

PEMANFAATAN VIDEO STREAMING PADA SISTEM KEAMANAN RUMAH DENGAN MENGGUNAKAN MOBILE DEVICES

Samuel Mahatmaputra

Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Bina Nusantara University
Jln. K.H. Syahdan No 9, Palmerah, Jakarta Barat 11480
samuelmt@binus.edu

ABSTRACT

This research has been produced a function prototype of video streaming with JPEG/MJPEG picture quality in a residential security system using mobile device (cell phone). The security system is an early warning system which server will deliver text messaging to client if there is an event in the sensor of home webcam to detect movement. The prototype is divided into two types, server and client. The client will be installed a mobile application that automatically activated if there is inbound message in specific port. Meanwhile, the application in user client is completed with functionality to visit particular webpage, one touch call to security phone numbers to do setting (previous research result). And, as the research objective this year, user will see the video streaming in a good quality from the object.

Keywords: cell phone, security system, client, server, video streaming.

ABSTRAK

Penelitian berikut ini telah menghasilkan sebuah prototype fungsi video streaming dengan kualitas gambar JPEG/MJPEG pada sistem keamanan hunian dengan menggunakan mobile device (ponsel). Sistem keamanan yang dimaksud adalah early warning system di mana server akan dengan segera mengirimkan pesan SMS kepada klien bila terjadi event pada sensor; dalam hal ini berupa home webcam untuk deteksi gerakan. Prototype tersebut dibagi menjadi dua jenis yaitu server dan klien. Pada klien akan di-install sebuah aplikasi mobile yang otomatis teraktifasi bila terdapat inbound SMS pada port tertentu. Aplikasi pada klien user dilengkapi dengan fungsionalitas untuk mengunjungi website tertentu, one touch call untuk nomor-nomor security tertentu, untuk melakukan setting (hasil penelitian terdahulu). Dan, yang merupakan fokus penelitian tahun ini, user dapat melihat hasil video streaming dengan kualitas gambar yang baik dari objek pemantauannya.

Kata kunci: ponsel, sistem keamanan, klien, server, video streaming.

PENDAHULUAN

Kebutuhan masyarakat akan sistem pemantauan keamanan hunian telah meningkat kepada kebutuhan akan pemantauan yang bersifat visual baik itu monitoring pada keadaan normal maupun ketika terjadi event pada sensor keamanan. Dalam hal ini dibutuhkan sebuah sistem yang dapat memonitor secara visual hunian/gedung dari lokasi yang jauh. Penerapan teknologi yang paling sesuai adalah streaming video menggunakan mobile communication pada hand phone dan berbagai variasinya. Pada prakteknya teknologi *video streaming* terbagi menjadi beberapa versi dan jenis *encoding*, namun penerapannya pada jaringan komunikasi di Indonesia memiliki tantangan tersendiri mengingat pangsa pasar terbesar pengguna mobile phone adalah user dengan bandwidth yang cukup sempit/kecil. Untuk itu diperlukan *customization* tertentu pada sisi klien sehingga tetap dapat menikmati fasilitas tersebut dengan sumber daya yang terbatas.

METODE

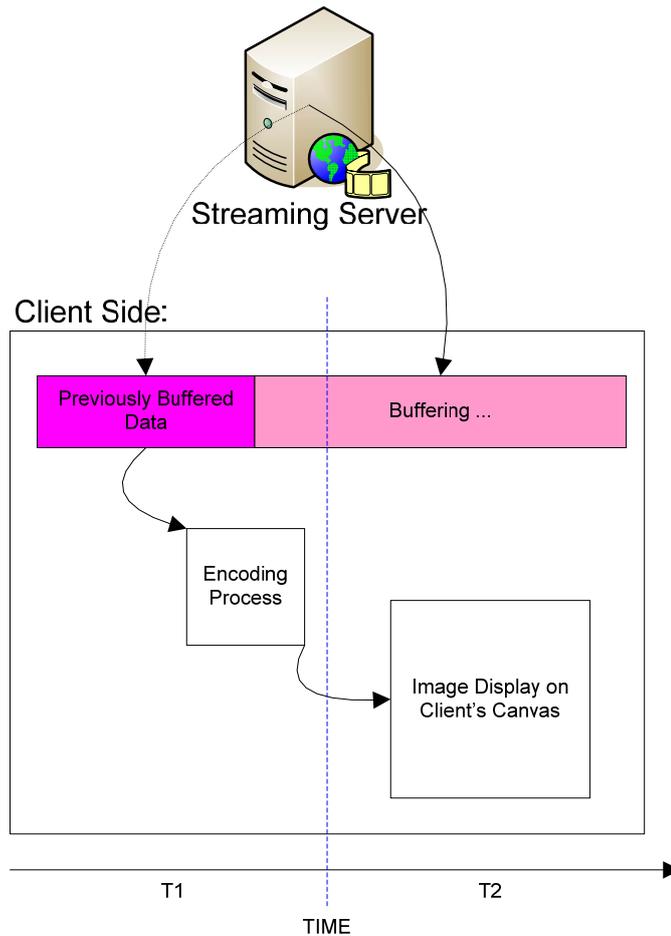
Penelitian ini akan menitik beratkan kepada streaming server sebagai penyedia *live monitoring* terhadap obyek hunian atau ruangan. Setelah dihasilkannya prototype sistem peringatan dini (*alert system*) pada penelitian terdahulu, streaming server ini akan melengkapi kemampuan server untuk mengirimkan object video ke *mobile devices* setiap user secara *live*.

