

# SISTEM PEMANTAUAN RUANGAN DENGAN WEBCAM MENGGUNAKAN METODE MOTION DETECTION

**Dani Rohpandi, Soleh Permana, Fahrurizal Muldiana**  
Program Studi Teknik Informatika  
Sekolah Tinggi Manajemen Informatika dan Komputer Tasikmalaya  
RE Martadinata 272A Kota Tasikmalaya Tasikmalaya Jawa Barat, Tlp (0265)310830  
danirtms@gmail.com, soldhoe@gmail.com, thelszal@yahoo.com

## Abstrak

Penelitian mengenai deteksi gerakan semakin banyak dilakukan dengan berbagai maksud dan tujuan, diantaranya untuk diterapkan dalam bidang keamanan, diantaranya untuk memantau suatu ruangan. Telah banyak peralatan yang dapat mendeteksi pergerakan suatu objek, tetapi harganya relative tinggi dan gambar yang direkam membutuhkan kapasitas media perekaman yang cukup besar. Penelitian ini menggunakan metode rekayasa, karena itulah penelitian ini bertujuan agar proses perekaman yang terjadi hanya dilakukan jika terdeteksi pergerakan suatu objek sehingga hasil gambar yang terrekam lebih menghemat kapasitas media penyimpanan data, serta kamera yang digunakan pun dapat digunakan yang harganya relatif terjangkau oleh masyarakat umum. Gambar yang direkam berbentuk file gambar statis. Kualitas kamera yang digunakan tentu saja berpengaruh terhadap jarak pantau yang dihasilkan.

**Kata kunci:** Deteksi gerakan, pemantau ruangan, gambar statis, efisiensi perekaman

## Abstract

There are many research on motion detection performed with different intents and purposes, including to be applied in the security field, including to monitor a room. Has a lot of equipment that can detect movement of an object, but the price is relatively high and the recorded image recording media capacity require a lot of sizable. This research is used the engineer method, that's why the research is intended that the recording process that occurs only if the detected movement of an object so that the image terrekam save more data storage media capacity, and the camera can be used which is relatively affordable by the general public. Images were recorded in static image files. The quality of the camera used of course monitor the effect on the resulting distance.

**Keywords:** Motion detection, room monitoring, static image, recording efficiency

## 1. Pendahuluan

Dalam beberapa tahun terakhir, deteksi gerak (*motion detection*) telah menarik minat yang besar dari para peneliti *Computer Vision* karena menjanjikan aplikasi di banyak bidang, seperti pengawasan video (*video surveillance*) [1], pemantauan lalu lintas (*traffic monitoring*) [2] atau tanda pengenalan bahasa. Namun, masih dalam tahap awal perkembangan dan kebutuhan untuk meningkatkan keandalannya bila diterapkan dalam lingkup permasalahan yang sangat kompleks.

Beberapa teknik untuk memindahkan deteksi objek telah diusulkan dalam beberapa penelitian [3]-[6] Aliran optik adalah perkiraan gerak gambar lokal dan menentukan berapa banyak setiap pixel gambar bergerak antara gambar yang berdekatan. Hal ini dapat mencapai keberhasilan mendeteksi gerakan di hadapan gerakan kamera atau latar belakang berubah. Menurut kendala kelancaran, poin-poin terkait dalam dua frame berturut-turut tidak harus bergerak lebih dari beberapa piksel. Untuk lingkungan yang tidak pasti, ini berarti bahwa gerak kamera atau latar belakang perubahan harus relatif kecil. Metode berdasarkan arus optik kompleks, tetapi dapat mendeteksi gerakan secara akurat bahkan tanpa mengetahui latar belakang

Pada penelitian sebelumnya membuktikan bahwa *webcam* dapat digunakan sebagai alat pendeteksi [7]. Demikian juga dengan penelitian yang berjudul "*Perancangan Pendeteksi Gerakan menggunakan Webcam untuk Sistem Pemantauan*" [8] telah merancang sebuah alat pendeteksi gerakan menggunakan *webcam* untuk sistem pemantauan. Sistem alat tersebut akan mengaktifkan alarm secara otomatis apabila *webcam* menangkap obyek yang diindikasikan sebagai pelaku yang akan berbuat kriminal

## 1.1 Permasalahan

Keamanan adalah bebas dari bahaya. Karena keamanan sudah menjadi kebutuhan yang diperlukan seiring dengan meningkatnya tindak kejahatan. Baik keamanan lingkungan, keamanan jasmani, keamanan rohani maupun keamanan harta benda dan berkas-berkas maupun dokumen-dokumen penting lainnya. Banyak juga masyarakat yang telah memperkerjakan tenaga keamanan untuk menjaga perumahan, perkantoran bahkan di instansi sekolah-sekolah, tetapi tetap saja pencuri lebih pintar dalam melakukan kejahatan.

Seiring dengan perkembangan teknologi informasi, berbagai macam cara telah dilakukan untuk menjaga keamanan dengan sebuah sistem keamanan ganda dimana untuk menghindari jika mengalami kegagalan keamanan maka diperlukan sebuah mekanisme yang dapat merekam suatu tindakan kejahatan sehingga dapat digunakan sebagai bahan evaluasi. Hal ini berguna untuk proses penyelidikan, sehingga bila pengamanan manusia gagal maka dapat diteruskan dengan keamanan yang dapat merekam suatu tindakan, sehingga dapat sebagai bahan bukti oleh pihak yang berwenang [1].

## 1.2 Manfaat dan Tujuan

Oleh sebab itu perlu alat pendeteksi gerakan untuk mengetahui objek-objek yang bergerak, dalam menjaga suatu ruangan kosong atau dengan memantau ruangan tersebut. Tetapi ada kalanya manusia juga tidak dapat memantau ruangan tersebut selama 24 jam, dan terkadang terjadi kelalaian yang mungkin ketiduran dan sebagainya, oleh karena itu diperlukan juga aplikasi yang dapat memberitahu jika terdeteksi adanya pergerakan bukan hanya tampilan saja dari hasil perekaman objek yang bergerak, tetapi diperlukannya suatu *alarm* yang bisa mengingatkan penjaga jika terjadi kelalaian, ketika ada objek yang bergerak.

Banyak aplikasi yang dirancang untuk berbagai keamanan. Salah satunya adalah aplikasi pendeteksi gerakan (*motion detection*). Aplikasi ini dapat dikembangkan karena dapat diaplikasikan di berbagai bidang permasalahan, salah satu contohnya adalah menjaga keamanan ruangan [9].

