

Jurnal ELTIKOM, Vol. 3, No. 2, Desember 2019, hal. 64-76  
ISSN 2598-3245 (Print), ISSN 2598-3288 (Online)  
Tersedia online di <http://eltikom.poliban.ac.id>  
DOI : <http://doi.org/10.31961/eltikom.v3i2.114>

## HAND GESTURE RECOGNITION SEBAGAI PENGGANTI MOUSE KOMPUTER MENGGUNAKAN KAMERA

Helda Yunita<sup>1)</sup>, Endang Setyati<sup>2)</sup>

<sup>1)</sup> STMIK Indonesia Banjarmasin

<sup>2)</sup> Institut Sains dan Teknologi Terpadu Surabaya  
e-mail: [hheldayunita@gmail.com](mailto:hheldayunita@gmail.com)<sup>1)</sup>, [endang@stts.edu](mailto:endang@stts.edu)<sup>2)</sup>

### ABSTRACT

*Nowadays, development of technology goes rapidly, so methods of interaction and communication between user and computer is one of the demands of technological development. So many technology try to minimize the tools. Therefore users require more communications equipment that is natural because it does not require direct contact with the equipment input. For example, the movement of the human body in front of the camera should be able to be interpreted by computers. To solve the problem, the writer does a research on the detection of hand gesture. The input is in the form of hand gestures and hand movements in front of the camera to make an mouse action or camera mouse. The method used is the convexhull algorithm. Using Convexhull algorithm we can get count the finger of user that can be reference mouse action. Actually, so many research on the camera mouse, but the implementation is still use a tools. This study develops the implementation of hand gesture recognition to set the mouse movements for realtime video. With hand gesture recognition and convexhull algorithm method the recognition about hand can be easier just using camera, just a few second mouse action in computer can be used better with an accuracy 68% from 75 attempts.*

**Keywords:** camera mouse, convexhull algorithm, hand gesture recognition

### ABSTRAK

*Akhir-akhir ini perkembangan teknologi semakin pesat, metode interaksi dan komunikasi antara pengguna dengan komputer adalah salah satu tuntutan perkembangan teknologi. Berbagai macam pembaharuan teknologi mengusahakan untuk meminimalisir berbagai macam perangkat menjadi satu agar lebih mudah digunakan. User lebih membutuhkan peralatan komunikasi yang bersifat alami karena tidak membutuhkan kontak langsung dengan peralatan input. Misalnya dengan gerakan dari tubuh manusia didepan kamera komputer sudah bisa menginterpretasikan. Untuk mengatasi masalah tersebut maka dilakukan suatu penelitian tentang deteksi isyarat tangan. Inputan berupa isyarat dan gerakan tangan didepan kamera dapat memberikan aksi pergerakan pada mouse yang diistilahkan dengan kamera mouse. Metode yang digunakan adalah convexhull algorithm. Melalui convexhull algorithm bisa didapatkan jumlah jari tangan yang kemudian dapat dijadikan acuan dalam pengerjaan aksi mouse. Sebenarnya sudah banyak penelitian tentang camera mouse, tetapi implementasinya masih banyak yang bergantung dengan peralatan tambahan. Penelitian ini mengembangkan penelitian yang sudah ada, yaitu hand gesture recognition dengan implementasi pergerakan mouse dari video secara realtime. Dengan hand gesture recognition dan menggunakan metode convexhull algorithm pengenalan tangan akan lebih mudah hanya dengan menggunakan kamera, hanya dengan hitungan detik aksi mouse pada komputer dapat berjalan dengan baik yaitu dengan tingkat akurasi sebesar 68 % dari 75 kali percobaan.*

**Kata Kunci:** algoritma convexhull, kamera mouse, pengenalan isyarat tangan

### I. PENDAHULUAN

**D**EWASA ini terdapat banyak minat dalam mengembangkan interaksi alami antara manusia dan komputer. Beberapa studi interaksi manusia dan komputer dalam komputasi universal sudah ramai diperkenalkan. Teknik interface berbasis visi-ekstrak informasi gerak tanpa peralatan biaya tinggi dari sebuah gambar video input. Dengan demikian, pendekatan berbasis vision memperhitungkan teknik yang efektif untuk mengembangkan sistem interface komputer manusia.

Telah banyak mobile phone yang menggunakan teknologi touch screen untuk dapat berinteraksi dengan penggunanya. Namun teknologi ini masih tergolong mahal untuk dapat digunakan pada komputer desktop dan laptop [1]. Dengan berbagai macam hardware pada komputer setiap tahunnya

mengalami kemajuan yang sangat pesat. Hal ini dapat dilihat dari beberapa generasi yang telah meluncurkan beberapa jenis processor core, motherboard, LCD, LED, sistem memori seperti harddisk dan flashdisk dengan kapasitas yang besar, proyektor, serta bluetooth. Namun ternyata tidak semua elemen komputer mengalami perkembangan. Salah satunya adalah mouse dan keyboard yang memiliki peranan penting dalam cara manusia berinteraksi dengan komputer [15].

Untuk interaksi manusia komputer yang berdasarkan kepada penglihatan, pengenalan kode pergerakan jari tangan masih belum banyak digunakan. Dari kode pergerakan jari tangan dapat digunakan sebagai input untuk mengurangi atau bahkan menggantikan perangkat mouse pada setiap komputer. Hal ini merupakan cara yang mudah untuk berinteraksi kepada komputer dengan pengenalan isyarat tangan (*hand gesture recognition*) dapat menggunakan sebuah kamera [2]. Suatu sistem dapat mengikuti *hand gesture* dan kemudian memberikan respon sesuai dengan *gesture* yang dikenali tersebut. Agar dapat melakukan ini dapat pula menggunakan *colour marker* yang diletakkan pada tangan [3].

Dengan menerima parameter pergerakan dari pengguna dalam suatu video yang terekam secara real-time, pointer mouse juga dapat dikontrol sesuai dengan parameter pergerakannya. Mouse memasukkan data berupa koordinat dari gerakannya dan diinterpretasikan menjadi gerakan kursor yang terlihat pada layar komputer. Bentuk dan jenis mouse saat ini sangat bermacam – macam, mulai dari mouse klasik dengan menggunakan bola sampai dengan mouse tanpa kabel yang menggunakan sensor CMOS sebagai pengganti bola tersebut. Penggunaan mouse yang cenderung sama dalam kurun waktu yang cukup lama membuat pengguna komputer semakin pandai dalam menggunakan mouse. Alat mouse baru ini dinamakan kamera mouse.

Secara visual pergerakan pointer mouse diterima dari parameter yang ditangkap dari video dan pengaturan pada tetikus komputer dapat ditegaskan sesuai aturan yang ditentukan. Dalam hal ini yang menjadi fokus pelajaran adalah tangan seseorang dapat diamati dari warna, tipe dan texture nya. Dari pengenalan tangan tersebut dapat pula dikenali struktur tangannya.