HASIL DAN PEMBAHASAN

Pendekatan Streaming

Di dalam dunia teknologi informasi, streaming adalah proses mengirimkan data melalui koneksi jaringan (TCP/IP) yang mana data dapat langsung dikonsumsi oleh klien tanpa harus menunggu seluruh data selesai di kirimkan oleh server. Metoda ini sangat sesuai dengan kebutuhan transfer data multimedia (video, suara) di mana data multimedia dapat langsung dinikmati seiring dengan proses transfer data yang sedang berlangsung secara bersamaan. Hal ini dimungkinkan oleh beberapa protocol yang melengkapinya, diantaranya (1) datagram atau yang sering juga di sebut UDP (User Datagram Protocol) packet - merupakan teknologi paling awal pada teknologi streaming. Perbedaan utama dengan TCP (*Transmission Control Protocol*) adalah data tidak ada mekanisme yang menjamin urutan kedatangan packet serta kepastian data sampai di tujuan; (2) *Real-time Streaming Protocol* (RTSP), Real-time Transport Protocol (RTP) and the *Real-time Transport Control Protocol* (RTCP) adalah teknologi yang dibangun diatas UDP untuk menyempurnakan proses streaming.

To stream dapat diartikan sebagai mengalirkan. Teknologi *streaming* memungkinkan data yang dialirkan dari server kepada klien dapat dinikmati seketika tanpa menunggu seluruh data selesai dialirkan (*end of file*), yang pada *live video streaming* (e.g. *live.sdp*) notabene tidak memiliki *end of file/image*. Aliran data visual (*video streaming*) merupakan aliran data biner yang akan di-encoder yang kemudian ditampilkan pada klien. Untuk dapat menikmati gambar bergerak dengan tingkat *delay* yang rendah, idealnya sebuah proses streaming akan ditampung terlebih dahulu pada memory klien (*buffering*) sehingga proses download rangkaian gambar selanjutnya dapat disamakan di belakan proses menampilkan isi *buffer* yang telah di-download sebelumnya.



Gambar 1 Proses Buffering pada Video Streaming

Pada gambar 1 di atas terlihat proses buffering (time = T2) terjadi pada latar belakang proses penampilan gambar (Image display on client's canvas). Rangkaian gambar yang ditampilkan pada time = T2 diambil dari proses buffering time = T1 setelah mengalami proses *encoding*.

RTP (Real-time Transport Protocol)

Merupakan protocol standar yang mendefinisikan format paket untuk mengirimkan (streaming) data suara dan gambar melalui Internet. Dikembangkan oleh IETF pada tahun 1996 dengan kode RFC (*request for comments*) RFC1889. RTP biasanya digunakan bersamaan dengan RTSP (sehingga memungkinkan *VCR like operation*). Service yang disediakan oleh RTP adalah (1) *payload-type identification*, indikasi data yang sedang dibawa; (2) *sequence numbering*, PDU *sequence number*; (3) *time stamping*, memungkinkan sinkronisasi dan *jitter*; (4) *delivery monitoring*. RTP didukung oleh RTCP untuk mengindikasikan kualitas penerimaan sehingga tindakan selanjutnya dapat diambil seperti menyesuaikan *data rate*.

