

PENDETEKSI GERAK BERBASIS KAMERA MENGGUNAKAN OPENCV PADA RUANGAN

Kurniawan Dwi Irianto

Jurusan Teknik Informatika, Fakultas Komunikasi dan Informatika,
Universitas Muhammadiyah Surakarta
Email : kdwiirianto@gmail.com

ABSTRACT

At this decade, the need of monitoring systems in every field is increase rapidly. Many of monitoring systems is applied to improve the security systems as well as the productivities. The implementation always based on the need to keep watch and record all activities that occurs in one location with purpose that if something emergency is happening, we can know and handle it immediately. A building is one of the place that very important to keep save if there are much valuable stuff. Surveillance systems that currently using is operator surveillance camera where many of error factor from human being is unprecise so it is not very efficient to get a maximum result. A surveillance system using camera that can detect the motion as well as record the motion automatically is a alternative method to improve the efficiency where it will more save memory and there is no operator. This paper is describe how to detect the motions and record them using opencv library with web camera. The method used background subtraction and frame differencing technique.

Kata kunci: *Motions, camera, background subtraction technique, frame differencing*

PENDAHULUAN

Saat ini kebutuhan sistem monitoring di berbagai sektor meningkat dengan pesat. Semakin banyak sistem monitoring diterapkan untuk tujuan peningkatan aspek keamanan dan produktivitas. Penerapan monitoring selalu berdasarkan pada kebutuhan pengawasan secara berkala dan merekam segala aktivitas yang berlangsung di lokasi tersebut dengan harapan ketika terjadi suatu hal kritis / penting, maka dapat segera diketahui dan ditangani.

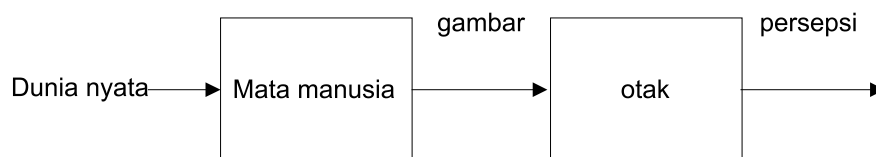
Sistem monitoring biasanya diterapkan untuk aspek keamanan sebagai contoh pada perbankan, pergudangan, perkantoran, berbagai fasilitas publik seperti bandara, stasiun, hingga digunakan pada rumah tinggal.

Sedangkan penerapan sistem monitoring untuk aspek produktivitas sebagai contoh diterapkan pada sektor manufaktur atau industri di mana manajemen dapat memonitor atau memantau aktivitas produksi para pekerja / buruh, mengontrol instrumentasi proses, instalasi permesinan, dan lain-lain. Dan tentunya masih banyak tujuan-tujuan lain yang mendasari penerapan sistem monitoring tersebut.

Oleh karena itu penggunaan kamera pada sistem pengawasan sangat dibutuhkan. Akan tetapi yang menjadi permasalahan adalah kamera yang dipasang selalu merekam meskipun tidak ada gerakan atau kejadian yang terjadi, akibatnya terjadi penggunaan

memori yang sia-sia dan salah satu alternatif untuk mengatasi permasalahan ini adalah dengan merancang suatu perangkat lunak yang dapat meningkatkan efisiensi kamera, sehingga kamera hanya akan mendeteksi dan merekam apabila ada gerak atau benda yang bergerak.

Dengan menggunakan teknik *background subtraction* dan *frame differencing* maka gerakan yang berada dalam kawasan pantauan kamera dapat dideteksi dan direkam ke dalam Harddisk. Teknik *background subtraction* mengurangkan frame sekarang (f_t) dengan frame background untuk mendapatkan objek yang bergerak sedangkan teknik *frame differencing* mengurangkan frame sekarang (f_t) dengan frame sebelumnya (f_{t-1}) untuk mendapat objek yang bergerak pula.