Kamera mengenali telapak tangan dengan jari – jari yang terbuka untuk mengatur pergerakan pointer mouse. Setelah pointer mouse berada dilokasi yang dituju, operasi mouse dapat dieksekusi seperti mengantup telapak tangan tersebut untuk menghentikan pergerakannya dan mengeluarkan telunjuk untuk klik kiri dan mengeluarkan jari telunjuk dengan jari tengah untuk klik kanan. Isyarat jempol dan jari telunjuk dapat untuk aksi double klik. Jari telunjuk, jempol dan jari tengah dikeluarkan bersama – sama untuk isyarat drag. Kecepatan tanggap dan ketepatan penggunaan camera mouse dapat dijadikan sebagai patokan. Hand gesture, visual hand tracking dan HSV color quantization merupakan metode yang dapat membuat camera mouse.

Dari berbagai masalah yang telah dijelaskan, penulis tertarik melakukan penelitian mengenai hand gesture recognition untuk pengenalan pergerakan jari tangan sebagai pengganti mouse dengan menggunakan kamera.

## II. METODE PENELITIAN

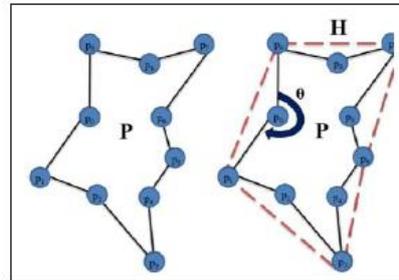
Fungsi komputer yang mendasar dapat berjalan karena adanya komunikasi antara manusia dengan komputer. Manusia dapat menciptakan dan memberikan perintah yang diinginkannya, lalu komputer menjalankannya. Hand gesture recognition atau yang disebut pula pengenalan isyarat tangan merupakan interaksi antara manusia dan komputer (mesin). Hal yang paling mendasar dalam pengenalan isyarat tangan adalah dengan membuat interaksi yang alami (sesuai kebiasaan) antara manusia dan komputer yang mana pengenalan isyarat tersebut dapat digunakan untuk mengontrol sebuah robot atau menyampaikan suatu informasi [4].

Sistem arsitektur mengenai hand gestur ini adalah penangkapan gambar, pergerakan tangan (*hand tracking*) dan pengenalan isyarat tangan dan *control mouse* [5]. Sistem pengenalan isyarat tangan (*hand gesture recognition*) dapat digunakan untuk beberapa pengaplikasian seperti terjemahan, virtual environment, smart surveillance, kontrol robot, sistem medis dan lain – lain. [6]

Convex hull merupakan persoalan klasik dalam komputasi geometri. Convex hull digambarkan secara sederhana dalam sebuah bidang sebagai pencarian subset dari himpunan titik pada bidang tersebut, sehingga jika titik-titik tersebut dijadikan poligon maka akan membentuk poligon yang konvek. [7] Suatu poligon dikatakan konveks jika garis yang menghubungkan antar kedua titik dalam poligon tersebut tidak memotong garis batas dari poligon. Convex hull suatu obyek P didefinisikan sebagai area poligon convex terkecil yang melingkupi P. Oleh karena itu, untuk suatu himpunan titik  $N = \{p_0, p_1, p_2, \dots, p_n\}$

... , pN} ∈ P, maka dapat dinyatakan bahwa hull H dapat disusun dengan M titik dari himpunan N untuk membuat suatu area konveks poligon minimum. [8].

Convex hull dibuat dengan mengambil sudut interior θ, dari tiga titik yang bersebelahan {p1, p0, p9}. Jika θ > π maka p0 dianggap sebagai titik refleksi dan p0 bukan anggota M. Himpunan akhir H adalah {p1, p9, p7, p5, p3} [9].

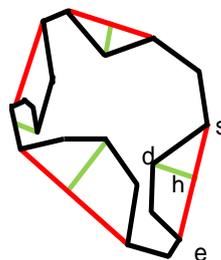


Gambar. 1. Generalisasi logika convex hull [8]

Pencarian convex hull dari sebuah himpunan titik Q (CH(Q)) yaitu dengan mencari sebuah convex set terkecil yang memuat seluruh titik pada Q. Convex hull dari sebuah himpunan titik Q (CH(Q)) pada n dimensi adalah seluruh irisan dari semua convex set yang mengandung Q. Sehingga untuk N buah titik {p1, p2, ..., pN} ∈ P, convex hull merupakan himpunan convex combination sesuai dengan Persamaan (1).

$$CH(Q) = \left\{ \sum_{j=1}^N \lambda_j p_j ; \lambda_j \geq 0; \sum_{j=1}^N \lambda_j = 1 \right\} \tag{1}$$

*Convexity defect*: adalah sebuah fitur dalam *OpenCV* yang berfungsi untuk menemukan *defect* antara *convex hull* yang terbentuk dengan kontur dari poligon. *Defect* tersebut berguna untuk menemukan *feature* pada sebuah poligon, salah satunya yaitu untuk mendeteksi jari tangan manusia.



Gambar. 2. Convexity defect sebuah poligon [8]

*Convex hull* digambarkan dengan garis merah yang menyelubungi poligon dengan garis konturnya yang berwarna hitam. Simbol “s” dan “e” menunjukkan “start point” dan “end point” dari *convexity defect* tersebut. Sedangkan simbol “d” melambangkan “depth point”, yaitu titik kontur yang terletak antara “s” dan “e” yang merupakan titik terjauh antara kontur dengan garis *convex hull* yang dilambangkan dengan “se”. Simbol “h” yaitu “depth” atau kedalaman dari *defect* yang merupakan jarak dari “d” hingga garis “se”. [10]

Segmentasi warna kulit (*skin detection*) banyak digunakan untuk aplikasi pengenalan wajah, deteksi badan, ataupun anggota badan. *Skin detection* ini bertujuan untuk dapat mendeteksi warna kulit dari setiap citra yang ditangkap oleh kamera. Citra yang diperoleh umumnya memiliki format RGB. Untuk memudahkan pendeteksian warna kulit, format RGB ini dapat diubah ke format warna lainnya. Segmentasi warna kulit dilakukan dengan cara menentukan nilai *range* dari Y, Cr, dan Cb. Sehingga untuk setiap piksel dalam citra, jika berada dalam *range* tersebut, maka akan dianggap sebagai warna kulit, sedangkan untuk yang berada di luar *range* tersebut akan dianggap sebagai *background* (latar belakang) [11]

Proses pencarian titik pusat citra tangan setelah disegmentasi dapat dilakukan dengan mengetahui  $x_i$  dan  $y_i$  yang merupakan koordinat x,y dari i piksel pada area tangan, dan k menunjukkan jumlah piksel

pada area tersebut. Setelah diketahui lokasi pusat tangan, selanjutnya dapat dihitung area telapak tangan untuk mendapatkan ukuran tangan dengan menggunakan Persamaan (2).

$$\bar{x} = \frac{\sum_{i=0}^k x_i}{k}, \frac{\sum_{i=0}^k y_i}{k} \tag{2}$$

Untuk mendapatkan ukuran tangan dapat dibuat suatu lingkaran dengan menaikkan jari-jari lingkaran dari pusat koordinat sampai lingkaran mencapai piksel hitam pertama. Saat algoritma telah mendapatkan piksel hitam pertama, akan kembali ke nilai jari-jari sebelumnya.

