

PENGENALAN ABJAD SISTEM ISYARAT BAHASA INDONESIA (SIBI) BERBASIS KAMERA *DEPTH*

Cucun Very Angkoso¹, Muhammad Fuad², Dian Rusydi Hadiwineka³

Program Studi Teknik Informatika, Universitas Trunojoyo Madura
Jl. Raya Telang, PO BOX 2, Kamal, Bangkalan - 69162
E-mail: cucunvery@gmail.com¹, ibrahim.fuad@gmail.com², d.rusydi@gmail.com³

Abstrak

Interaksi dalam kehidupan sehari-hari umumnya menggunakan bahasa verbal. Penyandang disabilitas tidak dapat menggunakan bahasa verbal, tetapi mereka menggunakan bahasa isyarat yang sulit untuk dimengerti. Sehingga, mereka membutuhkan seorang translator, namun dilain sisi translator tidak dapat memberi mereka privasi. Pada penelitian ini sistem pengenalan isyarat alfabet SIBI dilakukan dengan memanfaatkan kamera depth dari Microsoft kinect. Kinect merupakan sebuah teknologi baru yang dapat memindai gerakan manusia dan suara. Pemanfaatan kinect bertujuan untuk melakukan pengenalan bahasa isyarat secara real-time. Kamera depth menghasilkan citra 3D yang dapat digunakan dalam ruangan gelap dan memungkinkan proses deteksi lebih akurat. Dalam penelitian ini dilakukan segmentasi citra berdasarkan jarak antara obyek dan latar belakang dan hanya menangkap bagian tangan. Binerisasi adalah proses selanjutnya dengan otsu thresholding. Selanjutnya cropping untuk mengambil obyek diperlukan dan resize agar mempermudah dan mempercepat proses selanjutnya. Terakhir, proses pengenalan menggunakan metode euclidean distance berdasarkan nilai jarak terkecil antara template dan obyek. Pengujian template menghasilkan akurasi sebesar 96.538462%. Pengujian real-time menghasilkan akurasi yang baik jika tangan pengguna ditangkap kamera dengan hasil sama seperti template.

Kata Kunci: *kinect, SIBI, euclidean distance, kamera depth, otsu*

1. Pendahuluan

Bahasa isyarat merupakan bentuk komunikasi tanpa mengeluarkan suara, yaitu dengan menggunakan bahasa tubuh dan gerak bibir, serta kombinasi antara gerak tangan, lengan, dan tubuh. Di Indonesia terdapat dua sistem bahasa isyarat yang umum digunakan, yaitu BISINDO (Berkenalalan Dengan Sistem Isyarat Indonesia) dan SIBI (Sistem Isyarat Bahasa Indonesia).

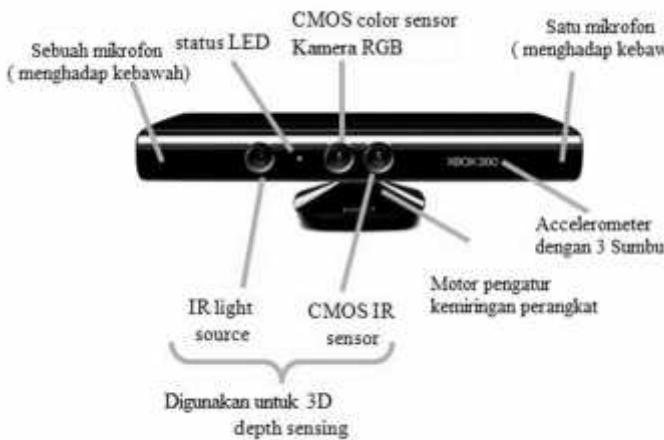
Dalam penelitian ini, sistem yang dibangun merupakan sistem pengenalan isyarat alfabet pada Sistem Isyarat Bahasa Indonesia secara *real-time*. Dari sistem ini, diharapkan dapat memudahkan dalam berkomunikasi dan dapat mengurangi kelemahan dari seorang *translator*. Penelitian Sistem Isyarat Bahasa Indonesia pernah dilakukan oleh Farida (Asriani, F., 2010), dengan mengolah data citra 2 Dimensi. Kekurangan dari penelitian tersebut adalah ketika citra jari memiliki warna yang sama dengan latar belakang, dan jarak antara jari dan telapak tangan terlalu dekat. Maka, pada penelitian ini digunakan Kinect karena memiliki sensor kedalaman. Kinect atau Kinect for Xbox 360 adalah