Gambar 1. Proses *computer vision* mendapatkan suatu persepsi

PENGOLAHAN CITRA DIGITAL

Secara harafiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera pemindai (*scanner*) dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam.

Meskipun sebuah citra kaya informasi, namun seringkali citra yang kita miliki

COMPUTER VISION

Vision secara bahasa bisa diartikan penglihatan. *Vision* juga bisa diartikan sebagai suatu proses pengamatan apa yang ada pada dunia nyata melalui panca indra penglihatan manusia. Adapun *computer vision* adalah suatu pembelajaran menganalisa gambar dan video untuk memperoleh hasil sebagaimana yang bisa dilakukan manusia. Pada hakikatnya, *Computer vision* mencoba meniru cara kerja sistem visual manusia (*Human Vision*). Manusia melihat objek dengan indera penglihatan (mata), lalu citra objek diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti objek apa yang tampak dalam pandangan matanya. Hasil interpretasi ini mungkin digunakan untuk pengambilan keputusan (misalnya menghindari kalau melihat mobil melaju di depan).

mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur (*blurring*) dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasikan karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami gangguan mudah diinterpretasikan (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (*image processing*).

OPENCV (INTEL® OPEN SOURCE COMPUTER VISION LIBRARY)

OpenCV merupakan singkatan dari *Intel Open Source Computer Vision Library* yang sekurang-kurangnya terdiri dari 300 fungsi-fungsi C, bahkan bisa lebih. Software ini gratis, dapat digunakan dalam rangka komersil maupun non komersil, tanpa harus membayar lisensi ke intel. OpenCV dapat beroperasi pada komputer berbasis windows ataupun linux.

Pustaka *OpenCV* adalah suatu cara penerapan bagi komunitas *open source vision* yang sangat membantu dalam kesempatan meng-update penerapan *computer vision* sejalan dengan pertumbuhan PC (*Personal Computer*) yang terus berkembang. Software ini menyediakan sejumlah fungsi-fungsi *image processing*, seperti halnya dengan fungsi-fungsi analisis gambar dan pola.

Beberapa contoh aplikasi dari OpenCV adalah pada *Human-Computer Interaction* (*Interaksi Manusia-Komputer*); *Object Indentification* (*Identifikasi Objek*), *Segmentation* (*Segmentasi*), dan *Recognition* (*Pengenalan*); *Face Recognition* (*Pengenalan Wajah*); *Gesture Recognition* (*Pengenalan Gerak Isyarat*), *Motion Tracking* (*Penjajakan Gerakan*), *Ego Motion* (*Gerakan Ego*), dan *Motion Understanding* (*Pemahaman Gerakan*); *Structure From Motion* (*Gerakan Dari Struktur*); dan *Mobile Robotics* (*Robot-Robot Yang Bergerak*).

TEKNIK BACKGROUND

SUBTRACTION DAN FRAME

DIFFERENCING

Yang dimaksud *background* adalah sejumlah piksel-piksel gambar yang diam

dan tidak bergerak didepan kamera. Model *background* yang paling sederhana mengasumsikan bahwa seluruh kecerahan piksel *background* berubah-ubah secara bebas, tergantung pada distribusi normalnya. Karakteristik *background* dapat dihitung dengan mengakumulasi beberapa jumlah *frame* sehingga akan menemukan jumlah nilai-nilai piksel dalam lokasi $s_{(x,y)}$ dan jumlah *square-square* $sq_{(x,y)}$ yang memiliki nilai untuk setiap lokasi piksel. Sedangkan *foreground* adalah semua objek yang ada selain *background* dan biasanya *foreground* ini ada setelah didapatkannya *background*.

