

Pengenalan Bahasa Isyarat Tangan Menggunakan *Depth Image*

Try Yuliandre Pajar, Djoko Purwanto, dan Hendra Kusuma

Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember (ITS)

E-mail: try13@mhs.ee.its.ac.id, djoko@ee.its.ac.id, hendraks@ee.its.ac.id

Abstrak—Sistem pengenalan bahasa isyarat menggunakan pengolahan visual digital. Pada studi ini sistem mengambil citra gambar menggunakan depth sensor. Depth sensor digunakan untuk mendapatkan gambar tangan sehingga tahap pengambilan kontur berbeda dibandingkan dengan kamera RGB. Depth sensor yang diatur jarak pembacaannya dapat menghasilkan gambar kontur tangan. Menggunakan metode *convex hull*, *convexity defects*, dan pusat massa gambar dapat menghasilkan nilai-nilai yang dapat dilatih untuk melakukan pengenalan pada tahap uji cobanya. Sistem ini dapat menangkap citra tangan dari jarak 50cm hingga 65cm. Sistem ini dilatih menggunakan *artificial neural network* dengan dua kondisi percobaan. Percobaan pertama menggunakan delapan output berdasarkan koordinat yang didapat. Percobaan kedua menggunakan tiga input berdasarkan panjang garis dan luas. Hasil yang dicapai sistem ini yaitu dapat mengenali gestur bahasa isyarat tangan berdasarkan hasil pelatihan. Hasil pelatihan ditentukan dari elemen penyusun *neural network* dan banyaknya iterasi yang dilakukan, pada ragam huruf yang sedikit akurasi hasil pelatihan dapat memenuhi target output sebaliknya jika ragam huruf bertambah banyak.

Kata Kunci—Bahasa Isyarat, Kinect, Machine Vision, Depth Sensor.

I. PENDAHULUAN

KETERBATASAN dalam berkomunikasi masih menjadi sebuah masalah yang dirasakan oleh orang-orang yang menderita gangguan berbicara seperti tunawicara, tunarungu, dan sejenisnya. Contohnya pada orang yang tidak mampu berbicara seperti tunawicara atau tunarungu menggunakan isyarat tangan. Permasalahannya adalah penggunaan isyarat tangan tidak selalu dimengerti oleh orang normal sehingga dibutuhkan perangkat tambahan yang mampu mengubah isyarat tangan menjadi sebuah keluaran baru yang dapat dipahami oleh orang normal.

Perkembangan teknologi machine vision dapat diterapkan dalam bermacam-macam persoalan, salah satunya pada studi ini. Penggunaan bahasa isyarat tangan dimana citra dari gestur tangan tersebut akan digunakan sebagai acuan dalam pemrosesan machine vision dimana gestur tangan yang dihasilkan akan diolah menjadi suara. Perangkat utama yang digunakan adalah kinect, personal computer (PC), dan speaker. Pengenalan gestur tangan (*Hand Gesture Recognition*) menggunakan Kinect ini diproses oleh personal computer (PC) dan output yang dihasilkan berupa suara menggunakan speaker. Perangkat Kinect dari Microsoft menyediakan cara untuk mengekstrak fitur diskriminatif untuk pengenalan isyarat

tangan. Pendekatan pengakuan dari Kinect terdiri dari dua tahap: tahap pelatihan dan tahap pengenalan [1]. Gestur tangan yang dikenali ini akan merepresentasikan tiap huruf alfabet dan nantinya output audio yang dihasilkan adalah pengejaan huruf tersebut.

II. DASAR TEORI

A. Bahasa Isyarat

Fungsi bahasa pada dasarnya terbagi menjadi tiga elemen yaitu pembicara, pendengar, dan sebuah sistem [2]. Bahasa isyarat merupakan suatu bentuk komunikasi yang menggunakan bahasa tubuh, tangan, dan gerak bibir namun tidak lisan untuk mengekspresikan pikiran penggunanya [3]. Kaum tunarungu dan tunawicara adalah kelompok utama yang menggunakan bahasa isyarat. Bahasa ini merupakan kombinasi dari bentuk tangan orientasi dan gerak jari, tangan, dan lengan.