controler-free gaming dan pengalaman hiburan buatan Microsoft dan Xbox 360 *video game platform*. Sensor kedalaman dapat memperoleh data antara jari tangan, dan jarak antara tangan dan badan.

2. Sistem Isyarat Bahasa Indonesia

Sistem Isyarat Bahasa Indonesia atau SIBI merupakan suatu standart nasional di Indonesia sebagai bahasa isyarat yang telah disepakati dan digunakan sebagai media komunikasi bagi penderita tuna rungu atau/dan tuna wicara. SIBI dibuat dari hasil rekayasa orang normal, bukan buatan Tuna Rungu sendiri, dan SIBI sama dengan bahasa isyarat Amerika (America Sign Language - ASL). Dalam SIBI, ada isyarat untuk kata dan huruf, pada penelitian ini hanya difokuskan untuk isyarat huruf. Gambar 2.8 merupakan pola isyarat alfabet SIBI dari huruf A hingga huruf Z. Pada isyarat huruf "J" dan "Z" terdapat gerakan tangan. Sedangkan isyarat pada huruf lainnya hanya perubahan susunan pola jari-jari tangan tanpa melakukan gerakan tangan. Beberapa huruf memiliki kemiripan pola, hanya berbeda

posisinya saja, seperti pada huruf G dan Q, kemudian K dan P. Pada isyarat huruf A dan S, lalu M dan N memiliki pola genggaman tangan yang mirip.



Gambar 2.1. Komponen Sensor Microsoft Kinect

Dalam penelitian ini, terdapat proses segmentasi yang dilakukan dengan mengambil objek sesuai jarak dengan kamera *depth* pada perangkat *Kinect*. Yang selanjutnya dilakukan proses *cropping*, *resize*, binerisasi, dan pencocokan data dari rentetan citra tangan menggunakan metode *Euclidean Distance*. Hasil pencocokan berdasarkan nilai *minimum distance* antara citra masukan dan citra *training*. Penggunaan metode ini diharapkan dapat mengenali isyarat tangan secara efisien dan mudah.

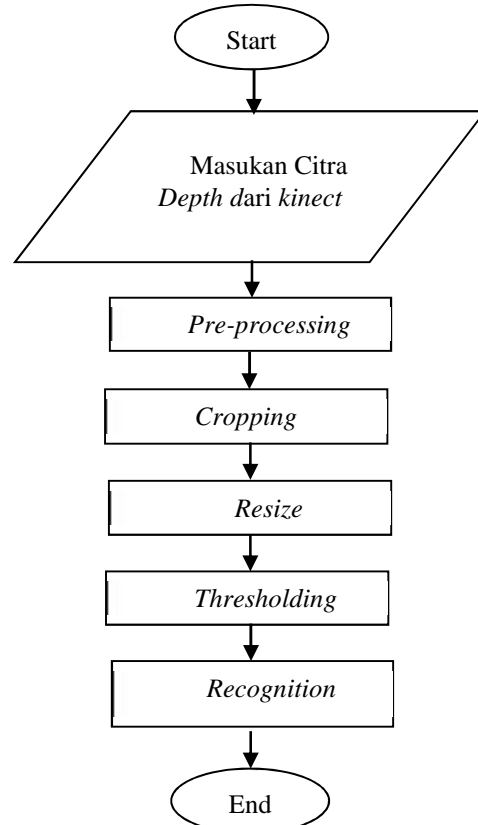


Gambar 2.2. Bentuk Isyarat Bahasa Indonesia (Depsos, 2015)

