

PERANCANGAN SISTEM PEMANTAU RUANGAN DENGAN WEBCAM MENGGUNAKAN BAHASA PEMROGRAMAN VISUAL BASIC .NET

Rian Aldy Hidayat^{*)}, Kodrat Imam Satoto^{*)}, Adian Fatchur Rochim^{*)}

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH. Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}Email : rian.aldy.hidayat@gmail.com, kodrat@gmail.com, adian@undip.ac.id

Abstrak

Saat ini, keamanan menjadi sesuatu yang susah didapat dan mahal harganya di kehidupan manusia. Tuntutan ekonomi yang semakin tinggi semakin menambah beban bagi warga yang kehidupan ekonominya dibawah standar. Hal ini mengakibatkan meningkatnya tindak kejahatan seperti pencurian. Sedangkan manusia tidak bisa terus menerus memantau tempat yang dianggapnya kurang aman. Teknologi keamanan yang banyak digunakan sekarang adalah CCTV (*Closed Circuit Television*). Namun, harga CCTV tidaklah murah dan tidak semua orang sanggup membelinya. Alternatifnya adalah menggunakan *web camera* atau *webcam* yang harganya jauh lebih murah dari CCTV. *Webcam* adalah kamera video digital kecil yang dihubungkan ke komputer melalui *port* USB maupun serial. Langkah yang dilakukan dalam menyusun Tugas Akhir ini adalah sebagai berikut. Langkah pertama adalah melakukan studi literatur tentang *webcam* dan bahasa pemrograman Visual Basic .NET, serta melakukan pengamatan pada aplikasi yang sudah ada untuk dijadikan acuan. Langkah kedua adalah melakukan perancangan dengan *Unified Modelling Language* (UML) untuk mewujudkan aplikasi tersebut. Langkah ketiga adalah mengimplementasikan menjadi sebuah produk berupa program aplikasi. Aplikasi dibuat dengan bahasa pemrograman Visual Basic .NET. Berdasarkan dari hasil pengujian yang dilakukan, diperoleh kesimpulan bahwa aplikasi ini dapat digunakan untuk memantau kondisi ruangan. Pengguna dapat menyimpan video selama proses pemantauan berlangsung dan dapat mengaktifkan fitur pendeteksi gerak yang apabila terdeteksi gerakan di area pemantauan dapat menghidupkan alarm dan menangkap gambar selama terdeteksi gerakan. Selain itu pengguna dapat menyiarkan video dari *server* dan melihat secara *streaming* di aplikasi *client* dimanapun pengguna berada.

Kata Kunci : Webcam, Visual Basic .NET, Pemantauan

Abstract

Today, security is something that hard to find and expensive in human life. Economic demand was increase from day to day add burden for people that have economy below for standard. This is cause increase crime such theft. While humans can not continuously monitor the placed that less secure. The popular security technology used is CCTV (*Closed Circuit Television*). However, CCTV price is not cheap and not everyone can buy it. An alternative is use web camera or webcam that cost far cheaper than CCTV. Webcam is a small digital video camera that connected to computer via USB or serial port. Steps taken in preparing this Final Exam is as follows. The first step is to conduct studies in the literature about the email client in order to find out how it works, as well as making observations on the application of an existing email client to be used as a reference. The second step is to design with *Unified Modeling Language* (UML) to realize these application. The third step is to implement the design into a product of the application program. Applications created with Visual Basic .NET. Based on the result, Based on the test results, it is concluded that this application can be used to monitor the condition of the house. Users can save the videos during the process of monitoring progress and can activate the motion detection feature that if movement is detected in the monitoring area to turn the alarm on and capture images during motion is detected. In addition, users can broadcast video from server and streaming at the client application wherever user is.

Keyword : Webcam, Visual Basic .NET, Monitoring

1. Pendahuluan

Semakin meningkatnya populasi penduduk Indonesia membuat tingginya tingkat pengangguran dan

kemiskinan. Tuntutan ekonomi yang semakin banyak menambah beban bagi warga miskin. Hal ini mengakibatkan meningkatnya tindak kejahatan seperti pencurian sehingga membuat orang semakin susah menemukan rasa aman. Sedangkan manusia tidak bisa

