

PENERAPAN KAMERA WEB SEBAGAI PENDETEKSI GERAKAN DENGAN ANTARMUKA *DIRECTSHOW*

Yonatan Santoso,^{*)} Iwan Setiyawan, dan Victor N. Papilaya

Fakultas Teknik Elektro dan Komputer, Universitas Kristen Satya Wacana, Salatiga 50711, Indonesia

^{*)}E-mail: yona_seven@yahoo.com

Abstrak

Salah satu cara untuk mengendalikan kamera *web* adalah dengan menggunakan *DirectShow* yang terintegrasi di dalam *DirectX*. *DirectShow* berperan sebagai API sehingga penggunaannya lebih mudah. Selanjutnya kamera *web* ini akan digunakan untuk mendeteksi gerakan obyek seperti halnya kamera pengawas. Metodenya adalah dengan membandingkan gambar acuan dengan gambar yang diambil kemudian. Jika ada perbedaan pada kedua gambar tersebut berarti sebuah gerakan telah terdeteksi.

Abstract

Implementation of Web Cam as Motion Detector with DirectShow Interfacing. One of many ways to take control a web cam is by using DirectShow which is integrated in DirectX. DirectShow acts like an API therefore it is easier to use. Afterwards, web cam will be implemented to detect object's motion as well as camera surveillance. The method is by comparing a reference image with other image which is taken lately. If there are differences between those image it means motion has been detected.

Keywords: motion detection, surveillance camera

1. Pendahuluan

Keamanan merupakan hal yang sangat penting akhir – akhir ini. Seiring dengan perkembangan teknologi, dibangunlah sistem keamanan berdasarkan sistem kendali elektronika atau terotomatisasi. Sistem keamanan yang populer adalah menggunakan kamera pengawas.

Kamera pengawas yang sering digunakan adalah jenis CCTV. Namun dalam pemasangannya membutuhkan hardware tambahan yang dipasang pada komputer selain dari kamera itu sendiri, yaitu *DVR card* atau *TV Tuner*. Di samping itu, jenis komputer yang dibutuhkan adalah komputer dengan spesifikasi tinggi. Kamera *web* dapat digunakan sebagai alternatif karena harganya relatif terjangkau, efektivitasnya sangat tergantung pada perangkat lunak yang mengendalikannya sehingga lebih fleksibel.

Berbagai penelitian dan percobaan telah dilakukan oleh beberapa pengembang sebagai berikut: 1) Perusahaan pengembangan perangkat lunak berbasis di Mesir bernama *Link Development* dipimpin A.Kirillov.

Penelitiannya menggunakan kamera *web* berbasis IP (*Internet Protocol*) untuk ditempatkan sebagai pengawas secara *real time*, 2) Pengembang Guy Watson (direktur *FlashGuru LTD*) menggunakan API dari produsen perangkat lunak terkenal *Adobe* untuk mengontrol kamera sebagai pendeteksi gerakan [1], 3) Pengembang bernama T Amin menggunakan algoritma perbandingan sederhana untuk mendeteksi gerakan, 4) Pengembang Eric Rudolph (*Microsoft*) menggunakan *DirectShow* yang merupakan salah satu komponen *Microsoft DirectX* yang terintegrasi pada sistem operasi komputer berbasis *Microsoft Windows* untuk hanya mengakses kamera *web*, 5) Dikembangkan kamera dengan pendeteksi gerakan menggunakan *PIR (Passive Infra Red)*.

Dari beberapa penelitian di atas terlihat ada beberapa macam cara membangun antar muka untuk mengendalikan kamera *web* yang selanjutnya digunakan untuk mendeteksi gerakan. Pada penelitian ini digunakan antar muka *DirectShow* karena lebih praktis (terintegrasi dalam sistem operasi *Windows* dan minim sumberdaya memori) [2].

