

**VIRTUAL MOUSE MENGGUNAKAN ALGORITMA HAARCASCADE
DAN HOUGH TRANSFORM SEBAGAI MEDIA INTERAKSI MANUSIA
DENGAN KOMPUTER BAGI PENYANDANG DISABILITAS**

Oleh:

Fandy Setyo Utomo¹, Moch. Hari Purwidianoro²

¹Dosen Program Studi Teknik Informatika, STMIK AMIKOM Purwokerto

²Dosen Amik Cipta Darma Surakarta

ABSTRAK

Komputer adalah salah satu media untuk pencarian informasi. Dalam mengoperasikan komputer seseorang harus dapat menggerakkan pointer pada layar komputer dengan mouse. Hal ini menjadi sulit untuk dilakukan bagi penderita disabilitas gerak tangan. Penelitian yang dilakukan bertujuan untuk membuat aplikasi virtual mouse yang dapat digunakan sebagai media interaksi antara penyandang disabilitas dengan komputer. Proses penjejakan wajah untuk mengendalikan pointer menggunakan algoritma Haarcascade Classifier, sedangkan untuk operasi click event mouse menggunakan deteksi kedipan mata dengan algoritma Hough Transform. Penelitian dilakukan dalam beberapa tahap yaitu melakukan analisis kebutuhan fungsional, membuat desain aplikasi, implementasi, dan pengujian aplikasi. Pengujian dilakukan terhadap kecepatan aplikasi dalam mendeteksi wajah berdasarkan sudut kemiringan wajah tertentu. Selain itu, pengujian dilakukan pula terhadap kecepatan aplikasi dalam mendeteksi iris mata berdasarkan nilai threshold tertentu. Berdasarkan hasil pengujian aplikasi dapat disimpulkan bahwa nilai pergeseran memiliki toleransi 10° hingga 25° untuk pendeteksian wajah, sedangkan nilai threshold yang baik untuk pendeteksian iris mata berkisar antara 90 sampai dengan 100.

Kata Kunci: Virtual Mouse, Haarcascade, Hough Transform, Blink detection, Face Tracking

A. PENDAHULUAN

Dalam Undang-undang Republik Indonesia Nomor 19, Tahun 2011, mengenai Pengesahan *Convention On The Rights Of Persons With Disabilities* (Konvensi Mengenai Hak-Hak Penyandang Disabilitas), menjelaskan bahwa penyandang disabilitas adalah mereka yang mempunyai kelainan fisik, mental dan intelektual, atau sensorik secara permanen yang dalam interaksinya dengan berbagai hambatan dapat merintangi partisipasi mereka dalam masyarakat secara penuh dan efektif berdasarkan pada asas kesetaraan dengan orang lain. Akibat kelainan tersebut, melalui penelitian yang dilakukan oleh Irwanto *et al.* (2010), mengenai analisis situasi penyandang disabilitas di Indonesia, menjelaskan bahwa penyandang disabilitas sering dianggap sebagai warga masyarakat yang tidak produktif, tidak mampu menjalankan tugas dan tanggung jawabnya sehingga hak-haknya diabaikan.

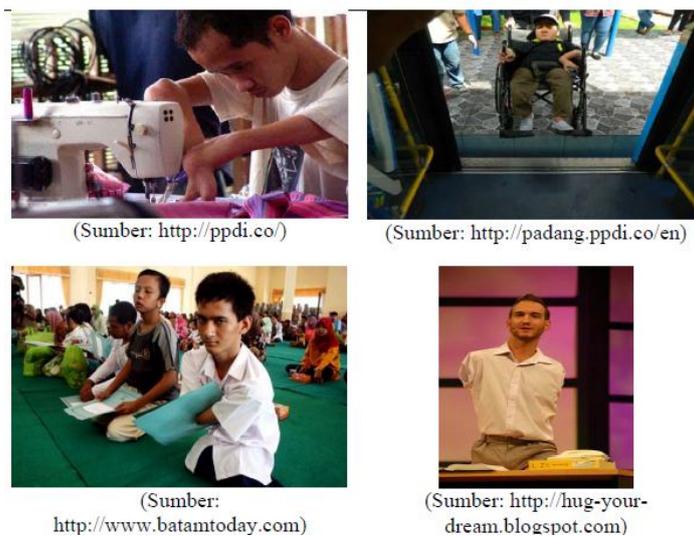
Penelitian yang dilakukan oleh Irwanto *et al.* (2010), menjelaskan bahwa Badan Pusat Statistik (BPS) membagi penyebab disabilitas menjadi 3 jenis, yaitu akibat kelainan kongenital atau cacat bawaan, kecelakaan, dan penyakit. Berdasarkan survey yang dilakukan oleh BPS di tahun 2008, pada program Pendataan Program Perlindungan Sosial (PPLS) untuk mengetahui informasi jenis kecacatan pada rumah tangga miskin, diperoleh data bahwa cacat anggota gerak memiliki jumlah terbesar di tiap kategori rumah tangga miskin. Tabel 1 menjelaskan hasil survey BPS secara rinci :

