

# Rancang Bangun Pengontrol Presentasi Berbasis Slide dengan Teknik Analisis Gerakan Jari dan Tangan

Maisevli Harika<sup>1</sup>, Diena Rauda Ramdania<sup>2</sup>

<sup>1</sup>Dept. of IT Convergence and Engineering,  
Pukyong National University, Rep. of Korea

45 Yongso-ro Daeyeon 3(sam)-dong, Nam-gu, Busan, South Korea.

<sup>2</sup>Jurusan Teknik Informatika, Fakultas Sains dan Teknologi  
Universitas Islam Negeri Sunan Gunung Djati Bandung

Jl. A.H. Nasution 105, Bandung 40614 Indonesia

<sup>1</sup>maisevli@pukyong.ac.kr, <sup>2</sup>dienarauda@pukyong.ac.kr

**Abstrak-** Gerakan tangan mampu membantu proses berpikir karena memberikan gambaran visual pada pembicara. Memanfaatkan gerakan untuk mengendalikan presentasi berbasis *slide* merupakan hal yang bermanfaat dan memudahkan pembicara. Analisis gerakan tangan dalam sinar proyektor sulit dilakukan karena artikulasi tangan, *occlusion*, dan pencahayaan yang berubah-ubah. Pendekatan yang diajukan dalam jurnal ini didesain untuk mengendalikan presentasi berbasis *slide* dan *pointer* dengan tangan kosong.

**Kata kunci-** *computer vision*, deteksi jari, deteksi tangan, HCI, kendali gerakan tangan, *slide* presentasi.

## I. PENDAHULUAN

Pernahkan anda melihat seorang presenter yang berbicara sambil menggerakkan tangan? Atau anda adalah orang tersebut? Penelitian menunjukkan, gerakan anggota tubuh dapat membantu proses berpikir seseorang ketika berbicara. Hal ini disebabkan karena gerakan tubuh dapat membantu memberikan petunjuk visual[4].

Ketika melakukan presentasi, waktu yang paling efektif untuk tetap menarik perhatian *audiens* adalah sekitar 15 menit. Beberapa tips dan teknik presentasi telah banyak disebar luaskan guna menjaga perhatian *audiens*. Tak ketinggalan, teknologi yang canggih juga turut meramaikan kebutuhan pada saat presentasi, misalnya *remote control*, *pointer*, dan kinetik *control*.

Kemudahan dalam menggunakan teknologi memang menjadi kebutuhan manusia. Namun, kendala teknis terkadang tak dapat dipungkiri. Seringkali *presenter* justru kehilangan waktu yang berharga ketika harus melakukan *setting* alat sebelum presentasi. Hal ini mengundang minat peneliti untuk memanfaatkan anggota tubuh manusia pada saat presentasi, misalnya tangan dan jari.

Kemampuan untuk melakukan sesuatu hanya dengan gerakan jari merupakan impian manusia sejak dulu. Dewasa ini, impian tersebut menjadi mungkin untuk diterapkan dengan memanfaatkan teknologi *computer vision*.

Menggunakan gerakan jari dan tangan untuk mengendalikan *pointer* dan navigasi *slide* diharapkan memudahkan pembicara yang menggunakan *slide* dalam presentasinya. Tulisan ini mempersembahkan desain dan implementasi dari analisis gerakan jari dan tangan menggunakan *webcam* untuk mengendalikan *pointer* dan navigasi *slide* presentasi.

## II. TINJAUAN PUSTAKA

### A. HSL COLOR MODEL

HSL adalah kependekan dari *Hue*, *Saturation* dan *Lightness*. Ketiga istilah memiliki hubungan dengan warna. *Hue*, *Saturation*, *Brightness* adalah aspek dari skema warna *Red*, *Green*, *Blue* (RGB).

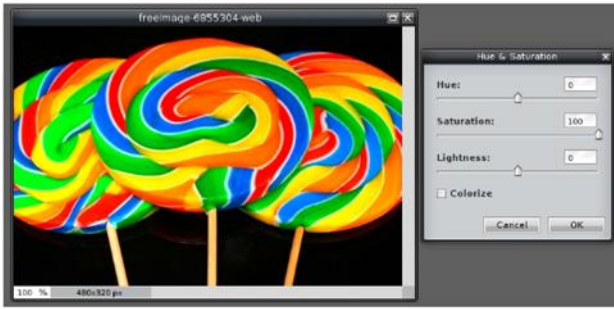
*Hue* adalah istilah yang digunakan untuk menunjukkan suatu warna-warna dasar seperti merah, biru, hijau, kuning, coklat dan sebagainya. Perbedaan antara warna adalah perbedaan dalam *hue*. Sehingga *hue* adalah penyebutan lain dari warna itu sendiri.