Algoritma ini menggunakan asumsi bahwa saat lingkaran mencapai piksel hitam pertama, setelah menggambar lingkaran yang semakin membesar, maka panjang dari pusat adalah jari-jari dari tangan [8].

Warna adalah hasil persepsi dari cahaya dalam spektrum wilayah yang terlihat oleh retina mata, dan memiliki panjang gelombang antara 400nm sampai dengan 700nm [12]. Sedangkan ruang warna adalah model matematis abstrak yang menggambarkan cara agar suatu warna dapat direpresentasikan sebagai baris angka biasanya dengan nilai-nilai dari tiga atau empat buah warna atau komponen. contohnya adalah ruang warna RGB, ruang warna CMY/CMYK, ruang warna YIQ, ruang warna YCbCr, ruang warna HSI, HSL, HSV, ruang warna CIELAB.

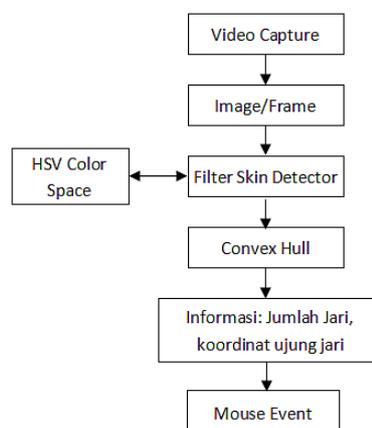
HSV mendefinisikan warna dalam terminologi Hue, Saturation dan Value. Keuntungan HSV adalah terdapat warna-warna yang sama dengan yang ditangkap oleh indra manusia. HSV memiliki 3 karakteristik pokok, yaitu Hue, Saturation dan Value. Hue : menyatakan warna sebenarnya, seperti merah, violet, dan kuning dan digunakan menentukan kemerahan (redness), kehijauan (greeness) (3). Saturation : kadang disebut chroma, adalah kemurnian atau kekuatan warna (4). Value : kecerahan dari warna (5). Nilainya berkisar antara 0-100 %. Apabila nilainya 0 maka warnanya akan menjadi hitam, semakin besar nilai maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut. Untuk mendapatkan setiap nilai dari HSV relatif lebih sederhana.

$$H = \tan\left(\frac{3(G - B)}{(R - G) + (R - B)}\right) \tag{3}$$

$$S = 1 - \left(\frac{\min(R, G, B)}{v}\right) \tag{4}$$

$$v = \frac{R + G + B}{3} \tag{5}$$

Pada penelitian ini bermaksud untuk mengembangkan sistem deteksi pengenalan isyarat tangan (hand gesture recognition) menggunakan citra dari hasil rekaman video mulai dari video capture yang dirubah menjadi image perframe dan setiap frame tersebut dideteksi kulit manusia dan dengan menggunakan metode *convexhull algorithm* dapat terdeteksi jumlah jarinya dan hasil akhirnya berupa aksi pada mouse. Alur proses tersebut selengkapnyanya dapat terlihat seperti Gambar 3.



Gambar. 3. Alur proses video capture hingga mendapatkan informasi jumlah jari

Langkah perancangan diawali dengan pengambilan citra gestur tangan dengan menggunakan *web-camera*. Citra hasil tangkapan kamera akan ditangkap dengan memperhatikan *sample color*. Kemudian membuat *binary representation* dari tangan tersebut. Langkah selanjutnya mendapatkan kontur dari tangan dengan mengurangi kontur yang mengganggu.

Dari kontur tangan tersebut dilakukan perhitungan dengan metode *convexhull algorithm*. Kemudian dilakukan pencocokan antara informasi yang didapat dengan data yang ada. Sehingga akan didapat hasil output berupa informasi mengenai tangan tersebut seperti jumlah jari tangan yang dapat digunakan untuk identifikasi perintah sebagai pengganti tombol klik pada *mouse*.

Proses penghitungan titik pusat tangan dan pendeteksi gerakan tangan untuk menentukan arah gerakan tangan pengguna yang akan digunakan untuk menggerakkan kursor. Untuk dapat mengambil citra secara realtime, maka sistem perlu terhubung ke kamera yang mana kamera tersebut dapat diletakkan diatas computer / laptop seperti Gambar 4.



Gambar. 4. Penempatan kamera

Untuk menghubungkan ke sistem, sistem dapat mendeteksi perangkat kamera yang terhubung ke laptop atau komputer. Dimulai dari import library DirectShow, menyiapkan variable, mendeteksi kamera yang digunakan kemudian dilanjutkan dengan capture library dari bantuan EmguCV. Proses query frame tersebut akan berjalan terus menerus hingga capture dihentikan. Hasil dari query frame berupa image atau frame dan fps tertentu yang akan digunakan dalam rangkaian tahap selanjutnya mulai dari skin detection hingga penemuan jumlah jari tangan seperti yang terlihat pada Gambar 3.

Setelah citra dari frame grabber melalui query frame telah diperoleh, langkah selanjutnya adalah melakukan segmentasi antara piksel skin dan non skin. Untuk tahap ini telah disiapkan terlebih dahulu sebuah class yang berisi prosedur dan parameter-parameter untuk mengisi nilai range dari HSV. Oleh sebab itu algoritma skin detection menggunakan HSV dapat dikerjakan dengan langkah seperti berikut:

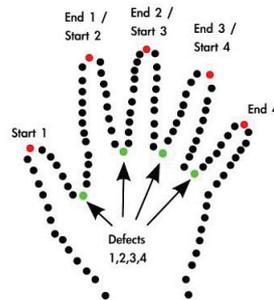
1. Citra input diperoleh dari query frame yang merupakan hasil dari video capture.
2. Citra input dalam ruang warna RGB diubah menjadi ruang warna HSV menggunakan transformasi HSV.
3. Histogram dari ketiga komponen dihitung dan dari histogram tersebut nilai threshold HSV ditentukan.
4. Masking diterapkan untuk piksel skin dalam citra.
5. Threshold diterapkan pada citra mask.
6. Lakukan smoothing pada citra hasil threshold.
7. Citra output hanya berisi piksel skin.