Tabel 1. Jumlah Penyandang Disabilitas Berdasarkan Rumah Tangga Miskin

Jenis Kecacatan	Status Kemiskinan			
	Sangat Miskin	Miskin	Hampir Miskin	Total
Tuna Netra/ Buta	46,146	82,242	78,699	207,087
Tuna Rungu/ Tuli	24,746	54,747	66,468	145,961
Tuna Wicara/ Bisu	20,678	33,822	27,054	81,554
Tuna Rungu & Wicara	7,616	13,700	12,703	34,019
Cacat Anggota Gerak	51,857	106,042	116,981	274,880
Lumpuh	19,985	42,167	45,755	107,907
Cacat Mental	39,439	76,280	66,571	182,290
Total Cacat	210,467	409,000	414,231	1,033,698

(Sumber: BPS – PPLS, tahun 2008)

Berdasarkan data statistik pada Tabel 1, maka fokus penelitian kami untuk membantu penyandang disabilitas dengan cacat fisik berjenis cacat anggota gerak agar mampu berinteraksi dengan komputer untuk kepentingan berkomunikasi secara sosial melalui jejaring sosial, media hiburan, dan menggali informasi yang ada di internet, sehingga para penyandang disabilitas mampu memenuhi hasratnya sebagai makhluk sosial.



Gambar 1. Penyandang Disabilitas dengan Cacat Anggota Gerak

Manusia sebagai makhluk sosial memiliki rasa saling ketergantungan antara yang satu dengan yang lain, akibatnya mereka pun saling berinteraksi. Manusia ingin mengetahui lingkungan sekitarnya, bahkan ingin mengetahui apa yang terjadi dalam dirinya. Rasa ingin tahu ini memaksa manusia perlu berkomunikasi. Komunikasi adalah suatu kebutuhan yang sangat fundamental bagi seseorang dalam hidup bermasyarakat. Di era teknologi informasi, komputer merupakan teknologi yang dapat digunakan antara lain untuk berinteraksi dan berkomunikasi, sebagai media hiburan, media belajar, serta sumber informasi.

Salah satu permasalahan yang dihadapi oleh penyandang disabilitas (seperti pada Gambar 1) dalam memanfaatkan komputer sebagai sarana untuk berinteraksi, berkomunikasi, belajar, hiburan, dan menggali informasi, yaitu mereka kesulitan mengendalikan kursor dan melakukan operasi *click event* menggunakan *mouse*. *Mouse* merupakan kategori unit masukan (*input device*) pada komputer. Tanpa perangkat *mouse*, manusia kesulitan untuk berinteraksi dengan komputer. Berdasarkan permasalahan tersebut, maka fokus penelitian

kami untuk membuat aplikasi *virtual mouse* menggunakan anggota tubuh penyandang disabilitas sebagai pengganti *mouse*.

Penelitian yang berkaitan dengan interaksi penyandang disabilitas dengan komputer telah dilakukan oleh beberapa peneliti terdahulu. Penelitian yang bertujuan membuat aplikasi untuk mengendalikan pointer dan melakukan operasi *click event* mouse telah dilakukan sebelumnya oleh Agustian *et al.* (2012) menggunakan algoritma *Haarcascade* untuk mengendalikan pointer dengan penjejak wajah dan algoritma SURF untuk operasi *click event* dengan deteksi kedipan mata. Penelitian berikutnya dilakukan oleh Królak dan Strumillo (2012), menggunakan algoritma *Haarcascade* untuk penjejak wajah dan algoritma *Template Matching* untuk deteksi kedipan mata. Sedikit berbeda dengan penelitian sebelumnya, penelitian yang dilakukan oleh Kocejko *et al.* (2009), Gupta *et al.* (2012), Syarif *et al.* (2013), berfokus pada pengendalian pointer dan operasi *click event* menggunakan pupil mata. Penelitian yang dilakukan oleh Kocejko *et al.* (2009), menggunakan *Longest Segment Algorithm* (LSD) dan *Modified Longest Segment Algorithm* (MLSD), Penelitian Gupta *et al.* (2012) menggunakan algoritma SSR filter (*Six Segmented Rectangular*) dan SVM (*Support Vector Machine*), sedangkan penelitian Syarif *et al.* (2013) menggunakan algoritma *Lucas Kanade*.

Perbedaan penelitian kami dengan penelitian terdahulu, yaitu pada penelitian ini algoritma untuk penjejak wajah (*face tracking*) menggunakan algoritma *Haarcascade*, sedangkan untuk mendeteksi kedipan mata (*blink detection*), melalui iris mata, menggunakan algoritma *Lucas Kanade*.

B. METODE PENELITIAN

Penelitian ini dapat digolongkan sebagai penelitian Terapan. Penelitian terapan adalah penelitian yang dikerjakan dengan maksud untuk menerapkan, menguji, dan mengevaluasi kemampuan suatu teori yang diterapkan dalam pemecahan permasalahan praktis.

1. Alat Penelitian

Pada penelitian ini menggunakan beberapa perangkat keras. Tabel 2 menjelaskan spesifikasi perangkat keras tersebut.

Tabel 2. Spesifikasi Perangkat Keras

No	Spesifikasi	Jumlah (Unit)
1.	PC (<i>Personal Computer</i>) : Processor Core i3 RAM DDR3 2 GB Harddisk SATA2 500 GB	1 Unit
2.	Genius FaceCam 320, 640x480 pixel	1 Unit
3.	Lux Meter	1 Unit

Perangkat Lux Meter digunakan untuk mengukur tingkat intensitas cahaya, sedangkan perangkat *Genius FaceCam* digunakan sebagai perangkat *input device* pada sistem aplikasi.