DirectShow adalah sebuah pustaka *COM* (*Component Object Model*) untuk mengambil, mengolah, dan memainkan media gambar dan suara. Beberapa aplikasi juga menggunakan *DirectShow* sebagai antar muka perangkat keras karena *DirectShow* adalah satu – satunya jalan kode aplikasi untuk dapat mengakses perangkat keras berbasis *Windows Driver Model* (*WDM*) seperti *web cam* [3].

Komponen penyusun *DirectShow* adalah perangkat lunak berupa komponen pengaliran data yang disebut filter. Filter adalah berupa obyek *COM* yang dapat mewakili devais perangkat keras, perangkat lunak enkoder atau dekoder, sebuah pembuat audio atau video, ataupun yang memiliki kemampuan memproses audio video .

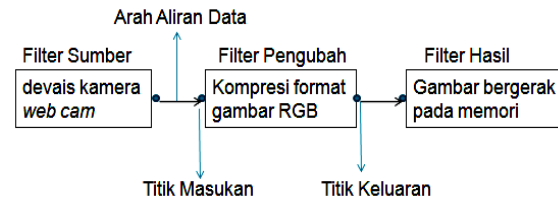
Filter dibagi menjadi 3 tipe, yaitu sumber, pengubah, dan hasil atau pembawa. Filter sumber bertugas mengakses inti *WDM* dan kemudian menciptakan data hasil akses tersebut untuk dapat dialirkan ke filter (atau sekelompok filter) selanjutnya. Filter hasil atau pembawa berada di ujung berlawanan dengan filter sumber. Filter ini bertugas menerima data dan kemudian mengirimkannya devais yang dapat berupa penulisan media ke dalam sebuah *file* atau perubahan audio video. Di antara filter sumber dan hasil terdapat filter pengubah yang menerima data dari filter sumber dan kemudian melakukan operasi pada data sumber tersebut agar menghasilkan data yang dapat digunakan oleh filter hasil [3].

Sebuah filter paling tidak memiliki sebuah titik penghubung. Filter saling terhubung dengan filter yang lain pada titik tersebut. Data bergerak di antara filter – filter melalui titik penghubung. Ada dua jenis titik penghubung, yaitu titik masukan dan titik keluaran. Filter sumber pada umumnya hanya memiliki titik keluaran. Filter hasil pada umumnya hanya memiliki titik masukan, dan filter pengubah biasanya memiliki kedua titik. Titik keluaran dari satu filter terhubung dengan sebuah titik masukan filter yang lain. Filter menerima masukan dan menghasilkan suatu keluaran. Dalam *DirectShow*, sebuah aplikasi mengerjakan tugas dengan cara menghubungkan serangkaian filter bersama-sama, sehingga keluaran dari satu filter menjadi masukan untuk filter lainnya. Saat dua filter terhubung, titik-titik penghubung menentukan tipe koneksi apa yang akan dibangun.

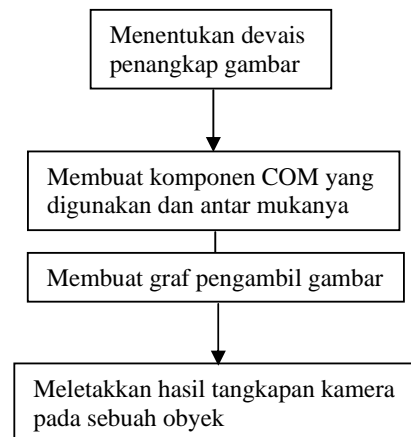
Kumpulan filter-filter yang terhubung membentuk graf disebut graf filter. Graf filter memiliki empat kemungkinan keadaan, yaitu: berhenti, jeda, berjalan, dan transisi. Sebagai contoh, seperti gambar 1, filter file sumber mengakses *WDM* kamera *web*. Filter pengubah melakukan kompresi data gambar *raw* menjadi *RGB* (*Red Green Blue*). Filter hasil berupa “tempat” untuk meletakkan data gambar [4].

Aplikasi tidak menangani semua aliran data. Sebagai gantinya, filter dikendalikan oleh komponen tingkat tinggi yang disebut Filter Graph Manager. Aplikasi membuat eksekusi *API* tingkat tinggi seperti memindahkan data melalui graf atau menghentikan aliran data.