Bahasa isyarat memiliki sistem berdasarkan region seperti amerika memiliki American Sign Language (ASL). Indonesia memiliki sistem bahasa isyarat Indonesian Sign Language (ISL) atau sistem Isyarat Bahasa Indonesia (SIBI), sistem ini memiliki kesamaan dengan ASL.

B. Kinect

Kinect adalah perangkat elektronik dengan teknologi software yang dikembangkan oleh Microsoft. Kinect awalnya digunakan sebagai kontroler tambahan pada konsol game Xbox 360. Perangkat Kinect memiliki kamera RGB, Depth Sensor, motorized tilt, dan multi-array microphone [3]. Depth Sensor pada kinect merupakan kombinasi dari laser inframerah dan sensor CMOS monokromatik. Kinect generasi pertama dirilis pada november 2010 kemudian pada february 2012 generasi kedua dari perangkat ini dirilis. Pada studi ini kinect yang digunakan adalah kinect generasi pertama. Kinect dapat dikembangkan dengan beberapa macam software, salah satunya adalah Microsoft SDK kinect yang diproduksi dari Microsoft sendiri. Selain SDK, kinect juga memiliki berbagai macam library yang dapat digunakan untuk pengembangannya. Pada studi ini library yang digunakan adalah OpenNI.

C. Kinect

Kinect adalah perangkat elektronik dengan teknologi software yang dikembangkan oleh Microsoft. Kinect awalnya digunakan sebagai kontroler tambahan pada konsol game Xbox 360. Perangkat Kinect memiliki kamera RGB, *Depth*

Sensor, motorized tilt, dan multi-array microphone [3]. *Depth Sensor pada* kinect merupakan kombinasi dari laser inframerah dan sensor CMOS monokromatik. Kinect generasi pertama dirilis pada november 2010 kemudian pada februari 2012 generasi kedua dari perangkat ini dirilis. Pada studi ini kinect yang digunakan adalah kinect generasi pertama. Kinect dapat dikembangkan dengan beberapa macam software, salah satunya adalah Microsoft SDK kinect yang diproduksi dari Microsoft sendiri. Selain SDK, kinect juga memiliki berbagai macam library yang dapat digunakan untuk pengembangannya. Pada studi ini library yang digunakan adalah OpenNI.

D. *Depth Sensor Kinect*

Depth sensor kinect atau sensor kedalaman kinect terdiri dari dua bagian yaitu pemancar inframerah (IR emitter) dan sensor kedalaman inframerah (IR depth sensor) yang merupakan sensor monokrom CMOS (*Complimentary Metal Oxide Semiconductor*). Pemancar inframerah berfungsi sebagai proyektor yang memancarkan cahaya inframerah dalam pola "pseudo-random dot"[4]. Dot atau titik ini tidak dapat dilihat oleh mata manusia dan terdapat pada permukaan objek yang berbeda-beda berdasarkan arah pemancarnya tetapi, titik ini merupakan informasi penting yang nantinya diolah oleh sensor kedalaman inframerah. Sensor kedalaman inframerah berfungsi sebagai pembaca dari dot yang dihasilkan pemancar, sensor akan mengukur jarak tiap dot sehingga terdapat perbedaan nilai tiap dot akan menghasilkan gambar kedalaman atau depthmap. Depthmap didukung oleh resolusi 640 x 480 piksel, 320 x 240 piksel, dan 80 x 60 piksel [4]. Teknologi dari sensor depth ini berasal dari PrimeSense. Cara pengolahan data depth bekerja sesuai diagram berikut.