Background subtraction merupakan salah satu tugas penting yang pertama kali dikerjakan pada aplikasi *computer vision*. Output dari *background subtraction* biasanya adalah inputan yang akan diproses pada tingkat yang lebih lanjut lagi seperti *tracking* objek yang teridentifikasi. Kualitas *background subtraction* umumnya tergantung pada teknik pemodelan *background* yang digunakan untuk mengambil *background* dari suatu layar kamera. *Background subtraction* biasanya digunakan pada teknik segmentasi objek yang dikehendaki dari suatu layar, dan sering diaplikasikan untuk sistem pengawasan.

Tujuan dari *background subtraction* itu sendiri adalah untuk menghasilkan urutan *frame* dari kamera dan mendeteksi seluruh objek *foreground*. Suatu deskripsi pendekatan yang telah ada tentang *background subtraction* adalah mendeteksi objek-objek *foreground* sebagai perbedaan yang ada antara *frame* sekarang dan gambar *background* dari layar statik. Suatu *piksel* dikatakan sebagai *foreground* jika :

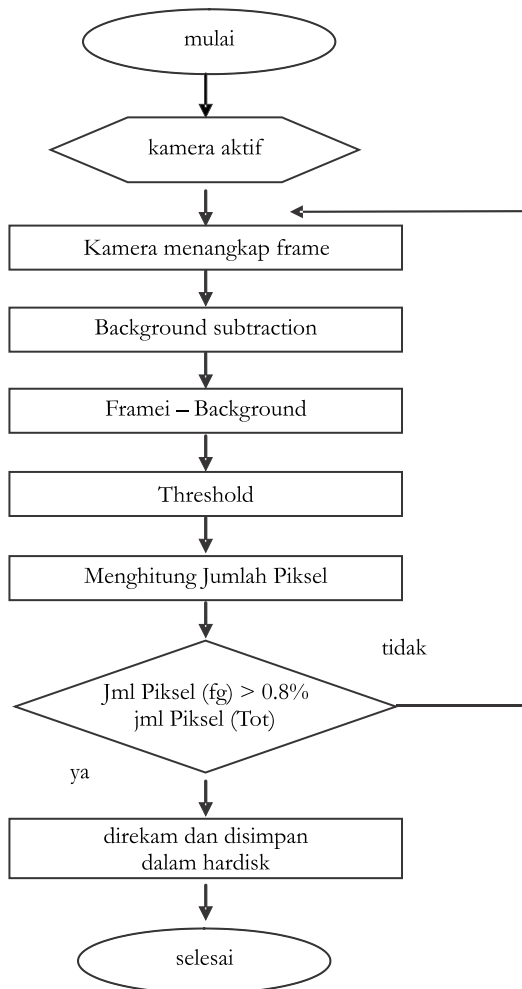
$$|Frame_i - Background_i| > Threshold ..(1)$$

Pendekatan ini sangat sensitif terhadap *threshold*, sehingga *threshold* dapat digunakan untuk mengatur sensitifitas suatu kamera dalam menangkap gambar.

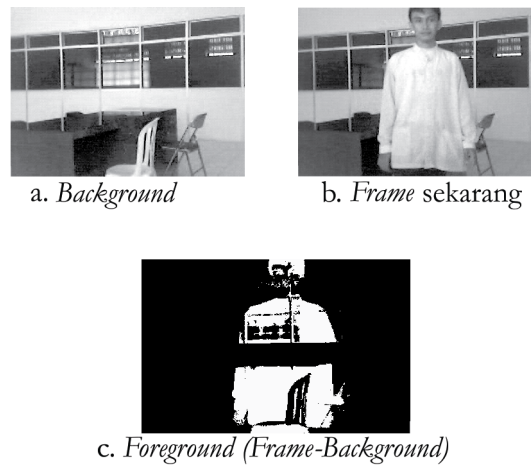
Suatu pendekatan juga dalam mendapatkan *background* adalah dengan Model *Running average*. Dibandingkan model yang lainnya seperti Model *Average* atau *Median*, model ini lebih unggul karena lebih membutuhkan sedikit memori daripada model-model yang lainnya. Model *running average* mempunyai rumus sebagai berikut :

$$B_{i+1} = \alpha * F_i + (1-\alpha) * B_i \dots \dots \dots (2)$$

α adalah tingkat pembelajaran, biasanya mempunyai nilai 0.05, sedangkan B_i adalah *background* dan F adalah *frame*.