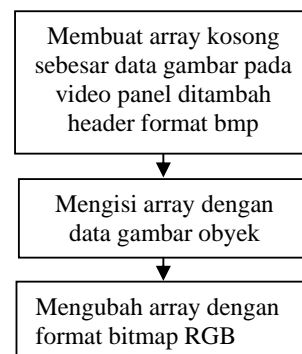
Hasil tangkapan yang diletakan pada obyek ini masih berupa aliran data *real – time* pada memori. Agar dapat diolah maka harus diubah menjadi file gambar. Data dalam obyek pada satu waktu dicuplik ke dalam sebuah array kosong dan diberi format gambar bitmap (*BMP*) [5].



Gambar 1. Graf Filter Mengakses Kamera Web



Gambar 2. Langkah Pengerjaan Tugas pada Aplikasi *DirectShow*



Gambar 3. Membuat Sebuah File Gambar

```

differ = 0;
if (p1 > p)
{
differ_temp = (int)(p1 - p);
}
else
{
differ_temp = (int)(p - p1);
}
if (differ < differ_temp)
    differ = differ_temp;
threshold = k;
if(differ>threshold)
    motion = true;

```

Gambar 4. Kode Bayangan Metode Pembandingan Dua Gambar

Ada berbagai macam metode untuk membandingkan dua buah gambar. Metode yang sederhana dan prosesnya tidak terlalu berat yaitu dengan mencari nilai perbedaan piksel tertinggi dan membandingkannya dengan suatu nilai batas. Jika nilai perbedaan lebih tinggi dari nilai batas maka disimpulkan terdapat perbedaan pada kedua gambar tersebut. Agar rentang perbedaan piksel tidak terlalu jauh maka sebaiknya dilakukan *greyscale* terlebih dahulu, yaitu dengan mengambil nilai rata – rata komponen RGB [6].

Jika p dan $p1$ masing – masing merupakan nilai piksel gambar acuan dan gambar yang akan dibandingkan, $differ$ merupakan nilai perbedaan piksel tertinggi, dan nilai batas $threshold$ ditetapkan bernilai k , maka jika nilai $k < differ$ berarti terdapat perbedaan pada kedua gambar. Nilai batas digunakan untuk mentoleransi nilai perbedaan yang dianggap terlalu kecil.

2. Metode Penelitian

Penelitian ini akan menguji sejauh mana efektivitas kamera *web* dengan pengendalian melalui *DirectShow* untuk berperan sebagai pendeteksi gerakan. Adapun spesifikasi kamera *web* yang digunakan adalah menghasilkan gambar 320 x 240 piksel, menggunakan antar muka *USB*, fokus diatur secara manual, dan *frame rate* maksimal 2 *fps*.

Pengujian dilakukan untuk mengetahui beberapa faktor yaitu: 1) Karakteristik ruangan, 2) Jarak kamera dengan obyek, 3) Karakteristik obyek bergerak [7].

Jika saat satu faktor diuji pengaruhnya, maka faktor lain diasumsikan tidak mempengaruhi hasil pengujian. Hal ini agar memudahkan dalam menarik kesimpulan.

Gambar acuan diasumsikan adalah gambar di mana tidak ada obyek bergerak di dalamnya. Jika ada

perbedaan pada kedua gambar tersebut maka disimpulkan bahwa suatu gerakan terjadi.

Pengujian akan dilakukan terhadap metode pembandingan gambar yaitu dengan membandingkan gambar di mana tidak ada obyek bergerak di dalamnya. Hal ini untuk mengetahui ketahanan metode terhadap kesalahan deteksi (*false detection*). Jika tingkat kesalahannya cukup tinggi yaitu kira-kira di atas 2% berarti metode dianggap gagal.

Pada pengujian selanjutnya obyek bergerak melewati kamera dengan waktu acak yaitu untuk mengetahui tingkat kegagalan metode dalam mendeteksi gerakan obyek. Jika terlalu banyak gerakan obyek yang tidak terdeteksi (*miss detection*) yaitu kira-kira di atas 2% maka metode dianggap gagal.

