3.3 Pengujian Kendali Motor

Pengujian dilakukan dengan memutar motor satu putaran penuh (360°) dan diperoleh 322 *step* (karena motor menggunakan *gearbox*). Jadi, untuk satu *step* pada motor besar sudutnya adalah  $360/322 = 1,1^\circ$ . Untuk mengetahui keakuratan *driver* motor, sistem diuji dengan cara komputer *client* mengirimkan perintah memutar motor dengan jumlah *step* tertentu. Hasil pengujiannya ditunjukkan pada tabel berikut.

Tabel 3. Hasil pengujian kendali motor *webcam*.

Jumlah <i>Step</i>	Hasil Perhitungan	Hasil Pengukuran	Selisih
5	$1,1^\circ \times 5 = 5,5^\circ$	$5,5^\circ$	$0^\circ$
10	$1,1^\circ \times 10 = 11^\circ$	$10,75^\circ$	$0,25^\circ$
15	$1,1^\circ \times 15 = 16,5^\circ$	$16,25^\circ$	$0,25^\circ$
30	$1,1^\circ \times 30 = 33^\circ$	$32,5^\circ$	$0,5^\circ$
60	$1,1^\circ \times 60 = 66^\circ$	$66,25^\circ$	$0,25^\circ$

## 4. KESIMPULAN

Sebuah alternatif implementasi sistem keamanan berupa monitoring suatu ruangan dengan memanfaatkan jaringan komputer LAN telah

dipaparkan. Sistem ini terdiri dari *webcam* yang dilengkapi 2 buah motor dan dapat dikendalikan oleh perintah yang dikirimkan menggunakan protokol TCP/IP. Dengan menggunakan 2 buah motor tersebut, seluruh area di dalam ruangan dapat dipantau. Dari hasil pengujian didapatkan bahwa dengan menggunakan *webcam* yang memiliki resolusi standar VGA dan dioperasikan dalam sebuah jaringan yang terisolasi, komputer *client* dapat menangkap gambar bergerak yang dipancarkan oleh *server* melalui mekanisme *video streaming* dengan *framerate* hingga 30 fps dan hanya memanfaatkan sekitar  $\frac{1}{4}$  dari kapasitas maksimum *bandwidth* jaringan. Jika sistem dioperasikan secara *online* dan terhubung dengan internet, kualitas *video streaming* sangat tergantung infrastruktur jaringan yang ada. Namun penempatan komputer *server* di lokasi yang strategis seperti pada *backbone* jaringan utama akan dapat meningkatkan aksesibilitas dari sistem dan menurunkan penggunaan *bandwidth* jaringan hingga hanya 4% saja. Dari hasil pengujian pengendalian motor juga didapatkan bahwa motor dapat dikendalikan sempurna dengan tingkat kesalahan hanya  $0,5^\circ$ .

## PUSTAKA

- Burns, K. (2003). *TCP/IP Analysis and Troubleshooting Toolkit*. Indianapolis: Wiley Publishing, Inc.
- Eldad, E. (2005). *Reversing: Secrets of Reverse Engineering*. Indianapolis: Wiley Publishing, Inc.
- Gorbatenko, V. (1999). *Wrapper for AVICap Window*. Diakses pada 3 Desember 2004 dari <<http://www.codeguru.com/Cpp/G-/multimedia/video/article.php/c1601/>>.
- Stallings, W. (2000). *Data and Computer Communications 6th Edition*. New Jersey: Prentice-Hall, Inc.
- Microsoft (2004). *Windows Sockets 2 Architecture*. Diakses pada 23 November 2004 dari <ms-help://MS.MSDNQTR.2002APR.1033/winsoc/ovrvw1\_5kky.htm>.