Berdasarkan diagram, chip PrimeSense akan mengirimkan sinyal ke pemancar untuk menyalakan lampu dari inframerah jika ada kamera memulai *capturing* Selain itu chip juga mengirimkan sinyal lain ke sensor inframerah untuk pengambilan data kedalaman sesuai jangkauan yang terlihat sensor. Kemudian pemancar memancarkan inframerah ke segala objek yang berada didepan kemudian sensor kedalaman membaca data berdasarkan jarak yang terukur antara titik dengan sensor, data ini kemudian diteruskan ke chip PrimeSense. Setelah chip PrimeSense menerima data dari sensor yang kemudian dianalisa sehingga dapat menampilkan gambar kedalaman.

E. *OpenNI*

OpenNI adalah perangkat lunak antarmuka untuk algoritma pengolahan data sensor kinect. Fungsi dari OpenNI adalah untuk mendefinisikan tipe data yang akan diolah pada kinect seperti warna, *depthmap*, dan *skeleton*. OpenNI juga memiliki *library* yang dapat digunakan untuk mengakses perangkat kinect.

F. *OpenCV*

OpenCV adalah suatu pustaka yang mengimplementasikan algoritma dan sintaks pada pengolahan citra. Dimana pengolahan citra terfokus ada mengekstrak suatu informasi dari gambar. OpenCV menyediakan algoritma yang dapat menemukan lokasi objek dan juga pendeteksian warna.

G. *Pengolahan Citra digital*

Pengolahan citra digital adalah suatu proses dengan masukan dan keluaran data yang berupa citra. Proses ini bertujuan untuk memanipulasi dan menganalisa suatu informasi citra dengan perangkat komputer. Data yang diolah

Pengolahan citra digital dapat memperbaiki suatu kualitas gambar dan juga dapat digunakan untuk pengenalan suatu objek. Pada studi ini pengolahan citra digital bertujuan untuk mengekstraks fitur dari gambar yang nantinya akan diolah oleh algoritma ANN.

H. *Erode*

Erode atau erosi adalah salah satu dari transformasi morfologi dimana proses morfologi ini mengikis batas atau tepi dari suatu gambar biner. *Erode* merupakan salah satu dari operator dasar pada morfologi [5], operator ini biasanya bekerja pada gambar biner. Pada gambar biner operator ini akan mengikis tepi gambar akan dibuang bergantung dari ukuran dan bentuk dari kernel. Operator *erode* bekerja dengan menggunakan dua data yaitu gambar masukan dan *structuring element* atau kernel

I. *Dilate*

Dilate atau pelebaran adalah satu dari transformasi morfologi dimana proses morfologi ini menebalkan batas atau tepi dari suatu gambar biner. *Dilate* merupakan salah satu dari operator dasar pada morfologi selain erode, operator ini menebalkan suatu tepi dari gambar biner berdasarkan ukuran dan bentuk dari kernel. Operator *dilate* bekerja dengan menggunakan dua data yaitu gambar masukan dan *structuring element* atau kernel.

J. *Convex Hull*

Convex hull adalah suatu himpunan yang terdiri dari beberapa titik dimana sebarang dari titik tersebut akan terhubung membentuk sebuah garis yang melingkupi semua. *Convex Hull* mampu meng-cover semua himpunan titik dengan jumlah garis yang paling minimal.

K. *Convexity Defects*

Convexity defects adalah pengolahan lebih lanjut dari *convex hull* dimana *convexity defects* terdiri dari 4 elemen vektor yang merepresentasikan suatu nilai yaitu *start index*, *end index*, *farthest point index*, *fix point depth* *Start index* adalah suatu titik terluar yang terhubung dengan *end index*, kedua titik ini terhubung mengitari kontur objek sesuai *convex hull*. *Farthest point* adalah titik yang memiliki nilai terjauh dari kontur terhadap garis *hull*. *farthest point* terletak diantara *start index* dan *end inde*. *Fixed point depth* adalah

L. *Neural Network*

Artificial Neural Network (ANN) adalah sebuah paradigma pengolahan informasi yang terinspirasi dari cara kerja sistem saraf secara biologis[6]. *Neural network* merupakan salah satu komponen penting dalam *artificial intelligence* (AI). Hal ini telah bertahun-tahun dipelajari dengan harapan mencapai seperti kinerja manusia di berbagai bidang [6].