+ Bits	0-1	2	3	4-7	8	9-15	16-31
0	Ver.	P	X	CC	M	PT	Sequence Number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifiers (optional)						

	...
96+(CC×32)	Extension header (optional).
96+(CC×32) + (X×((EHL+1)×32))	Data

Gambar 2 – Susunan RTP packet

Pada umumnya RTP dikonfigurasi pada *port number* dengan range: 16384 s/d 32767.

JPEG dan MJPEG

JPEG merupakan salah satu standar *image encoding* yang paling banyak digunakan pada dunia internet utamanya pada HTTP protocol. JPEG merupakan singkatan dari Joint Photographic Experts Group yaitu nama dari komite penyusun standar tersebut yang didirikan pada tahun 1986 dengan dukungan standarisasi ISO 10918-1 pada tahun 1994. Standar JPEG menjelaskan mekanisme *codec* dan *encodec* sebuah gambar kedalam byte stream dan kembali ke gambar, serta format file untuk menampung gambar tersebut.

MJPEG (*Motion JPEG*) merupakan proses *compression*/rangkaiannya gambar-gambar (JPEG) sebagai sebuah video. MJPEG menggunakan *intraframe coding technology* sehingga kualitas gambar pada video yang dihasilkan sangat baik - mendekati kualitas gambar tidak bergerak (*still image*) sebuah file JPEG. Namun kekurangannya adalah kemampuan kompresi data yang rendah yang dimiliki oleh teknologi *interframe prediction*, berakibat pada tingginya volume data. Untuk tujuan *video surveillance*, kualitas data/image merupakan prioritas utama mengingat transfer gambar dilakukan rata-rata satu kali dalam setiap detik (1 *fps-frame per second*) dengan kualitas gambar yang baik.

Multi-threading

Thread adalah mekanisme Java untuk menjalankan beberapa proses menggunakan processor yang sama namun seolah-olah dieksekusi secara parallel/bersamaan. Setiap thread merupakan sub-class dari *Thread class* ataupun *Runnable interface*. Berikut ini adalah contoh implementasi thread sebagai sub-class dari *Thread class*:

```
class MyThread extends Thread {
    MyThread(String param1) {
        // constructor
    }
    public void run() {
        // Program yang akan dieksekusi pada saat          // thread
dijalankan
        . . .
    }
}
```

Atau dengan mengimplementasikan *Runnable interface*:

```
class MyThread implements Runnable {
    MyThread(String param1) {
        // constructor
    }
    public void run() {
        // Program yang akan dieksekusi pada saat          // thread
dijalankan
        . . .
    }
}
```

Untuk membentuk sebuah object MyThread pada kedua contoh diatas dapat menggunakan kode berikut ini:

```
MyThread p = new MyThread("Thread pertama");  
p.start(); // run menggunakan Thread Class  
new Thread(p).start(); //run
```

GPRS (General Packet Radio Service)

Seperti halnya teknologi 2G lainnya teknologi GSM memiliki karakteristik yang dapat menangani suara secara efisien namun memiliki keterbatasan dalam kemampuan *transfer* data aplikasi internet. Komunikasi data pada GSM dijalin melalui mekanisme *circuit-switched connection* yang berarti hubungan diawali dengan *dial* dari pengguna dan diakhiri dengan pemutusan hubungan. Apabila pengguna ingin mengakses data lagi maka mereka harus melakukan *dial* ulang. Hal inilah yang menjadikan keterbatasan GSM dimana pengguna akan selalu dibebani biaya koneksi selama waktu mereka mengakses data. Untuk memecahkan masalah tersebut maka dibutuhkan teknologi paket data dengan menggunakan *packet-switched connection*. Teknologi *packet-switched connection* untuk GSM diperkenalkan pertama kali dengan diluncurkannya *General Packet Radio Service* (GPRS).

GPRS merupakan teknologi *overlay* yang disisipkan di atas jaringan *GSM* untuk menangani komunikasi data pada jaringan dengan *packet-switched connection*. Dengan kata lain dengan menggunakan *GPRS*, komunikasi data tetap berlangsung di atas jaringan *GSM* dimana komunikasi suara masih ditangani oleh *GSM* dan *transfer* data ditangani oleh *GPRS*.