harus terus menerus memantau lokasi dalam jarak pandang mata. Teknologi keamanan yang banyak digunakan sekarang adalah CCTV (*Closed Circuit Television*). CCTV adalah sebuah kamera yang digunakan untuk memantau dan merekam segala aktifitas yang terjadi di tempat dimana kamera tersebut ditempatkan. Perkembangan terbaru dari CCTV adalah *IP Camera*. *IP Camera* adalah CCTV kamera yang menggunakan *Internet Protocol (IP)* untuk mengirimkan data gambar dan sinyal kendali. Yang membedakan dengan CCTV adalah setiap kamera memiliki *IP* sendiri sehingga bisa di akses melalui jarak yang jauh dengan menggunakan koneksi internet. Namun, harga sebuah *IP Camera* yang mempunyai fitur-fitur canggih relatif lebih mahal dari CCTV dan tidak semua orang sanggup membelinya.

Salah satu alternatifnya adalah menggunakan *web camera* atau yang sering disebut dengan *webcam*. *Webcam* adalah kamera video digital kecil yang dihubungkan ke komputer melalui port USB atau serial. Fungsi *webcam* yang paling populer adalah melakukan *video conference* melalui internet. Dalam perkembangan selanjutnya, *webcam* tidak hanya sebagai alat *video conference* semata, namun juga untuk *home monitoring* atau untuk memantau rumah selama 24 jam. Oleh karena itu pada penelitian tugas akhir ini, penulis akan merancang perangkat lunak yang dapat membantu kegiatan pemantauan ruangan menggunakan *webcam*.

2. Visual Basic .NET

Visual Basic adalah salah satu bahasa pemrograman berbasis *desktop* yang dikeluarkan oleh perusahaan perangkat lunak komputer terbesar yaitu Microsoft. Pada dasarnya Visual Studio .NET didesain untuk menampung berbagai macam bahasa pemrograman dan terlingkup dalam Visual Studio .NET. Dengan demikian, dapat membangun aplikasi-aplikasi Windows di dalam Visual Studio .NET. Visual Basic .NET merupakan salah satu bahasa pemrograman yang dapat digunakan untuk membuat program aplikasi. Kelebihan Visual Basic .NET adalah sebagai berikut :

- Dukungan *Object Oriented Programming* : VB.NET adalah bahasa pemrograman yang *full Object Oriented*. Jadi VB.NET mendukung fitur-fitur OOP seperti *Inheritance*, *Interface*, *Method Overloading*, dan *Polymorphism*.
- *Structure Exception Handling* : untuk menggantikan perintah *OnError Goto* pada VB6, VB.NET menyediakan “*Try..Catch..Finally*” *error handling*. *Error handling* pada VB.NET ini lebih mudah digunakan karena anda hanya cukup menaruh kode yang akan dicek di area “*Try*”, dan menyiapkan *exception handling* nya di area “*catch*”.
- *NET Framework* : VB.NET mempunyai koleksi *class library* yang sangat banyak dan terorganisasi dengan baik sehingga mudah digunakan dan dicari. Dengan

menggunakan *class library* ini programmer tidak perlu membuat kode sendiri dari awal.

- GDI+ : GDI+ adalah *library Graphic* yang digunakan untuk mengembangkan aplikasi Windows *form*.
- *Web Services dan Web Form* : dengan VB.NET anda dapat membuat aplikasi berbasis web dengan menggunakan *Web Form (ASP.NET)*. anda juga dapat membuat aplikasi *web service* untuk membuat *three tier application*.
- GUI : Visual Basic .NET adalah bahasa pemrograman berbasis *Graphic User Interface (GUI)* dimana pengguna dapat menggunakan piranti dengan *drag and drop*. Hal ini memungkinkan pembuat program untuk menciptakan aplikasi-aplikasi berbasis Windows yang menarik.

