



## Pengenalan Pose Tangan Menggunakan *HuMoment*

Dina Budhi Utami<sup>1</sup>, Muhammad Ichwan<sup>2</sup>

<sup>1,2</sup>Jurusan Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Nasional

<sup>1,2</sup>Jalan PH.H. Mustafa No. 23, Bandung, Jawa Barat 40124, Indonesia

Email korespondensi: [dinabudi@itenas.ac.id](mailto:dinabudi@itenas.ac.id)

Dikirim 31 Januari 2017, Direvisi 8 Februari 2017, Diterima 13 Februari 2017

Abstrak – *Computer vision* yang didasarkan pada pengenalan bentuk memiliki banyak potensi dalam interaksi manusia dan komputer. Pose tangan dapat dijadikan simbol interaksi manusia dengan komputer seperti halnya pada penggunaan berbagai pose tangan pada bahasa isyarat. Berbagai pose tangan dapat digunakan untuk menggantikan fungsi *mouse*, untuk mengendalikan robot, dan sebagainya. Penelitian ini difokuskan pada pembangunan sistem pengenalan pose tangan menggunakan *HuMoment*. Proses pengenalan pose tangan dimulai dengan melakukan segmentasi citra masukan untuk menghasilkan citra ROI (*Region of Interest*), yaitu area telapak tangan. Selanjutnya, dilakukan proses deteksi tepi. Kemudian, dilakukan ekstraksi nilai *HuMoment*. Nilai *HuMoment* dikuantisasikan ke dalam bukode yang dihasilkan dari proses pelatihan menggunakan K-Means. Proses kuantisasi dilakukan dengan menghitung nilai *Euclidean Distance* terkecil antara nilai *HuMoment* citra masukan dan bukode. Berdasarkan hasil penelitian, nilai akurasi sistem dalam mengenali pose tangan adalah 88,57%.

Kata kunci – Pengenalan Pose Tangan, *Computer Vision*, *HuMoment*, *Depth Image*, *Kinect*

Abstract - *Computer vision* based on shape recognition open up a new possibility for future human-computer interaction. Hand poses can be used as a symbol of human interaction with computers as well as sign language. Various hand poses can be used to replace the mouse, motion control robots, and others. This study focused on the development of a hand pose recognition system using *HuMoment* quantization. The process begins with image segmentation to generate the image ROI (*Region of Interest*). Then the edge detection process is performed. After that, the value of *HuMoment* is extracted. The *HuMoment* value quantized into codebook which is resulted from the training process using K-Means. Quantization process is done by calculating the smallest *Euclidean Distance* between the *HuMoment* of input image and codebook. Based on this research, the recognition accuracy is 88.57%.

Keywords - Hand Pose Recognition; *Computer Vision*; *HuMoment*; *Depth Image*; *Kinect*

### I. PENDAHULUAN

Dengan semakin meningkatnya penggunaan komputer dalam kehidupan sehari-hari, maka muncul tantangan baru bagaimana memberikan kenyamanan dan kemudahan dalam menggunakan komputer, salah satunya adalah kebutuhan akan metode interaksi dan komunikasi yang lebih alami antara pengguna dan komputer.

Penggunaan pose tangan merupakan salah satu solusi untuk memenuhi kebutuhan manusia dalam interaksi manusia dan komputer (*Computer Human Interaction*) yang lebih intuitif, cepat, dan sejalan dengan fungsi natural anggota tubuh manusia. Tangan dapat menghasilkan beragam pose. Setiap pose

memiliki maksud dan makna tersendiri sesuai dengan kesepakatan umum ataupun kesepakatan diantara yang melakukan komunikasi. Oleh karena itu pose tangan dapat digunakan untuk menggantikan fungsi *mouse*, untuk mengontrol gerak robot, dan sebagainya.

*Depth-sensing* dan *gesture-tracking* seperti *Microsoft Kinect* atau *Leap Motion Controller* merupakan contoh inovasi teknologi dalam metode interaksi manusia dan komputer. Teknologi tersebut memberikan peluang kepada pengguna untuk berinteraksi dengan komputer menggunakan gerakan tangan atau badan. Namun, sistem pengenalan gerakan yang digunakan berdasarkan pada posisi skeleton bukan pose atau bentuk.