Pengembangan teknologi *GPRS* di atas *GSM* dapat dilakukan secara efektif tanpa menghilangkan infrastruktur lama, yaitu dengan penambahan beberapa *hardware* dan *upgrade software* baru pada terminal dan server *GSM*. Kecepatan *transfer* data *GPRS* dapat mencapai hingga 160 *kbps*. Teknologi *GPRS* memiliki 3 fitur keunggulan, yaitu (1) *always online*, *GPRS* menghilangkan mekanisme *dial* kepada pengguna pada saat ingin mengakses data, sehingga dikatakan *GPRS* selalu online karena *transfer* data dikirim berupa paket dan tidak bergantung pada waktu koneksi; (2) *an upgrade to existing networks* (*GSM*), sistem *GPRS* tidak perlu menghilangkan sistem lama karena *GPRS* dijalankan di atas infrastruktur yang telah ada (*GSM*); (3) *an integral part of EDGE and WCDMA*, *GPRS* merupakan inti dari mekanisme pengiriman paket data untuk teknologi 3G selanjutnya.

Riset Pendahuluan

Untuk dapat mengakses rangkaian gambar dari sebuah kamera pengawas (TCP/IP enabled Camera) user memiliki beberapa opsi yang sudah tersedia saat ini di pasar, diantaranya melalui browser ataupun streaming client pada PC/Laptop (HTTP dan RTP) dan melalui built-in media player pada handphone/ PDA (RTP).

Opsi pertama yaitu dengan menggunakan PC/Laptop telah menjadi opsi terbanyak user saat ini, hal ini diakibatkan oleh telah mature nya teknologi yang tersedia serta besarnya sumber daya (memory capacity, processor speed, graphic compatibility) yang dimiliki oleh penggunaan PC/Laptop. Namun dengan seiring berkembangnya teknologi mobile communication, maka tuntutan user pun semakin meningkat dengan pemanfaatan hand-held devices seperti handphone ataupun PDA.

Sejalan dengan hal tersebut, opsi kedua yaitu pemanfaatan builtin media player dari sebuah hanphone untuk dapat mengakses live video streaming telah menjadi trend baru di dalam dunia video steaming utamanya pada *live video streaming*. Hand-held devices seperti ponsel/PDA tersebut merupakan alat yang sangat portable walaupun dibandingkan dengan perangkat portable lainnya yaitu Laptop. Sebuah hanphone dapat mengakses streaming server melalui RTP (*Real Time Protocol*) diatas GPRS. Streaming yang dikirimkan memiliki 2 channel yaitu video dan sound, dengan kualitas gambar yang seadanya menimbang bandwidth yang sempit dan sharing dengan slot *bandwidth* untuk sound/suara. Format video streaming yang paling optimum saat ini adalah MPEG4, sebuah metode kompresi data dan suara yang paling terakhir oleh Moving Picture Experts Group (MPEG) dengan menggunakan standard ISO/IEC 14496.



(a)



(b)

Gambar 3 (a,b) – Perbandingan Kompresi MPEG4 dan JPEG
http://www.vivotek.com/products/live_demo.php

Perbedaan yang sangat kontras dapat terlihat dari kedua gambar diatas. Gambar (a) merupakan hasil streaming menggunakan RTP/RTSP dan MPEG4 dimana kualitas gambar tidak terjaga dengan baik dengan banyaknya *pixel aliasing*. Gambar (b) merupakan still image dalam format JPEG dari kamera pengawas yang sama dengan kualitas gambar yang lebih baik. Dari perbandingan kualitas gambar tersebut diatas, format JPEG/MJPEG merupakan format yang ideal untuk live video surveillance streaming. Berikut adalah tabel perbandingan kedua opsi tersebut:

Tabel 1 Perbandingan MPEG4 dan JPEG/MJPEG

	MPEG4	JPEG/MJPEG
image	poor	Excellent
sound	yes	no
available to PC	yes	yes
available to Mobile	yes	no
bandwidth	high	low