Perangkat lunak yang digunakan dalam penelitian ini, yaitu Microsoft Visual Studio 2012 dan Emgu CV v2.2. Emgu CV v2.2 digunakan sebagai *library Computer Vision*, algoritma *Haarcascade* dan *Hough Transform* diambil dari *library* tersebut. Microsoft Visual Studio 2012 digunakan sebagai *tools developer* untuk membangun aplikasi *virtual mouse*.

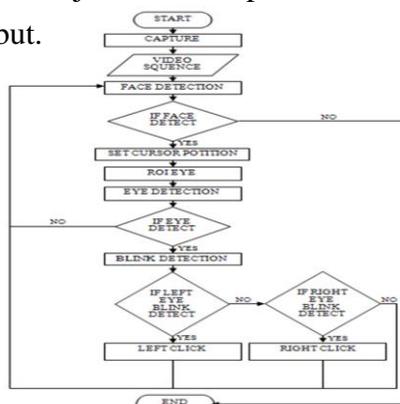
2. Tahapan Penelitian

a. Analisis Kebutuhan Fungsional

Pada tahapan ini dilakukan analisis kebutuhan fungsional yang membahas fitur aplikasi yang dibutuhkan oleh penyandang disabilitas untuk berinteraksi dengan komputer. Dokumentasi kebutuhan fungsional menggunakan diagram *use case*.

b. Desain Aplikasi

Terdapat beberapa tahapan dalam membuat aplikasi media interaksi dengan deteksi wajah dan kedip mata. Gambar 2 menjelaskan mengenai tahapan tersebut.



Gambar 2. Alur Kerja Aplikasi *Virtual Mouse*

Berdasarkan Gambar 2, dapat dijelaskan bahwa aplikasi menggunakan input berupa *video realtime* yang diambil dari *webcam*. Input video tersebut digunakan untuk mendeteksi wajah dengan algoritma *haarcascade classifier*, kemudian deteksi wajah digunakan sebagai navigasi untuk menggerakkan kursor pada layar monitor. Setelah wajah terdeteksi, dilakukan proses pencarian area mata. Setelah area mata ditemukan, langkah berikutnya yaitu mendeteksi mata dengan menggunakan algoritma *haarcascade classifier*. Proses selanjutnya yaitu *thresholding* atau binerisasi, dimana proses binerisasi dilakukan hanya pada area mata untuk mencari bentuk lingkaran iris mata. Selanjutnya dengan menggunakan metode morfologi berupa *erosi*, dilakukan pembuangan bagian yang tidak diperlukan, dan proses *dilasi* untuk menambah bagian yang diperlukan. Setelah dilakukan proses *thresholding* dan morfologi, langkah selanjutnya adalah deteksi kedip mata. Pada proses deteksi kedip mata, algoritma *Hough Transform* digunakan untuk mencari bentuk lingkaran iris mata. Setelah iris mata ditemukan, maka hasilnya diberi nilai integer 1 dimana mata tidak berkedip atau terbuka, dan 0 untuk mata berkedip.

1) Face Detection

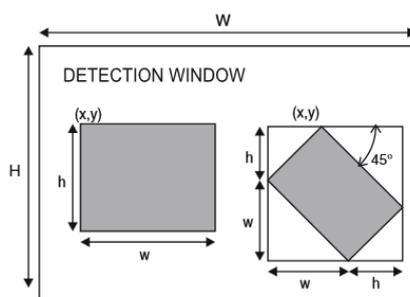
Face Detection (deteksi wajah) merupakan proses untuk mengidentifikasi keberadaan wajah dalam citra digital, apabila ditemukan citra wajah, maka sistem memberikan nilai kembalian berupa lokasi dan ukuran setiap citra wajah (Yang, 2009; Huang dan Hu, 2009; Bevilacqua *et al.*, 2008).

Pada penelitian ini deteksi wajah menggunakan algoritma *haarcascade*. Fitur *Haar-like* di dalam layar berukuran $W \times H$ Pixel dapat dirumuskan dengan persamaan berikut (Lienhart *et al.*, 2002; Kasiński dan Schmidt, 2007; Kasiński dan Schmidt, 2010) :

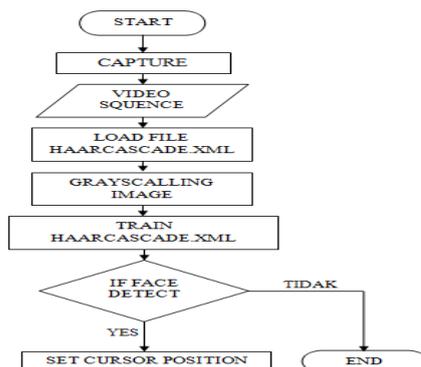
$$\text{feature} = \sum_{i=1}^N \omega_i \cdot \text{RecSum}(r_i)$$

Gambar 3. Kalkulasi *Haar-like Feature*

Dimana w_i adalah faktor bobot, $\text{RecSum}(r_i)$ merupakan penjumlahan nilai intensitas persegi secara tegak lurus atau berotasi yang terdapat di dalam *window*. Persegi didefinisikan memiliki 5 parameter, yaitu x , y , w , h , dan ϕ , dimana x dan y merupakan titik koordinat kiri atas persegi, w dan h merupakan dimensi dari persegi, dan ϕ merupakan sudut rotasi persegi. Gambar 4 menjelaskan deskripsi persegi tersebut.