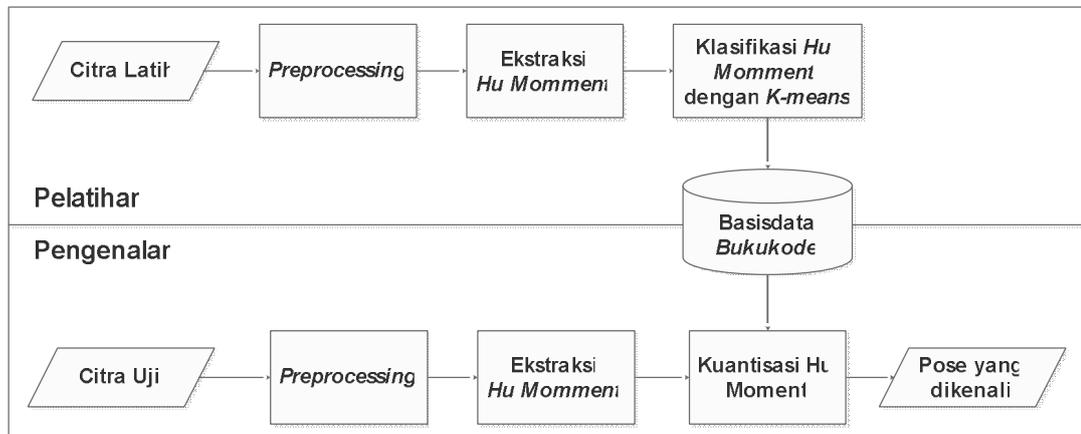
Selain teknologi-teknologi tersebut, penelitian mengenai penggunaan tangan dalam interaksi manusia dan komputer serta penelitian mengenai pengenalan pose tangan telah banyak diaplikasikan pada beberapa jurnal ilmiah diantaranya yaitu penelitian mengenai pengenalan gerakan tangan untuk mengendalikan robot tangan menggunakan *color glove* [1]. Penelitian lain yang sejenis adalah pengenalan gerakan tangan untuk mengendalikan robot menggunakan *depth camera* [2]. Citra yang dihasilkan dari *depth camera* diolah dan kemudian diklasifikasikan menggunakan algoritma *Support Vector Machine* (SVM). Selain itu, terdapat penelitian mengenai penggunaan tangan sebagai interaksi alamiah dengan *Augmented-Reality Interfaces* [3] dan penelitian mengenai pengenalan bahasa isyarat yaitu pengenalan American Sign Language (ASL) berdasarkan nilai estimasi pose tangan dari *depth sensor* secara *real time* [4]. Penggunaan *Kinect* dalam penelitian telah banyak dilakukan diantaranya penelitian mengenai metode untuk mendeteksi tangan dan gerakan tangan menggunakan *depth sensor Kinect* [5][6], penelitian untuk memperkirakan pose kepala secara *real time* menggunakan *depth sensor* [7], dan penelitian mengenai skema untuk mengekstraksi tangan dari *depth image* [8].

Penelitian ini difokuskan pada pembangunan sistem pengenalan pose tangan berdasarkan *depth image* yang dihasilkan *Kinect*. Perbedaan dengan

penelitian sebelumnya adalah sistem pengenalan gerakan tangan berdasarkan pada bentuk pose tangan bukan pada posisi skeleton. Selain itu, sistem pengenalan gerakan tangan tetap dapat mengenali pose tangan walaupun pose tangan berbeda posisi dan arah rotasi. Pada penelitian ini, sistem pengenalan pose tangan menggunakan nilai *HuMoment*. Nilai *HuMoment* dipilih berdasarkan pertimbangan bahwa nilai *HuMoment* invarian terhadap skala ukuran objek dan rotasi [9]. Penggunaan *HuMoment* telah digunakan dalam berbagai penelitian dalam area pengenalan objek, salah satunya adalah penggunaan *HuMoment* dan *Backpropagation* untuk mengenali objek *scattered* dalam citra [10]. Seperti penelitian-penelitian yang telah dilakukan, citra masukan dalam penelitian ini adalah *depth image* yang didapat dari *depth sensor Kinect*. Seluruh citra melalui proses pengolahan citra, ekstraksi ciri, dan pengenalan untuk mengenali bentuk pose tangan.