3. Hasil dan Pembahasan

Setelah proses pengujian yaitu dengan mengambil 900 gambar atau selama 7,5 menit tanpa adanya obyek bergerak dan keadaan ruangan khususnya tingkat pencahayaan tidak boleh berubah terlalu besar didapatkan hasil tidak ada kesalahan deteksi (0%). Untuk tingkat toleransi pencahayaan tidak dapat diuji karena keterbatasan alat yang digunakan.

Tabel 1. Tabel Hasil Percobaan

Jumlah gambar masing-masing 900 gambar	<i>False detection</i> (Tidak ada obyek bergerak)	<i>Miss detection</i> (Obyek bergerak)
Jumlah gambar	0	14
Persentase	0 %	1,55%



Gambar 5. Contoh Gambar Obyek Bergerak yang Tidak Terdeteksi

Ket.: Gambar kurang jelas karena berasal dari kamera *web*



Gambar 6. Contoh Gambar Obyek Terdeteksi
Ket.: Gambar kurang jelas karena berasal dari kamera *web*



Gambar 7. Contoh Gambar Acuan
Ket.: Gambar kurang jelas karena berasal dari kamera *web*

Percobaan selanjutnya obyek bergerak dengan acak melewati kamera (waktu atau jumlah gambar sama). Didapatkan hasil bahwa jarak maksimal obyek setinggi 1,5 m dapat terdeteksi dengan baik adalah 7 m. Jika melebihi dari jarak tersebut maka obyek tidak dapat terdeteksi. Perbandingan tinggi obyek dengan jarak pengamatan adalah 1 : 4,67.

Obyek bergerak tidak dapat dideteksi karena obyek memiliki warna menyerupai dengan latar belakangnya. Hal ini terlihat pada contoh gambar.

Obyek bergerak juga tidak dapat dideteksi karena obyek bergerak terlalu cepat, yaitu melebihi 3 m/s. Perhitungan didapatkan dengan mengambil contoh gambar obyek tidak terdeteksi yaitu panjang daerah yang diamati 3 m dan waktu yang diperlukan obyek untuk menempuh jarak tersebut adalah 1 detik.

4. Simpulan

Ada beberapa faktor yang dapat mempengaruhi metode pendeteksian gerakan dapat berfungsi dengan baik. Pencahayaan ruangan harus stabil. Untuk obyek yang diamati harus memiliki kontras warna yang berbeda dengan latar belakangnya, serta kecepatannya tidak melebihi kemampuan kamera memproses gambar. Jarak kamera dengan obyek yang diamati tidak boleh melebihi perbandingan dengan tinggi obyek yaitu 1:4,67.

Daftar Acuan

- [1] G. Watson, Webcam Motion Detection: Using the BitmapData API in Flash 8, Adobe Developer Connection, www.adobe.com/devnet/flash/articles/webcam_motion.html, 2005.
- [2] M.D. Pesce, Programming Microsoft® DirectShow® for Digital Video, Microsoft Press, Washington, 2003, p.29
- [3] E. Rudolph, How To Get Data from a Microsoft DirectShow Filter Graph, MSDN, <http://msdn.microsoft.com/enus/windows/bb190313.aspx>, 2003.
- [4] H.R. Myler, A.R. Weeks, The Pocket Handbook of Image Processing Algorithms, Prentice Hall, California, 1993, p.120.
- [5] C. Grauss, Image Processing For Dummies With C# and GDI+ Part 1 - 3, The Code Project, <http://www.codeproject.com/KB/GDIplus/csharpgraphicfilters11.aspx>, 2002.
- [6] A. K. Jain, Fundamentals of Digital Image Processing, Prentice Hall International, California, 1989, p.246.
- [7] Y. Amit, *2D Object Detection and Recognition*, Massachusetts Institute of Technology, Massachusetts, 2002, p.18.