Gambar 4. Persegi Tegak Lurus dan Persegi Berotasi 45° di Dalam Window (Sumber : Kasiński dan Schmidt, 2010)

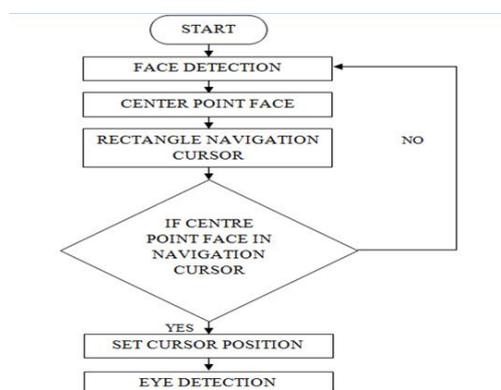


Gambar 5. Penerapan *Face Detection* pada Aplikasi

Gambar 5 menjelaskan pada aplikasi, proses deteksi wajah menggunakan *haarcascade classifier* yang tersimpan dalam bentuk file XML. Langkah pertama, melakukan perekaman *video Sequence*, selanjutnya dilakukan proses *grayscale* untuk proses deteksi wajah. Kemudian aplikasi mengakses *file haarcascade.xml* untuk melakukan *training file haarcascade.xml* untuk menentukan fitur *haar* pada video input dan mempelajari fitur *haar*, sehingga dapat menentukan gambar wajah dan bukan wajah.

2) Set Cursor Position

Setelah melakukan proses deteksi wajah, langkah berikutnya adalah melakukan penyetelan pergerakan *mouse pointer* menggunakan hasil dari deteksi wajah. Pada proses ini langkah pertama yang dilakukan menentukan *center point face*, setelah proses penentuan *center point face*, langkah selanjutnya membuat *rectangle* untuk navigasi pointer, jika *center point face* berada di dalam *rectangle* navigasi, maka *set cursor position* aktif. Gambar 6 menjelaskan mengenai proses *set cursor position*.



Gambar 6. *Setting Cursor Position*

3) Eye and Blink Detection

Proses deteksi mata menggunakan algoritma *haarcascade classifier*, namun untuk menghasilkan keakuratan dalam deteksi mata, terlebih dahulu dilakukan ekstraksi ROI (*Region Of Interest*) dari mata. Proses ekstraksi ROI mata menggunakan *facial grid*, seperti yang dilakukan oleh Poornima *et al.* (2014). Setelah penentuan ROI mata, langkah selanjutnya aplikasi melakukan training data menggunakan *file haarcascade.xml* untuk mengidentifikasi mata. Setelah mata terdeteksi, proses berikutnya adalah *blink detection*. *Blink detection* menggunakan algoritma *hough transform* untuk mencari bentuk lingkaran iris mata. *Hough Transform* dapat digunakan untuk menentukan parameter lingkaran ketika sejumlah poin yang jatuh pada perimeter diketahui (Rhody, 2004). Lingkaran dengan radius R dan pusat (a, b) dapat

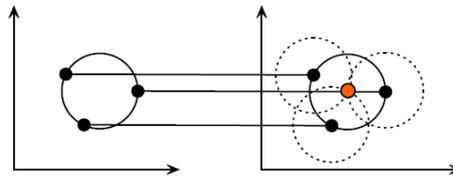
$$x = a + R \cos(\theta)$$

$$y = b + R \sin(\theta)$$

digambarkan dengan persamaan parametrik seperti pada Gambar 7 berikut:

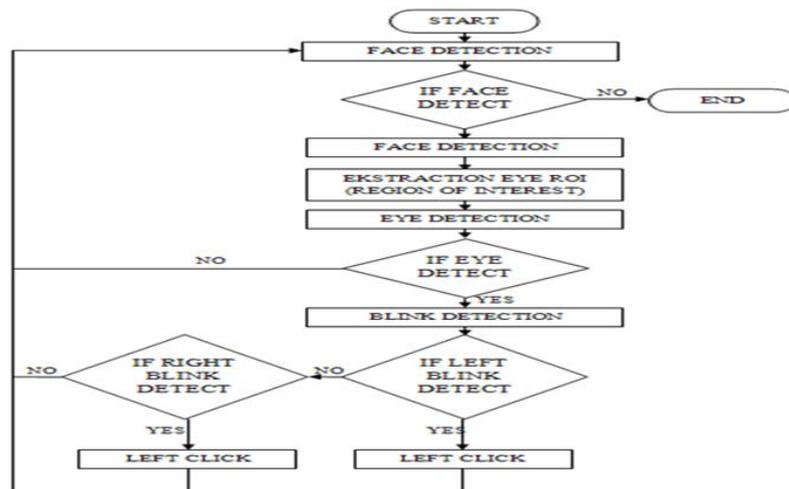
Gambar 7. Persamaan *Circle Hough Transform*

Berdasarkan persamaan tersebut, R adalah jari-jari lingkaran yang akan dicari, a dan b adalah titik pusat lingkaran. Gambar 8 merupakan ilustrasi *Circle Hough Transform*.