3. AVICAP32.dll

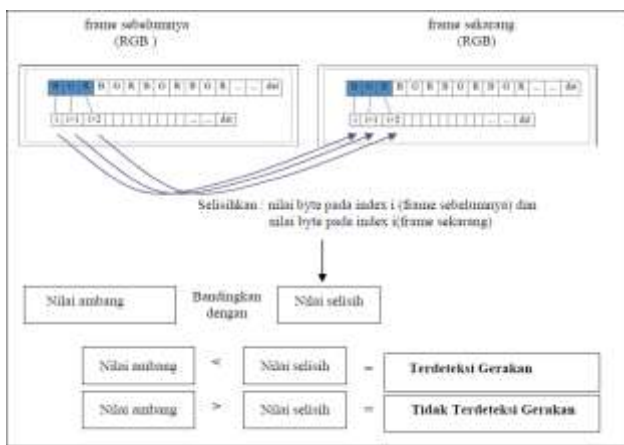
Windows pertama kali mengeluarkan teknologi *video capture* dalam bentuk API (*Application Programming Interface*) yang dikenal dengan VFW (*Video for Window*). Versi pertama yang dikeluarkan adalah 16 bit tersimpan dalam berkas *library* “*avicap.dll*”. Kemudian diperbaharui dalam versi 32 bit yang tersimpan dalam “*avicap32.dll*”. Teknologi VFW ini sangat mudah, baik dalam algoritmanya/prosedur pemrograman maupun dalam pemrogramannya. Hal ini memungkinkan VFW dapat diimplementasikan menggunakan berbagai bahasa pemrograman, asalkan dapat melakukan pemanggilan fungsi API. Berikut ini langkah-langkah yang harus dilakukan untuk menggunakan VFW:

1. Apakah ada kamera yang terpasang, dengan cara membaca semua *driver* kamera yang terpasang. *Driver* kamera diberi nomor 0 sampai seterusnya. Setiap nomor *driver* dibaca jenis dari kameranya, dan jika ada penjelasannya, maka ada *driver* yang terpasang. Pencarian akan terus dilakukan sampai tidak di temukan lagi *driver* terpasang.
2. Memilih, *driver* mana yang akan digunakan.
3. Membuka jendela untuk menampilkan “*preview*” dari kamera.
4. Menghubungkan *driver* dan menyambungkannya dengan jendela *preview*.
5. Melakukan beberapa pengaturan
 - a. Mengatur tampilan
 - b. Mengatur kecepatan

Apabila program berjalan dengan benar, maka setiap *video capture* berhasil menangkap satu gambar, maka data *frame* akan disimpan pada suatu *buffer* yang terdapat pada API *vidcap*. Setelah itu, API *vidcap* akan mengirimkan sinyal *trigger* yang akan mengaktifkan *event OnFrame*. Jika pada program Inisialisasi disertakan setting *OnFrame event callback*, maka program secara otomatis akan loncat ke rutin yang telah ditunjuk, misalkan pada suatu subrutin *OnFrameEvent*.

4. Metode Pendeteksi Gerakan

Deteksi gerakan secara sederhana dapat dilakukan dengan mencari beda antara 2 buah citra yang berurutan pada hasil pencitraan menggunakan *webcam*. Untuk mengetahui beda antara 2 buah citra tersebut maka operasi yang digunakan adalah pengurangan. Operasi pengurangan pada bagian yang tidak bergerak dalam citra akan menghasilkan nilai RGB (*red, green, blue*) per piksel sama dengan nol, sedangkan bagian yang bergerak dalam citra memberikan nilai RGB per piksel tidak sama dengan nol. Dengan mengevaluasi selisih nilai RGB per piksel, dapat diketahui apakah citra terdapat objek yang bergerak.



Gambar 1. Ilustrasi proses pendeteksi gerak

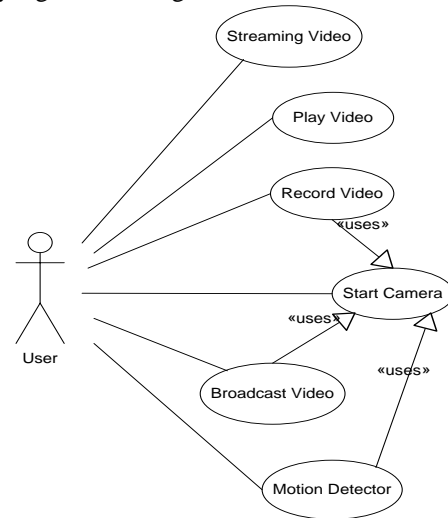
Langkah-langkah yang dilakukan dalam pendeteksi gerak yang pertama adalah menangkap warna pada citra dengan menangkap warna RGB (*red, green, blue*) per piksel pada *frame* saat ini dan *frame* sebelumnya. Selanjutnya, melakukan penghitungan selisih komponen nilai RGB dengan nilai ambang. Selisihkan nilai komponen RGB masing-masing index pada *frame* sekarang dengan *frame* sebelumnya (ilustrasi pada Gambar 2.7). Bandingkan masing-masing selisih dari tiap piksel dengan nilai ambang yang telah ditentukan. Apabila nilai selisih dari tiap piksel melebihi batas ambang yang telah ditentukan, maka dianggap telah terjadi pergerakan. Jadi semakin besar nilai ambang yang diatur, maka batas antara nilai selisih dengan nilai ambang menjadi jauh yang mengakibatkan apabila nilai selisih kecil, maka akan sukar mencapai nilai ambang.