*Saturation* adalah intensitas dari *hue*. Warna-warna dasar yang terang adalah warna dengan intensitas *saturation* tinggi. Sedangkan warna pastel/muda adalah warna dengan intensitas *saturation* rendah. Warna *monochrome* (hitam putih) sama sekali tidak mempunyai *saturation*.

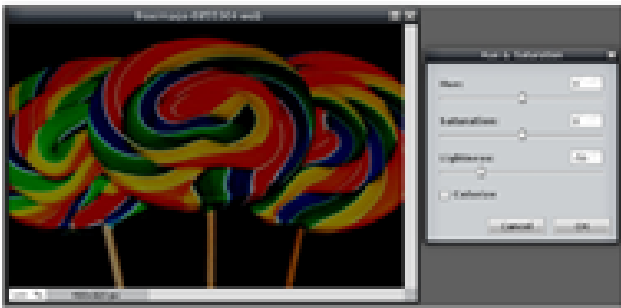
*Lightness* adalah ketajaman dari suatu warna ataupun tingkat hitam atau putih dalam skala warna. Istilah ini juga dikelan dengan *value* atau *tone*[5]. Gambar berikut ini mengilustrasikan perbedaan *Hue*, *Saturation* dan *Lightness*.



Gambar 1. *Hue* mengubah warna dasar menjadi biru.



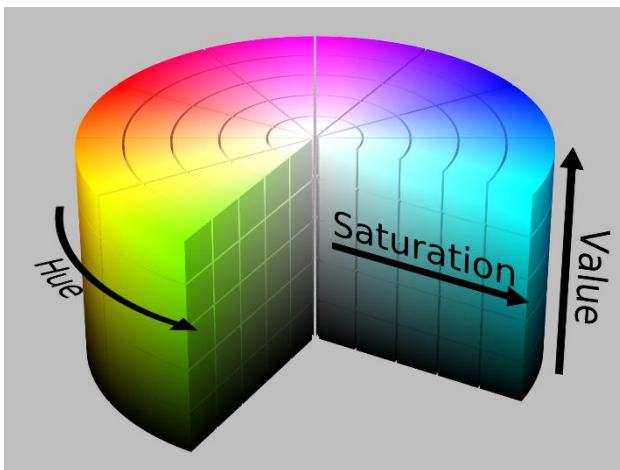
Gambar 2. Saturation.



Gambar 3. Lightness.

**B. HSV COLOR MODEL**

Model HSV pertama kali diperkenalkan oleh A.R Smith pada tahun 1978. HSV merupakan kependekan dari *Hue*, *Saturation*, dan *Value*. Value dapat diartikan sebagai kecerahan dari warna. Nilainya berkisar antara 0-100 %. Semakin besar nilai maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut. Apabila nilainya 0 maka warnanya akan menjadi hitam[7]. Gambar 4 mengilustrasikan HSV.



Gambar 4. HSV.

**C. KALMAN FILTER**

Kalman Filter pertama kali diperkenalkan oleh Rudolf E. Kalman pada tahun 1960. Metode ini merupakan sebuah perhitungan matematis. Secara singkat, metode ini berfungsi mengolah data yang mengandung noise menjadi data yang lebih sedikit mengandung noise. Pengolahan data tersebut mengakibatkan adanya waktu tunda (delay)[7].

**III. PENELITIAN TERKAIT**

Dewasa ini, telah banyak riset berkaitan dengan mengenali gerakan tubuh. Beberapa teknik telah ditemukan dan masing-masing memiliki pro dan kontra tersendiri. Hal ini wajar karena mengenali gerakan jari dan tangan membawa sejumlah masalah yang signifikan dan sulit untuk diselesaikan dengan algoritma yang ada sampai saat ini.