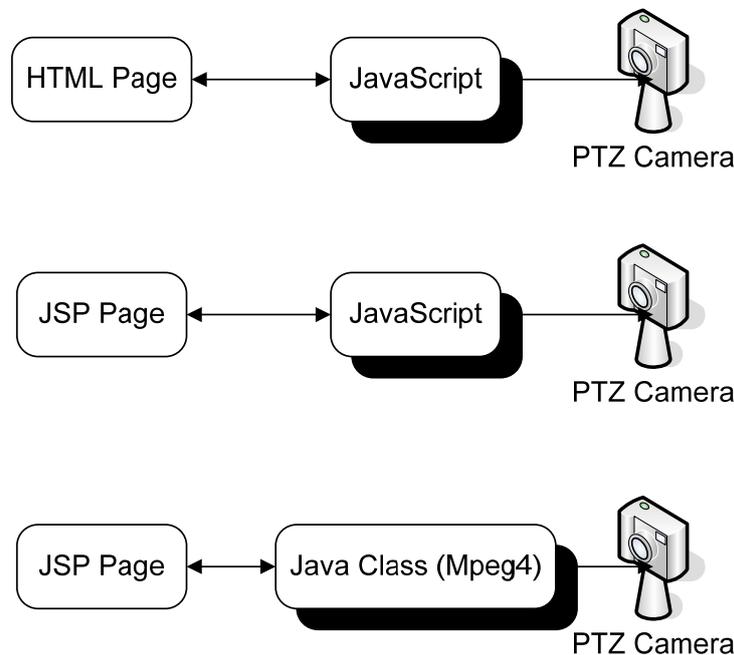
Sampai saat ini hanya terdapat satu produk komersil yang mendukung JPEG/MJPEG pada *mobile device* yang baru saja diluncurkan dengan beberapa keterbatasan, yaitu (1) *passive activation*, tidak dapat aktif secara otomatis bila terdapat event pada sensor/IP Camera, user harus secara manual mengaktifkan program setiap kali hendak mengakses video streaming; (2) *low security*, hanya dapat mengakses IP Camera yang tidak memiliki user name, dan password; (3) rasio gambar mengandalkan built-in ratio pada streaming server, bila terdapat kamera yang tidak mendukung resolusi rendah maka gambar yang besar akan ditampilkan terpotong pada layar mobile phone yang kecil; (4) tanpa suara.

Penelitian ini akan menjawab ketiga point pertama. Fasilitas suara pada saat ini telah tersedia dengan cukup memadai menggunakan RTP/RTSP.

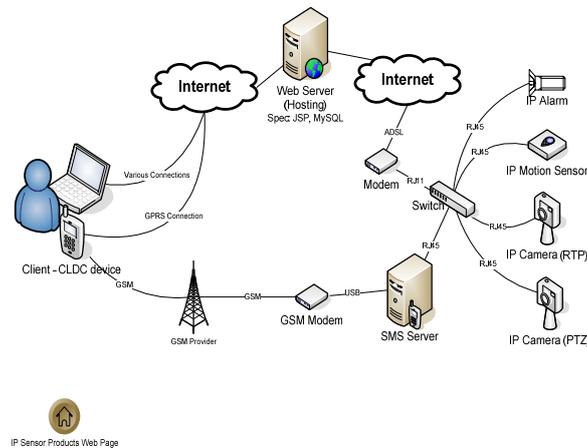
Perancangan Infrastruktur Streaming System

Awal perancangan streaming sistem dimulai dari klien yang direncanakan akan memanfaatkan built-in browser pada mobile phone, sehingga dapat langsung menampilkan active-X object (VB/c++) dari IP Camera. Namun tidak semua *mini web browser* yang built-in pada mobile devices memiliki kemampuan tersebut. Telaah berikutnya adalah usaha untuk memanfaatkan kemampuan javascript pada mini browser sebuah hand phone, namun gambar yang dihasilkan tidak dapat *stay on the screen* pada saat gambar pada frame selanjutnya sedang didownload, karakteristik umum sebuah web client/browser. Solusi selanjutnya yang menjadi pertimbangan adalah menggunakan java class pada JSP untuk menampilkan hasil streaming, namun dengan hasil yang serupa.

Pertimbangan yang akhirnya diambil adalah dengan tidak menggunakan built-in browser pada mobile phone dan menggantinya dengan menampilkan gambar hasil streaming menggunakan *canvas* pada aplikasi MIDLet (java MIDP/J2ME). Dengan menggunakan MIDLet gambar dapat *stay on the screen* pada saat back-ground process sedang berlangsung.



Gambar 4 Tiga Opsi streaming terdahulu menggunakan internet klien



Gambar 5 Gambar Konfigurasi Sistem pada SMS server dan Streaming Server

Pada gambar sistem konfigurasi diatas dapat dilihat konfigurasi yang digunakan pada penelitian ini. Beberapa alternatif konfigurasi lainnya dapat dilihat pada Lampiran-2. Konfigurasi yang digunakan memisahkan secara fisik antara web server, streaming server dan SMS server. Pemisahan tersebut dimaksudkan agar pengembangan serta implementasi masing-masing komponen dapat dilakukan secara modular.