ANN memiliki berbagai macam definisi, seperti menurut Simon Haykin ANN adalah prosesor dalam jumlah besar yang

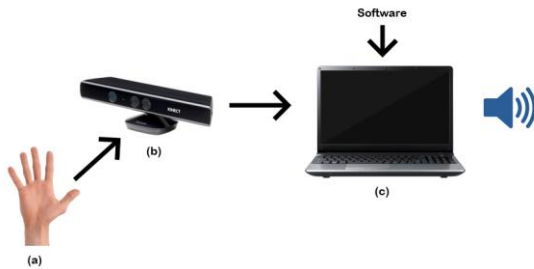
terpasang secara paralel dan memiliki unit pemroses yang sederhana, unit pemroses ini menyimpan hasil pembelajaran yang dapat digunakan kembali [7].

III. PERANCANGAN SISTEM

Perancangan sistem ini akan dijelaskan sistem secara keseluruhan terutama pada bagian perangkat lunak karena pada studi ini sistem berperan banyak pada bagian perangkat lunak yang akan mengolah informasi visual. Sistem pengenalan bahasa isyarat ini terdiri dari empat komponen utama yaitu pengguna alat, kamera kinect, pengolahan citra digital, dan speaker sebagai output.

A. Perancangan Hardware

Prinsip kerja dari sistem pengenalan bahasa isyarat ini dimulai dari beberapa tahap. Dimulai dari pengambilan gambar gestur bahasa isyarat pengguna. Gestur tangan yang ditangkap akan diolah berdasarkan jarak yang sudah diatur atau di-threshold pada program sehingga akan didapatkan kontur tangan pengguna. Kemudian kontur tangan yang didapat akan diolah dengan *convex hull*, *convexity defects*, dan *moment* untuk didapatkan fitur yang nantinya dapat diolah oleh algoritma *artificial neural network* sehingga fungsi sistem dapat mengenali gestur dari bahasa isyarat tangan.



Gambar 1. Ilustrasi cara kerja sistem (a) tangan pengguna; (b) kinect yang terhubung laptop dan suplai ; (c) laptop dan speaker

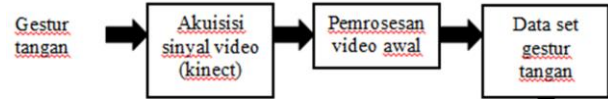
B. Perancangan Software

Pada perancangan *software* atau perangkat lunak sistem akan dirancang untuk mengambil data dan pembelajaran data. Pada proses pengambilan data dilakukan serangkaian proses sebagai berikut:

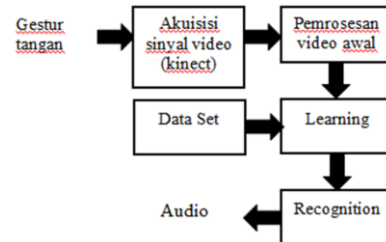
1. Depth sensor menangkap citra tangan, pada tahap ini sensor depth sudah di atur jarak minimum dan maksimum agar citra tangan lebih mudah didapatkan.
2. Citra tangan yang didapatkan akan diproses menggunakan transformasi morfologi, hasil dari morfologi menghasilkan citra tangan dengan kontur yang lebih halus. Hasil dari proses morfologi akan didapatkan pusat massa dan convex hullnya. Pada convex hull nilai convexity defect didapatkan, nilai-nilai ini terdiri dari *start point*, *end point*, *farthest point*.
3. Hasil dari proses sebelumnya menghasilkan nilai-nilai yang akan dijadikan data set dari gestur tangan. Data set ini digunakan untuk proses pembelajaran pada *artificial neural network*.