## II. METODE PENELITIAN

Masukan pada sistem pengenalan pose jari tangan adalah *depth image* yang didapat dari *depth sensor Kinect*. Dari citra tersebut kemudian diambil nilai *HuMoment* telapak tangan. Kemudian nilai *HuMoment* dikuantisasi untuk menghasilkan keluaran yaitu pose yang dikenali. Pada sistem pengenalan pose jari tangan, terdapat 2 alur proses, yaitu alur proses pelatihan dan alur proses pengenalan seperti pada Gambar 1.



Gambar 1. Diagram Alir Proses Pelatihan Dan Pengenalan Pose Tangan

Proses pelatihan merupakan proses mengumpulkan data sampel pose tangan dan kemudian disimpan sebagai referensi untuk proses pengenalan. Proses pelatihan dimulai dengan mengumpulkan data sampel. Setiap citra data sampel melalui proses *preprocessing* untuk memperbaiki kualitas citra dan mempersiapkan citra agar sesuai dengan kebutuhan proses ekstraksi ciri. Kemudian dilakukan ekstraksi nilai *HuMoment*. Selanjutnya nilai *HuMoment* tersebut diklasifikasikan dengan menggunakan algoritma *K-Means*. Proses klasifikasi tersebut menghasilkan bukukode (*codebook*) setiap *cluster* yang kemudian disimpan

dalam basisdata. Nilai bukukode ini digunakan pada proses pengenalan.

Proses pengenalan dimulai dengan mengambil citra data uji. Citra data uji melalui proses *preprocessing* untuk memperbaiki kualitas citra dan mempersiapkan citra agar sesuai dengan kebutuhan proses ekstraksi ciri. Kemudian dilakukan ekstraksi nilai *HuMoment*. Selanjutnya nilai *HuMoment* dikuantisasi terhadap bukukode (*codebook*) setiap *cluster* yang telah disimpan dalam basisdata. Proses kuantisasi dilakukan dengan menghitung kedekatan nilai *HuMoment* dengan setiap buku kode. Proses kuantisasi menghasilkan prediksi pose tangan.

### A. Pengambilan Citra

Pada sistem pengenalan pose tangan, masukan berupa *depth image* (Gambar 2) dari *Kinect Xbox One*. Pengambilan *depth image* dilakukan oleh *Kinect* dengan menggunakan *Library OpenNI*. Pada sensor *Kinect* terdapat 3 kamera yang terdiri dari sebuah kamera RGB dan dua buah kamera *infrared*. Kamera RGB akan menghasilkan citra RGB sedangkan kamera *infrared* akan menghasilkan *depth image*. Dalam *Library OpenNI* ini terdapat fungsi pengambilan *depth image* dan citra RGB, serta fungsi untuk menentukan posisi skeleton. Pada penelitian ini, *Kinect* hanya digunakan untuk menghasilkan *depth image* dan menentukan posisi skeleton telapak tangan (*palm*) dan pergelangan tangan (*wrist*).



Gambar 2. Contoh *Depth Image*

### B. Preprocessing

Selanjutnya, *depth image*, posisi skeleton telapak tangan (*palm*) dan pergelangan tangan (*wrist*) yang dihasilkan oleh *Kinect* digunakan sebagai masukan untuk proses *preprocessing*. *Preprocessing* berisi proses pengolahan citra dengan menggunakan *library OpenCV*. Tujuan dilakukan *preprocessing* adalah untuk mempersiapkan citra sesuai dengan yang dibutuhkan pada proses ekstraksi ciri. Pada proses *preprocessing* dilakukan segmentasi citra dan deteksi tepi.

Pertama, dilakukan segmentasi pada *depth image* (Gambar 3 atas) untuk memisahkan latar depan (*foreground*) dan latar belakang (*background*) sehingga dihasilkan citra latar depan (*foreground*) (Gambar 3 bawah). Proses segmentasi latar depan menggunakan operasi *thresholding* pada *library OpenCV*.

Dari hasil segmentasi latar depan (Gambar 4 atas), kemudian dilakukan segmentasi area ROI (*Region Of Interest*) yaitu area telapak tangan. Proses ini menghasilkan citra yang hanya memuat ROI (Gambar 4 bawah). Informasi area telapak tangan didapat berdasarkan posisi skeleton *palm* dan *wrist* yang dihasilkan *Kinect* sebelumnya.