Gambar 8. Ilustrasi *Circle Hough Transform* (Supriyanti *et al.* (2012))

Aplikasi mampu mendeteksi kedipan mata melalui beberapa tahapan. Proses awal *blink detection*, yaitu melakukan binerisasi citra mata yang terdeteksi, kemudian dilakukan proses erosi dan dilasi. Pada proses akhir digunakan algoritma *hough circle* untuk mendeteksi bentuk lingkaran mata. Gambar 9 menjelaskan tentang proses deteksi mata dan deteksi kedipan mata.



Gambar 9. Alur Kerja Aplikasi untuk Deteksi Kedipan Mata

c. Implementasi

Implementasi terhadap desain aplikasi yang telah dibuat. Pada fase ini, implementasi menggunakan *tools developer*, Visual Studio.NET dengan *platform* aplikasi desktop.

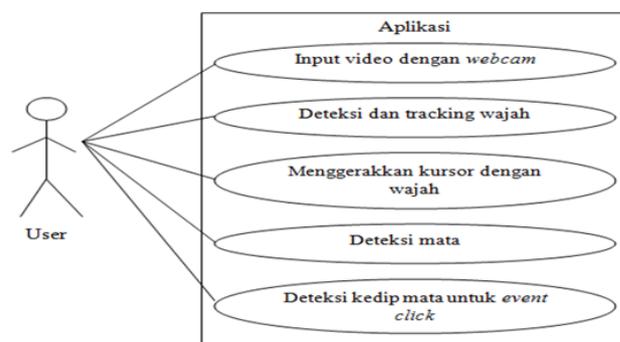
d. Pengujian

Pengujian aplikasi dilakukan terhadap kecepatan aplikasi dalam mendeteksi wajah berdasarkan sudut kemiringan wajah tertentu. Selain itu, pengujian dilakukan pula terhadap kecepatan aplikasi dalam mendeteksi iris mata berdasarkan nilai *threshold* tertentu.

C. HASIL DAN PEMBAHASAN

1. Analisis Kebutuhan Fungsional

Penyandang disabilitas gerak tangan memiliki kesulitan untuk berinteraksi dengan komputer. Hal ini dikarenakan keterbatasan mereka dalam mengoperasikan *mouse*. Gambar 10 menjelaskan tentang kebutuhan fungsional aplikasi *virtual mouse* bagi penyandang disabilitas.



Gambar 10. Kebutuhan Fungsional Aplikasi *Virtual Mouse*

Berdasarkan Gambar 10, dapat dijelaskan bahwa aplikasi virtual mouse memiliki kemampuan untuk menerima input berupa video, mendeteksi dan melakukan penjejakan wajah, mengendalikan kursor dengan gerakan wajah, mendeteksi mata, dan mendeteksi kedipan mata untuk *event click*.

2. Implementasi Aplikasi

a. Face Detection

Beberapa tahapan yang dilakukan untuk mendeteksi wajah pada aplikasi *virtual mouse*, yaitu :

1) Greyscale Image

Sebelum melakukan pendeteksian wajah, terlebih dahulu dilakukan proses konversi citra berwarna ke dalam citra *greyscale*, hal ini dikarenakan proses pengklasifikasian dilakukan pada citra *greyscale*.

2) Klasifikasi *Haar Feature*

Proses klasifikasi fitur *Haar* menggunakan file *Haarcascade.xml* yang terdapat pada pustaka *computer vision* EmguCV. Fitur *Haar* pada wajah akan digunakan dalam proses pendeteksian wajah. Proses perhitungan nilai fitur menggunakan *integral image*.

3) Face Training

Proses *face training* pada EmguCV menggunakan beberapa parameter untuk mendeteksi objek dengan algoritma *haarcascade*, parameter tersebut, yaitu *scale factor*, *min neighbors*, *flag*, dan *min size*. *Scale factor* merupakan parameter yang menentukan seberapa besar ukuran gambar diturunkan nilainya pada tiap skala gambar, *min neighbors* merupakan jumlah minimal persegi di sekitar yang akan membentuk suatu objek, *flag* merupakan tipe pendeteksi *Haarcascade* yang akan digunakan, sedangkan *min size* merupakan parameter untuk menentukan ukuran minimal wajah yang dapat terdeteksi.

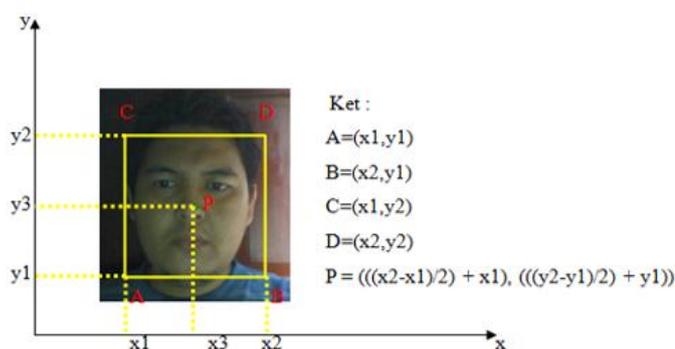
File *haarcascade_frontalface_alt.xml* yang terdapat pada pustaka EmguCV digunakan untuk melatih dan mendeteksi wajah. Pada penelitian ini ukuran *sub window* yang digunakan 20x20 (*min size*), nilai *min neighbors*=3, nilai *scale factor* 1.1, sedangkan *flag* yang digunakan bertipe `CV_HAAR_FIND_BIGGEST_OBJECT` untuk mencari satu objek dengan ukuran besar.