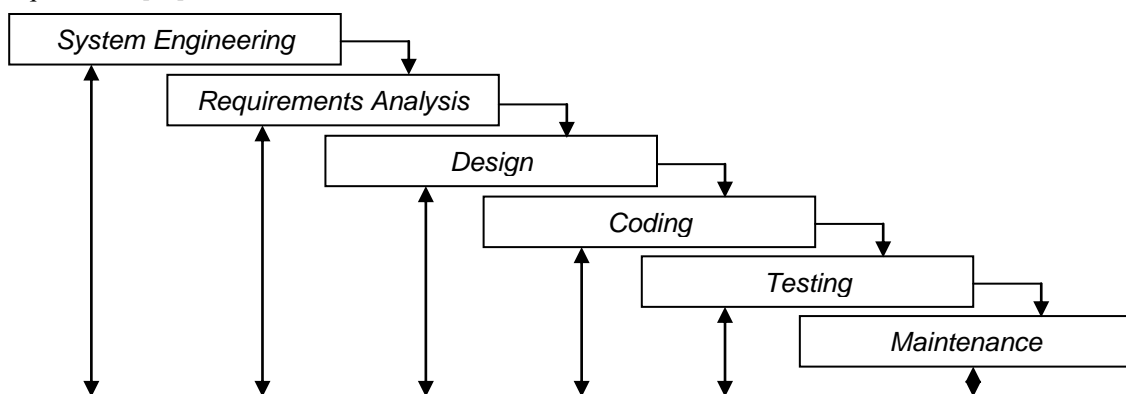
Peralatan yang dimanfaatkan harganya relatif murah dan mudah untuk digunakan karena cukup menggunakan peralatan standar yang banyak beredar di masyarakat umum. Demikian pula dengan aplikasi yang dibuat, selain mudah dalam penggunaannya juga dapat menghemat kapasitas penyimpanan data karena hanya akan melakukan perekaman gambar jika terdeteksi adanya suatu pergerakan dari objek yang tertangkap kamera.

## 2. Metode Rekayasa

Metode yang digunakan adalah metode rekayasa, dimana Penulis melakukan eksperimen atau ujicoba terhadap rekayasa perangkat lunak yang dibuat untuk mengetahui batasan kemampuan optimum dari output yang dihasilkan.

### 2.1 Model Waterfall

Dalam tahapan perancangan program aplikasi, Penulis menggunakan tahapan model *waterfall*. Dimana model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lama menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintenance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement* [10].



Gambar 1. Model Waterfall

Gambar diatas adalah tahapan umum dari model *waterfall*, yang terbagi menjadi 6 tahapan. Berikut adalah penjelasan dari tahap-tahap ada di dalam model *waterfall* :

- a. *System Engineering*, pemodelan ini diawali dengan mencari kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dsb. *Software* yang dibuat harus dapat mengakses kamera *webcam* yang digunakan dan dapat menangkap gambar yang dihasilkannya. Kemudian jika adanya pergerakan yang terjadi, maka *software* harus dapat melakukan perekaman gambar yang ditangkap kamera.
- b. *Requirements Analysis*, proses pencarian kebutuhan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti tentang domain informasi dari *software*. Misalnya fungsi yang dibutuhkan, *user interface*, dsb. Tentunya untuk dapat melakukan perekaman pergerakan yang terjadi dalam bentuk gambar membutuhkan suatu alat, pengendali alat, dan fasilitas untuk menampilkan gambar dalam *software* yang dibuat.
- c. *Design*, proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi *representasi* ke dalam bentuk “*blueprint*” *software* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.
- d. *Coding*, untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap *design* yang secara teknis nantinya dikerjakan oleh *programmer*.
- e. *Testing*, sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, yang sudah didefinisikan sebelumnya.
- f. *Maintenance*, pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja ada *errors* kecil yang ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari *eksternal* perusahaan seperti ketika ada pergantian sistem operasi atau perangkat lainnya.

## 2.2 Webcam

Web camera (webcam) adalah sebuah alat yang terhubung ke komputer berfungsi untuk mengambil citra dari lensa teleskopik yang terdapat pada alat tersebut. Citra yang diambil ini kemudian dikirimkan ke komputer melalui kabel USB (Universal Serial BUS) atau kabel serial dengan kecepatan realtime (kurang lebih 15 frame per detik). Web camera saat ini dipergunakan secara luas pada aplikasi online video conference, pemantauan, sebagai kamera digital, dan banyak lagi. Resolusi (ketajaman) maksimum citra hasil dari web camera adalah 640 x 480 pixel

Sebuah *web camera* yang sederhana terdiri dari sebuah lensa standar, dipasang di sebuah papan sirkuit untuk menangkap sinyal gambar; *casing* (*cover*), termasuk *casing* depan dan *casing* samping untuk menutupi lensa standar dan memiliki sebuah lubang lensa di *casing* depan yang berguna untuk memasukkan gambar, kabel *support*, yang dibuat dari bahan yang *fleksibel*, salah satu ujungnya dihubungkan dengan papan sirkuit dan ujung satu lagi memiliki *connector*, kabel ini dikontrol untuk menyesuaikan ketinggian, arah dan sudut pandang *web camera*.

*Webcam* yang digunakan dalam sistem ini berfungsi untuk menangkap gambar yang akan dilihat oleh pemakai, dan juga sebagai gambar yang akan dipindai oleh perangkat lunak untuk mendeteksi adanya gerakan.



Gambar 2 Genius i-Slim 300

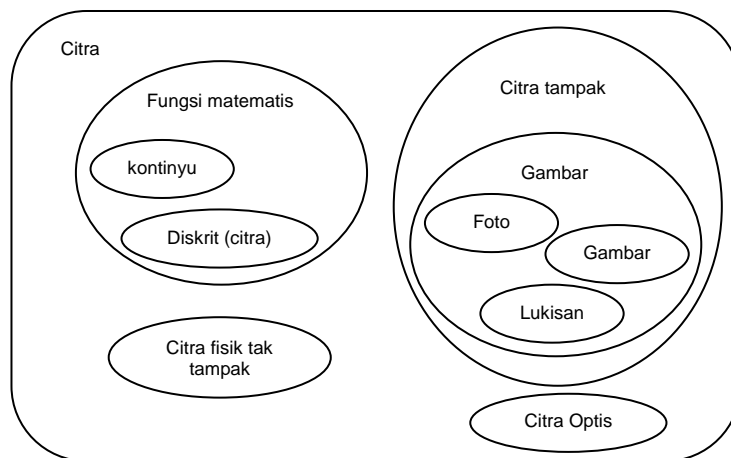
Sebagai contoh, kamera yang akan digunakan dalam sistem ini adalah Genius i-Slim 300. Dengan spesifikasi kamera, sebagai berikut : *Resolusi 300K pixel*, Lensa tipe *Manual Focus*, *White balance : Auto/Manual*, *Interface USB 1.1*, *File format : JPEG/WMV*, Kecepatan frame *30 frame per detik* (pada sistem yang direkomendasikan), *Mikropone* yang sudah terpasang didalamnya (*built-in*) untuk *chatting*, *Still Image in Genius Application : 3360 x 2520, 1600 x 1200, 1280 x 960, 800 x 600* (*Software interpolation*), *640 x 480, 352 x 288, 320 x 240, 176 x 144, 160 x 120 pixel*, Disertai *software : CrazyTalk Avatar Creator and Cam Suite* untuk *special effects*, *WebMate image editing*. Selain *webcam*, unit komputer juga merupakan bagian dari perangkat keras yang digunakan dalam sistem ini. Unit komputer juga dilengkapi penguat suara, serta dilengkapi dengan sistem operasi *Windows XP Profesional SP 2*