Setelah citra didapat dari video, citra yang masih berupa RGB tersebut dirubah menjadi bentuk HSV. Setelah histogram dihitung dapat dilakukan teknik masking, threshold dan smoothing. Sampai pada akhirnya citra outputnya adalah piksel dari kulit yang terdeteksi. Melalui algoritma tersebut dapat diperoleh citra keluaran dengan model HSV. Metode deteksi skin dengan menggunakan ruang warna HSV dengan nilai true positive 91,1% dan true negative 88,1 terbukti paling cocok untuk mendeteksi kulit tipikal India [13].

Metode convex hull dan convexity defect bertujuan untuk mendeteksi jari tangan pengguna yang tertangkap oleh kamera [8]. Untuk dapat mengekstrak fitur tangan dengan baik diperlukan dua tahap proses, yaitu :

1. *Pembentukan convex hull menggunakan algoritma Sklansky*: *Convex hull* merupakan himpunan titik-titik yang membentuk sebuah poligon konveks yang melingkupi seluruh himpunan titik tersebut. Algoritma Sklansky yaitu sebuah algoritma linear sekuensial yang digunakan untuk membentuk *convex hull* dari himpunan titik atau sebuah poligon sederhana [9].

2. Menentukan defect dari convex hull yang telah terbentuk: Defect sendiri dapat didefinisikan sebagai area terkecil dalam convex hull yang mengitari tangan. Pencarian defect ini berguna untuk menemukan feature pada tangan, sehingga nantinya dapat dideteksi jari tangan dan diketahui berapa jumlahnya yang dapat dihitung seperti Gambar 5 berikut.



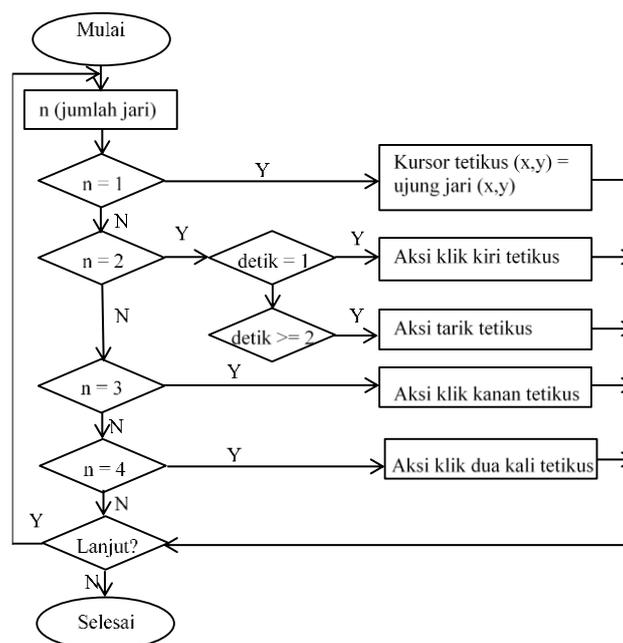
Gambar. 5. Convexity defect citra tangan [17]

Langkah-langkah algoritma penghitungan jumlah jari tangan adalah sebagai berikut :

1. Koordinat titik tengah(x,y) persegi panjang terkecil (*minimum area rectangle*) dari kontur tangan (*box*).
2. Titik koordinat (x,y) startpoint dan depthpoint dari defect yang ditemukan.
3. Jumlah Jari=0.
4. Untuk  $i = 0$  hingga  $i <$  jumlah defect  
 Jika ( $startpoint.y < box.center.y$  atau  $depthpoint.y < box.center.y$ ) dan ( $startpoint.y < depthpoint.y$ ) dan ( $jarak\ startpoint\ dan\ depthpoint > tinggi\ box / 6,5$ ) Maka, jumlah jari = jumlah jari + 1
5. Jumlah jari yang terdeteksi ditampilkan.

Pada tahap penerapan algoritma convexhull, dapat diperoleh informasi mengenai jumlah jari dan koordinat setiap jari pada setiap frame yang terjadi. Fungsi-fungsi mouse yang dijalankan mengikuti jumlah jari yang terdeteksi. Informasi jumlah jari digunakan sebagai syarat memberi perintah mouse.

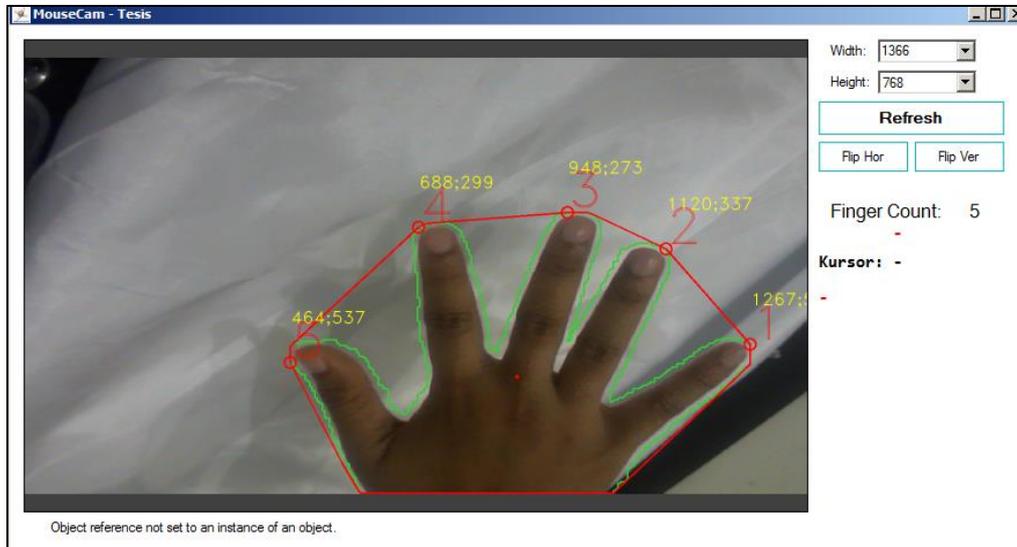
Koordinat dari 1 jari digunakan untuk mengubah koordinat kursor mouse agar mengikuti gerak dari satu jari tersebut. Pada sistem ini menggunakan satu jari untuk menggerakkan kursor mouse (sorot), dua jari untuk melakukan perintah klik kiri dan drag, tiga jari untuk melakukan perintah klik kanan, empat jari untuk melakukan perintah klik ganda yang terlihat alurnya seperti Gambar 6.