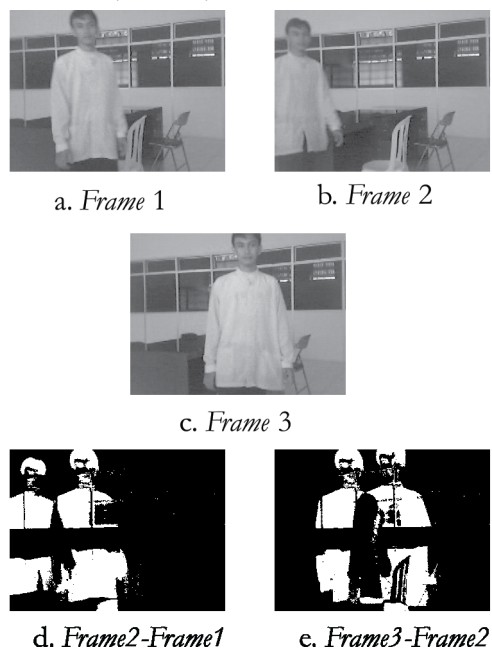
Gambar 2. Sistem Perancangan Metode Background Subtraction



Gambar 4. Metode Background Subtraction (Lokasi di Lab Teknik Elektro UMS)

Sedangkan teknik *frame differencing* atau perbedaan frame adalah menghitung selisih antara frame yang satu dengan frame yang lainnya, dalam hal ini adalah frame sekarang dikurangi frame sebelumnya sehingga mendapatkan objek yang diinginkan.

$$|Frame_i - Frame_{i-1}| > Threshold \dots \dots (3)$$



PENGUJIAN DENGAN KAMERA

Kamera yang digunakan adalah kamera CMOS (*Complementary Metal Oxide Semi-Conductor*) dan CCD (*Charge-Coupled Device*) Logitech yang mempunyai resolusi 320x240 dan 640x320. Langkah-langkah pengujian dapat dilakukan sebagai berikut : Kamera dipasang pada sebuah ruangan yang diasumsikan mempunyai pencahayaan yang tetap. Kemudian kamera dihubungkan ke komputer agar kamera bisa aktif. Aktifkan program pendeteksi gerakan yang telah dirancang. Berikan sebuah aksi yaitu gerakan yang berada di dalam jangkauan kamera untuk mencoba apakah sistem yang dirancang dapat bekerja, gerakan bisa berupa benda atau orang yang bergerak. Atur *threshold*nya pada nilai 50. Kemudian lihat hasilnya sesuai dengan nilai *threshold* dari gerakan tersebut, berupa rekaman video yang direkam ke dalam hardisk.

Percobaan pertama dilakukan dengan menguji program ini menggunakan metode *background subtraction* dengan kamera CCD untuk merekam aktivitas yang ada pada Lab. Teknik Elektro pada hari Jum'at, 16 Februari 2007 selama 24 jam yang dimulai pada jam 10.00 WIB – 10.00 WIB keesokan harinya. Memori yang dibutuhkan untuk menyimpan hasil rekaman adalah sebesar 40 MB. Ini juga tergantung pada banyak-sedikitnya aktivitas yang terjadi, semakin banyak aktivitas maka semakin banyak memori yang dibutuhkan.

Percobaan kedua dilakukan dengan menggunakan 2 kamera dan metode yang berbeda, yaitu metode *background subtraction* dan metode *frame differencing*, pada hari Senin 26 Februari 2007 yang dimulai pada jam 16.00 WIB sampai hari Selasa 27 Februari 2007 jam 15.00 WIB. Dikarenakan keterbatasan alat kamera sehingga percobaan

dilakukan dengan menggunakan kamera yang berbeda. Tujuan percobaan ini adalah untuk melakukan perbandingan terhadap 2 metode ini. Hasil rekaman metode *background subtraction* menggunakan kamera CMOS 320x240 membutuhkan memori sebesar 260MB sedangkan hasil rekaman metode *frame differencing* menggunakan kamera CCD 640x320 membutuhkan memori sebesar 90MB.

