

## Tracking Arah Gerakan Telunjuk Jari Berbasis Webcam Menggunakan Metode *Optical Flow*

Ubaidillah Umar, Reni Soelistijorini, Haryadi Amran Darwito  
Jurusan Teknik Telekomunikasi - Politeknik Elektronika Negeri Surabaya  
Institut Teknologi Sepuluh Nopember (ITS) Surabaya  
Kampus PENS-ITS, Keputih, Sukolilo, Surabaya.  
Telp : +62+031+5947280; Fax. +62+031+5946011  
u64i@student.eepis-its.edu

### Abstrak

*Deteksi gerakan merupakan subyek penting dalam bidang computer vision yang digunakan oleh banyak sistem pada aplikasi video pengawas (Surveillance System), monitoring traffik, kompresi video, perhitungan kecepatan dan lain-lain.*

*Gerakan adalah suatu pusat perhatian yang digunakan manusia ataupun hewan untuk mengenali suatu obyek dari suatu latar yang tidak teratur. Dalam aplikasi pencitraan, gerakan muncul dari perpindahan tempat antara sistem pendeteksi dan lingkungan yang sedang dilihat, seperti aplikasi robotika, navigasi otomatis, dan analisis lingkungan dinamis.*

*Pada penelitian ini dijelaskan tentang deteksi gerakan dengan melakukan tracking arah gerakan telunjuk jari berbasis webcam dengan menggunakan Metode Optical Flow.*

*Webcam terhubung dengan komputer server untuk melakukan proses capture kondisi ruangan atau pengambilan gambar dan disimpan dalam file .jpg pada periode waktu tertentu. Dengan menggunakan metode Optical Flow, webcam akan mendeteksi dan mentracking jika ada arah gerakan telunjuk jari. Berdasarkan hasil penelitian, dari 120 percobaan yang telah dilakukan 117 diantaranya dinyatakan baik sehingga dapat diambil nilai keberhasilan penelitian sebesar 97,5%.*

**Kata Kunci :** *tracking, metode Optical Flow, webcam, telunjuk jari.*

### 1. Pendahuluan

Seiring perkembangan jaman, teknologi komputer mengalami kemajuan yang sangat pesat. Perkembangan ini mendorong berkembangnya teknologi *computer vision* dan *image processing* yang tidak hanya memungkinkan komputer untuk dapat melakukan proses *filtering* pada citra (gambar) digital, namun juga untuk mengenali citra tersebut. Salah satu contohnya adalah dibidang interaksi manusia dengan komputer

Teknik interaksi berbasis visi komputer menjadi kandidat teknik interaksi yang bersifat alami. Teknik ini tidak membutuhkan kontak langsung pengguna dengan peralatan *input*, melainkan komputer menangkap gerakan pengguna melalui kamera video

dan menginterpretasikannya. *Input* secara *visual* dapat memberikan kemampuan penginderaan pada komputer sebagaimana manusia melakukan penginderaan menggunakan mata. Interaksi antar manusia secara alami tidak membutuhkan *mouse* atau *keyboard*, melainkan diantaranya menggunakan tangan, mata, dan telinga untuk mendapatkan informasi dari lingkungan.

Interaksi manusia dengan komputer merupakan salah satu bidang penelitian yang banyak dilakukan, bahkan hingga saat ini karena memiliki berbagai aplikasi. Dibidang ini, pengambilan data dengan menggunakan kamera dapat digunakan sebagai proses *input* data. Beragam gerakan manusia, baik anggota badan yang melakukan gerakan ataupun *pose* yang dihasilkan, dapat diinterpretasikan terhadap beragam makna. Salah satu alternatif anggota badan manusia yang dapat digerakan untuk menghasilkan beragam *pose* adalah telunjuk jari. Metode *Optical Flow* mampu mengetahui pergerakan objek ketika ada objek yang bergerak (berpindah tempat). Hal ini berbeda dengan *computer vision*, dimana hasil perekaman alat optik (kamera) tidak dapat langsung diterjemahkan, didefinisikan dan dikenali oleh komputer atau sistem *visual* robot. *Computer vision* membutuhkan proses pengolahan (*image processing*) terlebih dahulu, seperti segmentasi, labelisasi, filterisasi, dan deteksi objek