b. Set Cursor Position

Setelah proses pendeteksian wajah dilakukan langkah selanjutnya adalah mengalokasikan *cursor position*. Pengalokasian *cursor position* dilakukan pada sebuah *cursor navigation* dimana alokasi *cursor* bertumpu pada titik tengah dari wajah. Beberapa tahapan untuk mengendalikan kursor pada aplikasi dengan gerakan wajah, yaitu sebagai berikut :

1) Penentuan Titik Tengah Wajah

Pencarian titik tengah wajah sendiri adalah dengan membagi dua lebar dan tinggi *rectangle*. Adapun proses penghitungan secara matematik adalah dengan menggunakan perhitungan koordinat cartesius, dimana sebuah frame diasumsikan sebagai koordinat sumbu x dan sumbu y.

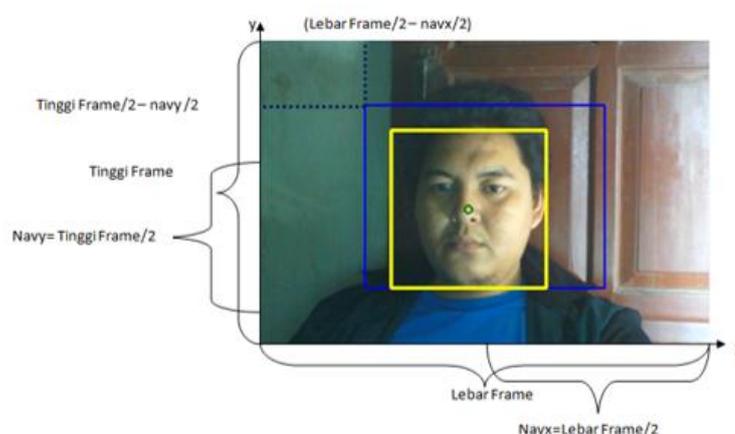


Gambar 11. Penentuan Koordinat Titik Tengah

Gambar 11 menjelaskan bahwa koordinat P (titik tengah wajah) diperoleh dari perhitungan $x3 = ((x2-x1)/2) + x1$; dan $y3 = ((y2-y1)/2) + y1$; sehingga $P(((x2-x1)/2) + x1), (((y2-y1)/2) + y1)$.

2) Proses Penentuan *Rectangle Navigation*

Rectangle navigation digunakan sebagai navigasi kursor. Navigasi ini dibuat dengan menentukan nilai lebar dan panjang *rectangle* dan menentukan titik awal untuk menggambar *rectangle*. Proses penentuan panjang dan lebar *rectangle* dengan membagi dua, lebar dan panjang *image*, sedangkan langkah selanjutnya adalah menentukan titik awal penggambaran *rectangle*.



Gambar 12. *Rectangle Navigation*

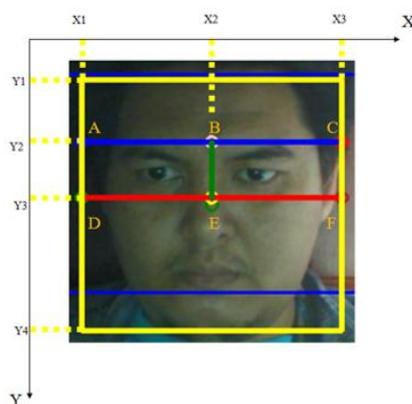
Berdasarkan Gambar 12, dapat dijelaskan bahwa $Navx$ merupakan lebar frame, sedangkan $Navy$ merupakan tinggi frame. Posisi awal untuk menggambar rectangle yaitu, $lebar\ frame/2 - navx/2$; dan $tinggi\ frame/2 - navy/2$, dimana penggambaran *rectangle* dimulai pada titik yang ditunjukkan dengan garis patah berwarna biru.

c. Eye and Blink Detection

Setelah proses pengendalian kursor, tahapan selanjutnya, aplikasi memiliki kemampuan untuk mendeteksi kedipan mata pengguna untuk melakukan *click event mouse*. Tahapan dalam mendeteksi mata, yaitu :

1) Ekstraksi ROI Mata

Pada penelitian ini ekstraksi mata menggunakan *facial grid*. Dalam *facial grid* terdapat tiga garis yang akan membentuk area mata kanan dan area mata kiri. Diantara garis tersebut ialah garis batas atas, garis batas bawah, dan garis pemisah mata kanan dan mata kiri. Gambar 13 menunjukkan ketiga garis tersebut.



Gambar 13. Pencarian ROI Mata

2) Deteksi Mata

Setelah melakukan pencarian ROI mata, langkah selanjutnya adalah mendeteksi mata. Algoritma *haarcascade classifier* digunakan untuk pendeteksian mata. File EmguCV yang digunakan *haarcascade_eye.xml*, pada file tersebut pengklasifikasian dilakukan sebanyak 24 kali.