Gambar. 6. Alur proses jumlah jari terhadap aksi mouse

### III.HASIL DAN PEMBAHASAN

Berikut adalah langkah-langkah yang perlu dilakukan untuk mendapat citra streaming dari kamera. Pertama mempersiapkan dan memasang perangkat pada laptop atau PC dengan meletakkannya sesuai desain sistem. Dari peletakkan kamera, diharapkan bagian jari hingga pergelangan tangan dapat terlihat dengan baik agar tahap berikutnya berhasil baik seperti Gambar 7.



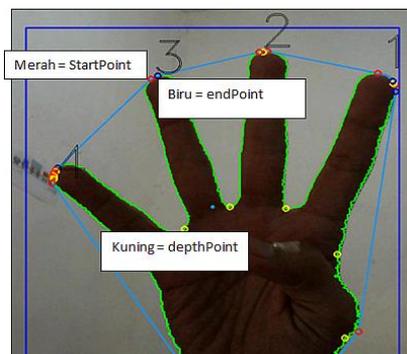
Gambar. 7. Hasil Capture kelima jari

Tangan terbuka secara jelas dan sebaiknya terdapat area yang kosong disekitar tangan agar memudahkan tangan untuk bergerak menggerakkan kursor mouse nantinya. Untuk objek disekitar tangan sebaiknya berlatar warna putih bukan cokelat atau krim dengan threshold yang digunakan dalam sistem ini memiliki range awal dalam RGB minimum (0,45,0) dan maksimum (40,255,255). Yang kemudian nilai RGB tersebut dirubah ke ruang warna HSV dengan menggunakan rumus (3), (4) dan (5). Hal ini dikarenakan agar tidak mengganggu proses pendeteksian tangan yang memperhatikan pada warna kulit orang asia [14]. Kemudian proses capture dapat terjadi setelah form dijalankan dan user menekan tombol F2 sehingga terlihat perubahan label pada form mengenai status kameramenjadi ON.

Pada Gambar 7 tampak hasil ekstraksi kontur dan hull. Garis warna hijau (lime) merupakan garis kontur tangan, garis merah merupakan garis polyline dari kontur tangan sedangkan kotak biru menunjukkan area biggest contour dimana dari kotak tersebut dapat dicari pusat dari kontur area tangan.

Langkah selanjutnya yang dilakukan pada setiap frame adalah tahap perhitungan jari. Pada tahap ini sekaligus juga dapat ditampilkan titik-titik start dan end dari hull dan defect.

Pada saat StartPoint, EndPoint dan DepthPoint telah didapatkan, titik puncak dan defect dapat ditentukan pula dari titik yang tertinggi dan terendah. Titik puncak jari tersebut dianggap sebagai perwakilan jari (perwakilan fingertip). Dari titik puncak jari tersebutlah dapat dihitung jumlah jari yang terdeteksi. Pengurutan penomoran dari kanan ke kiri seperti Gambar 8.



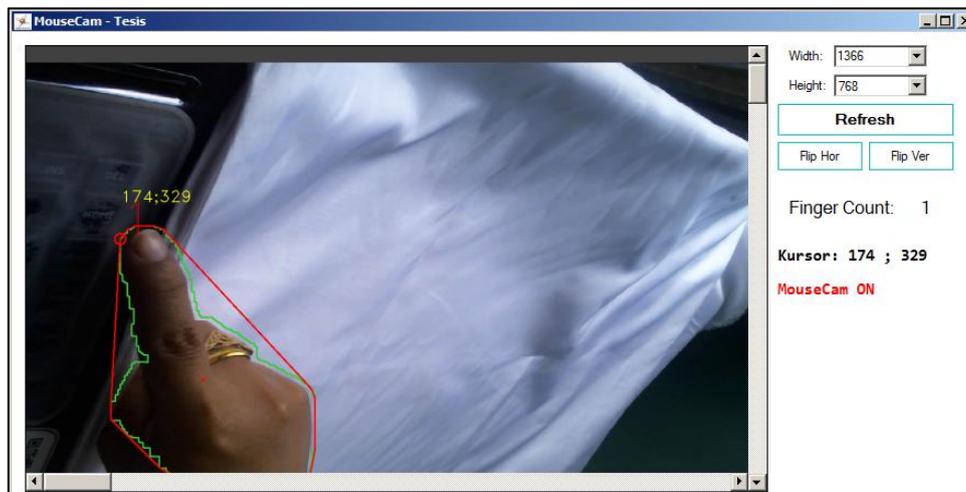
Gambar. 8. Perhitungan jari dan letak StartPoint, depthPoint, Endpoint hull dan defect

Uji coba dilakukan untuk mengetahui seberapa besar tingkat keberhasilan dari penelitian yang dilakukan. Dalam laporan ini, presentasi uji coba disajikan dalam aksi mouse pada layar komputer atau laptop. Data pengujian adalah berupa pergerakan cursor mouse, klik kiri, klik kanan, double klik dan drag pada komputer atau laptop. Uji coba dapat dilakukan pada pengukuran cahaya, posisi tangan, posisi kamera dan uji coba dengan beberapa pengguna yang berbeda.

Uji coba dilakukan dengan beberapa percobaan pada berbagai situasi, seperti pencahayaan :

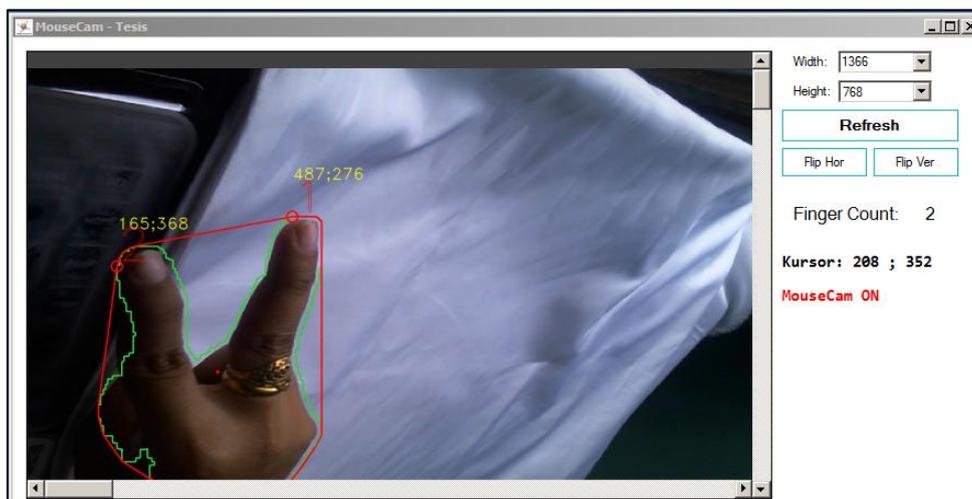
1. Sangat terang, diukur menggunakan light meter dengan ukuran cahaya yang lebih dari 100 lux.
2. Terang, diukur menggunakan lightmeter dengan ukuran cahaya < 100 lux.
3. Kurang terang, diukur menggunakan lightmeter dengan ukuran cahaya < 40 lux.
4. Cahaya redup, diukur menggunakan lightmeter dengan ukuran cahaya < 15 lux.