Beberapa penelitian telah dilakukan berkaitan dengan deteksi gerakan manusia dengan implementasi pada beragam aplikasi diantaranya penelitian yang pernah dilakukan tahun 2010 oleh Achmad Zaini<sup>[1]</sup> pada proyek akhirnya dengan judul Implementasi video *surveillance* untuk Sistem keamanan rumah berbasis JMF dan J2ME

CCTV terhubung dengan komputer *server* untuk melakukan proses *capture* kondisi ruangan pada periode tertentu dimana setelah kamera CCTV mendeteksi adanya gerakan komputer *server* akan menghubungi *handphone client* melalui komunikasi serial yang kemudian *handphone client* yang didukung teknologi Java dan GPRS berfungsi untuk mengakses hasil *capture* ruangan. Dalam mendeteksi gerakan yang terjadi pada penelitian ini menggunakan metode yang sederhana yaitu metode *thresholding* hasil deteksi tepi dengan mencari beda antara dua citra yang berurutan pada hasil pencitraan.

## 2. Optical Flow

### 2.1 Gambaran Optical Flow

*Optical Flow* adalah perkiraan gerakan suatu bagian dari sebuah citra berdasarkan turunan intensitas cahayanya pada sebuah sekuen citra. Pada ruang 2D hal ini berarti seberapa jauh suatu piksel citra berpindah di antara dua *frame* citra yang berurutan. Sedangkan pada ruang 3D hal ini berarti seberapa jauh suatu *volume* piksel (*voxel*) berpindah pada dua *volume* yang berurutan. Perhitungan turunan dilakukan berdasarkan perubahan intensitas cahaya pada kedua *frame* citra maupun *volume*. Perubahan intensitas cahaya pada suatu bagian citra dapat disebabkan oleh gerakan yang dilakukan oleh obyek, gerakan sumber cahaya, ataupun perubahan sudut pandang.

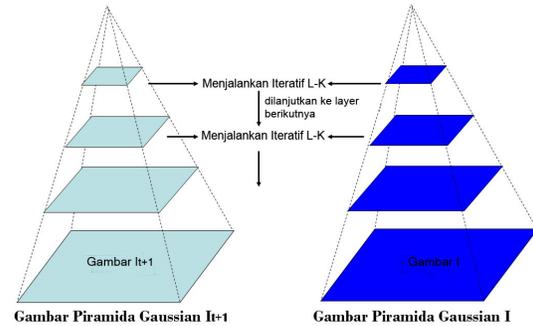
### 2.2 Algoritma Lucas-Kanade

Algoritma Lucas-Kanade, pertama kali diajukan pada tahun 1981, awalnya adalah sebuah usaha untuk mencari teknik registrasi citra yang cepat dengan memanfaatkan *gradient* intensitas spasial. Pada perkembangannya, algoritma ini kemudian menjadi salah satu algoritma *optical flow* yang penting. Berbeda dengan algoritma *Horn-Schunk* yang bekerja berbasis pada keseluruhan citra, algoritma ini bekerja berdasar pada informasi lokal yang diturunkan dari *window* kecil (*patch*) disekeliling titik yang diperhitungkan. Kelemahan digunakan *window* lokal kecil pada algoritma Lucas-Kanade adalah tidak terdeteksinya gerakan-gerakan yang besar karena gerakan-gerakan tersebut jatuh diluar *window*. Permasalahan ini kemudian dapat diatasi dengan mengimplementasikan penyelesaian dengan prinsip piramida, yaitu *pyramidal Lucas-Kanade*. Prinsip ini merupakan penyelesaian berdasar iterasi dari *level* detail citra paling rendah hingga *level* detail citra paling tinggi.

### 2.3 Pyramidal Lucas-Kanade

Penyelesaian algoritma Lucas-Kanade dengan pendekatan piramida, atau disebut *Pyramidal Lucas-Kanade* diajukan pertama kali oleh Jean-Yves Bouguet (Bouguet, 2000). Pendekatan ini menggunakan prinsip piramida, yaitu bekerja dari detail citra paling rendah hingga detail citra paling tinggi (Gambar 1). Tujuannya adalah agar gerakan yang “besar” dapat diperhitungkan. Sementara asumsi yang digunakan pada algoritma Lucas-Kanade adalah gerakan yang “kecil” dan koheren, sehingga tidak dapat menangkap gerakan yang “besar”. Solusi untuk dapat menangkap gerakan yang “besar” pada algoritma Lucas-Kanade adalah dengan menggunakan *window* yang besar. Tetapi penggunaan *window* yang besar sering kali membuat gerakan yang ditangkap adalah gerakan yang tidak koheren. Algoritma *Pyramidal Lucas-Kanade* menyelesaikan permasalahan tersebut tanpa menghilangkan asumsi gerakan yang koheren.