5. Metode

Dalam penelitian ini, metode penelitian yang dilakukan adalah memodelkan sistem dengan *Unified Modelling Language* (UML) menggunakan *use case diagram*, *class diagram*, *sequence diagram*, dan pemodelan bisnis.

5.1 Use Case Diagram

Use case diagram adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat pengguna. Yang ditekankan dalam *use case* adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. *Use case* mempresentasikan sebuah interaksi antara aktor dan sistem. *Use case* dapat digunakan selama proses analisis untuk menangkap persyaratan sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use case* diagram berperan untuk mendapatkan perilaku sistem saat diimplementasikan. Komponen pembentuk diagram *use case* yang pertama adalah *actor*. *Actor* mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan inputan pada sistem, hanya menerima informasi dari sistem atau keduanya. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. *Actor* digambarkan dengan *stick man*. Selain itu komponen pembentuk lainnya adalah *use case*. *Use case* adalah gambaran fungsionalitas dari sistem sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.



Gambar 2. Use case diagram

Untuk mendeskripsikan *use case* diatas dan aktor-aktor yang terlibat didalamnya, dapat dilihat pada tabel 1.

Tabel 1. Identifikasi Use Case

No	Aktor	Use Case	Deskripsi
1	User	Start Camera	Pengguna mengaktifkan <i>webcam</i> agar semua fitur-fitur yang ada di aplikasi ini dapat berjalan sebagaimana mestinya. Karena fitur-fitur yang ada di aplikasi ini berhubungan langsung dengan tampilan yang dihasilkan oleh <i>webcam</i> .

2	User	Broadcast Video	Pengguna mengaktifkan untuk server mulai <i>Listening</i> permintaan dari <i>client</i> untuk mengirimkan data guna keperluan <i>streaming video</i>
3	User	Record Video	Pengguna merekam gambar yang ditampilkan <i>webcam</i> untuk disimpan di media penyimpanan di komputer <i>server</i> .
4	User	Motion Detector	Pengguna mengaktifkan untuk mendeteksi apabila terdeteksi gerakan di area pemantauan, maka akan menghidupkan alarm dan secara otomatis akan menangkap gambar pada saat terdeteksi gerakan.
5	User	Play Video	Pengguna memainkan atau memutar video yang telah direkam sebelumnya.
6	User	Streaming Video	Pengguna dapat streaming video secara langsung pada saat pemantauan dengan memasukkan <i>IP address</i> dan <i>port server</i> .

6. Pengujian dan Analisa

6.1 Pengujian Broadcast Video

Broadcast video ini digunakan untuk *streaming video* pada sisi *client*. Apabila *broadcast video* ini tidak diaktifkan, maka *streaming video* tidak bisa dilakukan di sisi *client*. Subrutin `New, serverSocket = New TcpListener(System.Net.IPAddress.Any, Port)`, `TcpListener` merupakan kelas dari `namespace System.Net.Socket` yang memrepresentasikan aliran *TCP socket* dimana *server* dapat mendengarkan *client* melalui *IP address* dan *Port* yang telah ditentukan. Subrutin `StartConnection()` dengan metode `serverSocket.Start()` ini digunakan untuk mulai mendengarkan jika ada permintaan dari *client*. Selanjutnya, menetapkan koneksi antara *server* dan *client*. *Server* akan mendengarkan terus menerus sampai ada permintaan dari *client* dan *server* membuat koneksi ke *client* pada saat *server* menerima permintaan. Subrutin `serverSocket.Start()` dengan metode `clientSocket = serverSocket.AcceptTcpClient()` menjelaskan ketika *server* menerima permintaan, metode `AcceptTcpClient()` memanggil metode `Accept` dari `TcpListener` dibawah `Socket` untuk membuat koneksi. Apabila *server* dan *client* sudah terhubung, maka subrutin `Bmp = Cam.CopyFrame(PictureBox, New RectangleF(0, 0, CameraOutputSize.Width, CameraOutputSize.Height))` ini menjelaskan bahwa `PictureBox` sebagai target (Gambar preview dari webcam), akan di gandakan dengan ukuran tinggi dan lebar sesuai dengan `PictureBox`, lalu akan di kirim di fungsi `SendMessage` dengan metode `Dim Img As Image = Image.FromStream(ms) If SendMessage(Img)`. Subrutin diatas dipanggil di kelas `MainMenu` dengan senarai `serverCam = New Server (PictureBoxMainCam, 8080)`. Potongan senarai