3.1. Metode

Metode penelitian yang dilakukan secara umum terdapat pada Gambar 1, diawali dengan mengaktifkan sensor yang terdapat pada perangkat *Kinect*, sensor yang diaktifkan adalah sensor RGB dan sensor *depth*, tetapi hanya sensor *depth* yang digunakan untuk proses selanjutnya. Hasil tangkapan dari sensor RGB digunakan untuk menampilkan lingkungan yang ada di depan perangkat *Kinect*. Sedangkan hasil tangkapan dari sensor *depth* yang berupa citra bitmap digunakan sebagai masukan untuk tahap *pre-processing*. Pada

tahap *pre-processing*, dilakukan segmentasi citra untuk mengatur dan menentukan jarak tangkap obyek yang dilakukan oleh sensor *Kinect*. Pada area jarak yang telah ditentukan tersebut obyek ditangkap, sedangkan obyek yang berada di luar area tersebut diabaikan dan dijadikan latar belakang berwarna putih. Selanjutnya tahap *cropping*, yaitu pemotongan citra menjadi berukuran 150x150 piksel. Kemudian dilakukan *resize* untuk mengubah ukuran citra hasil *cropping* pada tiap citra yang dihasilkan menjadi berukuran 25x25 piksel. Citra yang telah berukuran 25x25 piksel dilanjutkan dengan proses *thresholding*, proses ini melakukan konversi citra awal yang berupa citra *greyscale* menjadi citra biner dengan cara mengatur nilai ambang tiap piksel pada citra *greyscale*. Jika nilai piksel dibawah nilai ambang yang telah ditentukan, maka piksel diberi nilai 0, dan jika piksel berada diatas nilai ambang yang telah ditentukan, piksel diberi nilai 1. Setelah berubah menjadi citra biner, proses yang dilakukan adalah proses pencocokan citra menggunakan metode *euclidean distance*, proses ini dilakukan dengan menghitung nilai antara citra masukan dengan citra *template* yang kemudian dicari nilai terkecil dari hasil perhitungan *euclidean distance* tersebut. Nilai terkecil yang didapat merupakan hasil citra masukan yang memiliki kemiripan dengan citra *template*.



Gambar 3.1. Garis besar sistem

Keterangan :

- a. *Input Depth image* : data berupa citra dari sensor *depth* Kinect. Data citra masukan berupa citra bitmap berukuran 640×480 dalam 10 *frames per second* (FPS). Citra tersebut merupakan citra yang digunakan untuk diolah dalam sistem.
- b. *Preprocessing* : merupakan tahap *segmentation*. Citra yang ditangkap hanya citra dengan jarak yang telah ditentukan, dan obyek yang diperlukan saja yang diambil untuk proses selanjutnya. Obyek yang tidak diperlukan dalam proses selanjutnya dan obyek yang tidak berada dalam area jarak tangkap diabaikan.
- c. *Cropping* : memotong citra pada bagian tengah berukuran 150×150 piksel dengan memberikan frame untuk meletakkan tangan, sehingga citra yang diambil hanya terdapat obyek citra tangan.
- d. *Resize* : mengkonversi ukuran citra menjadi 25×25 piksel.
- e. *Thresholding* : citra masukan yang berupa citra *greyscale* dikonversi menjadi citra biner.
- f. *Recognition* : melakukan perhitungan *euclidean distance* untuk mencari nilai jarak minimum antara citra masukan dan citra *template*. Citra masukan akan dibandingkan satu persatu dengan tiap citra *template*, yang kemudian dicocokkan kemiripannya berdasarkan nilai terkecil dari hasil perhitungan.