## 2.3 Pengolahan Citra

### 2.3.1 Definisi Citra

Citra adalah suatu *representasi*, kemiripan, atau imitasi dari suatu obyek atau benda [11]. Selain itu juga didalam sebuah citra juga terdapat *Kompresi* Citra adalah aplikasi *kompresi* data yang dilakukan terhadap citra digital dengan tujuan untuk mengurangi *redundansi* dari data-data yang terdapat dalam citra sehingga dapat disimpan atau ditransmisikan secara efisien. Citra dapat dikelompokkan menjadi citra tampak dan citra tak tampak, sebagaimana disajikan pada gambar 3.

Contoh citra tampak dalam kehidupan sehari-hari adalah foto keluarga, gambar yang nampak pada layar *monitor* dan televisi, serta hologram (citra optis). Sedangkan contoh citra tak tampak adalah data gambar dalam *file* (citra digital) dan citra yang merepresentasikan menjadi fungsi matematis. Di samping itu ada juga citra fisik tak tampak, misalnya citra *distribusi* panas di kulit manusia serta peta densitas dalam suatu material. Untuk dapat dilihat mata manusia, citra tak tampak ini harus diubah menjadi citra tampak, misalnya dengan menampilkannya di *monitor*, dicetak di atas kertas, dan sebagainya.



Gambar 3. Pengelompokan Jenis-Jenis Citra

### 2.3.2 Komponen Citra Digital

Setiap citra digital memiliki beberapa karakteristik, antara lain ukuran citra, *resolusi*, dan *format* nilainya. Umumnya citra digital berbentuk persegi panjang yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau *pixel*, sehingga ukuran citra selalu bernilai bulat.

*Format* citra *digital* ada bermacam-macam. Karena sebenarnya citra mempresentasikan informasi tertentu, sedangkan informasi tersebut dapat dinyatakan secara bervariasi, maka citra yang mewakilinya dapat muncul dalam berbagai *format*. Citra yang mempresentasikan informasi yang hanya bersifat *biner* untuk membedakan 2 keadaan tentu tidak sama citra dengan informasi yang lebih kompleks sehingga memerlukan lebih banyak keadaan yang diwakilinya. Pada citra digital semua informasi disimpan dalam angka, sedangkan penampilan angka tersebut biasanya dikaitkan dengan warna.

Citra digital tersusun atas titik-titik yang biasanya berbentuk persegi panjang atau bujursangkar (pada beberapa sistem pencitraan, piksel-piksel penyusun citra ada pula yang berbentuk segi enam) yang secara beraturan membentuk barisan-barisan dan kolom-kolom. Setiap titik memiliki *koordinat* sesuai

dengan posisinya dalam citra. *Koordinat* ini biasanya dinyatakan dalam bilangan bulat *positif*, yang dapat dimulai dari 0 atau 1 bergantung pada sistem yang digunakan.

Setiap titik juga memiliki nilai berupa angka *digital* yang mempresentasikan informasi yang diwakili titik tersebut. *Format* nilai piksel sama dengan *format* citra keseluruhan. Pada kebanyakan sistem pencitraan, nilai ini biasanya berupa bilangan bulat *positif* juga.

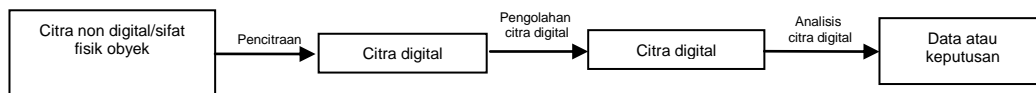
### 2.3.3 Representasi Citra

Komputer dapat mengolah isyarat-isyarat elektronik *digital* yang merupakan kumpulan sinyal *biner* (bernilai dua: 0 dan 1). Untuk itu, citra *digital* harus mempunyai *format* tertentu yang sesuai sehingga dapat mempresentasikan obyek pencitraan dalam bentuk kombinasi data *biner*.

Pada kebanyakan kasus, terutama untuk keperluan penampilan secara *visual*, nilai data *digital* tersebut mempresentasikan warna dari citra yang diolah, dengan demikian *format* data citra digital berhubungan erat dengan warna. *Format* citra *digital* yang banyak dipakai adalah citra *biner*, skala keabuan, warna, dan warna berindeks.

### 2.3.4 Pengolahan Citra

Kegiatan untuk mengubah informasi citra fisik non *digital* menjadi *digital* disebut sebagai pencitraan (*imaging*). Citra *digital* dapat diolah dengan komputer karena berbentuk data *numeris*. Suatu citra *digital* melalui pengolahan citra *digital* (*digital image processing*) menghasilkan citra *digital* yang baru; termasuk di dalamnya adalah perbaikan citra (*image restoration*) dan peningkatan kualitas citra (*image enhancement*). Sedangkan analisis citra *digital* (*digital image analysis*) menghasilkan suatu keputusan atau suatu data; termasuk di dalamnya adalah pengenalan pola (*pattern recognition*), sebagaimana disajikan pada Gambar 2.3.



Gambar 4. Urutan Pengolahan Citra Digital

### 2.3.5 Operasi Pengolahan Citra

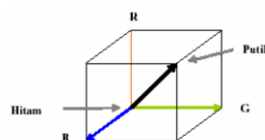
Pengolahan citra pada dasarnya dilakukan dengan cara memodifikasi setiap titik dalam citra tersebut sesuai dengan keperluan. Secara garis besar, modifikasi tersebut dikelompokkan menjadi :

1. Operasi titik, dimana setiap titik diolah secara tidak gayut terhadap titik-titik yang lain.
2. Operasi global, dimana karakteristik global (biasanya berupa sifat statistik) dari citra digunakan untuk memodifikasi nilai setiap titik.
3. Operasi temporal/berbasis bingkai, dimana sebuah citra diolah dengan cara dikombinasikan dengan citra lain.
4. Operasi geometri, dimana bentuk, ukuran, atau orientasi citra dimodifikasi secara geometris.
5. Operasi banyak titik bertetangga, dimana data dari titik yang bersebelahan (bertetangga) dengan titik yang ditinjau ikut berperan dalam mengubah nilai.
6. Operasi morfologi, yaitu operasi yang berdasarkan segmen atau bagian dalam citra yang menjadi perhatian.