tersebut berarti kelas `MainMenu` membuat koneksi *socket*, dengan target gambar yang ada di `PictureBoxMainCam` dengan port 8080. Untuk mengaktifkannya, `MainMenu` mendeklarasikan `serverCam` sebagai kelas `Server`, dan memanggil subrutin `serverCam.StartConnection()` untuk memulai *server* mendengarkan permintaan dari *client* dan `MsgBox("Broadcast Dimulai. Mulai Mendengarkan Client di Port 8080")` digunakan sebagai pemberitahuan bahwa *broadcast* sudah aktif. Gambar 8 menunjukkan bahwa fitur *broadcast video* telah aktif dan siap menunggu permintaan dari *client*.



Gambar 3. Tampilan broadcast video

6.2 Pengujian Record Video

Pengujian ini bermaksud untuk mengetahui fungsi rekam video serta pengujian kompresi video untuk gambar bergerak dan gambar diam. Perekaman video adalah salah satu fitur pada aplikasi ini yang memungkinkan pengguna untuk dapat melihat kejadian yang terjadi sebelumnya pada saat pemantauan.



Gambar 4. Tampilan pemberitahuan pengaktifan kamera

Sesuai kode program yang dibuat apabila kamera belum diaktifkan, pengguna yang akan melakukan perekaman video akan muncul pemberitahuan untuk mengaktifkan terlebih dahulu. Telah disebutkan sebelumnya bahwa penyimpanan default dari library `AVICAP32.dll` adalah `C:\CAPTURE.AVI`. Untuk aplikasi ini sudah merubah

dengan senarai CapturePathAndFileName = ("C:\Users\Equal\Videos\VideoServer\" & Format(Date.Now, "ddMMMMyyyyhhmmss") & ".avi" yang artinya file ditempatkan di di folder C:\Users\Equal\Videos\VideoServer dengan nama tanggal dan waktu yang sekarang yang diambilkan dari sistem penanggalan dan waktu di Windows (contoh : 01November2012071001.avi). Proses perekaman sudah dimulai ditandai dengan tulisan "RECORDING" yang dapat dilihat pada gambar 10.



Gambar 5. Tampilan proses perekaman sedang berlangsung

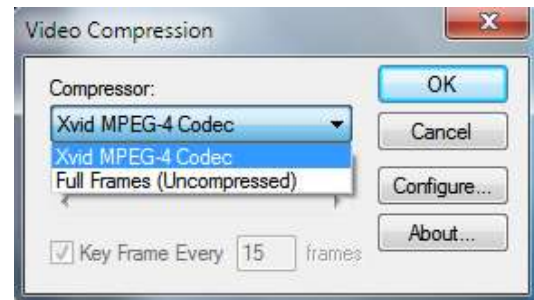
6.2.1 Pengujian Kompresi Video (Gambar Diam dan Gambar Bergerak)

Secara default, library AVICAP32.dll dengan perintah WM_CAP_SEQUENCE merekam video dengan format AVI yang asli dan tidak terkompresi. Hal ini mengakibatkan hasil file perekaman video menjadi sangat besar dan menjadi kelemahan bagi sistem komputer yang mempunyai harddisk yang kecil. Dengan ukurannya yang sangat besar, AVI dianggap sebagai format video yang belum terkompresi. Padahal, AVI merupakan salah satu jenis kompresi juga. Fungsi SendMessage(Handle, WM_CAP_DRIVER_CONNECT, VideoSource, 0) yang menjelaskan bahwa webcam harus sudah terhubung dahulu dan sudah menampilkan preview gambar. Apabila webcam belum aktif maka MessageBox.Show("Aktifkan Kamera Terlebih Dahulu", "", MessageBoxButtons.OK, yang akan muncul pemberitahuan bahwa kamera harus diaktifkan terlebih dahulu. Gambar 11 merupakan tampilan pemberitahuan apabila pengguna belum mengaktifkan kamera.