3) Deteksi Kedipan Mata

Setelah mata terdeteksi selanjutnya adalah deteksi kedipan mata. Tahapan dalam deteksi mata, yaitu *thresholding*, erosi dan dilasi, serta operasi *hough circle*. Gambar 14 menunjukkan hasil *thresholding* citra, sedangkan Gambar 15 menunjukkan hasil proses erosi dan dilasi.



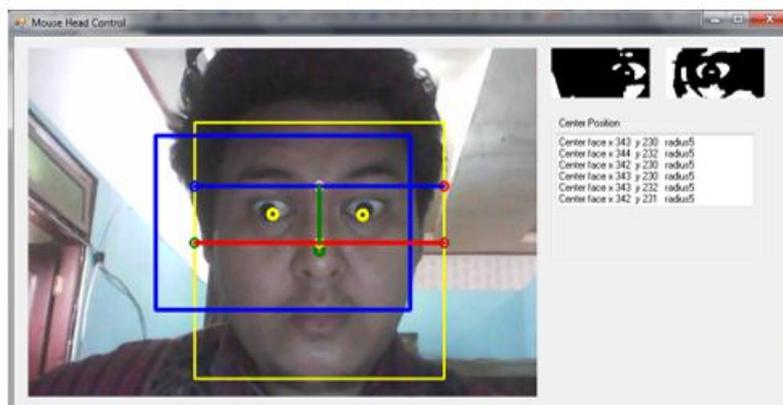
Gambar 14. Hasil *Thresholding*



Gambar 15. Hasil Erosi dan Dilasi

Hough circle digunakan untuk mendeteksi bentuk lingkaran iris mata, dimana *hough circle* bernilai 1 yang menandakan mata terbuka dan bernilai 0 untuk menandakan mata tertutup.

d. Hasil Akhir



Gambar 16. Tampilan Antarmuka Pengguna

Pada Gambar 16, *rectangle* kuning menunjukkan wajah yang terdeteksi. *Rectangle* biru adalah merupakan *rectangle* navigasi. Dua garis biru diatas mata merupakan garis batas pencarian ROI mata, 2 garis merah di bawah mata merupakan garis batas bawah pencarian ROI mata, sedangkan garis vertical warna hijau merupakan garis batas ROI mata kanan dan mata kiri. Gambar pada dua *image box* merupakan representasi gambar mata yang telah di *thresholding*.

3. Pengujian Aplikasi

Pengujian aplikasi dilakukan terhadap kecepatan aplikasi dalam mendeteksi wajah berdasarkan sudut kemiringan wajah tertentu dan kecepatan aplikasi dalam mendeteksi iris mata berdasarkan nilai *threshold* tertentu. Pada proses pengujian deteksi wajah dan iris mata, pengujian dilakukan dalam pencahayaan sebesar 29 lux dengan jarak 30 cm, *background* berwarna orange, dan *sample* 7 wajah yang berbeda.

a. Pengujian Deteksi Wajah

Tabel 3. Hasil Pengujian Kecepatan Deteksi Wajah

wajah \ sudut	10°	15°	20°	25°	30°	35°	40°	45°
Wajah 1	0.038	0.045	0.060	0.121	0.066	0.507	-	-
Wajah 2	0.029	0.027	0.035	0.027	0.036	0.038	0.042	-
Wajah 3	0.023	0.034	0.025	0.030	0.033	-	-	-
Wajah 4	0.041	0.046	0.046	0.041	0.044	-	-	-

Wajah 5	0.038	0.031	0.039	0.044	-	-	-	-
Wajah 6	0.037	0.036	0.038	0.045	0.042	0.049	-	-
Wajah 7	0.035	0.035	0.031	0.038	-	-	-	-

Dari hasil percobaan pada Tabel 3, disimpulkan bahwa nilai pergeseran terbaik dimana wajah dapat terdeteksi antara 10° dengan kecepatan rata-rata 0.035 *millisecond* sampai 25° dengan rata-rata kecepatan 0.049 *millisecond*, sedangkan untuk nilai kemiringan $>30^\circ$ wajah masih dapat terdeteksi hanya saja kestabilan deteksi wajah sudah mulai tidak baik, sedangkan untuk nilai kemiringan $>35^\circ$ wajah sulit terdeteksi.

b. Pengujian Deteksi Iris Mata

Tabel 4. Hasil Pengujian Kecepatan Deteksi Iris Mata

Threshold \ Wajah	30	40	50	60	70	80	90	100
Wajah 1	-	-	0.002	0.001	0.001	0.0035	0.002	0.001
Wajah 2	-	-	-	0.0025	0.002	0.0015	0.001	0.001
Wajah 3	-	-	0.002	0.001	0.0015	0.003	0.0029	0.001
Wajah 4	-	0.002	0.001	0.0015	0.001	0.001	0.0015	0.001
Wajah 5	-	-	0.002	0.0025	0.003	0.001	0.0015	0.001
Wajah 6	-	-	-	0.001	0.001	0.001	0.001	0.001
Wajah 7	-	-	-	-	-	0.001	0.001	0.001

Dari hasil percobaan pada Tabel 4, menunjukkan bahwa kecepatan deteksi iris mata terbaik dengan nilai threshold 90, dengan kecepatan rata-rata 0.00155 *millisecond*, dan nilai threshold 100 dengan kecepatan rata-rata 0.001 *millisecond*.