### Estimasi detail citra paling kecil ke citra paling besar Optical Flow

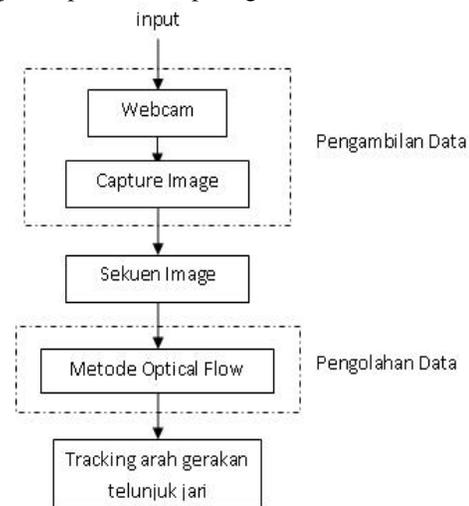


Gambar 1. Pyramidal optical flow

Algoritma *Pyramidal Lucas-Kanade* pertama bekerja pada layer piramida paling tinggi. Kemudian hasilnya digunakan sebagai titik awal untuk bekerja pada layer dibawahnya. Hal ini berlanjut hingga mencapai *level* paling rendah.

## 3. Metode Penelitian

Langkah-langkah yang dilakukan dalam penelitian ini secara umum dapat dibagi menjadi dua kegiatan utama, yaitu pengambilan data, dan pengolahan data. Pengambilan data adalah pekerjaan yang dilakukan pertama kali. Hasilnya adalah sekuen *image* gerak tangan. Sekuen *image* tersebut kemudian menjadi *input* bagi proses pengolahan data. Alur pekerjaan diperlihatkan pada gambar 2.



Gambar 2. Blok Diagram Perancangan Program

### 3.1 Pengambilan data

Untuk tahap pengambilan data webcam akan menerima perintah dari program untuk melakukan *capture* obyek secara kontinyu kemudian disimpan ke dalam *server*. Gambar atau *image* yang disimpan haruslah memenuhi persyaratan yang telah ditentukan. Adapun gambar disimpan dalam *type* BMP. Dan

gambar yang disimpan tersebut adalah gambar berwarna (*true color*) karena metode yang digunakan untuk mencari daerah warna merah adalah metode *color detection* dimana metode ini bertujuan untuk mendeteksi lokasi warna dari gambar dengan membuang seluruh bagian dari gambar yang tidak memiliki warna merah.

Pada saat pengambilan data, telunjuk jari didepan kamera dengan (Gambar 3). Gerakan yang dibuat meliputi gerakan ke kiri, gerakan ke kanan, gerakan ke atas dan gerakan ke bawah. Pengujian dilakukan pada keadaan intensitas cahaya dan kondisi *obyek* yang berbeda – beda.



Gambar 3. Situasi Pengambilan Data

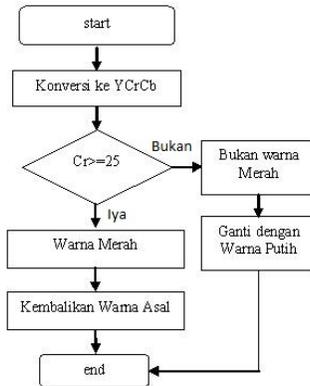
### 3.2 Pengolahan Gambar

Dalam tahap ini dibuat sebuah program untuk melakukan *tracking* terhadap telunjuk jari tangan. Metode *tracking* yang digunakan berbasis *optical flow*. Proses *tracking* diawali dengan identifikasi warna merah yang kemudian deteksi masing-masing tepi obyek dan deteksi arah pergerakan telunjuk jari.