Salah satu metode yang diajukan adalah dengan menggunakan *color detection* untuk mengenali jari dan tangan. Teknik *skin color detection* digunakan oleh Rajesh *et al*[1] dan Jadhav *et al*[2] untuk mengendalikan navigasi presentasi berbasis *slide*. Rajesh menggunakan HSV *color model* sedangkan Jadhav *et al* menggunakan HSL *color model*. Perbedaan yang paling mendasar adalah penggunaan Kalman *filter* untuk membantu menghaluskan gerakan *pointer* dan HSL *color model* dalam teknik *adaptive skin color detection*. Sedangkan kedua tim lainnya menentukan ambang batas warna kulit yang telah ditentukan untuk proyek mereka. Tabel 1 berikut adalah matriks perbandingan metode yang digunakan dengan penelitian yang terkait.

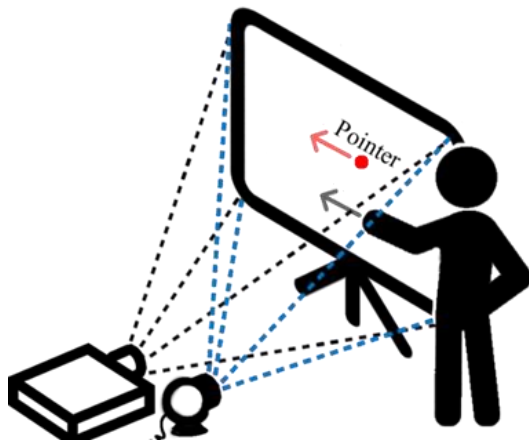
Tabel 1. Matriks perbandingan sistem

Peneliti	Alat	Posisi kamera	Metode segmentasi tangan	Metode kendali slide
Ram Rajesh <i>et al</i>	Webcam	Telapak tangan sejajar dengan kamera	Static skin color dalam HSV <i>color model</i>	Menghitung jumlah jari yg berdiri
Dnyanada Jadhav <i>et al</i>	Webcam	Telapak tangan sejajar dengan kamera	Static skin color dalam HSL <i>color model</i>	Menggunakan centroid tangan untuk menggerakkan pointer dan mengganti slide.
Darko Martinovi kj <i>et al</i>	Microsoft Kinect	Seluruh badan sejajar dengan kamera	Skeleton data stream	Deteksi gerakan dengan pemetaan 3D
Proposed research	Webcam atau built in laptop camera	Kamera menghadap ke layar monitor	Adaptive skin color dari sampel awal	Mengendalikan pointer dengan satu jari dan centroid untuk mengganti slide

**IV. PERANCANGAN**

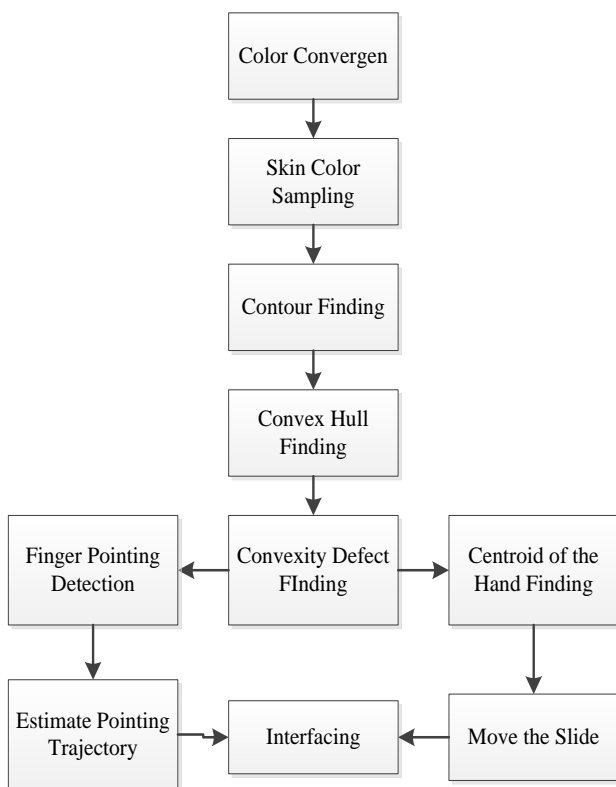
**A. DESKRIPSI SISTEM**

Sistem didesain menggunakan webcam atau built-in kamera laptop. Kamera diletakkan didepan layar. Ketika melakukan presentasi, pembicara haruslah membuat gerakan tangan di depan layar untuk mengendalikan slide. Gerakan tersebut haruslah sejajar dengan layar agar gerakan dapat dianalisis dengan baik. Ketika kamera menangkap gerakan dalam cahaya proyektor, analyzer akan menganalisis gerakan tersebut dan mengeluarkan perintah ke aplikasi sesuai dengan gerakan tersebut. Konfigurasi sistem dapat dilihat pada Gambar 5 dibawah ini.