Sensor Kinect sebagai perangkat input citra, diatur melakukan pengambilan gambar secara otomatis setelah sistem dinyalakan sebanyak 10 citra tiap detik. Citra yang diambil berupa citra *greyscale* dari sensor kedalaman pada Kinect. Citra masukan merupakan citra 3D dengan x, y, dan z yang diketahui.

Sebelum data dapat digunakan, data terlebih dahulu harus melalui *preprocessing*, didalamnya terdapat proses antara lain :

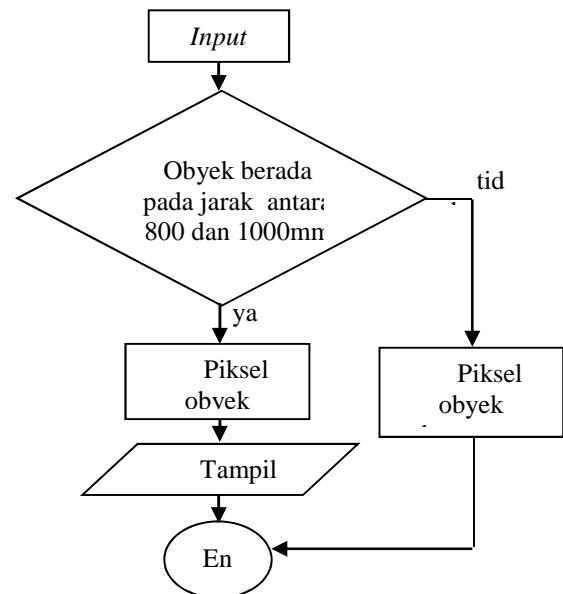
1. Mengatur jarak tangkap

Dengan menggunakan sensor *depth* Kinect, obyek yang ditangkap harus berada dalam rentang jarak 800-1000mm dari sensor, dapat dilihat seperti pada Gambar 2.15. Selain pada jarak tersebut obyek diberi warna putih, dan diabaikan untuk proses

selanjutnya. Pemberian rentang jarak sebagai area kerja bertujuan untuk menghasilkan citra yang akurat dan tidak memiliki perbedaan yang besar. Jika obyek jauh dari kamera, maka citra yang dihasilkan akan kecil dan begitupula sebaliknya.

2. Menampilkan objek

Citra yang ditangkap hanyalah citra tangan, dengan cara menampilkan warna piksel pada obyek jika berada pada area kerja sensor kinect yang telah ditentukan.



Gambar 3.2. Preprocessing

Pada Gambar 4, input citra dari sensor *depth* Kinect jika terdapat obyek berada dalam jarak antara 800mm dan 1000mm maka obyek tersebut masuk dalam area tangkap. Yang kemudian obyek tersebut diambil untuk proses selanjutnya. Jika tidak, obyek tersebut diabaikan dengan cara mengatur piksel obyek menjadi berwarna putih.

3.2. Thresholding

Thresholding adalah metode sederhana segmentasi citra. Dari citra *grayscale*, *thresholding* dapat digunakan untuk membuat citra biner.

Selama proses *thresholding*, setiap piksel dalam foto ditandai sebagai "obyek" piksel jika nilainya adalah lebih besar dibandingkan nilai ambang (asumsi obyek menjadi lebih terang daripada latar belakang) dan ditandai sebagai "latar belakang" untuk piksel yang lain. Konvensi ini dikenal sebagai ambang di atas. Variasi termasuk di bawah ambang batas yang berlawanan dari ambang di atas, di dalam batas, dimana piksel yang berlabel "obyek" jika ada di antara dua nilai *thresholds*; dan di luar batas, yang merupakan kebalikan dari dalam

ambang (Shapiro, L., 2000). Biasanya, obyek piksel diberi nilai "1" sedangkan piksel latar belakang diberi nilai "0." Akhirnya, biner gambar yang dibuat oleh setiap piksel warna putih atau hitam, tergantung pada piksel labelnya.