Pada proses pembelajaran atau *learning*, sistem akan melakukan tahap yang sama pada proses akuisisi data dan

pemrosesan video awal. Hasil dari data set yang sudah melalui tahap *learning* akan digunakan untuk mengenali gestur tangan yang diujikan.



Gambar 2. Diagram blok sistem untuk mendapatkan data set



Gambar 3. Diagram blok sistem learning hingga ke output Audio

```

1 mengakses kinect depth sensor
2 capturing depth data
3 atur jarak threshold minimum dan maksimum depth sensor
4 lakukan operator morfologi erode
5 lakukan operator morfologi dilate
6 dapatkan kontur gambar yang didapat
7 gambarkan garis tepi kontur
8 dapatkan pusat massa kontur
9 dapatkan convex hull
0 gambarkan garis hull
1 dapatkan convexity defect
2 gambarkan garis start point
3 gambarkan garis end point
4 gambarkan titik start point
5 gambarkan titik farthest point
6 gambarkan titik pusat massa
7 masukkan nilai-nilai hasil pelatihan:
8 weight
9 bias
0 input layer (fitur dari defect dan pusat massa)
1 lakukan normalisasi input
2 lakukan pemrosesan hidden layer
3 lakukan pemrosesan output layer
4 case output:
5     jika output 00000 = A
6     jika output 00001 = B
7     .
8     .
9     jika output 11001 = Z
    
```

Gambar 4. Algoritma Sistem

C. Thresholding jarak depth sensor

Pada pemrograman digunakan *library* OpenNI untuk mengakses kamera *depth kinect*. Kemudian dilakukan percobaan untuk mendapatkan jarak yang ideal untuk pengambilan data dan pengujian sistem. Berdasarkan percobaan, jarak minimum yang dipilih adalah 50cm dan jarak maksimum adalah 65cm. Gambar depth yang dihasilkan oleh OpenNI dan dan OpenCV tampak berbeda karena OpenCV hanya mengakses dari segi dua dimensi sedangkan OpenCV sudah terintegrasi oleh perhitungan sensor IR pada kinect sehingga tampak seperti tiga dimensi.



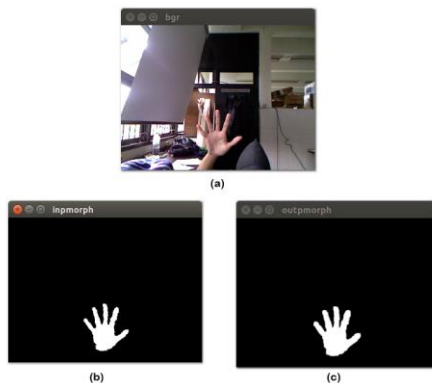
Gambar 5. Hasil gambar *depth* yang ditampilkan ketika objek memasuki area *threshold* dengan jarak yang berbeda-beda

Pengujian radius dilakukan untuk mengetahui hasil dari kontur tangan yang didapatkan ketika diberikan *threshold* jarak pada *depth* sensor kinect.

Mengikuti urutan gambar tangan (kiri ke kanan). Pada percobaan pertama tangan memasuki area *threshold* hingga batas siku, pada gambar kedua tangan hingga pergelangan tangan, dan yang ketiga ketika kepala dan bahu juga masuk ke daerah *threshold* sensor.

D. Pembentukan Kontur Menggunakan Morfologi

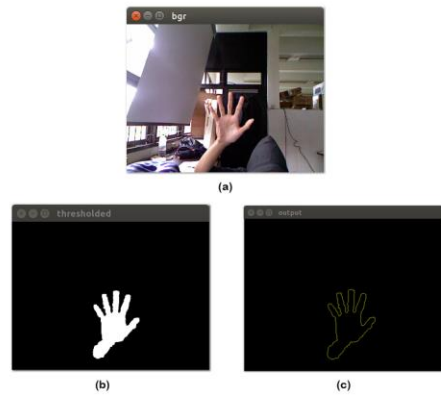
Hasil gambar tangan yang sudah memasuki area *threshold* masih akan ditransformasi menggunakan morfologi untuk mendapatkan bentuk yang lebih baik. Pada studi ini dilakukan tiga kali morfologi yaitu *erode*, *closing* (*erode* kemudian *dilate*), dan *dilate*. Bentuk kernel yang digunakan adalah *rectangular* dengan ukuran 3x3 untuk *erode* dan 8x8 untuk *dilate*.