Batasan yang dipakai dalam pengolahan citra yang difokuskan pada format citra skala keabuan 8 bit dengan warna hitam pekat untuk nilai minimum (0) dan warna putih cemerlang untuk nilai maksimal (255), serta citra warna *true color*.

### 2.3.6 Pixel

*Pixel* adalah bagian terkecil dari sebuah gambar. *Pixel* kependekan dari *Picture Element*. *Pixel* menggambarkan posisi koordinat dan mempunyai intensitas yang dapat dinyatakan dengan bilangan. Intensitas ini menunjukkan warna gambar, dalam bentuk RGB (*Red, Green, Blue*).



Gambar 5. Koordinat RGB

## 2.4 Motion Detection

*Motion detection* merupakan penelitian yang penting dalam keilmuan *computer vision*. Banyak metode *motion detection* yang telah ditemukan, satu diantaranya adalah dengan menghitung perbedaan nilai-nilai intensitas pada suatu piksel dari dua *frame* gambar yang diambil secara berturut-turut yang kemudian dilakukan proses *thresholding* untuk menentukan adanya perubahan objek atau tidak.

Sekalipun metode ini sangat sederhana dalam proses implementasinya, tapi metode ini merupakan metode dasar dari proses *motion detection*. Hanya saja metode ini kurang efektif untuk menentukan pergerakan objek secara keseluruhan, terutama bagian dalam dari objek yang bergerak, akan tetapi secara umum metode ini sudah mampu mengidentifikasi adanya perubahan objek. Pengurangan nilai-nilai intensitas setiap piksel yang ada pada *background* terhadap suatu *image* baru merupakan metode yang paling populer untuk proses *motion detection*.

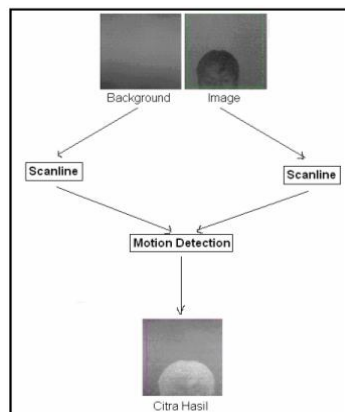
Untuk mengetahui adanya perubahan pada objek yang bergerak maka operasi yang digunakan adalah pengurangan. Operasi pengurangan pada bagian yang tidak bergerak dalam citra akan menghasilkan nilai RGB (*red, green, blue*) per piksel sama dengan nol, sedangkan bagian yang bergerak dalam citra memberikan nilai RGB (*red, green, blue*) per piksel tidak sama dengan nol. Dengan mengevaluasi selisih nilai RGB per piksel, dapat diketahui apakah pada citra terdapat obyek yang bergerak.

Nilai RGB per piksel pada *koordinat* x,y citra pertama dikurangkan dengan nilai piksel koordinat x,y citra kedua. Seperti dijelaskan di atas jika hasil pengurangan bernilai nol maka dinyatakan tidak ada gerakan, begitu pula sebaliknya. Untuk mengantisipasi nilai kurang dari nol maka hasil dari pengurangan ditambahkan suatu konstanta.

Obyek yang ditangkap dibuat bercahaya untuk mempermudah pendeteksian gerakan. Dengan obyek yang bercahaya perbedaan citra dari sisi nilai RGB per piksel akan lebih jelas dan pendeteksian gerakan jadi lebih mudah.

Pada gambar 6 sebelah kiri atas merupakan gambar asli atau *current image (background)*, sebelah kanan atas merupakan gambar dengan adanya objek atau *previous image*, sedangkan gambar dibawahnya merupakan hasil dari proses *motion detection*. Langkah-langkah *motion detection* adalah :

1. Memasukkan gambar pertama sebagai *background*.
2. Memasukkan gambar kedua sebagai gambar yang ada objeknya.
3. Gambar pada langkah kesatu dan kedua lakukan *scanline*, fungsinya adalah untuk menemukan apakah ada perubahan objek atau tidak ada perubahan. *Scanline* dilakukan dari kiri atas gambar sampai dengan kanan bawah gambar.
4. Lakukan perbandingan antara gambar *background* dengan gambar satunya lagi, hal ini dilakukan supaya bisa mengetahui adanya perubahan atau adanya perbedaan antara gambar kesatu dan gambar kedua. Pada langkah ini biasanya ada proses *thresholding* dan deteksi pinggiran jika diperlukan. Ini juga disebut proses *motion detection*.
5. Setelah melakukan proses diatas maka akan dihasilkan citra hasil *motion detection*.



Gambar 6 *Motion detection*

## 2.5 Program Aplikasi (Software)

Program merupakan sekumpulan instruksi atau perintah yang tersusun secara teratur dan sistematis, dimana urutan perintah itu menunjukkan urutan proses yang akan dijalankan. Program aplikasi adalah penerapan dari kumpulan instruksi atau perintah terperinci yang sudah dipersiapkan supaya komputer melakukan fungsinya dengan cara yang sudah ditentukan. Sekumpulan intruksi yang digunakan untuk mengatur perangkat keras komputer agar melaksanakan tindakan tertentu[12].

Seperti halnya manusia yang menggunakan bahasa untuk berkomunikasi, komputerpun juga memiliki bahasa tersendiri. Supaya manusia dapat berkomunikasi dengan komputer. Programmer perlu menuliskan program dalam bahasa yang bisa dimengerti oleh komputer. Bahasa inilah yang disebut dengan bahasa pemrograman atau bahasa komputer. Bahasa pemrograman terdiri atas sekumpulan

instruksi yang ditunjukkan agar orang bisa menuangkan perintah yang nantinya akan dijalankan oleh komputer.

## 2.6 Visual Basic 6.0

Visual Basic 6.0 merupakan salah satu perangkat lunak pendukung pemrograman *visual*. Kelebihan yang dimiliki oleh Visual Basic 6.0 adalah memiliki kemampuan membuat program berorientasi objek atau biasa dengan OOP (*Object Oriented Programming*) berbasis *Windows™*. Untuk memudahkan OOP Visual Basic 6.0 dilengkapi dengan sejumlah aplikasi pengembangan seperti *Database Designer*, *Table Designer*, *Program Editor*, *Project Manager*, *Debugger*, *Class Design*, *Report Designer*, *Menu Design*, *View Designer*, dan *Form Designer* sehingga dapat menciptakan kemudahan dalam membuat sebuah program”[13].