D. KESIMPULAN DAN SARAN

1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan yaitu :

- a. Nilai pergeseran wajah yang masih baik untuk dapat mendeteksi wajah antara 10° sampai 25° .
- b. Nilai minimal *threshold* yang baik untuk mendeteksi iris mata antara 90 sampai 100

2. Saran

Berdasarkan hasil yang ditemukan dan dicapai pada penelitian ini, maka terdapat beberapa saran yang dapat diberikan, antara lain:

- a. Melakukan pengujian dengan intensitas cahaya yang berbeda
- b. Melakukan pengujian berdasarkan pengaruh jarak terhadap intensitas pencahayaan.
- c. Menambahkan fungsi *mouse event drag and drop* ke dalam aplikasi.
- d. Menambahkan fungsi *speech to text* ke dalam aplikasi sebagai *virtual keyboard* untuk penyandang disabilitas.

DAFTAR PUSTAKA

- Agustian, I., Hidayat R., dan Widodo, Th. S., 2012, Mouse Kamera dengan Deteksi Wajah Realtime dan Deteksi Kedip Berbasis Metode Haarcascade dan SURF, Conference on Information Technology and Electrical Engineering (CITEE 2012), Gadjah Mada University : Yogyakarta, ISSN : 2085-6350, pp. 146-151
- Bevilacqua V., Filograno G., dan Mastronardi G., 2008, Face Detection by Means of Skin Detection, Advanced Intelligent Computing Theories and Applications-With Aspects of Artificial Intelligence Vol. 5227, pp. 1210-1220
- Gupta, A., Rathi, A., dan Radhika, Y., 2012, "Hands-Free PC Control" Controlling of Mouse Cursor Using Eye Movement, International Journal of Scientific and Research Publications Vol. 2, Issue : 4, pp. 1-5.
- <http://www.batamtoday.com>, diakses tanggal 29 April 2014
- <http://www.hug-your-dream.blogspot.com>, diakses tanggal 29 April 2014
- <http://www.padang.ppdi.co/en>, diakses tanggal 29 April 2014
- <http://www.ppdi.co/>, diakses tanggal 29 April 2014
- Huang, Han-min., dan Hu, G., 2009, A Face Detection Based on Face Features, Fuzzy Information and Engineering Vol.2, Advances in Intelligent and Soft Computing Vol. 62, pp. 173-180
- Irwanto., Kasim, E.R., Fransiska A., Lusli M., dan Okta S., 2010, Analisis Situasi Penyandang Disabilitas di Indonesia: Sebuah Desk-Review, Pusat Kajian Disabilitas Fakultas Ilmu-Ilmu Sosial dan Politik Universitas Indonesia

- Kasiński, A., dan Schmidt, A., 2007, The Architecture of the Face and Eyes Detection System Based on Cascade Classifiers, *Computer Recognition Systems 2-Advances in Soft Computing* Vol. 45, pp 124-131
- Kasiński, A., dan Schmidt, A., 2010, The architecture and performance of the face and eyes detection system based on the Haar cascade classifiers, *Pattern Analysis and Applications* Vol. 13, Issue : 2, pp. 197-211
- Kocejko T., Bujnowski A., dan Wtorek J., 2009, Eye Mouse for Disabled, Human-Computer System Interaction, *Advances in Intelligent and Soft Computing* Vol. 60, pp. 109-122
- Królak A., dan Strumillo P., 2012, Eye Blink Detection System for Human-Computer Interaction, *Universal Access in The Information Society* Vol. 11, Issue : 4, pp. 409-419
- Lienhart, R., Kuranov, A., dan Pisarevsky, V., 2002, Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection, Link : http://videoanalysis.org/Prof._Dr._Rainer_Lienhart/Publications_files/MR-L-TR-May02-revised-Dec02.pdf, diakses tanggal 2 Mei 2014
- Poornima, K.M., Danti, A., dan Narasimhamurthy, S.K., 2014, Novel Feature Extraction for Face Recognition using Multiscale Principal Component Analysis, *International Journal of Computer Science and Engineering Communications- IJCSEC* Vol.2, Issue : 1, pp. 121-128
- Rhody, H., 2005, "Lecture 10: Hough Circle Transform", Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology
- Supriyanti, R., Setiawan, B., Widodo, H. B., dan Murdyantoro, E., 2012, Detecting Pupil and Iris under Uncontrolled Illumination using Fixed-Hough Circle Transform, *International Journal of Signal Processing, Image Processing and Pattern Recognition* Vol. 5, No. 4, pp. 175-188
- Syarif, Y., Isnanto, R.R., dan Satoto, K.I., 2013, Perangkat Lunak Pengendali Pointer Menggunakan Pelacakan Mata (Eye Tracking) dengan Algoritma LUCAS KANADE, *Jurnal TRANSIENT* Vol. 2, No. 3, pp. 714-719
- Undang-undang Republik Indonesia Nomor 19, Tahun 2011, mengenai Pengesahan Convention On The Rights Of Persons With Disabilities
- Yang, Ming-Hsuan., 2009, Face Detection, *Encyclopedia of Biometrics*, Springer : USA, pp. 303-308.