Uji coba yang dapat dilakukan pertama kali adalah dengan pengenalan isyarat tangan untuk mengatur posisi cursor mouse. Uji coba kali ini adalah dimulai dengan mengarahkan mouse ke posisi dari tengah ke sebelah kiri (mengarah ke sebuah icon pada desktop) seperti Gambar 9.



Gambar. 9. Uji coba untuk cursor mouse

Uji coba yang dilakukan selanjutnya adalah dengan pengenalan isyarat tangan untuk klik kiri mouse. Setelah posisi cursor mouse berada di salah satu icon (MS Word) pada desktop dapat dilakukan isyarat klik kiri dengan mengeluarkan 2 jari.



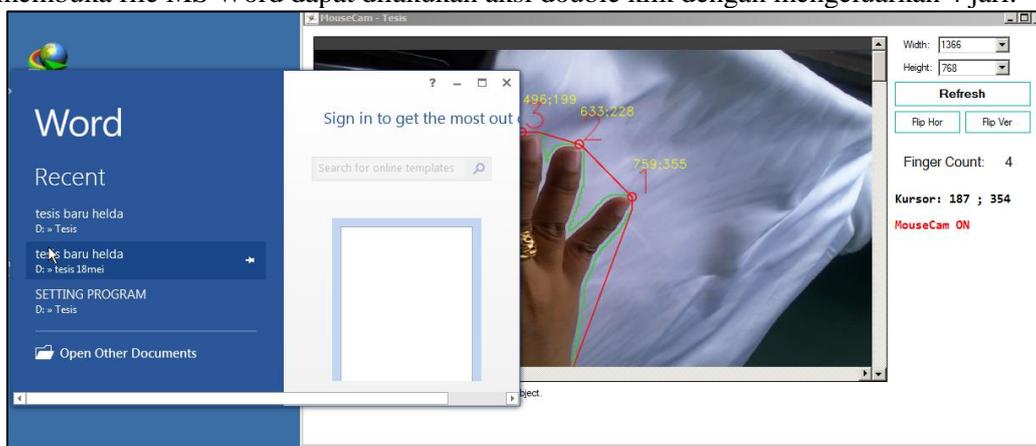
Gambar. 10. Uji coba untuk klik kiri mouse

Uji coba yang dilakukan selanjutnya adalah dengan pengenalan isyarat tangan untuk klik kanan mouse. Setelah cursor mouse mengarah ke salah satu icon pada desktop (MS Word) dapat dilakukan aksi klik kanan dengan mengeluarkan 3 jari seperti Gambar 11.



Gambar. 11. Uji coba untuk klik kanan mouse

Uji coba selanjutnya adalah dengan pengenalan isyarat tangan untuk double klik. Uji coba dilakukan untuk membuka file MS Word dapat dilakukan aksi double klik dengan mengeluarkan 4 jari.



Gambar. 12. Uji coba untuk double klik mouse

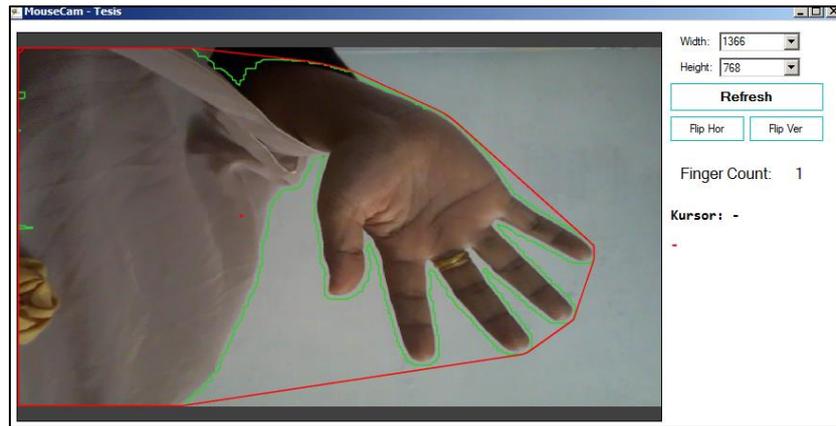
Uji coba selanjutnya adalah dengan pengenalan isyarat tangan untuk drag. Pada saat salah satu drive D terbuka dapat dilakukan perintah drag pada beberapa folder tersebut dengan mengeluarkan dua jari selama 2 detik kemudian mengeluarkan satu jari sambil digerakkan ke arah seleksi folder. Dengan berbagai percobaan pada cahaya tersebut dan dicatat waktu pengerjaannya didapatkan hasil pada Tabel I berikut.

TABEL I  
PENCATATAN WAKTU KERJA MOUSE PADA KONDISI CAHAYA BERBEDA

Keterangan Cahaya	Jumlah Keberhasilan Aksi (Detik)				
	Cursor	Klik Kiri	Klik Kanan	Double Klik	Drag
Sangat Terang	1,73	1,60	1,03	1,57	8,15
Terang	1,92	1,68	1,57	1,88	6,38
Kurang Terang	1,49	3,15	3,35	2,15	7,65
Cahaya Redup	2,39	8,06	9,59	4,59	9,97

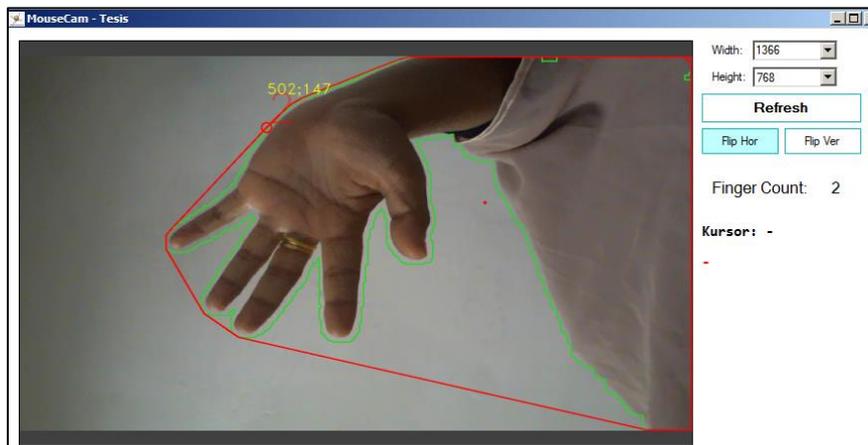
Dari Tabel 1 diatas terlihat hasilnya pada pencahayaan yang sangat terang dan terang hasil yang didapatkan adalah sangat baik dan cepat. Pada kondisi cahaya kurang terang hasil yang didapatkan adalah baik namun seringkali pengenalan contour tangan kurang yang pas, karena adanya bayangan dari tangan yang warnanya hampir sama dengan warna tangan. Begitu pula pada kondisi cahaya redup, hasil yang didapatkan adalah kurang baik karena warna tangan yang dikenali semakin gelap dan hampir sama dengan warna bayangan dari tangan.