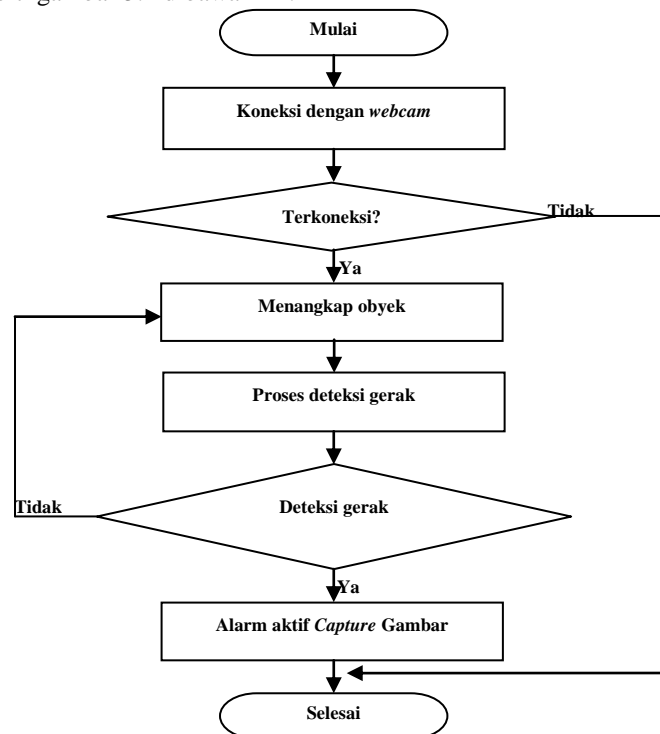
Sebagai suatu *tool programming*, Visual Basic 6.0 memiliki keunggulan dalam hal kelengkapan fungsi dan perintah serta fasilitas yang berhubungan dengan pengolahan data. Dengan Visual Basic inilah aplikasi pemantau ruangan ini dapat dibuat karena memiliki fitur penunjang untuk menangkap citra yang dihasilkan oleh kamera dan mampu melakukan proses-proses yang dibutuhkan.

## 2.7 Algoritma Deteksi Gerak (Secara Umum)

Berdasarkan cara kerja sistem operasi deteksi gerakan (*motion detection*) ini terdiri dari beberapa tahap, yaitu:

1. Mengaktifkan kamera (webcam).
2. Kamera mengambil obyek ditampilkan dalam bentuk video.
3. Hasil dari pengambilan obyek tersebut di pilih kemudian tampilan berbentuk gambar diam dalam format file JPG.
4. Proses penampilan gambar dilakukan berdasarkan pengaturan timer.
5. Timer menentukan kapan citra ditampilkan sebagai citra pertama dan kedua, proses ini dilakukan secara kontinyu.
6. Setelah citra terambil dilakukan proses perbandingan dengan menggunakan metode deteksi gerakan. Apabila ada perbedaan nilai di identifikasikan ada suatu gerakan.
7. Identifikasi ada perubahan atau tidak adalah dengan penyimpanan citra dengan bentuk JPG dan aktifnya alarm.
8. Proses ini terus berjalan sampai aplikasi dimatikan.

Dengan demikian sesuai cara kerja sistem operasi deteksi gerakan, maka menghasilkan *flowchart motion detection* seperti gambar 5.2 dibawah ini:



Gambar 7 Flowchart Mendeteksi Gerak

Berdasarkan tahapan cara kerja sistem operasi deteksi gerakan (*motion detection*) dan diagram alir mendeteksi gerak, maka dapat digambarkan alur dari algoritmanya sebagai berikut :

For i = 0 To 640 / inten - 1  
For j = 0 To 480 / inten - 1

1

c = Camera1.Point(i \* inten \* Tppx, j \* inten \* Tppy)

2

R = c Mod 256  
G = (c \ 256) Mod 256  
B = (c \ 256 \ 256) Mod 256

3

c2 = P(i, j)

4

R2 = c2 Mod 256  
G2 = (c2 \ 256) Mod 256  
B2 = (c2 \ 256 \ 256) Mod 256

3

If Abs(R - R2) < Tolerance And Abs(G - G2) < Tolerance And Abs(B - B2) < Tolerance Then

5

Ri = Ri + 1  
POn(i, j) = True  
Else  
Wo = Wo + 1

6

P(i, j) = Camera1.Point(i \* inten \* Tppx, j \* inten \* Tppy)  
Camera1.PSet (i \* inten \* Tppx, j \* inten \* Tppy), vbRed  
POn(i, j) = False  
End If

RealRi = 0  
For i = 1 To 640 / inten - 2  
For j = 1 To 480 / inten - 2  
If POn(i, j) = False Then  
If POn(i, j + 1) = False Then  
If POn(i, j - 1) = False Then  
If POn(i + 1, j) = False Then  
If POn(i - 1, j) = False Then  
RealRi = RealRi + 1  
Camera1.PSet (i \* inten \* Tppx, j \* inten \* Tppy), vbGreen  
End If  
End If  
End If  
End If  
End If  
Endfor  
Endfor

7

Keterangannya :

1. Logika tersebut menjelaskan resolusi umum pada webcam yaitu 640 x 480.
2. Memanggil nilai untuk pembandingan yang pertama.
3. Memecahkan nilai dari RGB dimana Penulis mengambil kedalaman bit dalam gambar yaitu sebesar 8 bit, sehingga akan menghasilkan 256 warna yang dihasilkan dalam 8 bit pada kedalaman gambar yang pertama.
4. Untuk pembandingan nilai yang ke 2.