Thresholding dapat dirumuskan seperti Persamaan 1, yaitu :

$$f(x,y) = \begin{cases} 1, & f(x,y) \geq T \\ 0, & f(x,y) < T \end{cases}$$

Pada Persamaan 2, T adalah nilai ambang atau *threshold*. *Thresholding* digunakan sebagai acuan untuk proses binerisasi. *Thresholding* digunakan untuk menentukan suatu piksel dengan derajat keabuan tertentu untuk diubah nilainya menjadi nilai biner. Namun, menggunakan rumus pada persamaan 2 dapat beresiko. Karena dengan menetapkan nilai ambang T pada suatu nilai tertentu dapat mengakibatkan munculnya suatu piksel yang mengganggu. Adanya perbedaan tingkat keabuan yang sangat tipis sehingga menyulitkan proses selanjutnya. Maka, untuk menghasilkan nilai ambang yang baik dan secara otomatis, digunakan metode *otsu*.

Metode *otsu* membagi histogram citra *gray level* ke dalam dua daerah yang berbeda secara otomatis tanpa perlu bantuan pengguna untuk memasukkan nilai ambang. Pendekatan yang dilakukan oleh metode *otsu* adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul. Analisis Diskriminan memaksimumkan *variable* tersebut agar dapat membagi menjadi dua daerah, yaitu latar depan (*foreground*) dan latar belakang (*background*) (Otsu, N,1979)

Ada beberapa perhitungan yang harus dilakukan untuk mendapatkan nilai *threshold*. Pertama membuat histogram dari citra agar diketahui jumlah piksel untuk setiap tingkat keabuan. Tingkat keabuan citra dinyatakan dengan i sampai dengan L . Level ke i dimulai dari 1, yaitu piksel 0. Untuk L , maksimal level adalah 256 dengan piksel bernilai 255. Dari histogram tersebut dapat diketahui jumlah piksel pada level ke i , yang dinyatakan dengan n_i . Perhitungan pertama adalah mencari nilai probabilitas piksel ke i yang dirumuskan pada Persamaan ke-3, yaitu :

$$p_i = \frac{n_i}{N} \quad (3)$$

Di mana p_i merupakan probabilitas piksel ke- i , jumlah level pada piksel ke- i merupakan n_i , dan N merupakan jumlah piksel keseluruhan. Dilanjutkannya dengan mencari nilai *weight*, *mean*, dan *mean global*. Untuk mendapatkan nilai tersebut, dilakukan beberapa perhitungan, yaitu :

Menghitung jumlah kumulatif (*cumulative sum*) menggunakan rumus pada persamaan 4. yang

dinyatakan dengan $\omega(k)$, untuk piksel ke $i = 0,1,2,\dots,L-1$, yaitu :

$$\omega(k) = \sum_{i=0}^k p_i \quad (4)$$

Menghitung rerata kumulatif (*cumulative mean*) menggunakan rumus pada persamaan 5, yang dinyatakan dengan $\mu(k)$, untuk piksel ke $i = 0,1,2,\dots,L-1$, yaitu :

$$\mu(k) = \sum_{i=0}^k i \cdot p_i \quad (5)$$

Menghitung rerata intensitas global menggunakan rumus pada persamaan 6, yang dinyatakan dengan $\mu_T(k)$, untuk piksel ke $i = 0,1,2,\dots,L-1$, yaitu :

$$\mu_T(k) = \sum_{i=0}^{L-1} i \cdot p_i \quad (6)$$

Rentang piksel i dari 0 hingga $L-1$ dihitung menggunakan Persamaan 4, Persamaan 5, dan Persamaan 6. Dari hasil perhitungan tersebut, dilakukan perhitungan *between class variance* (varian antar kelas). Persamaan untuk perhitungan tersebut ditampilkan pada Persamaan 7 dengan $\sigma_B^2(k)$ merupakan *between class variance*, yaitu :

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \quad (7)$$

Dari seluruh hasil perhitungan *between class variance*, dicari nilai tertinggi atau nilai maksimal seperti Persamaan 8. Nilai tersebut akan digunakan sebagai nilai ambang atau *threshold*.