Gambar 6. Tampilan pemberitahuan pengaktifan kamera

WM_CAP_DLG_VIDEOCOMPRESSION yaitu menampilkan codec yang terpasang di komputer. Jadi dalam penggunaan kompresi ini, untuk jenis kompresi yang digunakan mengikuti codec yang terpasang di masing-masing komputer. Komputer satu dengan yang lain bisa saja memiliki jenis codec yang berbeda-beda.



Gambar 7. Tampilan jenis-jenis kompresi video

Dalam pengujian ini, yang akan dilakukan adalah merekam video dengan gambar bergerak dan gambar diam selama 60 detik dengan kompresi Xvid MPEG-4 codec, dan Full Frames (uncompressed) lalu hasil yang didapat dibandingkan dengan hasil penelitian yang sudah pernah dilakukan oleh orang lain. Berikut adalah hasil pengujian yang dapat dilihat pada tabel 2.

Tabel 2. Hasil pengujian kompresi video dengan codec Xvid MPEG-4

Codec	Gambar Bergerak	Gambar Diam
Xvid MPEG-4 codec		
Pengujian 1	35,7 MB	32,4 MB
Pengujian 2	35,2 MB	34,8 MB
Pengujian 3	36,3 MB	33,2 MB

Tabel 3. Hasil pengujian kompresi video dengan codec Full Frame (Uncompressed)

Codec	Gambar Bergerak	Gambar Diam
Full Frame (Uncompressed)		
Pengujian 1	440 MB	404 MB
Pengujian 2	466 MB	393 MB
Pengujian 3	417 MB	390 MB

Dari hasil pengujian dan perbandingan antara kompresi video dengan codec Xvid MPEG-4 dengan codec Full Frame (Uncompressed) menyimpulkan, ukuran berkas video yang paling kecil baik untuk gambar bergerak maupun gambar diam dihasilkan oleh kompresi dengan jenis Xvid MPEG-4. Sedangkan untuk ukuran berkas video yang paling besar baik untuk gambar bergerak maupun gambar dian dihasilkan oleh kompresi dengan jenis Full Frame. Hal ini dikarenakan tidak ada proses kompres sama sekali, masih berbentuk berkas AVI asli.

6.3 Pengujian Pendeteksi Gerak

Fitur ini digunakan untuk mendeteksi gerakan yang ada di area pemantauan. Apabila terdeteksi gerakan, maka akan menghidupkan alarm dan secara otomatis menyimpan gambar yang terdeteksi. Metode yang digunakan dalam pendeteksian gerak ini adalah membandingkan dua buah gambar (PictureBox1 dan PictureBox2) melalui *pixel* vertikal karena gambar dua dimensi memiliki permukaan X dan Y. Suatu gambar dihasilkan oleh titik-titik warna yang dinamakan *pixel*. Dalam hal ini, setiap warna dalam titik dibentuk dari 3 dasar warna yaitu RGB (*Red Green Blue*). Lalu dipisahkan masing-masing komponen RGB untuk setiap PictureBox 4 dan PictureBox 3 ke variabel yang berbeda sehingga menghasilkan 6 variabel. Tiga dasar warna di PictureBox4 dan PictureBox3 menjadi acuan untuk dimasukkan ke variabel integer. Lalu menghitung perbedaan antara PictureBox di setiap komponen yang disebut Delta. Apabila pendeteksi gerak diaktifkan, maka sistem akan memeriksa apakah delta tersebut tidak lebih tinggi dari jumlah nilai ambang yang diatur oleh pengguna. Jika komponen delta ini melampaui batas yang diatur, maka hal ini dianggap terdeteksi gerakan. Jadi, semakin tinggi nilai ambang yang diatur, maka akan semakin besar batas delta di setiap komponen yang mengakibatkan pendeteksi gerak ini kurang sensitif. Pada pengujian yang dilakukan yaitu uji coba dengan tingkat kesalahan yang berbeda-beda dengan kecepatan objek tinggi, sedang, dan rendah

Tabel 4 Hasil pengujian pendeteksi gerak

Nilai Ambang	Objek Manusia		
	Berlari	Berjalan Cepat	Berjalan Lambat
100	Tidak Terdeteksi	Tidak terdeteksi	Terdeteksi
90	Tidak Terdeteksi	Terdeteksi	Terdeteksi
80	Tidak Terdeteksi	Terdeteksi	Terdeteksi
70	Tidak Terdeteksi	Terdeteksi	Terdeteksi
60	Tidak Terdeteksi	Terdeteksi	Terdeteksi
50	Terdeteksi	Terdeteksi	Terdeteksi
40	Terdeteksi	Terdeteksi	Terdeteksi
30	Terdeteksi	Terdeteksi	Terdeteksi