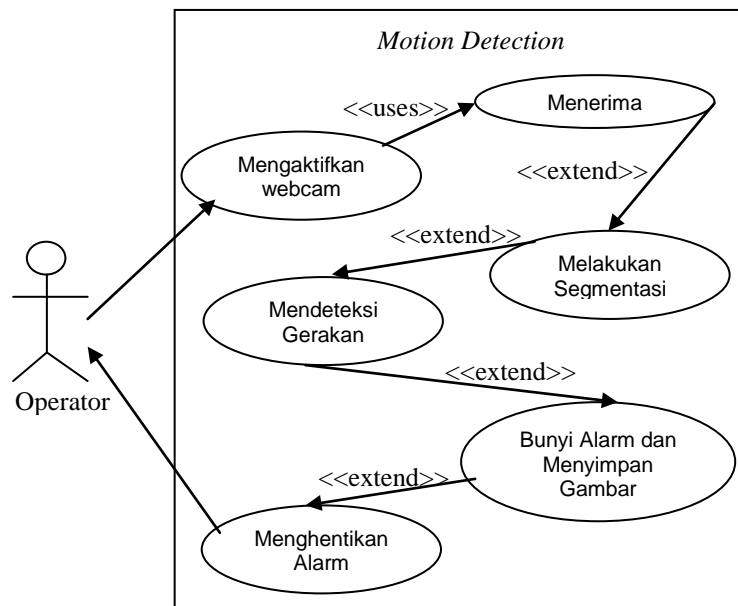
5. Perbandingan antara nilai RGB dengan toleransi nilai RGB berikutnya untuk nilai pixel pada gambar yang kedua.
6. Untuk mencari pergerakan, kemudian akan menampilkan titik-titik merah jika terjadi pergerakan.
7. Selain titik-titik merah akan muncul juga titik-titik hijau, dimana titik-titik hijau tersebut mempunyai fungsi untuk memperjelas bentuk dari suatu objek yang ditangkap oleh *webcam*.

### 2.8 File pendukung *Capture Citra*

Terdapat beberapa cara untuk melakukan *capture* citra dari kamera video. Diantaranya penggunaan *Direct-X* dan *AviCap*. Pada *AviCap*, koneksi kamera memanfaatkan *library* yang telah ada pada *Operating Sistem (OS) Windows* berupa *avicap32.dll* dan *user32.dll*. Pada penulisan Skripsi ini, digunakan komponen *avicap32.dll* untuk koneksi kamera pada program aplikasi. Sedangkan untuk proses *capture*, digunakan *windows device context (WindowDC)*. *WindowDC* diatur untuk menangkap tampilan layar (*screen capture*) yang diarahkan pada posisi komponen *avicap32.dll*. Karena komponen *avicap32.dll* menampilkan *preview* gambar video dari kamera, maka hasil *screen capture*-nya adalah citra yang didapatkan dari hasil pencitraan kamera video.

### 3. Hasil dan Analisa

Dibawah ini merupakan bentuk dari *use case* pada sistem pemantauan ruangan dengan *webcam* menggunakan metode *motion detection* :



**Gambar 8.** Use Case Pemantauan Ruangan

#### 3.1 Proses Pendeteksi

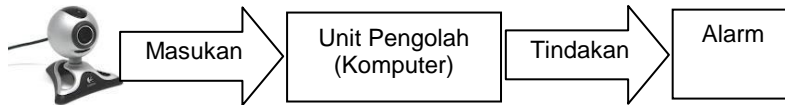
Pada proses pendeteksi ini *webcam* mencoba menangkap objek. Ketika objek berhasil ditangkap oleh *webcam* maka akan tersimpan pada komponen gerak yaitu *windows device context (WindowDC)*. Gambar akan terus *diupdate* untuk mendeteksi apakah adanya gerakan yang terjadi. Dapat dikatakan bergerak adalah jika suatu objek berpindah tempat dari suatu posisi ke posisi lainnya atau dari suatu titik ke titik lainnya.

Terdeteksi adanya suatu perpindahan atau pergerakan dari suatu titik ke titik lainnya pada objek yang ditangkap oleh *webcam* maka akan memunculkan titik-titik merah, dimana titik merah ini sebagai penanda adanya suatu objek yang mengalami pergerakan. Jika pergerakan semakin banyak dan jelas maka titik-titik merah tersebut akan bertambah banyak.

Selain titik-titik merah yang muncul, akan muncul titik-titik hijau yang berfungsi untuk memperjelas bentuk dari suatu objek yang ditangkap oleh *webcam*. titik-titik merah dan titik-titik hijau tersebut akan selalu mengikuti objek yang bergerak. Titik-titik merah dan titik-titik hijau tersebut tercipta dari fungsi menggambar gerak, dimana fungsi tersebut dipanggil pada saat objek yang tersimpan pada komponen gerak *WindowDC* dan dibantu dengan bahasa pemrograman Visual Basic 6.0 sebagai bahasa penghubung antara manusia, komputer dan alat pendukung yang digunakan yaitu *webcam*.

### 3.2 Arsitektur Sistem *Motion Detection*

Sistem yang akan di bangun oleh Penulis terdiri dari unit kamera video untuk pengambilan gambar, unit komputer sebagai pengolah data serta *speaker* untuk alarm keamanan. Arsitektur sistem *motion detection* secara umum dapat digambarkan sebagai berikut :



Gambar 9. Diagram Rancangan Sistem *Motion Detection*

### 3.3 Perancangan dan Pembuatan GUI Perangkat Lunak

Tujuan dari perancangan GUI (*Graphical User Interface*) adalah untuk mendapatkan hasil kinerja maksimal dari interaksi manusia dan komputer. Tampilan GUI yang baik diharapkan dapat memberikan kemudahan dan efisiensi kepada penggunaanya.

Dari tujuan diatas berikut ini rancangan program yang akan dibangun oleh penulis dengan mempunyai ketentuan-ketentuan sebagai berikut :

1. Mampu menampilkan *streaming video* dari perangkat *webcam* ke program aplikasi.
2. Mampu melakukan proses *motion detection* terhadap objek.

Mampu memberikan *respon* dari proses *motion detection* yaitu berupa hasil rekaman gambar dan bisa juga mengaktifkan *alarm* jika diperlukan.

### 3.4 Pembuatan GUI *Motion Detection*

Agar hasil kinerja maksimal antara interaksi manusia dan komputer Penulis membuat GUI dengan menggunakan bahasa pemrograman visual basic 6.0. Berikut ini *form* GUI yang penulis buat :

The form contains the following elements:

- Text labels: % Movement, Real Movement :0, Completed in : 0
- Two large empty rectangular boxes for video display.
- Checkboxes:  RECORD,  ALARM
- Buttons: START, SHOW IMAGE, REFRESH LIST, CLEAR IMAGE, and a vertical E X I T button.
- A rounded rectangular container labeled HASIL CAPTURE containing a grey rectangular area.