$$\sigma_B^2(k^*) = \max_{1 \leq x \leq L} \sigma_B^2(k) \quad (8)$$

Nilai *between class variance* tertinggi digunakan sebagai nilai ambang atau *threshold* yang digunakan untuk merubah citra *grayscale* menjadi citra biner

3.3. Proses Pengenalan

Euclidean distance adalah perhitungan jarak dari 2 buah titik dalam *Euclidean space*. *Euclidean space* diperkenalkan oleh seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. *Euclidean* ini biasanya diterapkan pada 2 dimensi dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi. Persamaan 9 merupakan rumus *euclidean distance* 2 dimensi (Gonzales, R. C.,2002)

$$d(i, j) = \frac{1}{\sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2}}$$

Dimana d merupakan *distance* atau nilai jarak yang dicari. Sedangkan i dan j merupakan koordinat titik.

4.1. Uji Coba Dan Pembahasan

Uji coba bertujuan untuk menganalisis dan mengevaluasi dari sistem yang telah dibuat. Tabel.1 merupakan lingkungan uji coba dari system.

Tabel.4.1. Lingkungan Uji Coba

Perangkat	Nama Komponen	Spesifikasi
Perangkat Keras	Brand	ASUSTeK COMPUTER INC. K46CM
	Processor	Intel Core i3 3217U @ 1.80GHz
	Memory	4.00GB Single-Channel DDR3 @ 794MHz
	Kamera	XBOX 360 KINECT SENSOR
Perangkat Lunak	Sistem Operasi	Windows 8 Pro 64-bit
	Perangkat Lunak	Visual Studio Ultimate 2013, Kinect for Windows SDK (Software Development Kit) v1.8, AForge.NET framework 2.2.5

Uji coba dilakukan terhadap citra *template*. Citra *template* merupakan citra yang diambil dengan menggunakan kamera *depth kinect*. Setiap huruf memiliki 10 variasi citra yang berbeda posisi letak dan sudut kemiringan tangannya. Total dari semua *template* adalah 260 citra *template*.

Setiap citra *template* melalui proses segmentasi, *cropping*, *resize*, binerisasi, dan *recognition*. Pada proses *recognition*, digunakan *euclidean distance* untuk menghitung jarak antara *template* yang diuji dengan keseluruhan *template*. Gambar 4.3 merupakan gambar tampilan sistem setelah sistem dijalankan. Sistem menganalisis citra tangan yang berada pada kotak.



Gambar 4.1. Tampilan Sistem yang Dijalankan dengan Posisi Tangan Berada pada Bingkai

Tabel 4.2. Citra input a1

Citra Depth	Cropping Citra
 Ukuran 640x480	 Ukuran 150x150
Resize Citra	Binerisasi
 Ukuran 25x25	

4.2. Hasil Uji Coba

Pada Tabel 2. terdapat salah satu citra, yakni citra a1 yang melalui semua proses kecuali *recognition*. Cita a1 dibandingkan dengan seluruh citra *template* dan dihitung nilai *euclidean distance* nya. Setelah mendapatkan nilai *euclidean distance*, dicari nilai rata-rata dari setiap huruf yang masing-masing berjumlah 10 huruf dan telah dibandingkan dengan citra a1.

Hasil dari rata-rata tersebut dapat dilihat pada Tabel 2.