Tombol flip horizontal dan vertical pada program dapat digunakan untuk merotasi posisi tangan yang ditangkap kamera. Tombol flip horizontal gunanya untuk merotasi tampilan secara mendatar. Tombol flip vertical gunanya untuk merotasi tampilan secara menurun. Kedua tombol ini dapat berguna untuk menyesuaikan orientasi tangan yang akan ditangkap kamera.



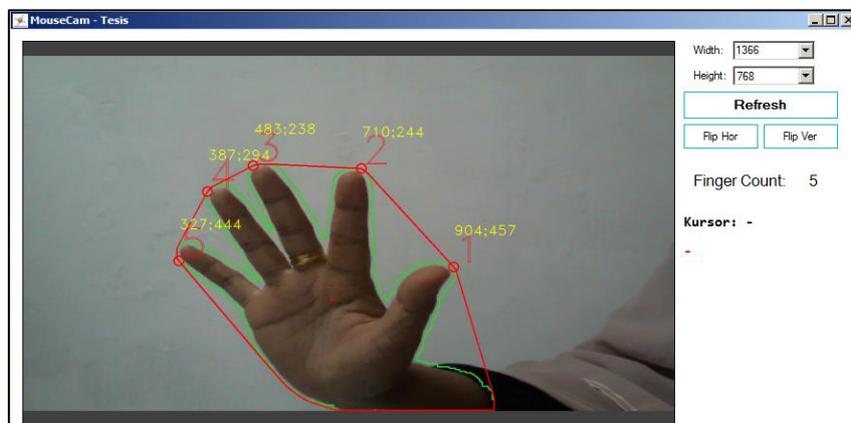
Gambar. 13. Hasil Tangkap Tangan Didepan Kamera / Laptop Sesudah Flip Secara Vertical

Pada dasarnya kamera menangkap objek yang berada disamping laptop, namun apabila orientasi tangan yang ditangkap bukan diletakkan disamping laptop contohnya seperti persis didepan laptop tanpa melakukan flip secara vertical, maka tangan yang tertangkap kamera tidak dapat dikenali jari-jarinya. Begitu pula apabila pemutaran yang dilakukan secara horizontal terlihat hasilnya yang masih belum sesuai harapan.



Gambar. 14. Hasil Tangkap Tangan Didepan Kamera / Laptop Sesudah Flip Secara Vertical dan Horizontal

Apabila pada saat pertama tadi dilakukan pemutaran secara vertical dan horizontal terlihat hasilnya seperti pada Gambar 15. Dari gambar tersebut terlihat hasilnya untuk penangkapan jumlah jari yang sudah hampir sesuai dengan yang diharapkan namun arah pergerakan tangan berlawanan dengan arah gerak yang sebenarnya. Sehingga pergerakan tetikus menjadi tidak sesuai dengan keinginan pengguna.



Gambar. 15. Hasil Tangkap Tangan Didepan Kamera / Laptop Sesudah Flip Secara Horizontal

Dari beberapa uji coba diatas dapat terlihat bahwa posisi tangan berpengaruh terhadap program, secara otomatis tangan yang ditangkap adalah dengan orientasi kebawah. Algoritma hanya dapat melakukan perhitungan dengan posisi jari yang menghadap keatas. Selain dari itu, maka perlu dilakukan flip sesuai dengan posisi tangan yang ditangkap oleh kamera.

Dengan menggunakan kamera mouse ini diharapkan dapat memberikan kontribusi pemanfaatan yang baik kepada user. Berikut dilakukan uji coba kepada 15 user yang berbeda dengan masing – masing user melakukan percobaan sebanyak 10 kali dan dicatat waktu serta keberhasilan pengerjaan aksinya yang terlihat seperti pada Tabel II.

TABEL II  
PENCATATAN UJI KERJA KAMERA MOUSE DENGAN USER BERBEDA

User	Jumlah Keberhasilan Aksi				
	Cursor	Klik Kiri	Klik Kanan	Double Klik	Drag
1	8	8	8	8	6
2	6	8	8	6	4
3	8	8	7	7	5
4	6	8	7	8	5
5	10	6	7	7	5
6	8	8	5	5	4
7	9	9	7	9	3
8	5	9	7	6	4
9	8	8	7	8	4
10	7	7	8	7	4
11	9	8	6	8	5
12	6	7	8	7	5
13	6	9	9	7	4
14	7	7	6	6	5
15	7	9	6	8	5

Dengan menggunakan Persamaan (6) didapatkan hasil error pengerjaannya seperti pada Persamaan (7).

$$Err = \sum_{i=1}^n \left( \frac{10 - Xi}{10} \right) \times 100\% \tag{6}$$

Dengan i = percobaan (1 s.d 75). n = jumlah percobaan (75). 10 = jumlah keberhasilan terbaik untuk uji coba dalam 1 aksi.

$$Err_1 = \left( \frac{10 - 8}{10} \right) \times 100\% = 20\% \tag{7}$$

Secara lebih singkatnya hasil error dengan rumus diatas dapat dilihat seperti pada Tabel III.

TABEL III  
HASIL PERHITUNGAN ERROR

User	Jumlah Keberhasilan Aksi				
	Cursor	Klik Kiri	Klik Kanan	Double Klik	Drag
1	20 %	20 %	20 %	20 %	40 %
2	40 %	20 %	20 %	40 %	60 %
3	20 %	20 %	30 %	30 %	50 %
4	40 %	20 %	30 %	20 %	50 %
5	0 %	40 %	30 %	30 %	50 %
6	20 %	20 %	50 %	50 %	60 %
7	10 %	10 %	30 %	10 %	70 %
8	50 %	10 %	30 %	40 %	60 %
9	20 %	20 %	30 %	20 %	60 %
10	30 %	30 %	20 %	30 %	60 %
11	10 %	20 %	40 %	20 %	50 %
12	40 %	30 %	20 %	30 %	50 %
13	40 %	10 %	10 %	30 %	60 %
14	30 %	30 %	40 %	40 %	50 %
15	30 %	10 %	40 %	20 %	50 %
<i>Rata-rata</i>	26,667 %	20,667 %	29,333 %	28,667 %	54,667 %

Dari Tabel III diketahui rata – rata error untuk cursor adalah sebesar 26,667%, rata – rata error untuk klik kiri adalah sebesar 20,667%, rata – rata error untuk klik kanan adalah sebesar 29,333%, rata – rata error untuk double klik adalah sebesar 28,667% dan rata – rata error untuk drag adalah sebesar 54,667%. Untuk nilai hasil akurasi dapat terlihat seperti pada Tabel IV.