Gambar 10. Rancangan *Form Motion Detection*

Pengujian program aplikasi *motion detection* ini, sebagai berikut :

1. Pasangkan USB *webcam* pada slot USB yang tersedia pada *laptop*.
2. Jalankan aplikasi *motion detection*
3. Masukkan *user name* dan *password* sebagai berikut :

*User Name* : SCRT0001

*Password* : Admin

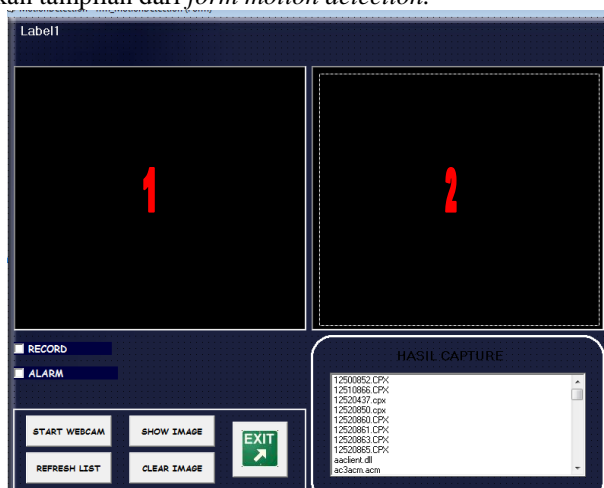


**Gambar 11.** Pemasukan Data Pengguna Pada *Form Login* program aplikasi *motion detection*



**Gambar 12.** Tampilan Menu Utama

4. Jika *user name* dan *password* yang dimasukan oleh pengguna benar, maka akan langsung masuk ke *menu utama*, berikut ini tampilan dari *menu utama*.
5. Di dalam *menu utama* terdapat *menu bar master*, yang didalamnya terdapat *file motion detection*.
6. Berikut ini merupakan tampilan dari *form motion detection*.



**Gambar 13.** Tampilan *Form Motion Detection*

Keterangan gambar :

- a. Pada *form motion detection* terdapat 5 tombol, diantaranya tombol *start webcam* yang berfungsi untuk mengaktifkan *webcam*, *show image* mempunyai fungsi untuk menampilkan dan meriview hasil tangkapan *webcam* dalam bentuk gambar, tombol *refresh file* mempunyai fungsi untuk merefresh pada kotak hasil *capture*, tombol *clear image* mempunyai fungsi untuk menghapus *file* gambar, sedangkan tombol *exit* mempunyai fungsi untuk keluar dari *form motion detection*.
- b. Selain tombol *start webcam*, *show image*, *refresh file*, *clear image*, dan *exit* terdapat juga 2 kotak yang mempunyai fungsi dan peranan yang berbeda, yaitu kotak yang bernomor 1 mempunyai fungsi untuk menampilkan dalam bentuk *video* keadaan suatu ruangan yang sedang dipantau, sedangkan kotak yang bernomor 2 mempunyai fungsi untuk melihat hasil dari penangkapan objek yang tertangkap oleh *webcam* dalam bentuk gambar (*image*).
- c. Pada program aplikasi *motion detection* juga terdapat *checkbox record* yang mempunyai fungsi sebagai pilihan jika hasil pemantauan ingin di simpan, sedangkan *checkbox alarm* mempunyai fungsi untuk mengaktifkan *alarm* berfungsi sebagai pengingat jika terjadi pergerakan pada objek yang sedang dipantau.
- d. Terdapat juga *file list box* yang mempunyai fungsi untuk menampilkan nama *file* yang tertangkap oleh *webcam* karena adanya pergerakan.

### 3.5 Data Hasil Pengujian

Ketika aplikasi *motion detection* dijalankan dengan mengikuti langkah pengujian yang telah disebutkan sebelumnya, program aplikasi *motion detection* akan menampilkan keadaan ruangan kantor dalam bentuk *video*, dimana tampilan tersebut sebagai tahapan pencarian objek yang bergerak, sehingga jika terdeteksi adanya objek yang bergerak maka aplikasi *motion detection* akan menyimpan hasil pendeteksian tersebut dalam bentuk gambar (*image*).

Pengujian dilakukan di ruangan dengan beberapa pengujian diantaranya pengujian terhadap jarak antara *webcam* dengan objek yang bergerak. Berikut ini data hasil dari pengujian :

**Tabel 1. Pengujian Terhadap Jarak**

Jarak (m)	Respon (ya/tidak)	Movements (%)
1	Ya	32 %
1,5	Ya	28 %
2	Ya	18 %
2,5	Ya	15 %
3	Ya	12 %
3,5	Ya	10 %
4	Ya	9 %
4,5	Ya	7 %
5	Ya	3 %
5,5	Tidak	0
6	Tidak	0

Pengujian kedua masih dilakukan di ruangan dengan mengamati jenis pergerakan dari objek yang bergerak. Berikut ini data hasil dari pengujian :

1. Pengujian dengan mengamati jenis pergerakan dengan jarak 1 m.

**Tabel 2. Pengujian Jenis Pergerakan Berdasarkan Jarak**

Jarak (meter)	Jenis Pergerakan	Movements (%)	Respon (ya/tidak)	Alarm
1	Jalan ke kiri	32 %	Ya	Hidup
	Jalan ke kanan	32 %	Ya	Hidup
	Lari ke kiri	32 %	Ya	Hidup
	Lari ke kanan	32 %	Ya	Hidup
2	Jalan ke kiri	18 %	Ya	Hidup
	Jalan ke kanan	18 %	Ya	Hidup
	Lari ke kiri	18 %	Ya	Hidup
	Lari ke kanan	18 %	Ya	Hidup
3	Jalan ke kiri	12 %	Ya	Hidup
	Jalan ke kanan	12 %	Ya	Hidup
	Lari ke kiri	12 %	Ya	Hidup
	Lari ke kanan	12 %	Ya	Hidup
4	Jalan ke kiri	9 %	Ya	Hidup
	Jalan ke kanan	9 %	Ya	Hidup
	Lari ke kiri	9 %	Ya	Hidup
	Lari ke kanan	9 %	Ya	Hidup
5	Jalan ke kiri	3 %	Ya	Hidup
	Jalan ke kanan	3 %	Ya	Hidup
	Lari ke kiri	3 %	Ya	Hidup
	Lari ke kanan	3 %	Ya	Hidup
6	Jalan ke kiri	0	Tidak	Tidak
	Jalan ke kanan	0	Tidak	Tidak
	Lari ke kiri	0	Tidak	Tidak
	Lari ke kanan	0	Tidak	Tidak

Pengujian ketiga masih dilakukan di ruangan dengan mengamati jumlah pengambilan gambar yang tertangkap oleh *webcam* pada objek yang bergerak. Berikut ini data hasil dari pengujian :

1. Pengujian jumlah pengambilan gambar dengan jarak 1 m.

**Tabel 3. Pengujian Jumlah Pengambilan Gambar Berdasarkan Jarak**

Jarak (meter)	Jenis Penelitian	Objek berjalan ke kiri dan kanan	Objek berlari ke kiri dan kanan
1	Jumlah Gambar	16	6
	Movements %	32	26
2	Jumlah Gambar	10	7
	Movements %	18	16
3	Jumlah Gambar	7	5
	Movements %	12	7
4	Jumlah Gambar	7	5
	Movements %	9	5
5	Jumlah Gambar	7	3
	Movements %	3	3
6	Jumlah Gambar	-	-
	Movements %	0	0

### 3.6 Analisis Data Hasil Pengujian

Pengujian aplikasi *motion detection* yang dilakukan di ruangan pada siang hari dengan keadaan ruangan terang. Pengujian ini menguji pendeteksian dengan jarak objek ke *webcam* yang mempunyai jarak yang berbeda-beda. Dengan objek yang menjadi bahan pengujian ialah dengan objek manusia.