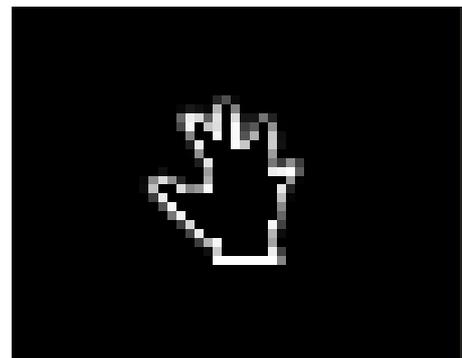
Selanjutnya, dilakukan deteksi tepi pada citra ROI. Proses deteksi tepi citra ROI menghasilkan bentuk kontur tangan seperti pada Gambar 5. Pada sistem ini, deteksi tepi dilakukan dengan metode *Canny* pada *library OpenCV*.



Gambar 3. Contoh Citra Latar Depan (*Foreground*) Hasil Segmentasi *Depth Image*



Gambar 4. Contoh Citra ROI (Bawah) Hasil Segmentasi Dari Citra Latar Depan (Atas)



Gambar 5. Contoh Tepi (Kontur) Citra ROI Tangan

### C. Ekstraksi Ciri dengan HuMoment

Ciri yang diambil dalam sistem ini adalah nilai *HuMoment* ( $\Phi$ ) yang didapat dengan menghitung momen kedua dan ketiga dari citra kontur tepi tangan. Teknik ini menghasilkan ciri yang tidak berubah terhadap perlakuan rotasi, penskalaan dan translasi (*Rotation Scale Translation (RST)-invariant*). Proses perhitungan nilai *HuMoment* ( $\Phi$ ) adalah sebagai berikut.

- a) Menghitung momen orde 0 ( $m_{00}$ ) dan momen orde 1 ( $m_{10}$  dan  $m_{01}$ ) dengan persamaan berikut:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y) \quad (1)$$

M : jumlah baris/resolusi panjang citra  
N : jumlah kolom/resolusi lebar citra  
p, q : orde momen  
x, y : kordinat piksel  
f(x, y) : intensitas piksel dititik x, y

- b) Menghitung pusat kordinat dari area atau massa ( $\bar{x}, \bar{y}$ ) dengan persamaan sebagai berikut:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (2)$$

- c) Menghitung momen pusat ( $\mu_{pq}$ ) orde 2 ( $\mu_{11}, \mu_{20}, \mu_{02}$ ) dan orde 3 ( $\mu_{21}, \mu_{12}, \mu_{30}, \mu_{03}$ ) dengan persamaan berikut:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (3)$$

- d) Menghitung normalisasi momen pusat ( $\eta_{pq}$ ) orde 2 ( $\eta_{11}, \eta_{20}, \eta_{02}$ ) dan orde 3 ( $\eta_{21}, \eta_{12}, \eta_{30}, \eta_{03}$ ) dengan persamaan berikut:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \gamma = \frac{p+q}{2} + 1 \quad (4)$$

- e) Setelah menghitung normalisasi momen pusat ( $\eta_{pq}$ ), nilai *HuMoment* ( $\Phi$ ) dihitung dengan persamaan berikut:

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} + \eta_{02})^2 + \eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \\ &\quad [(\eta_{30} + \eta_{12})^2 + 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03}) \\ &\quad (\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \\ &\quad [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30}) \\ &\quad (\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (5)$$

### D. Klasifikasi HuMoment dengan K-Means

Algoritma *K-Means* mengelompokkan satu set vektor kedalam sejumlah K kelompok atau dikenal dengan istilah *clustering*. *K-Means* digunakan pada proses pelatihan untuk mengelompokkan data sampel sehingga didapat bukukode pose tangan. Bukukode tersebut disimpan dalam basisdata untuk digunakan pada proses pengenalan. Proses pengelompokan

dengan menggunakan *K-Means* adalah sebagai berikut:

- Proses pengelompokan dimulai dengan menentukan jumlah kelompok atau cluster (K) yang diinginkan.
- Proses selanjutnya adalah menentukan *centroid* setiap *cluster* yang digunakan sebagai bukukode (*codebook*) awal. Pada iterasi awal, proses penentuan *centroid* setiap *cluster* dapat dilakukan secara acak dimana pada  $m=0$  (iterasi ke- $m$ ). Pilih sejumlah K set vektor masukan yang kemudian dijadikan *centroid* awal setiap *cluster* K.