PENGUJIAN DENGAN REKAMAN VIDEO

Langkah-langkah pengujian yang dilakukan sama seperti pengujian dengan kamera akan tetapi yang berbeda adalah aksi atau input yang diberikan berupa rekaman video yang sebelumnya telah direkam. Dan juga tidak harus menggunakan kamera karena tidak digunakan. Atur nilai *threshold* pada 10, 20, 50, 100, 150, 200 sebagai perbandingan. Gambar direkam dengan kamera yang sama yaitu CMOS Logitech dengan resolusi 320x240.

ANALISA. METODE

BACKGROUND SUBTRACTION DAN *FRAME DIFFERENCING*

Inputan yang diberikan berupa 2 file rekaman video yang berbeda. Inputan file rekaman video pertama berisi gerakan orang yang berjalan lewat depan kamera dengan menggunakan pakaian yang mempunyai corak agak gelap. Sedangkan pada file rekaman video kedua berisi gerakan orang yang berjalan lewat depan kamera dengan menggunakan

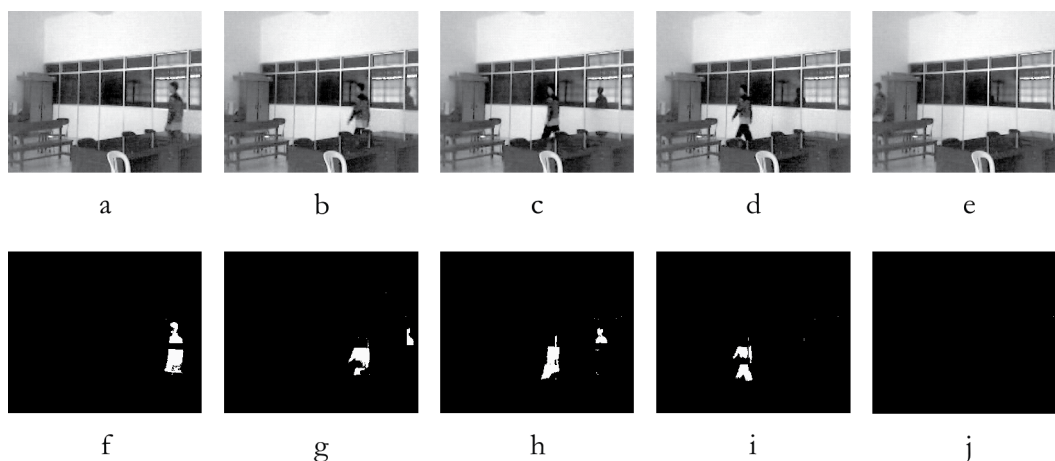
pakaian yang mempunyai corak yang agak berbeda dengan gambar *background*. Apabila kita melihat hasil dari kedua input tersebut akan mempunyai hasil yang berbeda.

Threshold atau ambang batas mempunyai pengaruh pada hasil pengujian dimana *threshold* berhubungan dengan kepekaan kamera dalam mendeteksi objek, semakin rendah nilai *threshold*nya maka semakin tinggi kepekaan kamera, sehingga jika ada perubahan sedikit saja kamera akan mendeteksi. Dikarenakan perubahan cahaya sekitar dianggap gerakan maka tidak akan efektif apabila nilai *threshold* terlalu rendah. Sedangkan semakin tinggi *threshold*nya maka semakin rendah kepekaan kamera sehingga kamera sulit untuk mendeteksi gerakan. Oleh karena itu nilai *threshold* diatur sesuai dengan

keadaan sekitarnya sehingga kepekaan kamera menjadi efektif.