Tabel 4.2. Hasil rata-rata *euclidean distance* pada citra a1 terhadap citra *template*

<i>Template</i>	Dikenali	Avg.ED	<i>Template</i>	Dikenali Sebagai Huruf	Rata-Rata ED
a1	A	1196.0408	a1	R	2152.0373
a1	H	1792.2668	a1	L	2173.3823
a1	G	1906.4196	a1	I	2212.5295
a1	Y	1934.7193	a1	E	2256.8348
a1	S	2001.991	a1	P	2322.8544
a1	T	2039.2249	a1	Q	2344.7968
a1	J	2076.512	a1	D	2378.4555
a1	N	2076.5225	a1	W	2414.0395
a1	M	2091.5322	a1	F	2469.8691
a1	K	2101.9953	a1	V	2502.0527
a1	O	2106.7623	a1	U	2511.3917
a1	X	2143.4644	a1	B	2608.7952
a1	Z	2148.0038	a1	C	2765.939

4.3. Analisa

Kesalahan-kesalahan tersebut dapat dipengaruhi oleh berbagai hal, seperti cahaya, posisi kinect, posisi tangan, ukuran tangan, jarak, dan metode *euclidean distance*.

5. Kesimpulan Dan Saran

5.1 Kesimpulan

Dari hasil percobaan sistem Pengenalan SIBI dapat diketahui bahwa *Euclidean Distance* cukup baik digunakan untuk proses pengenalan *template* isyarat alfabet SIBI. Uji coba yang dilakukan pada *template* menghasilkan nilai dengan tingkat akurasi sebesar 96.538462%. Beberapa *template* tidak dikenali sesuai huruf yang diwakilkannya karena kemiripan pola isyarat. Seperti pada *template* a5 yang dikenali sebagai huruf T dan *template* aa2 dikenali sebagai huruf S.

Sedangkan untuk uji coba sistem secara *real time*, sistem dapat mengenali dengan benar jika pengguna melakukan isyarat SIBI sama seperti *template* atau mendekati pola yang ada pada *template*. Hal ini disebabkan karena metode *euclidean distance* yang digunakan dalam proses pengenalan hanya dapat mengenali dengan baik apabila citra tangkapan sama dengan *template*. Perbedaan besar kecil tangan, jarak antara tangan dan kamera serta posisi tangan dapat mempengaruhi hasil pengenalan.

5.2. Saran

Sistem pengenalan pada bahasa isyarat yang lebih kompleks terutama yang didalamnya ada gerakan obyek kompleks perlu dilakukan pada penelitian selanjutnya. Pada penelitian ini, sistem yang dibuat berbasis pada kumpulan tiga citra statis (*depth image*) hasil akuisisi sensor kinect. Khusus pada isyarat huruf J dan Z merupakan isyarat yang menggunakan citra bergerak sehingga citra yang digunakan pada *template* hanya berupa posisi isyarat yang terakhir. Metode penurunan dimensi fitur perlu dilakukan untuk mengurangi kompleksitas komputasi.

Daftar Pustaka

- Asriani, F., Susilawati, H. Pengenalan Isyarat Tangan Statis pada Sistem Isyarat Bahasa Indonesia Berbasis Jaringan Syaraf Tiruan Perambatan Balik. MAKARA. 14 : 150-154. 2010.
- <http://www.who.int/topics/disabilities/en/>, diakses pada tanggal 3 des 2015.
- <http://melati.depsos.go.id/modules.php?name=Content&pa=showpage&pid=14>, Sistem Isyarat Bahasa Indonesia (SIBI), diakses pada tanggal 3 des 2015.
- Shapiro, L., Stockman, G. Computer Vision. Prentice Hall. 2000Fuad, M. *Pengembangan Deteksi Gesture Tangan Berbasis Citra Depth Menggunakan Pencocokan Fitur*. Prosiding-Seminar Nasional Ilmu Komputer. 2014.
- Sekar, K., Remimol, A.M. *A Method of DWT with Bicubic Interpolation for Image Scaling*. International Journal of Computer Science Engineering (IJCSE). Vol. 3 No.02. 2014.
- Catuhe, D. Programming with The Kinect for Windows Software Development Kit. Washington : Microsoft Press. 2012.
- Gonzales, R. C., Woods, R. E., & Eddin, S. L. (2002). Digital Image Processing Using Matlab. New Jersey: Pearson Education.
- Otsu, N, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics. Vol SMC-9 No.1. Page : 62-66. 1979.