$$y_i = x_i, \quad 1 \leq i \leq K \quad (6)$$

$y_i$  : *centroid cluster* ke- $i$

$x_i$  : vektor masukan ke- $i$

K : jumlah *cluster*

- Menghitung kemiripan dengan *Euclidean Distance* setiap vektor masukan terhadap seluruh *centroid* setiap *cluster*. Seperti pada persamaan 29.
- Mengelompokkan vektor masukan ( $x$ ) kedalam K *cluster* menggunakan aturan *nearest neighbor*. *Nearest neighbor* didapat dari kemiripan yang paling minimum.

$$x_i \in C_k \text{ jika } d_{ik} \leq d_{im}, \quad m \neq k \quad (7)$$

$C_k$  : *cluster* ke- $k$

$x_i$  : vektor masukan ke- $i$

$d_{ik}$  : nilai *Euclidean Distance* vektor masukan ke- $i$  dan *centroid cluster* ke- $k$

$d_{im}$  : nilai *Euclidean Distance* vektor masukan ke- $i$  dan *centroid cluster* ke- $m$

- Kemudian akan dilakukan pengecekan apakah terjadi perpindahan cluster dari setiap vektor masukan, jika terjadi perpindahan maka proses iterasi dilanjutkan, namun apabila tidak terjadi perpindahan cluster maka proses iterasi dihentikan dan dianggap telah mendapatkan model *cluster* yang stabil.
- Proses iterasi diulangi dengan memperbaharui nilai *centroid*. Namun untuk proses iterasi  $m \neq 1$ , nilai *centroid* dari suatu *cluster* didapat dengan menghitung nilai rata-rata (*mean*) dari seluruh vektor masukan yang dipetakan pada *cluster* tersebut.

$$x_i \in C_k \text{ jika } d_{ik} \leq d_{im}, \quad m \neq k \quad (8)$$

$y_i$  : Nilai *centroid cluster* ke- $i$

$x_i$  : Nilai vektor masukan yang berada di *cluster* ke- $i$

N : Banyaknya vektor masukan yang berada di *cluster* ke- $i$

- g) Selanjutnya ulangi proses ke-3 untuk iterasi  $m \neq 1$ .

#### E. Kuantisasi Vector *HuMoment*

Pada sistem ini dilakukan proses pemetaan vektor *HuMoment* ( $\Phi$ ) yang didapat dari proses ekstraksi ciri terhadap bukode (b). Bukode (b) berisi nilai *HuMoment* ( $\Phi$ ) setiap pose yang disimpan di dalam basisdata hasil dari proses pelatihan. Proses pemetaan dilakukan dengan menghitung *Euclidean Distance* (d) terkecil terhadap seluruh bukode (b).

$$d = \sqrt{(\phi_1 - \phi_{1b})^2 + (\phi_2 - \phi_{2b})^2 + \dots + (\phi_7 - \phi_{7b})^2} \quad (8)$$

Proses kuantisasi dilakukan untuk mengenali pose tangan sesuai dengan bukode pose tangan yang disimpan dalam basisdata. Pose yang dikenali merupakan pose pada bukode yang memiliki nilai *Euclidean Distance* (d) terkecil.

### III. HASIL DAN PEMBAHASAN

Pertama, dilakukan pengujian apakah nilai *HuMoment* dapat digunakan sebagai ciri dari bentuk pose tangan dan apakah nilai *HuMoment* untuk satu pose tidak berubah-ubah secara signifikan. Sebagai contoh, diambil 10 sampel nilai *HuMoment* pose mengepal seperti pada Gambar 6. Nilai *HuMoment* untuk 10 sampel pose mengepal dapat dilihat pada Tabel 1 dan grafik perbandingan nilai *HuMoment* pose tangan mengepal pada Gambar 7.