Pada inputan pertama atau video1.avi, objek yang bergerak menggunakan pakaian yang memiliki corak warna agak gelap sehingga ketika objek bergerak melewati gambar *background* yang berwarna gelap juga maka tidak dianggap gerakan karena warna objek dan *background* sama.

Sedangkan pada inputan kedua atau video2.avi, objek yang bergerak menggunakan pakaian yang corak warnanya agak berbeda dengan gambar *background* sehingga mudah untuk mendeteksi gerakan. Oleh karenanya, dapat kita lihat pada tabel diatas bahwa dengan nilai *threshold* yang sama (Th=20, 30, 50) menghasilkan output yang berbeda terhadap dua inputan yang berbeda.



Gambar 6. Urutan video1.avi beserta dengan foregroundnya

Kamera mempunyai resolusi 320x240 sehingga untuk menghitung banyaknya piksel gambar tersebut adalah :

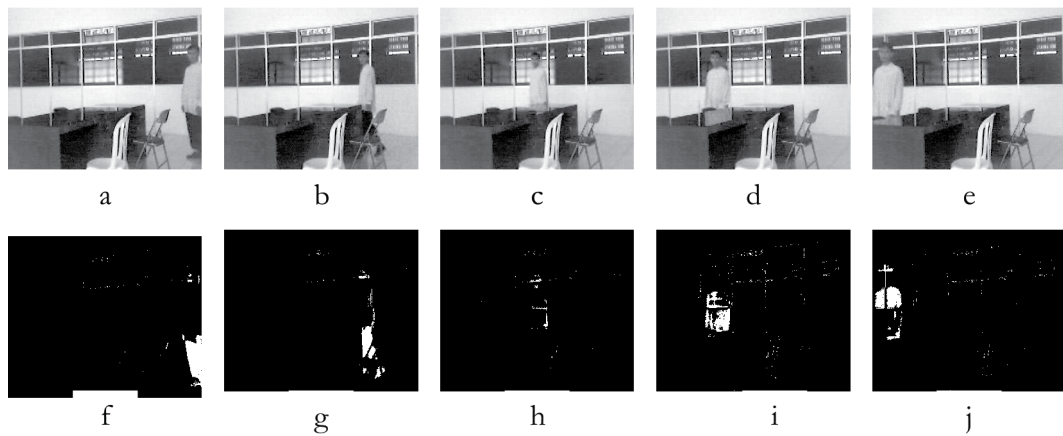
$$\begin{aligned}
 \text{Jml_Pksl_Tot} &= \text{width} \times \text{height} \dots\dots (\text{R.4}) \\
 &= 320 \times 240 \\
 &= 76800 \text{ piksel}
 \end{aligned}$$

Kamera akan mendeteksi gerakan apabila jumlah piksel yang ada lebih banyak dari pada 0,3 % dari jumlah piksel keseluruhan

sehingga kamera akan merekam apabila :

$$\begin{aligned}
 \text{Jml_Pksl_Min} &> 0.003 \times \text{Jml_Pksl_Tot} \dots (\text{R.5}) \\
 \text{Jml_Pksl_Min} &> 0.003 \times 76800 \\
 \text{Jml_Pksl_Min} &> 230,4 \text{ piksel}
 \end{aligned}$$

Apabila jumlah piksel kurang dari *Jml_Pksl_Min* atau 230,4 piksel maka dianggap tidak terjadi gerakan sehingga kamera hanya standby saja dan tidak merekam.