Perancangan Fungsi Video Streaming (JPEG/MJPEG) MIDLet

Terdapat dua opsi untuk dapat mengirimkan gambar ke sebuah mobile klien yaitu dengan format JPEG dan MJPEG. Pada opsi pertama klien akan meminta kode biner dalam format satu buah gambar (*still image*) JPEG dari server (*camera streaming server*). Pada opsi ini proses *network handshake* dilakukan pada awal dan akhir pengiriman setiap gambar. Permintaan terhadap gambar JPEG akan dilakukan berulang-ulang - hingga user melakukan perintah stop, yang memberikan efek gambar bergerak pada layar klien. Akibat proses handshake yang dilakukan 2 kali untuk setiap gambar, maka durasi frame untuk gambar bergerak yang dihasilkan cukup lambat. Beberapa percobaan membuktikan durasi akan berlangsung 0.25 – 1 fps (*frame per second*). Namun hal ini cukup mendukung pemanfaatannya dalam pemantauan keamanan hunian secara visual dimana kecepatan penampilan frame yang dibutuhkan adalah pada order 1 fps.

Opsi kedua yaitu dengan menggunakan format MJPEG. Opsi ini merupakan *the real video streaming* karena kode-kode biner yang dikirimkan oleh server merupakan rangkaian beberapa buah gambar, dalam format MJPEG, sekaligus dalam satu buah aliran/ streaming yang tidak terputus. Handshake hanya dilakukan pada awal dan akhir proses streaming, yaitu pada saat klien melakukan request dan pada saat melakukan perintah stop. Durasi yang dihasilkan sangat baik yaitu 2 – 5 fps sehingga gambar terlihat lebih hidup. Kelemahan sistem ini adalah pada jaringan yang tidak stabil seperti beberapa fasilitas GPRS di Indonesia (utamanya jaringan komunikasi diluar pulau Jawa dan Bali). Untuk hasil yang optimal disarankan untuk menggunakan bandwidth dengan kapasitas 512 Kbps *download* (klien side), dan 128 upload (*server side*). Proses buffering pada jaringan dengan spesifikasi dibawah rekomendasi tersebut akan memakan waktu cukup lama (2 -5 menit), dan pada kasus yang lebih buruk akan mengakibatkan gagal sistem.

Tabel 2 Perbandingan JPEG dan MJPEG

	JPEG	MJPEG
Network robustness	high	low
Frame per second	0.25 - 1 fps	2 - 5 fps
Upload bandwidth (recommended)	32 Kbps	128 Kbps
Download bandwidth (recommended)	128 Kbps	512 Kbps
Average cost per second	cost_unit x 0.625 x	cost_unit x 3.5 x

Untuk jaringan di dalam pulau Jawa dan Bali, opsi kedua menggunakan MJPEG dapat merupakan solusi yang baik, namun untuk jangkauan yang lebih luas sangat disarankan untuk menggunakan opsi pertama yaitu format gambar JPEG. Pada tabel diatas dapat dilihat perbandingan antara penggunaan JPEG dan MJPEG utamanya dalam hal biaya perdetik yang berbeda kurang lebih empat kali lipat. Proses yang terjadi pada saat streaming menggunakan format JPEG dapat dilihat melalui *program snippet (Java)* berikut ini:

```
while (true) { // repeat forever
    Get_picture_from_http(url,
        isUsePassword );
    myThread.sleep(100);
    Paint_picture();
}
```

Program tersebut di tempatkan pada method run() sebuah Thread sehingga proses koneksi ke server dan proses menggambar pada client's canvas dapat dilakukan bersamaan. Untuk menggunakan opsi kedua proses yang terjadi pada method run() adalah sebagai berikut:

```
Connect_to_server(url, isUsePassword)
while (true){ // repeat forever
    Get_frame_size();
    Get_JPEG();
    Get_JPEG_Boundary();
    Paint_picture();
}
```

Pada kedua code snippets diatas terlihat jelas perbedaan koneksi internet ke server dimana pada opsi pertama koneksi terjadi berulang-ulang untuk setiap gambar. Hal ini selain berakibat negatif dengan lambatnya penampilan setiap gambar juga berakibat positif yaitu sistem dapat digunakan pada bandwidth yang rendah dan tidak stabil, karena setiap gambar merupakan transaksi (*handshake*) yang berbeda. Sedangkan pada opsi kedua method connect_to_server() hanya digunakan sekali pada sebuah sesi streaming. Gambar-gambar JPEG akan disarikan keluar dari kode-kode biner MJPEG yang notabene merupakan kumpulan rangkaian gambar JPEG. Berikut ini adalah gambar hasil streaming kedua metode tersebut di atas Gambar 8.