Gambar 6. Pose Tangan Mengepal

Tabel 1. Nilai *HuMoment* Pose Tangan Mengepal

Sampel	$\phi_1$	$\phi_2$	$\phi_3$	$\phi_4$	$\phi_5$	$\phi_6$	$\phi_7$
1	0.175279	0.002033	8.00E-04	9.58E-06	-1.82E-10	3.02E-08	8.18E-10
2	0.177005	0.002734	7.85E-04	1.29E-05	4.51E-10	2.45E-07	1.22E-09
3	0.174767	0.002164	7.07E-04	8.42E-06	-8.13E-11	-3.05E-08	6.44E-10
4	0.180146	0.003475	9.69E-04	2.28E-05	1.63E-09	5.61E-07	2.98E-09
5	0.178675	0.002685	1.07E-03	1.83E-05	1.09E-09	2.82E-07	2.33E-09
6	0.177061	0.002552	8.73E-04	1.29E-05	2.81E-10	1.62E-07	1.34E-09
7	0.180426	0.00369	9.01E-04	2.45E-05	2.28E-09	8.25E-07	2.85E-09
8	0.180238	0.00368	9.58E-04	2.30E-05	2.22E-09	7.49E-07	2.59E-09
9	0.177432	0.003315	7.77E-04	2.07E-05	2.06E-09	8.03E-07	1.62E-09
10	0.177876	0.002645	9.21E-04	1.43E-05	5.51E-10	2.18E-07	1.55E-09
<b>rata-rata</b>	<b>0.17789</b>	<b>0.002897</b>	<b>0.000876</b>	<b>1.67E-05</b>	<b>1.03E-09</b>	<b>3.84E-07</b>	<b>1.79E-09</b>



Gambar 7. Grafik Perbandingan Nilai *HuMoment* Pose Tangan Mengepal

Berdasarkan Gambar 7, terlihat bahwa nilai *HuMoment* tidak berubah-ubah secara signifikan

untuk setiap sampelnya. Sehingga dapat disimpulkan bahwa nilai *HuMoment* untuk suatu pose bersifat konstan.

Selanjutnya dilakukan pengujian untuk menentukan apakah nilai *HuMoment* dapat digunakan untuk membedakan beberapa bentuk pose tangan dan apakah nilai *HuMoment* dari pose tangan invarian terhadap rotasi dan ukuran gambar tangan. Pengujian dilakukan dengan membandingkan nilai *HuMoment* 9 pose tangan. Perbandingan dilakukan dengan menghitung *Euclidean Distance* nilai *HuMoment* suatu pose tangan terhadap 9 pose tangan lainnya. Perbandingan nilai *Euclidean Distance* *HuMoment* dari 9 pose tangan dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Nilai *Euclidean Distance* Humoment Dari 9 Pose Tangan

Pose	Pose 1: 5 jari terbuka	Pose 2: 4 jari	Pose 3: 3 jari	Pose 4: 3 jari ok membulat	Pose 5: 2 jari telunjuk dan jari tengah	Pose 6: 1 jari telunjuk	Pose 7: 1 jari telunjuk rotasi 90°	Pose 8: Mengepal	Pose 9: Mengepal lebih dekat (zoom)
Pose 1: 5 jari terbuka	0	0.01112	0.01009	0.01132	0.04723	0.09637	0.09081	0.05048	0.05160
Pose 2: 4 jari	0.01112	0	0.01699	0.01789	0.05679	0.10605	0.10053	0.03939	0.04049
Pose 3: 3 jari	0.01009	0.01699	0	0.00333	0.03999	0.09007	0.08443	0.05459	0.05541
Pose 4: 3 jari ok membulat	0.01132	0.01789	0.00333	0	0.03968	0.08870	0.08313	0.05492	0.05575
Pose 5: 2 jari telunjuk dan jari tengah	0.04723	0.05679	0.03999	0.03968	0	0.05365	0.04775	0.09406	0.09471
Pose 6: 1 jari telunjuk	0.09637	0.10605	0.09007	0.08870	0.05365	0	0.00624	0.14218	0.14292
Pose 7: 1 jari telunjuk rotasi 90	0.09081	0.10053	0.08443	0.08313	0.04775	0.00624	0	0.13683	0.13755
Pose 8: Mengepal	0.05048	0.03939	0.05459	0.05492	0.09406	0.14218	0.13683	0	0.00206
Pose 9: Mengepal lebih dekat	0.05160	0.04049	0.05541	0.05575	0.09471	0.14292	0.13755	0.00206	0

Nilai *Euclidean Distance* menyatakan nilai perbedaan dua buah vektor. Semakin kecil nilai *Euclidean Distance* berarti bahwa dua buah vektor yang dibandingkan semakin mirip. Hasil pengujian pada Tabel 2 membuktikan bahwa nilai *HuMoment* setiap pose tangan relatif berbeda terhadap pose lain. Nilai ambang perbedaan adalah 0,01. Hal ini berarti jika nilai *Euclidean Distance* dua vektor atau dua pose lebih besar dari 0,01, maka kedua vektor atau pose tersebut dianggap berbeda. Namun jika nilai *Euclidean Distance* lebih kecil dari 0,01, maka kedua vektor atau pose dianggap sama.