Gambar 5.2. Urutan video2.avi beserta dengan foregroundnya

Input	Output					
	Rekaman Hasil Deteksi Gerak					
	Th=10	Th=20	Th=50	Th=100	Th=150	Th=200
Video1.avi (6,019 KB)	100% (5,334 KB)	95% (2,502 KB)	90% (1,881 KB)	30% (253 KB)	0% (0 KB)	0% (0 KB)
Video2.avi (3,064 KB)	100% (4,358 KB)	100% (3,991 KB)	95% (2,540 KB)	50% (470 KB)	0% (0 KB)	0% (0 KB)

Tabel 1. Hasil Pengujian *Background Subtraction*

Input	Output					
	Rekaman Hasil Deteksi Gerak					
	Th=10	Th=20	Th=50	Th=100	Th=150	Th=200
Video1.avi (6,019 KB)	100% (10,38 KB)	95% (4,53 KB)	90% (2,09 KB)	30% (104 KB)	0% (0 KB)	0% (0 KB)
Video2.avi (3,064 KB)	100% (4,60 KB)	100% (4,60 KB)	90% (1,35 KB)	50% (254 KB)	0% (0 KB)	0% (0 KB)

Tabel 2. Hasil Pengujian *Frame Differencing*



Gambar 8. Perbedaan *Background Subtraction* Dan *Frame Differencing*

KESIMPULAN

1. Pendeteksi gerakan berbasis kamera ini dapat bekerja dengan menggunakan kamera maupun dengan rekaman video untuk mendeteksi gerakan yang terjadi.
2. Kepekaan kamera dalam mendeteksi objek tergantung kepada nilai thresholdnya. Semakin rendah nilai threshold maka semakin tinggi kepekaan kamera dan semakin tinggi nilai threshold maka semakin rendah kepekaan kamera.
3. Pendeteksi gerakan berbasis kamera ini lebih efektif karena hanya merekam ketika terjadi gerakan saja dan tidak merekam sepanjang waktu.
4. Perubahan cahaya dianggap objek yang bergerak.
5. Objek yang memiliki warna yang sama dengan background dianggap sama, sehingga walaupun objek tersebut bergerak tidak dianggap gerakan.
6. Teknik *Background Subtraction* mengurangi antara frame sekarang (F_t) dengan backgroundnya untuk menghasilkan objek yang bergerak.
7. Teknik *Frame Differencing* mengurangi antara frame sekarang (F_t) dengan frame sebelumnya (F_{t-1}) untuk menghasilkan objek yang bergerak.

DAFTAR PUSTAKA

- Gurbuz,Zehra.2006.*Traffic Flow Speed Detection*.Yildiz Technical University Electric-Electronic Faculty Computer Engineering Department.Istambul.
- Hassan-Shafique, Khurram.Spring 2003.*Computer Vision*.
- Heikkila M,Pietikainen M,Heikkila J.*A Texture-based Method for Detecting Moving Objects*.Machine Vision Group Infotech Oulu and Department of Electrical and Information Engineering University of Oulu, Finland.
- Jogiyanto,H M.1993.*Konsep Dasar Pemrograman Bahasa C*. Andi Offset.
- Levin,Golan.2004.*Computer Vision For Artists Dan Designers : Pedagogic Tools And Techniques For Novice Programmers*.Carnegie Mellon University School of Art.
- McDanie,Troy L.*Intel Open Source Computer Vision Library Version 4.0-Beta Installation and Getting Started Guide for Windows*.Center for Cognitive Biquitous Computing Arizona State University
- McIvor,Alan M. *Background Subtraction Techniques*. Reveal Ltd PO Box 128-221, Remuera, Auckland, New Zealand (e-mail : alan.mcivor@reveal.co.nz)
- Munir, Rinaldi.2004.*Pengolahan Citra Digital*.Informatika Bandung.
- Piccardi, Massimo.*Background Subtraction Techniques : A Review*. Computer Vision Research Group (CVRG)University of Technology, Sydney (UTS).
- Reference Manual.2001. *Open Source Computer Vision Library*. USA Intel Corporation.
- Shah, Mubarak.2003.*Review Of Computer Vision Education*. IEEE Transactions On Education, vol 46, No 1,February 2003. www.hci.iastate.edu/575x/doku.php