20	Terdeteksi	Terdeteksi	Terdeteksi
10	Terdeteksi	Terdeteksi	Terdeteksi
1	Terdeteksi	Terdeteksi	Terdeteksi

Dari hasil percobaan diatas terlihat bahwa kecepatan objek juga mempengaruhi terdeteksinya gerakan. Untuk objek manusia dengan berlari, dapat terdeteksi pada nilai ambang 50 hingga 1. Pada objek manusia dengan berjalan cepat, pada nilai ambang 100 tidak dapat terdeteksi. Sedangkan pada objek manusia dengan berjalan pada nilai ambang 1 hingga 100 dapat terdeteksi.

6.4 Pengujian Streaming Video

Pada pengujian *streaming* video ini terdapat dua pengujian, pengujian *streaming* video dan pengujian *throughput streaming* video. Subrutin *Connect* dengan metode `clientSocket.Connect(IP, Port)` digunakan untuk membuat koneksi dari *client* ke *server* dengan memasukkan *IP address server* dan *port* dari I. Jika `ConnectedHost = True` yang berarti koneksi berhasil, `clientSocket` menggunakan metode `GetStream()` untuk memperoleh `NetworkStream` yang berfungsi untuk menulis ke dan membaca dari *server*. Pada sisi *server* yang menjadi target pengiriman adalah `PictureBox`, maka pada sisi *client* data yang akan diterima juga adalah `pictureBox` dengan metode `Dim img As Image = CType(Formatter.Deserialize(netStream), Image) PictureBox.Image = img PictureBox.Invalidate()`. Gambar 14 mengilustrasikan *streaming* video sedang berlangsung yang telah terhubung dengan *server* yang ditandai dengan adanya tulisan "Connected" dan munculnya gambar.



Gambar 8. Tampilan streaming video

6.4.1 Pengujian Throughput Streaming Video

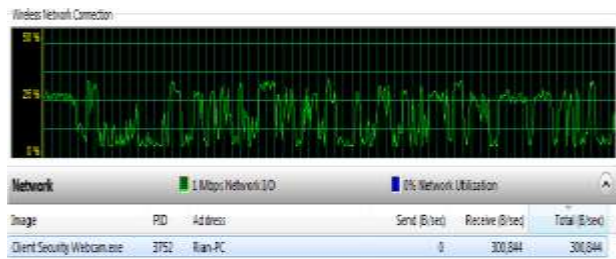
Pengujian ini dilakukan untuk mengetahui besarnya aliran data yang diterima *client* dalam proses *streaming* video. Pengujian ini dimaksudkan untuk mengetahui apakah gambar diam dan gambar bergerak mempunyai aliran data

yang sama atau tidak. Pengujian dilakukan dengan jaringan LAN menggunakan IP 192.168.0.0/30 dan port 8080. Port 8080 adalah port HTTP untuk protokol TCP. Untuk konfigurasi alamat IP dan port pada *server* dan *client* dapat dilihat pada tabel 5

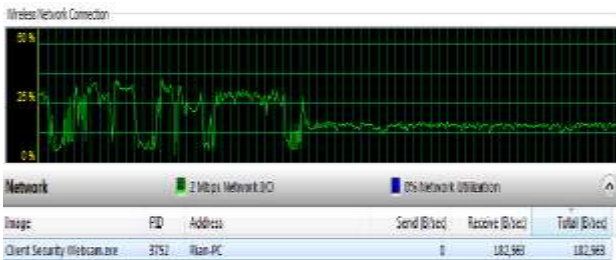
Tabel 5 Konfigurasi IP address dan port

Server (Subnet 192.168.0.0/30)	Client
1. IP : 192.168.0.1	1. IP : 192.168.0.2
2. Subnet mask : 255.255.255.252	2. Subnet mask : 255.255.255.252
3. Port : 8080	

Hasil pengujian dapat dilihat pada gambar 14 dan gambar 16.