Untuk kepentingan keamanan atau privasi maka setiap IP Camera sebaiknya mengaktifkan login sistemnya, sehingga hanya authorized users saja yang dapat mengakses gambar tersebut. Untuk dapat mengambil streaming dibelakang sebuah login sistem maka MIDlet ini telah diberikan kemampuan untuk melakukan user name dan password decode pada saat melakukan request ke sebuah steaming server. Bila picture resolution yang dihasilkan memberikan gambar dengan luasan melewati ukuran layar sebuah mobile device maka gambar tersebut akan di resize mengikuti ukuran layar klien device tersebut. MIDlet tersebut juga telah dilengkapi dengan *autorun* (penelitian terdahulu: Samuel, Perancangan Pemantauan Sistem Keamanan Rumah dengan Menggunakan Mobile Devices) bila terdapat pesan SMS yang dikirimkan oleh server untuk trigger yang terjadi pada sensor kamera.



Gambar 8 Hasil streaming pada Java mobile phone emulator
(address <http://60.248.39.152:1025/>)

PENUTUP

Dapat disimpulkan bahwa melalui penelitian ini pemanfaatan media video streaming pada sistem pemantauan/ monitoring untuk tujuan keamanan melalui berbagai mobilie devices (hand phone/ PDA/ Laptop) telah dapat dimungkinkan dengan dukungan Java Technology yang telah *built-in* pada masing-masing device khususnya *device* dengan sumber daya terbatas seperti ponsel.

Penelitian ini belum memasukkan suara sebagai media streaming. Untuk itu disarankan pada penelitian selanjutnya untuk dapat menggunakan suara pada setiap frame streaming yang dikirimkan beserta dengan proses sinkronisasi antara suara dan gambar yang dihasilkan.

DAFTAR PUSTAKA

- Anonymous. (2001). A MIDlet that displays the Doggy animation. Java2S, from <http://www.java2s.com/Code/Java/J2ME/AMIDletthatdisplaystheDoggyanimation.htm>
- Anonymous. Embedding Streaming Video in a Web Page. Mediacollege.com, from <http://www.mediacollege.com/video/streaming/embed/>
- Anonymous. (2007). Learning The Java Language - The Java Tutorials. Sun Micro System, from <http://java.sun.com/docs/books/tutorial/index.html>
- Knudsen, J. (March, 2003). Creating 2D Action Games with the Game API. Sun Developer Network, from <http://developers.sun.com/mobility/midp/articles/game/>

- Knudsen, J. Image Loader. Apress. ISBN: 1590590775, from <http://www.java2s.com/Code/Java/J2ME/ImageLoader.htm>
- Knudsen, J., & Nourie, D. (January, 2006). Wireless Development Tutorial. Sun Developer Network from <http://developers.sun.com/mobility/midp/articles/wtoolkit/>
- Künneht, T. (February, 2005). Low-Level Display Access in MIDlets. Java.Net, from <http://today.java.net/pub/a/today/2005/02/02/J2ME-4.html>
- Mahmoud, Q. H. (June, 2003). The J2ME Mobile Media API. Sun Developer Network, from <http://developers.sun.com/mobility/midp/articles/mmapioverview/>
- Network Working Group. Real Time Streaming Protocol (RTSP). RFC: 2326 / Standards Track, from <http://tools.ietf.org/html/rfc2326>
- Perdita, S., & Pooley, R. (2000). Using UML Software Engineering with Objects and Components. Addison-Wesley.
- Stevens, R., Brook, P., Jackson, K., & Arnold, S. (1998). System Engineering Coping with Complexity. Pearson – Prentice Hall.
- Zukowski, J. (1998). Java Tip 47: URL Authentication Revisited. JavaWorld, from <http://www.javaworld.com/javaworld/jvatips/jw-jvatip47.html>