Untuk membuktikan bahwa nilai *HuMoment* tangan invarian terhadap rotasi, maka dilakukan dengan menghitung *Euclidean Distance* nilai *HuMoment* pose 6 yaitu 1 jari telunjuk (Gambar 8) dan pose 7 yaitu 1 jari telunjuk yang berotasi 90° (Gambar 9). Nilai *Euclidean Distance* pose 6 dan pose 7 adalah 0.00624. Nilai tersebut lebih kecil dari nilai ambang yaitu 0.01. Oleh karena itu dapat disimpulkan bahwa pose 6 (Gambar 8) sama dengan pose 7 (Gambar 9).



Gambar 8. Pose Tangan Satu Jari Telunjuk Terbuka

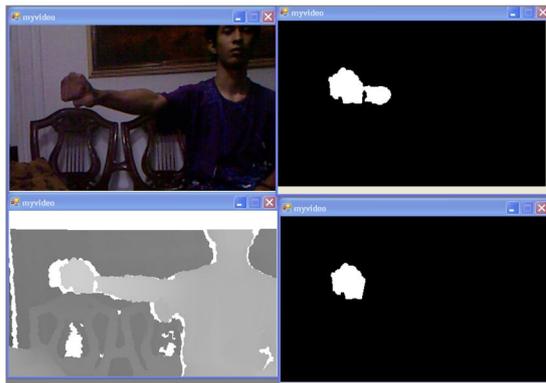


Gambar 9. Pose Tangan Satu Jari Telunjuk Terbuka Rotasi 90°

Kemudian, dilakukan pengujian bahwa nilai *HuMoment* tangan invarian terhadap skala. Hasil pengujian menunjukkan bahwa nilai *Euclidean Distance* pose 8 yaitu mengepal (Gambar 10) dan pose 9 yaitu mengepal lebih dekat (Gambar 11) adalah 0.00206. Nilai tersebut lebih kecil dari nilai ambang yaitu 0.01, sehingga pose 8 dan pose 9 adalah sama.



Gambar 10. Pose Tangan Mengepal



Gambar 11. Pose Tangan Mengepal Lebih Dekat (*Zoom*)

Sedangkan, nilai *HuMoment* untuk pose lain yaitu pose 1, 2, 3, 4, dan 5 memiliki nilai *Euclidean Distance* > 0.01, sehingga pose-pose tersebut adalah berbeda. Dari hasil pengujian tersebut dapat disimpulkan bahwa nilai *HuMoment* dapat membedakan bentuk pose tangan.

Tabel 3. Hasil Pengujian Pengenalan Pose Tangan Dengan Kuantisasi Nilai *HuMoment*

Pose Yang Diuji	Citra Pose Yang Diuji	Jumlah Pose yang Dikenali	Tingkat Akurasi (%)
Pose 1 (5 jari terbuka)		10	100 %
Pose 2 (jari jempol tertutup dan 4 jari lain terbuka)		10	100 %
Pose 3 (jari jempol dan kelingking tertutup dan 3 jari lain terbuka)		8	80%
Pose 4 (jari jempol dan telunjuk tertutup dan 3 jari lain terbuka)		10	100 %
Pose 5 (jari telunjuk dan jari tengah terbuka, 3 jari lain tertutup)		6	60%

Pose Yang Diuji	Citra Pose Yang Diuji	Jumlah Pose yang Dikenali	Tingkat Akurasi (%)
Pose 6 (jari telunjuk terbuka dan jari lain tertutup)		9	90%
Pose 7 (Seluruh jari tertutup/ mengepal)		9	90%

Pengujian kinerja sistem dalam mengenali pose tangan dilakukan terhadap 7 pose tangan yang berbeda. Percobaan dilakukan sebanyak 70 kali dengan 10 data pengujian untuk setiap pose. Setiap data pengujian memiliki variasi arah rotasi, posisi, dan skala ukuran objek tangan. Hasil pengujian pengenalan pose tangan dapat dilihat pada Tabel 3.