TABEL IV  
HASIL PERHITUNGAN AKURASI

User	Jumlah Keberhasilan Aksi					Total Rata-rata
	Cursor	Klik Kiri	Klik Kanan	Double Klik	Drag	
1	0,8	0,8	0,8	0,8	0,6	
2	0,6	0,8	0,8	0,6	0,4	
3	0,8	0,8	0,7	0,7	0,5	
4	0,6	0,8	0,7	0,8	0,5	
5	1	0,6	0,7	0,7	0,5	
6	0,8	0,8	0,5	0,5	0,4	
7	0,9	0,9	0,7	0,9	0,3	
8	0,5	0,9	0,7	0,6	0,4	
9	0,8	0,8	0,7	0,8	0,4	
10	0,7	0,7	0,8	0,7	0,4	
11	0,9	0,8	0,6	0,8	0,5	
12	0,6	0,7	0,8	0,7	0,5	
13	0,6	0,9	0,9	0,7	0,4	
14	0,7	0,7	0,6	0,6	0,5	
15	0,7	0,9	0,6	0,8	0,5	
<i>Rata-rata</i>	0,733	0,793	0,707	0,713	0,453	0,680

Dari tabel IV terlihat rata – rata akurasi untuk cursor adalah sebesar 73,3%. Rata – rata akurasi untuk klik kiri adalah sebesar 79,3%. Rata – rata akurasi untuk klik kanan adalah sebesar 70,7%. Rata – rata akurasi untuk double klik adalah sebesar 71,3%. Rata – rata akurasi untuk drag adalah sebesar 67,3%. Untuk total rata-rata keseluruhan akurasi adalah 72,4%.

Camera mouse berusaha menggantikan perangkat mouse yang telah biasa dipergunakan. Melalui kamera gerakan serta aksi mouse dapat diatur oleh pengguna melalui beberapa isyarat tangan yang telah ditentukan sebelumnya. Berikut ini adalah hasil perbandingan input menggunakan mouse dan input dengan isyarat tangan.

#### IV. KESIMPULAN

Berdasarkan perancangan dan hasil analisis yang dilakukan dalam penelitian, dapat disimpulkan bahwa dengan metode convexhull dapat mendeteksi objek tangan dengan baik serta dapat mengenali isyarat tangan sesuai dengan yang telah diperintahkan dengan tingkat akurasi sebesar 68 % dari 75 kali percobaan. Dalam hal natural user interface mouse dengan input isyarat tangan berbeda dari kebiasaan / perilaku menggunakan hardware mouse sehingga masih belum lebih baik / semudah daripada menggunakan hardware mouse serta gerakan cursor mouse yang masih belum lembut. Metode convexhull ini dapat melakukan deteksi gerakan dan isyarat tangan dengan arah jari yang keatas agar posisi telapak tangan jelas dan tidak ada yang tertutupi. Apabila letak kamera diatas laptop dan cahaya datang dari depan laptop, maka jari bisa saja terlindung dari cahaya yang ada sehingga pendeteksian masih sulit dilaksanakan.

Beberapa saran dalam hal ini adalah mengenai penggunaan alas mouse / background disarankan menggunakan kain yang tidak memantulkan cahaya / tidak ada bayangan dari objek lain, karena pada kain yang berwarna putih sudah terdapat banyak bayangan dari tangan. Hal ini dapat mengganggu pendeteksian tangan. Aplikasi masih sulit membedakan obyek dengan background yang memiliki warna hampir sama dengan kulit dan background yang tidak polos (bermotif). Masih perlu latihan terlebih dahulu kepada user dari camera mouse untuk kemudahan menggunakannya. Diharapkan agar penelitian ini dapat dilanjutkan sebagai suatu teknologi terbaru yang dapat dipergunakan dengan mudah dalam berbagai macam aplikasi untuk menggantikan teknologi *touchscreen* menjadi pengendali jarak jauh.

#### DAFTAR PUSTAKA

- [1] Abhik Banarjee, "Mouse Control Using a Web Camera Based on Colour Detection," *International Journal of Computer Trends and Technology (IJCTT)*, pp. 15-20, 2014.
- [2] Nyoman Hermawan, "Purwarupa Mouse Keypad Nirkabel Akselerometer Berbasis Mikrokontroler," *IJEIS*, 2012.

- 
- [3] Vivek Veeriah J, "Robust Hand Gesture Recognition Algorithm for Simple Mouse Control," *International Journal of Computer and Communication Engineering*, pp. 219-221, 2013.
- [4] Jishmi Jos Choondal, "Design and Implementation of a Natural User Interface Using Hand Gesture Recognition Method," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, pp. 249-254, 2013.
- [5] G. R. S. Murthy, "A Review of Vision Based Gestures Recognition," *International Journal of Information Technology and Knowledge Management*, pp. 405-410, 2009.
- [6] Mariam-Avram VINCZE, "Hand Gesture Mouse Cursor Control," *Scientific Bulletin of The Petru Maior "University of Tirgu Mures*, pp. 46-49, 2014.
- [7] Rafiqul Zaman Khan, "Hand Gesture Recognition: A Literature Review," *International Journal of Artificial Intelligence & Applications (IJAA)*, pp. 161-174, 2012.
- [8] Gary and Adrian Kaehler Bradski, "Learning OpenCV," *O'Reilly Media, Inc.*, 2008.
- [9] Rudy Hartanto, "Perancangan Awal Antarmuka Gesture Tangan Berbasis Visual," *JNTETI*, pp. 36-43, 2012.
- [10] Menatoallah M Youssef, "Hull Convexity Defect Features for Human Action Recognition," *Dissertation, The School of Engineering of the University of Dayton*, 2011.
- [11] Intel, "Open Source Computer Vision Library," *Intel Corporation*, 2000-2001.
- [12] Hojoon Park, "A Method for Controlling Mouse Movement Using Real-Time Camera," 2010.
- [13] C. A. Poynton. (1997) Frequently Asked Questions about Color. [Online]. <http://www.poynton.com/PDFs/ColorFAQ.pdf>
- [14] M. R. Tabassum, "Comparative Study of Statistical Skin Detection Algorithms for Sub-Continental Human Images," *Information Technology Journal*, pp. 811-817, 2010.
- [15] Khamar Basha Shaik, "Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space," in *Procedia Computer Science*, 2015.
- [16] Thomas Royal, *A High Level Description of Two Fingertip Tracking Techniques: k-curvature and convexity defects.*, 2016.