#### 3.2.1 Deteksi Warna Merah

Dalam tahap ini dibuat sebuah program untuk melakukan pendeteksian warna merah dengan menggunakan metode *color detection* dimana metode ini bertujuan untuk mengetahui piksel mana yang termasuk piksel warna merah dan mana piksel yang bukan piksel warna merah. Tahap ini adalah tahap dimana kita harus mencari daerah yang memiliki warna merah. Untuk mendapatkan daerah ini kita harus melakukan proses untuk mencari daerah warna merah dan memproses sampel untuk warna merah.

Untuk memisahkan warna merah dari warna lainnya, dilakukan pengkondisian/*thresholding* pada nilai  $C_r$  dengan nilai *threshold* sebesar 25, proses ini digambarkan seperti pada gambar 4. Jika nilai  $C_r$  lebih besar daripada 25, maka warna tersebut merupakan warna merah. Jika tidak, maka warna tersebut bukan warna merah dan akan diganti dengan warna putih. Operasi tersebut dilakukan terhadap setiap piksel pada gambar.



Gambar 4. Flowchart fungsi deteksi warna merah

Sebenarnya tetapan ini tidak harus mutlak bernilai 25 melainkan ketetapan ini diambil dari hasil percobaan dengan sampel yang telah kita buat sebelumnya. Dan angka 25 tersebut adalah nilai yang dianggap tepat untuk mewakili batas antara warna merah dan bukan. Gambar 5 menggambarkan hasil proses deteksi warna.

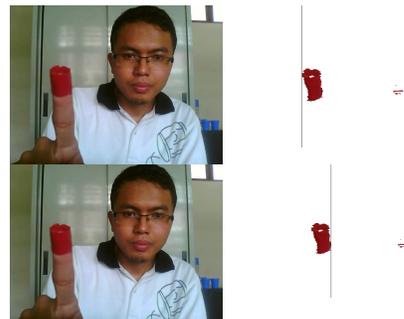


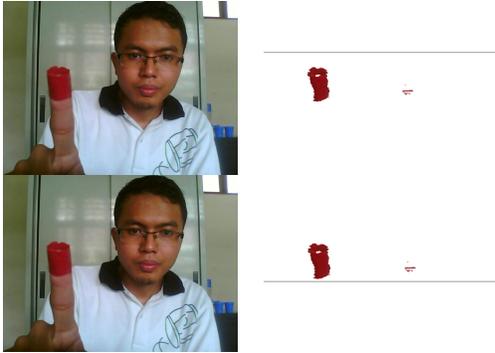
Gambar 5 Warna merah dan bukan warna merah

#### 3.2.2 Deteksi sisi lokasi obyek

Pendeteksian sisi lokasi obyek merupakan tahapan dalam mencari nilai masing-masing sisi obyek seperti sisi kiri, atas, kanan dan bawah obyek. Obyek yang digunakan sebagai masukan (*input*) adalah gambar hasil deteksi warna merah, sehingga apabila proses deteksi warna merah gagal, maka dapat dipastikan bahwa proses deteksi tepi obyek ini akan gagal pula.

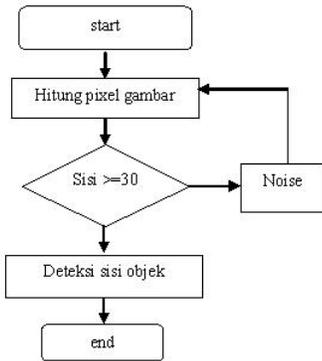
Proses deteksi lokasi tepi obyek dilakukan dengan menelusuri nilai pada masing-masing sisi obyek, sehingga akan didapat lokasi dari batas kiri, atas, dan kanan obyek seperti pada gambar 6.





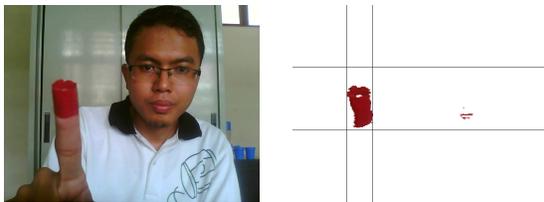
**Gambar 6.** Pencarian nilai sisi kiri, kanan, atas dan bawah obyek

Untuk membedakan antara noise dan obyek, dilakukan pengkondisian/*thresholding* pada nilai masing-masing sisi obyek sebesar  $\geq 30$ . Jika nilai sisi obyek lebih besar daripada 30, maka warna tersebut merupakan obyek. Jika tidak, maka warna tersebut bukan obyek atau noise seperti yang diperlihatkan pada alur gambar 7.