Gambar 5. Konfigurasi sistem.

Sistem terdiri dari pengambilan sampel warna untuk kulit, deteksi ujung-ujung jari, mengenali gerakan. Gambar 6 memperlihatkan pengorganisasian sistem.



Gambar 6. Organisasi sistem

**B. IMAGE PROCESSING**

Image processing digunakan hampir pada keseluruhan sistem dimulai dari pengambilan sampel warna kulit, penentuan ujung-ujung jari dan tangan dengan *contour detection*, serta *convexity defect*.

Pengambilan warna kulit dimulai dengan kalibrasi warna kulit yang terkena efek dari cahaya proyektor. Sampel warna digunakan untuk menentukan ambang batas dari warna kulit untuk modul selanjutnya. Untuk menentukan ambang batas perlu ditentukan *pixel* dengan warna kulit dan bukan warna kulit. Proses penentuan warna kulit ini dapat dilihat pada algoritma 1 *Color sampling*.

**Algorithm 1** *Color Sampling* (input: color image, output: binary image)

1. *Initiation*: perform Equation (1), and (2)
2.  $src = input, dst = output$  at time  $t$ ,
3. **for**  $t = 1$  to 30
4. convert RGB to HSL
5.   **for**  $k = 0$  to 5
6.     define a square ROI
7.     get  $(src.h, src.s, src.l)$  values in ROI //HSL
8.   **for** each HSL channel
9.     get the median
10.  $avg +=$  the median values
11. **end**
12. generate  $dst$  vector from  $avg$
13. **end**
14. **End**

Pengambilan sampel warna dilakukan dalam waktu tiga puluh frame, selama pengambilan sampel sistem mengambil beberapa sampel warna, menghitung rata-rata dari masing-masing variabel H, S, dan L pada setiap ROI (*Region of Interest*), kemudian menentukan nilai ambang batas tertinggi dan terendah untuk setiap variabel. Pengambilan sampel warna dapat dilihat pada gambar dibawah ini.



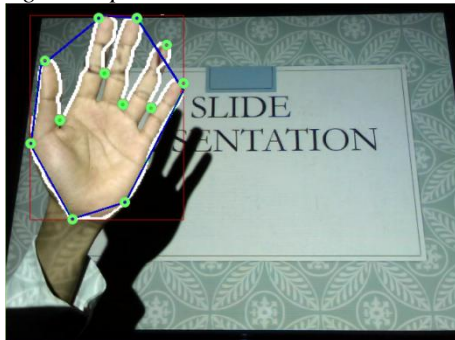
Gambar 6. Pengambilan Sampel Warna.

Proses dilanjutkan dengan me-negatifkan warna-warna lain yang dianggap bukan warna kulit (gambar biner). Batas-batas antara gambar biner tersebut di-extract menggunakan algoritma K[3], *contour finding* dan didapatkan tangan (*hand detection*).



Gambar 7. *Input* (kiri atas), *input* dalam HSL (kanan atas), dan gambar biner (bawah).

Pada *contour* yang didapat diterapkan teknik *convex hull* dan *convexity defect* untuk mendapatkan jari. Hasil dari teknik inilah yang selanjutnya diterapkan untuk *interfacing desktop*.



Gambar 8. *Extracted Contour* dan *Contour Points*.

Navigasi *slide* dan mengendalikan *pointer* dapat dilakukan dengan memanfaatkan titik-titik *contour* yang didapat pada proses implementasi *convex hull* dan *convexity defect*. Dari titik-titik tersebut dapat ditentukan ujung-ujung jari. Jumlah jari-jari ini digunakan untuk memicu navigasi dan *pointing*, untuk lebih jelasnya dapat diperhatikan pada algoritma berikut ini.