Dari pengujian pertama dengan pengujian respon program aplikasi jika terjadi pergerakan pada jarak yang berbeda-beda bisa di dapatkan hasil bahwa jarak antara *webcam* ke objek hanya sampai batas jarak 5 meter, sedangkan ketika jarak di tambah menjadi 5,5 dan 6 meter maka aplikasi tidak bisa menangkap objek meskipun objek yang diuji bergerak.

Pengujian kedua dengan pengujian jenis pergerakan oleh objek yang berjalan ke kiri ke kanan serta berlari ke kiri ke kanan dengan jarak dari 1 meter sampai 6 meter menghasilkan bahwa objek dengan jenis pergerakan berjalan ke kiri kanan dan berlari ke kiri kanan, selama objek tertangkap oleh *webcam* maka aplikasi tetap mendeteksi bahwa adanya pergerakan, tetapi yang membedakannya yaitu besarnya *movements* pada jenis pergerakan yang diteliti.

Sedangkan pada pengujian yang ketiga yaitu menguji jumlah gambar pergerakan yang berhasil di *capture* oleh program aplikasi dengan jarak 1 sampai 6 meter dan dengan dua jenis pergerakan yaitu objek berjalan ke kiri ke kanan dan objek bergerak dengan berlari ke kiri dan ke kanan.

Maka dari pengujian tersebut di dapat bahwa jumlah gambar yang berhasil di *capture* oleh program aplikasi berbeda-beda di karenakan pergerakan tersebut mempunyai *movements* yang berbeda sehingga pengujian dengan jarak 1 meter mempunyai jumlah gambar dari hasil *capture* yang lebih banyak dibandingkan dengan pengujian yang berjarak 2 sampai 5 meter, sedangkan objek yang berjarak 6 meter dengan *webcam* tidak ter*capture*.

### 4. Kesimpulan

Dengan penerapan deteksi gerakan (*motion detection*) maka proses perekaman gambar bisa diatur agar hanya terjadi perekaman gambar pada saat adanya suatu pergerakan objek saja, sehingga dapat menghasilkan efisiensi media penyimpanan data. Sistem ini dapat dimanfaatkan dalam hal pengamanan untuk mengawasi suatu ruangan. Data yang tersimpan berbentuk file gambar dengan format JPEG. Implementasi sistem ini dapat dilakukan menggunakan peralatan standar yang umum digunakan oleh masyarakat. Pada penelitian ini penulis menggunakan peralatan kamera yang memiliki kualitas standar, sehingga output yang dihasilkan kurang optimal, jarak optimum untuk area pengawasan relatif masih dekat, diharapkan dengan menggunakan kualitas peralatan kamera yang baik maka area pengawasan dapat bertambah dengan kualitas gambar yang semakin bagus. Penelitian ini dapat dikembangkan supaya kualitas gambar yang dihasilkan menjadi semakin baik dan jarak jangkauan yang lebih jauh, atau bahkan data yang disimpan dapat berbentuk gambar bergerak (video).

### Daftar Pustaka

- [1] Y.L. Tian and A. Hampapur, "Robust Salient Motion Detection with Complex Background for Real-time Video Surveillance," IEEE Computer Society Workshop on Motion and Video Computing, Breckenridge, Colorado, January 5 and 6, 2005.

- [2] C. Bahlmann, Y. Zhu, Y. Ramesh, M. Pellkofer, T. Koehler, "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information," IEEE Intelligent Vehicles Symposium, Proceedings, 2005, pp. 255-260.
  - [3] A. Manzanera and J.C. Richefeu, "A new motion detection algorithm based on  $\Sigma$ - $\Delta$  background estimation," Pattern Recognition Letters, vol. 28, n 3, Feb 1, 2007, pp. 320-328.
  - [4] Y. Ren, C.S. Chua and Y.K. Ho, "Motion detection with nonstationary background," Machine Vision and Applications, vol. 13, n 5-6, March, 2003, pp. 332-343.
  - [5] J. Guo, D. Rajan and E.S. Chng, "Motion detection with adaptive background and dynamic thresholds," 2005 Fifth International Conference on Information, Communications and Signal Processing, 06-09 Dec. 2005, pp. 41-45.
  - [6] A. Elnagar and A. Basu, "Motion detection using background constraints," Pattern Recognition, vol. 28, n 10, Oct, 1995, pp. 1537-1554
  - [7] Firdausy, K., Saudi, Y, "Deteksi Api *Real-Time* dengan Metode Thresholding Rerata RGB", Jurnal Ilmiah TELKOMNIKA Teknik Elektro UAD, Vol. 5 No. 2, Agustus 2007 : 42-48
  - [8] Daryono, "Perancangan Pendeteksi Gerakan menggunakan Webcam Untuk Sistem Pemantauan", Skripsi S-1, Teknik Elektro, Universitas Ahmad Dahlan, Yogyakarta, 2008
  - [9] Kartika Firdausy, Selamat Riyadi, Tole Sutikno, Muchlas, "Aplikasi Webcam Untuk Sistem Pemantauan Ruang Berbasis Web", Jurnal Ilmiah TELKOMNIKA Teknik Elektro UAD, Vol. 6, No. 1, April 2008 : 39 – 48
  - [10] Kristanto, Andri, "Rekayasa Perangkat Lunak (Konsep Dasar)", Edisi Pertama, Yogyakarta : Gava Media, 2004.
  - [11] Basuki, Ahmad, "Pengolahan Citra Digital menggunakan Visual Basic", Yogyakarta : Graha Ilmu, 2005 : 1.
  - [12] Tim Penelitian dan Pengembangan, "Pedoman Praktis Pemrograman Visual Foxpro 9,0", Yogyakarta, Wahana Computer, 2006 : 15
  - [13] Kadir Abdul, "Pemrograman Basis Data dengan Visual Basic", Jilid 1 & 2, Yogyakarta : Andi Offset, 2005).
-