**Gambar 7** Memisahkan antara obyek dan noise

Setelah sisi obyek sebelah kanan, kiri, atas dan bawah obyek dapat terdeteksi maka akan diperlihatkan hasil deteksi lokasi obyek seperti pada gambar 8.



**Gambar 8** Deteksi sisi lokasi obyek

#### 4. Hasil dan Pembahasan

##### 4.1 Pengambilan data

*Output* dari proses pengambilan data adalah sekuen *image* gerakan telunjuk jari. Gerakan yang dilakukan tangan meliputi gerakan ke atas, ke bawah, ke kiri dan ke kanan. Gerakan dilakukan dengan telunjuk jari menghadap kamera (*frontal view*).

Gambar 9 memperlihatkan beberapa frame dari sekuen *image* tersebut.



**Gambar 9.** Beberapa Frame Sekuen *Image* Gerak Tangan

##### 4.2 Deteksi warna merah

Deteksi warna merah bertujuan untuk mengetahui piksel mana yang termasuk piksel warna merah dan mana piksel yang bukan piksel warna merah

**Tabel 1.** Deteksi Warna Merah

Perco baan	Berhasil				Tidak Berhasil				JP
	Kr	Kn	At	Bw	Kr	Kn	At	Bw	
1	9	9	10	10	1	1	0	0	40
2	9	10	10	10	1	0	0	0	40
3	8	9	8	10	2	1	2	0	40
	<b>Jumlah</b>								<b>120</b>

Keterangan :  
Kr=Kiri, Kn=Kanan, At=Atas, Bw=Bawah dan  
JP=Jumlah Percobaan

Tabel 1. menunjukkan hasil dari deteksi warna untuk beberapa percobaan pada waktu yang berbeda-beda. Deteksi warna merah dikatakan baik apabila dapat dibedakannya antara piksel yang termasuk warna merah dengan piksel yang bukan piksel warna merah sedangkan hasil dikatakan tidak baik apabila tidak dapat dibedakannya antara piksel yang termasuk warna merah dengan piksel yang menghasilkan warna merah yang terbuang atau bahkan tidak terdeteksi sama sekali.

Berdasarkan hasil percobaan terlihat dari 120 percobaan yang telah dilakukan 113 diantaranya dinyatakan baik dan 8 dinyatakan tidak baik sehingga dapat diambil nilai keberhasilan sebesar 93,3%. Hasil deteksi yang tidak baik diakibatkan oleh pencahayaan pada waktu pengambilan gambar yang dapat mempengaruhi hasil deteksi warna merah.

### 4.3 Deteksi lokasi sisi obyek

Pendeteksian lokasi sisi obyek merupakan tahapan dalam mencari nilai masing-masing sisi obyek seperti sisi kiri, atas, kanan dan bawah obyek

**Tabel 2.** Deteksi sisi lokasi obyek

Percobaan	Berhasil				Tidak Berhasil				JP
	Kr	Kn	At	Bw	Kr	Kn	At	Bw	
1	10	10	10	10	0	0	0	0	40
2	10	10	10	10	0	0	0	0	40
3	9	9	9	10	1	1	1	0	40
<b>Jumlah</b>									<b>120</b>

Keterangan :

Kr=Kiri, Kn=Kanan, At=Atas, Bw=Bawah dan JP=Jumlah Percobaan

Dari tabel 2. menunjukkan hasil dari deteksi lokasi sisi obyek. Deteksi lokasi obyek dikatakan baik apabila semua nilai sisi kiri, kanan, atas dan bawah obyek dapat dicari nilainya sedangkan deteksi lokasi obyek dikatakan tidak baik apabila ada salah satu atau bahkan nilai dari masing-masing sisi obyek tidak terdeteksi.

Berdasarkan hasil percobaan terlihat dari 120 percobaan yang telah dilakukan 117 diantaranya dinyatakan baik sehingga dapat diambil nilai keberhasilan sebesar 97,5% sedangkan 3 diantaranya dinyatakan tidak baik, hal ini dikarenakan deteksi warna merah pada waktu pengambilan gambar mempengaruhi hasil deteksi lokasi dimana ketika piksel warna merah yang terdeteksi kurang dari 30 maka warna merah ini dianggap sebagai noise sehingga tidak terdeteksi sebagai obyek.