Gambar 6. Proses morfologi (a) gambar BGR; (b) gambar *depth* yang belum diolah morfologi; (c) gambar *depth* setelah diolah morfologi

E. Pembentukan Garis Kontur

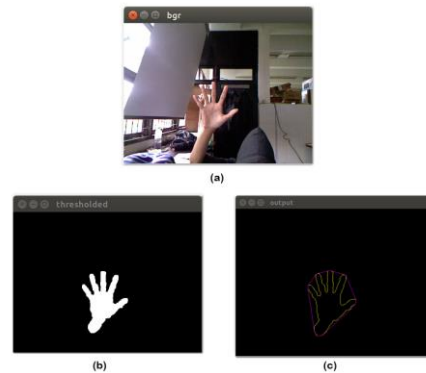
Setelah gambar melalui proses morfologi dilakukan proses untuk mendapatkan garis kontur menggunakan fungsi dari OpenCV yaitu *findContours*(). Hasil dari *findContours*() kemudian diproses lagi untuk menghasilkan garis tepi dengan fungsi *drawContours*().



Gambar 7. *Draw line contour* (a) gambar BGR; (b) gambar *depth*; (c) hasil penggambaran garis tepi kontur tangan

F. Pembentukan Garis Hull

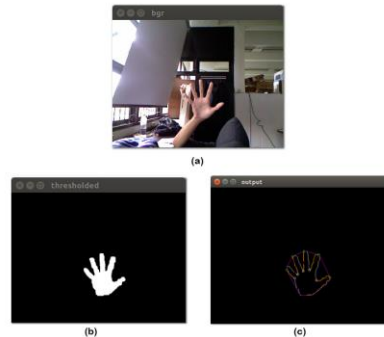
Kontur yang didapatkan fungsi *findContours*() dapat diolah untuk mendapatkan garis hull dengan fungsi *convexHull*() dimana fungsi ini menghubungkan titik-titik terjauh yang didapatkan dari output fungsi *findContours*(). Kemudian *convex* yang didapat akan digambarkan oleh fungsi *drawContours*() sehingga didapatkan garis poligonal yang melingkupi gambar tangan pada tampilan.



Gambar 8. *Draw convex hull* (a) gambar BGR; (b) gambar *depth*; (c) hasil penggambaran garis hull

G. Pembentukan Defects

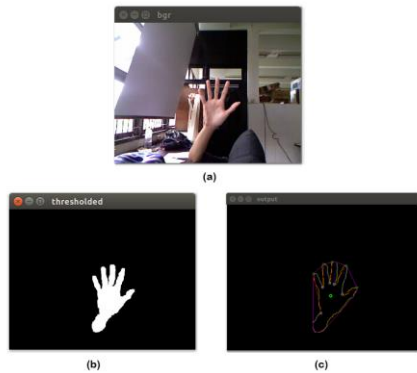
Pembentukan *defects* didapatkan oleh fungsi *convexityDefects*() dalam iterasi fungsi *convexHull*() dimana fungsi ini menghasilkan 4 elemen vektor. Tiga elemen vektor dari fungsi *convexityDefects*() dapat digambar dan diolah nilainya yaitu *start index*, *end index*, *farthest point index*.