**Algorithm 2** *Gesture analysis and finger pointing*

1. Initialization: Algorithm 1
2. **While** button *q* not press
3. Capture the images
4. Hand segmentation
5. Get fingertips ( $Fg_n$ )
6. **If** fingertips = 1
7. Get fingertips coordinate (*x*, *y*), perform Equation 3&4
8. Convert the equation 4 result with equation 7
9. **Else if** fingertips = 5
10. Get the centroids, perform equation 5, 6
11. Get centroid coordinate (*x*, *y*) set as initiation
12. **If** *x* initiation coordinate < current coordinate
13. Sends key move the slide forward
14. **Elseif** *x* initiation coordinate > current coordinate
15. Sends key move the slide backward
16. **End if**
17. **End if**
18. **End if**
19. **End if**
20. **End**

**C. NAVIGASI SLIDE**

Navigasi memanfaatkan deteksi jari dan *centroid* dari tangan. Navigasi *slide* dipicu oleh terdeteksinya lima jari dan arah perpindahan *centroid* menjadi penentu *slide* digerakkan maju atau mundur.

*Centroid* dideteksi saat tangan dibuka dan kelima jari terdeteksi. Titik dimana *centroid* kali pertama terdeteksi ditetapkan sebagai titik awal, ketika *centroid* bergerak titik terakhir akan dibandingkan dengan titik awal, nilai dari perbandingan ini dijadikan tolak ukur untuk menggeser *slide*. *Centroid* ditentukan menggunakan

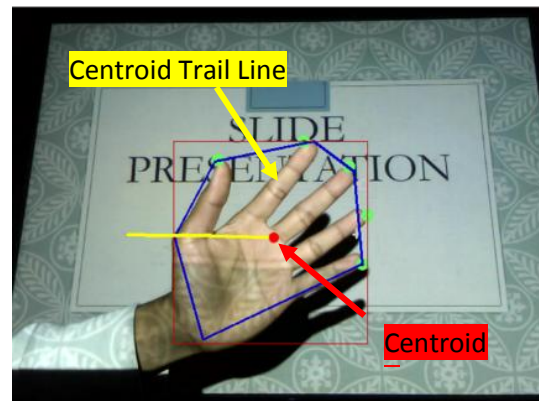
*moment*, penentuan *moment* dari titik-titik *contour* erupakan salah satu teknik *image processing*.

Penulis menghitung *moment* menggunakan formula:

$$hand\_centroid = \left( \frac{\mu_{1,0}}{\mu_{0,0}}, \frac{\mu_{0,1}}{\mu_{0,0}} \right) \quad (1)$$

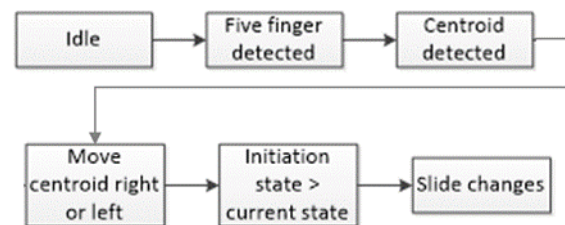
Koordinat-koordinat tersebut didapat dari rata-rata koordinat *x* dan *y*.

$$\mu_{1,0} = \frac{sum_x}{\mu_{0,0}} \text{ and } \mu_{0,1} = \frac{sum_y}{\mu_{0,0}} \quad (2)$$



Gambar 9. Menggerakkan Navigasi *Slide* dengan *Centroid*.

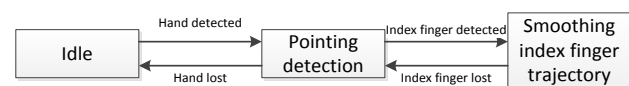
Lebih ringkasnyacara kerja navigasi *slide* presentasi dapat dilihat pada Gambar 10 berikut.



Gambar 10. Skenario untuk menggerakkan *slide* maju atau mundur.

**D. KENDALI POINTER**

Untuk mengendalikan skenario yang dirancang adalah menggunakan satu jari, tujuan untuk mengendalikan *pointer* ini adalah sebagai *highlight* pada bahasan yang dibahas seperti penggunaan cahaya merah pada *pointing device*. Pada gambar berikut ditampilkan diagram alur kendali *pointer*.