Berdasarkan Tabel 3 terlihat bahwa tingkat akurasi pengenalan setiap pose tangan lebih dari 50% dengan rata-rata tingkat akurasi 88.57%. Hal ini membuktikan bahwa nilai *HuMoment* dapat dijadikan sebagai ciri dari bentuk pose tangan dalam proses pengenalan pose tangan. Selain itu dapat dilihat bahwa walaupun data pengujian memiliki variasi arah rotasi, posisi, dan skala ukuran objek tangan, namun sistem tetap dapat mengenali pose tangan tersebut.

#### IV. PENUTUP

##### A. Kesimpulan

Berdasarkan hasil penelitian, sistem pengenalan pose tangan menggunakan *HuMoment* dapat mengenali pose tangan dengan tingkat akurasi pengenalan 88.57%. Sistem dapat mengenali pose tangan berdasarkan bentuk pose tangan bukan posisi. Selain itu, sistem dapat mengenali pose tangan walaupun dilakukan perubahan arah (rotasi), perubahan posisi, dan perubahan ukuran (diperbesar atau diperkecil).

##### B. Saran

Citra yang dihasilkan dari *depth sensor Kinect* tidak stabil, maka disarankan untuk memperbaiki kualitas citra sehingga didapat bentuk kontur tangan yang lebih jelas dan stabil. Kualitas bentuk kontur tangan dapat meningkatkan nilai akurasi pengenalan.

#### DAFTAR PUSTAKA

[1] Matthias Schröder, Christof Elbrechter dll, "Real-time hand tracking with a color glove for the actuation of anthropomorphic robot hands", 2012 12th IEEE-RAS International Conference on Humanoid Robots

- (Humanoids 2012), pp. 262 - 269 , 2012, ISSN: 2164-0580.
- [2] Lijun Zhao, Xiaoyu Li, Peidong Liang dll, "Intuitive robot teaching by hand guided demonstration", 2016 IEEE International Conference on Mechatronics and Automation, pp. 1578 – 1583, 2016, ISSN: 2152-744X.
- [3] Mark Billingham, Tham Piumsomboon, dan Huidong Bai, "Hands in Space: Gesture Interaction with Augmented-Reality Interfaces", IEEE Computer Graphics and Applications, vol. 34, Issue: 1, Jan.-Feb. 2014, pp. 77 – 80, 2014.
- [4] C. Keskin, F. Kirac, Y.E. Kara, L. Akarun, "Real Time Hand Pose Estimation Using Depth Sensors". Computer Vision Workshops (ICCV Workshop) 2011 IEEE International Conference on, pp. 1228-1234, 2011.
- [5] Yan Wen, Chuanyan Hu, Guanghui Yu, Changbo Wang, "A robust method of detecting hand gestures using depth sensors", Haptic Audio Visual Environments and Games (HAVE) 2012 IEEE International Workshop on, pp. 72-77, 2012.
- [6] Jesus Suarez, Robin R. Murphy, "Hand gesture recognition with depth images: A review", RO-MAN 2012 IEEE, pp. 411-417, 2012, ISSN 1944-9445.
- [7] Ti-zhou Qiao, Shu-ling Dai, "Fast head pose estimation using depth data", Image and Signal Processing (CISP) 2013 6th International Congress on, vol. 2, pp. 664-669, 2013.
- [8] Hui Liang, Junsong Yuan, Daniel Thalmann, "Parsing the Hand in Depth Images", Multimedia IEEE Transactions on, vol. 16, pp. 1241-1253, 2014, ISSN 1520-9210.
- [9] Nelly Indriani Widiastuti dan Restu Suhendar, "Scattered object recognition using Hu Moment invariant and backpropagation neural network", Information and Communication Technology (ICoICT ), 2015 3rd International Conference on, pp. 578 – 583, 2015.
- [10] Zhihu Huang dan Jinsong Leng, " Analysis of Hu's Moment Invariants on Image Scaling and Rotation, Proceedings of 2010 2<sup>nd</sup> International Conference on Computer Engineering and Technology (ICCET), pp. 476-480, 2010.