### 4.4 Deteksi gerakan menjauh dari kamera

Pada pendeteksian lokasi sisi obyek ini dilakukan pendeteksian terhadap sisi kiri, kanan, atas dan bawah obyek dengan gerakan yang dilakukan semakin menjauhi kamera.

**Tabel 3.** Deteksi sisi lokasi obyek gerakan menjauh

Percobaan	Arah Gerakan	Nilai	Hasil deteksi/jumlah	Jumlah Percobaan
1	Kiri	47	Baik	5
		45	Baik	
		41	Baik	
		31	Baik	
		<31	Jelek	
2	Kanan	47	Baik	5
		33	Baik	
		35	Baik	
		31	Baik	
		<31	Jelek	

3	Atas	79	Baik	5
		71	Baik	
		55	Baik	
		50	Baik	
		<50	Baik	
4	Bawah	110	Baik	5
		108	Baik	
		103	Baik	
		102	Baik	
		<102	Baik	
<b>Jumlah</b>				<b>20</b>

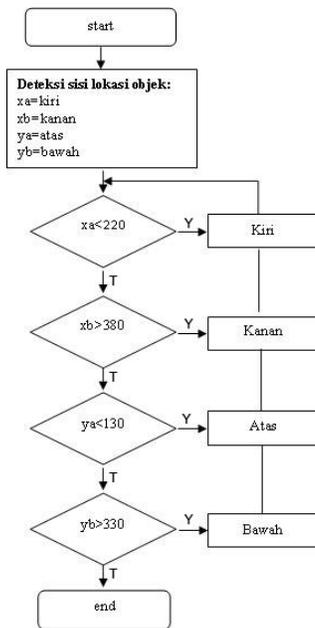
Berdasarkan hasil percobaan deteksi sisi lokasi obyek gerakan menjauh pada tabel 3 terlihat dari 20 percobaan yang telah dilakukan, hasil percobaan arah gerakan yang dilakukan 5 kali pada setiap percobaan mempunyai nilai yang berbeda-beda, hal ini dipengaruhi oleh gerakan telunjuk jari dimana semakin gerakan menjauhi kamera maka nilai yang dihasilkan akan semakin kecil. Pada percobaan ini gerakan 1 ke gerakan 5 pada masing-masing percobaan posisi telunjuk jari semakin menjauhi kamera.

Pada Percobaan gerakan ke kiri, dari percobaan 1 sampai percobaan 4 hasil deteksi baik tetapi nilai dibawah 31 hasil deteksi tidak baik, hal ini dikarenakan nilai dibawah 31 merupakan nilai yang dianggap sebagai noise demikian juga untuk percobaan 2 gerakan ke kanan dimana ketika nilai kurang dari 31 hasil deteksi tidak baik karena dianggap sebagai noise. Kemudian untuk percobaan 3 gerakan ke atas nilai dianggap noise ketika nilai kurang dari 50 dan untuk percobaan 4 gerakan ke bawah nilai dianggap noise ketika nilai kurang dari 102.

### 4.4 Deteksi Arah Gerakan Telunjuk Jari

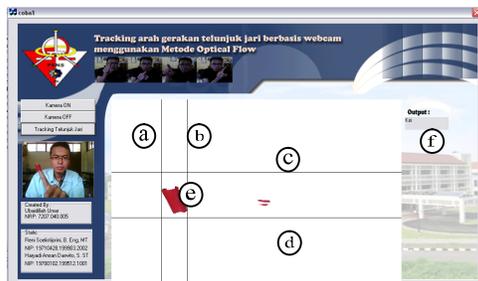
Tahap ini adalah tahap akhir dimana penentuan arah pergerakan telunjuk jari dilakukan. Setelah lokasi masing-masing sisi obyek terdeteksi dengan baik maka dapat dilakukan pendeteksian arah pergerakan dari masing-masing sisi obyek ini. Pendeteksian arah pergerakan telunjuk jari ini bertujuan untuk melakukan deteksi arah pergerakan dari telunjuk jari seperti arah ke kanan, ke kiri, atas dan bawah.