Gambar 11. Diagram alur pengendalian *pointer*

Gerakan tangan, karena efek pencahayaan dibutuhkan *Kalman filter* untuk memuluskan pergerakan *pointer*, karena pergerakan yang tertangkap akan *shaky* karena pengaruh *noise* dan pencahayaan. Kami menentukan gerakan tangan sebagai input dan rumus

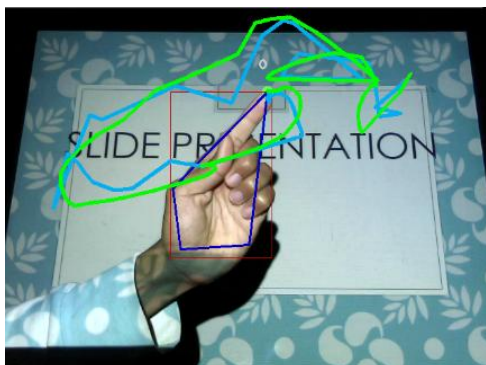
*smoothing trajectory* menggunakan *Kalman filter* adalah sebagai berikut.

$$y_t = Bx_t + v_t \tag{3}$$

dengan  $B$  merupakan *matrix* opservasi dari vektor  $x_t$  dan  $v_t$  digunakan untuk mengukur *noise* pada waktu  $t$ . Vektor opservasi  $y_t$  merupakan posisi jari yang diinginkan,  $x_{t+1}$  digunakan untuk *smoothing trajectory* dari posisi jari, rumus sebagai berikut.

$$x_{t+1} = Ax_t + w_t \tag{4}$$

$A_x$  merupakan pemodelan yang digunakan untuk prediksi ditambah *Gaussian white noise process*  $w_t$  untuk menentukan nilai  $x$  pada waktu  $t+1$ .



Gambar 12. *Finger pointing trajectory and estimation trajectory.*

**E. INTERFACING**

Proses terakhir adalah membangun antarmuka atau *interfacing*. Kita harus membangun antarmuka antara *vision system* dan perangkat lunak Powerpoint. Perintah-perintah dari *vision system* dikirim ke perangkat lunak untuk presentasi ini menggunakan *sends key*, salah satu fungsi dari Microsoft Windows untuk mengirim input ke sistem operasinya.

Koordinat *pointing gesture* akan sangat berbeda dan dibutuhkan penyesuaian pada *desktop*, untuk memetakan lokasi jari ke ukuran monitor presentasi digunakan fungsi berikut.

$$f: p \rightarrow d = \left( \frac{Px}{dx} W, \frac{Py}{dy} H \right) \tag{5}$$

Dengan,  $p$  merupakan posisi *pointer* pada *desktop* presentasi,  $d$  posisi *pointing* pada sistem,  $P_x$  dan  $P_y$  merupakan koordinat pada sistem,  $dx$  dan  $dy$  merupakan dimensi tampilan, dan  $W$  dan  $H$  merupakan dimensi pergerakan tangan pada sistem.

**V. PERCOBAAN DAN ANALISIS**

Hasil evaluasi dari sistem digunakan untuk mengukur performa dari sistem. Beberapa percobaan telah dilakukan untuk mengukur kemampuan dari sistem. Analisis sistem ini dilakukan untuk menentukan apa saja yang dibutuhkan untuk meningkatkan kemampuan dari sistem untuk penelitian lanjutan.

Analisis pertama adalah untuk mengukur kemampuan sistem dalam menangkap warna kulit, analisis kedua adalah mengukur seberapa akurat sistem menangkap satu jari. Selanjutnya penelitian untuk mengukur keakuratan

sistem untuk menggeser slide ke kiri atau ke kanan. Percobaan-percobaan dilakukan oleh 5 orang sampel yang diambil secara acak.

**A. DETEKSI WARNA KULIT**

Penelitian pertama ini untuk mengukur kemampuan deteksi warna kulit.

Tabel 2. Hasil Deteksi warna kulit.

User	Percobaan 1		Percobaan 2		Percobaan 3		Success Rate
	Success	Fail	Success	Fail	Success	Fail	
1	41	9	27	23	39	11	72.4 %
2	39	11	31	19	37	13	
3	45	5	37	13	44	6	
4	41	9	32	18	40	10	
5	40	10	29	21	36	14	

**B. DETEKSI JARI POINTING**

Percobaan ini dilakukan untuk mengukur tingkat kesuksesan deteksi *pointing gesture* dengan menghitung jumlah *frame* dimana *pointing gesture* terekam dalam satu menit.