Gambar 9. Grafik aliran data gambar bergerak



Gambar 10. Grafik aliran data gambar diam

Berdasarkan hasil pengujian, tercatat *receive data* tertinggi adalah 293,79 kByte/s untuk gambar bergerak yang dikirimkan dari *server*. Sedangkan pada gambar 16 adalah grafik aliran data untuk gambar diam dan tercatat *receive data* tertinggi adalah 178,28 kByte/s. Hal ini dikarenakan pada gambar bergerak, byte-byte data gambar yang bergerak lebih banyak dibandingkan dengan byte-byte data gambar diam. Sehingga aliran data gambar bergerak lebih banyak dibandingkan gambar diam.

7. Kesimpulan

Berdasarkan hasil pengujian aplikasi sistem pemantau ruangan dengan *webcam*, pembuatan aplikasi dapat menggunakan Visual Basic .NET dengan memanfaatkan *library* *avicap32.dll* untuk melakukan koneksi ke *driver webcam*, inisialisasi *capture windows* dan fungsi *callback* dari *capture windows*. Sedangkan untuk fitur *broadcast video* dan *streaming video* menggunakan *constructor*, *properties*, dan *method* yang terdapat pada *class socket*.

Berdasarkan hasil pengujian kompresi video dengan uji coba selama 60 detik dengan jenis *codec* Xvid MPEG-4, dan Full Frames (*Uncompressed*), yang menghasilkan ukuran berkas paling kecil adalah *codec* Xvid MPEG-4 untuk gambar bergerak dan gambar diam. Sedangkan jenis *codec* Full Frame (*Uncompressed*) menghasilkan ukuran berkas lebih besar dari *codec* Xvid MPEG-4. Berdasarkan hasil pengujian pendeteksi gerak, nilai ambang mempengaruhi tingkat keberhasilan dalam pendeteksian. Semakin besar nilai ambang, maka akan mempengaruhi tingkat pendeteksian karena apabila nilai selisih RGB kecil, maka akan semakin jauh mencapai batas nilai ambang yang diatur sehingga berdampak tidak terdeteksi gerakan walaupun sebenarnya terdapat gerakan. Pengujian di sisi *client* dapat menampilkan *streaming video* dari *server* menggunakan jaringan *Wireless LAN* dengan *IP address* 192.168.0.0.1 dan *port* 8080. Berdasarkan hasil pengujian *throughput data streaming video* menggunakan jaringan *Wireless LAN* dengan *Ad-Hoc*, data yang diterima *client* untuk gambar bergerak sebesar 293,79 kByte/s lebih besar dibandingkan data gambar diam yaitu sebesar 178,28 kByte/s.

Referensi

Textbooks :

- 1] Arifianto, Deni dan Wijaya Ariyana, *Cara Top Bikin Komputer Top*, PT. Kawan Pustaka, Jakarta, 2009.
- 2] Arifianto, Deni dan Ari Funatik, *Antigaptek Hardware Komputer*, PT. Kawan Pustaka, Jakarta, 2009.
- 3] Binanto, Iwan., *Multimedia Digital Dasar Teori + Pengembangan*, PT. Andi Offset, Yogyakarta, 2010.
- 4] Komputer, Wahana, *Cara Mudah Membangun Jaringan Komputer dan Internet*, Media Kita, Jakarta, 2010.
- 5] S. Mulyanta, Edi., *Pengenalan Protokol Jaringan Wireless Komputer*, CV. Andi Offset, Yogyakarta, 2005.
- 6] Yulwardian, Elga., *Editing Video Praktis Dengan Ulead VideoStudio 7.0*, PT. Elex Media Komputindo, Jakarta, 2004.
- 7] Suyanto, M., *Multimedia Alat Untuk Meningkatkan Keunggulan Bersaing*, CV. Andi Offset, Yogyakarta, 2005.
- 8] Deitel, *Visual Basic .NET How To Program Second Edition*, Prentice-Hal International, London, 2002.
- 9] Hirin, A.M., *Belajar Tuntas (Dari Dasar Sampai Mahir) VB.NET 2010*, Prestasi Pustaka, Jakarta, 2011.

Internet :

- 10] --, Ngobrol Dunia Biner, <http://duniabiner.co.cc>, September 2012.
- 11] --, Standar Kompresi Video, <http://digilib.itelkom.ac.id>, September 2012.
- 12] --, AVI, <http://videodigital.polinema.org>, Oktober 2012.
- 13] --, AVI, <http://www.jmcgowan.com>, Oktober 2012.
- 14] --, Application Port, <http://bandwidthcontroller.com>, Oktober 2012.
- 15] --, Video Compression for the web, <http://accad.osu.edu>, Oktober 2012.