Untuk menentukan arah gerakan, dilakukan pengkondisian/*thresholding* pada masing-masing sisi obyek. Gambar 10 menampilkan alur pengkondisian/*thresholding* pada masing-masing sisi obyek, untuk sisi kiri obyek sebesar 220, sisi kanan obyek sebesar 380, sisi atas obyek sebesar 130 dan sisi bawah obyek sebesar 330. Jika nilai sisi kiri obyek pada saat deteksi sisi lokasi obyek lebih kecil 220 maka telunjuk jari bergerak ke kiri, jika sisi kanan obyek pada saat deteksi sisi lokasi obyek lebih besar dari 380 maka telunjuk jari bergerak ke kanan, jika nilai sisi obyek atas pada saat deteksi sisi lokasi obyek lebih kecil dari 130 maka telunjuk jari bergerak ke atas dan jika deteksi lokasi obyek sebelah bawah lebih besar dari 330 maka telunjuk jari bergerak ke bawah.



**Gambar 10.** Flowchart deteksi arah gerakan telunjuk jari

Berdasarkan hasil percobaan, keberhasilan pendeteksian arah gerakan sangat dipengaruhi oleh hasil deteksi lokasi obyek. Bila deteksi obyek berhasil maka dapat dipastikan 100% deteksi arah gerakan akan menunjukkan arah gerakan yang benar. Sehingga dari 120 percobaan yang telah dilakukan didapatkan nilai keberhasilan pendeteksian arah gerakan sebesar 97,5%.



**Gambar 11.** Hasil deteksi arah gerakan telunjuk jari, (a) sisi kiri obyek, (b) sisi kanan obyek, (c) sisi atas obyek, (d) sisi bawah obyek, (e) obyek yang bergerak ke kiri, dan (f) output.

## 5. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat dibuat beberapa kesimpulan sebagai berikut :

1. Untuk membedakan antara noise dan obyek, dilakukan pengkondisian/*thresholding* berupa nilai minimal untuk dikatakan sebagai obyek.
2. Obyek dapat terdeteksi ketika nilai piksel warna merah yang terdeteksi kurang dari 31 untuk arah ke kanan dan ke kiri, kemudian 50 untuk arah ke atas dan 102 untuk arah ke bawah.
3. Dari 120 percobaan yang telah dilakukan 117 diantaranya dinyatakan baik sehingga dapat diambil nilai keberhasilan deteksi gerakan sebesar 97,5%.
4. Kelanjutan dari penelitian ini akan bisa dikembangkan untuk *tracking* obyek bergerak seperti Aplikasi *realtime object tracking*.

## Daftar Pustaka :

- [1] Zaini, A., "Implementasi *vide surveillance* untuk keamanan rumah berbasis *JMF* dan *J2ME*", Proyek Akhir PENS-ITS, 2009.
- [2] Mahtarami, .A, " *Tracking gerak tangan Berbasis pyramidal lucas-kanade*", URL: <http://digilib.its.ac.id/public/ITS-Undergraduate10509-Paper.pdf>, 2010.
- [3] Ramadijanti, N., " *Color detection* ", URL : [http://lecturer.eepisits.edu/~nana/index\\_files/materi/Prak\\_Citra/Prak9ColorDetection.pdf](http://lecturer.eepisits.edu/~nana/index_files/materi/Prak_Citra/Prak9ColorDetection.pdf), 2010.
- [4] Amalia, Rizky., " *Menilai kepribadian seseorang melalui Wajah*", Proyek Akhir PENS-ITS, 2006.
- [5] Baron, J.L., dan Thacker, N.A., " *Tutorial: Computing 2D and 3D optical flow*", Tina Memo Internal No. 2004-12, Manchester, UK. 1995.
- [6] Bouguet, J.Y., " *Pyramidal Implementation of The Lucas-Kanade Feature Tracker*", Intel. 2000.
- [7] Lu, N. , G., et. al, " *An Improved motion Detection Method for Real-Time Surveillance*" International Journal of Computer Science, 35:1, IJCS\_35\_1\_16, 2008.
- [9] Catalano, G., et. al., " *optical flow*", URL : <http://www.cvmt.dk/education/teaching/f09/VGIS8/AIP/opticalFlow.pdf>, 2009.
- [10] Aleix M. Martinez " *optical flow*", URL: <http://www.ece.osu.edu/~aleix/OpticalFlow.pdf>, 2003.