Tabel 3. Hasil pengujian deteksi *pointing gesture*.

User	Frames	Total Fingertips	Failed Fingertip	Accuracy
1	1739	1306	433	74.0%
2	1693	1254	439	
3	1823	1353	470	
4	1984	1385	599	
5	2074	1595	479	

**C. NAVIGASI SLIDE**

Tingkat kesuksesan mengendalikan *slide* menggunakan gerakan tangan diukur dengan keberhasilan menggeser *slide* maju dan mundur sebanyak 50 kali.

Tabel 4. Hasil pengujian navigasi *slide*.

Trial	Move Forward		Move Backward		Accuracy	
	Success	Fail	Success	Fail	Forward	Backward
1	39	11	42	8	76.4%	77.2%
2	40	10	38	12		
3	37	13	38	12		
4	38	12	37	13		
5	37	13	38	12		

**D. KENDALI POINTER**

Penelitian dilakukan dalam waktu 100 detik dengan status keberhasilan dinilai dari tidak hilangnya *pointer* saat *pointing*.

Tabel 5. Hasil pengujian *pointer pointing*.

Trial	Success	Fail	Success rate
1	89	11	80.0%
2	78	23	
3	84	16	
4	61	39	
5	88	12	

**VI. KESIMPULAN DAN SARAN**

Berdasarkan penelitian yang dilakukan, terbukti metode yang digunakan mampu untuk menggantikan peralatan antarmuka yang biasa digunakan dalam

presentasi (*mouse* dan *keyboard*), tanpa memandang warna kulit dan ras penggunanya.

Penulis menyarankan untuk meningkatkan kemampuan aplikasi yang menggunakan teknologi *computer vision* dengan memanfaatkan *background subtraction* bersamaan dengan metode yang digunakan saat ini mungkin mengatasi masalah dengan presentasi yang kaya akan warna. *Background-subtraction* mengambil *foreground* dari *background-area*. Dengan memanfaatkan hal ini akan didapat *foreground* dari gerakan tangan. *Foreground* ini mencakup siluet tangan yang bergerak dan bayangan tangan, karena bayangan tangan juga menjadi dominan di dalam sinar proyektor. Karena hal inilah metode *background-subtraction* masih membutuhkan metode lain misalnya *color-detection*. Selanjutnya penambahan skenario untuk fitur aplikasi seperti menggambar *shape*, *free draw* atau menulis dengan gerakan tangan merupakan hal yang menarik untuk dilakukan.

## VII. REFERENSI

- [1] Rajesh, Ram., Nagajunan. Arunachalam. Aarthi. (2012). Distance Transform Based Hand Gestures Recognition for Powerpoint Presentation Navigation. *Proceedings of the Advanced Computing: An International Journal*, vol.3, pp 41 – 48.
- [2] Jadhav, Dnyanda. dan L.M.R.J. Lobo. (2015). *Proceedings International Journal of Science and Research (IJSR)*, vol. 4, pp 883 – 837.
- [3] Suzuki, S., dan Abe, K., (1985). Topological Structural Analysis of Digitized Binary Images by Following, *Computer Vision, Graphics, and Image Processing*, vol. 30., pp 32 – 46.
- [4] Dcience Daily, (2011). Gerakan anggota tubuh saat berbicara bantu proses berpikir. <http://nationalgeographic.co.id/berita/2011/01/gerakan-anggota-tubuh-saat-berbicara-bantu-proses-berpikir>, diakses tanggal: 14 Desember 2016.
- [5] Al-Arkam, (2015). Mengenal Hue, Saturation, dan Lightness. <http://mbah4us.blogspot.co.id/2015/12/mengenal-hue-saturation-lightness.html?m=1>, diakses tanggal: 12 Desember 2016.
- [6] Abd Fauzan (2015). Ruang Warna HSV serta Konversinya. <http://www.charisfauzan.net/2015/01/ruang-warna-hue-saturation-value-hsv.html>, diakses tanggal: 12 Desember 2016.
- [7] Greg Welch, Gary Bishop, (2001), An Introduction to The Kalman Filter, Course 8. [http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001\\_CoursePack\\_08.pdf](http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf). Diakses tanggal: 10 Desember 2015.