Gambar 9. *Draw convexity defects* (a) gambar BGR; (b) gambar *depth*; (c) penambahan titik *start point*, *end point*, dan *farthest point*

H. Menentukan Pusat Massa

Pusat massa pada suatu gambar dapat ditemukan menggunakan fungsi moments() dari OpenCV. Moments menggunakan metode rata-rata massa dari intensitas piksel gambar[]]. Secara matematis perhitungan pusat



Gambar 10. Draw convexity defects (a) gambar BGR; (b) gambar depth; (c) penambahan titik moments sebagai pusat massa pada kontur tangan

I. Perancangan ANN

Fitur-fitur yang didapatkan dari convexity defects dan moments merupakan nilai yang dapat diolah oleh algoritma ANN. Berdasarkan fitur-fitur yang didapatkan akan dirancang pengujian ANN berdasarkan jumlah input, hidden layer, dan node. Pengujian berdasarkan jumlah input akan dibagi menjadi 2 yaitu menggunakan 8 input dan 3 input. Pada ANN 8 input, digunakan nilai ptStart.x, ptStart.y, ptEnd.x, ptEnd.y, ptFar.x, ptFar.y, mc.x, dan mc.y. Pada ANN 3 input, digunakan nilai jarak mc-start, mc-far, dan luas dari kontur tangan dengan bentuk rectangular. Input-input ini akan diolah pada ANN dengan output sebanyak 5. Berikut rancangan target dari ANN.

Tabel 1. Perencanaan target output pada ANN

Huruf	Target	Huruf	Target
A	00000	N	01101
B	00001	O	01110
C	00010	P	01111
D	00011	Q	10000
E	00100	R	10001
F	00101	S	10010
G	00110	T	10011
H	00111	U	10100
I	01000	V	10101
J	01001	W	10110
K	01010	X	10111
L	01011	Y	11000
M	01100	Z	11001

IV. PENGUJIAN DAN ANALISIS

A. Kondisi Ideal Pengujian

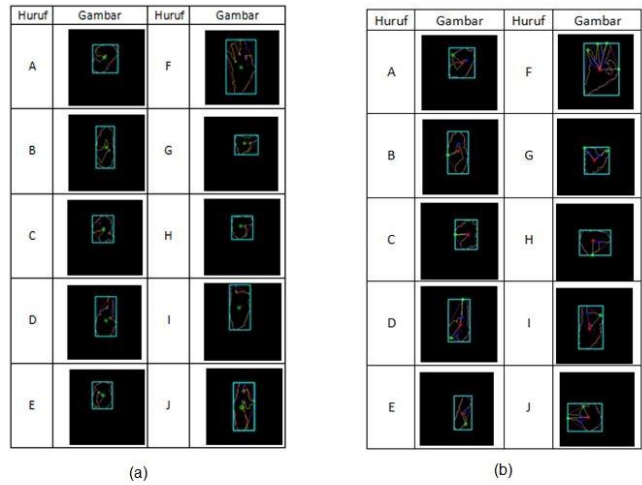
Pada studi ini ditentukan kondisi ideal untuk melakukan pengujian yaitu:

1. Hanya ada satu orang di depan kinect.
2. Pengguna hanya bisa mengulurkan tangan ke area

pemrosesan kamera.

3. Gambar tangan yang tertangkap berada tepat ditengah.

B. Pengujian ANN



Gambar 11. (a) Tabel input mulai huruf A sampai J untuk ANN 8 input; (b) Tabel input mulai huruf A sampai J untuk ANN 8 input

Tabel 2. Hasil Pengujian ANN 8 Input Untuk 4 Huruf dan 10 Training Set

Pengujian	Pertama	Ke-2	Ke-3
Jumlah Input	8	8	8
Jumlah Output	5	5	5
Jumlah Huruf	4	4	4
Training Set Perhuruf	10	10	10
Banyak Iterasi	5000	5000	5000
Jumlah Layer	2	8	3
Jumlah Node	8--2--5	8--8--5	8--4--4--5
Fungsi Aktivasi	Log Sigmoid - Linear	Log Sigmoid - Linear	Log Sig-Log Sig-Linear
Target Sesuai	40	40	40
Target Tidak Sesuai	0	0	0
Persentase Target	100%	100%	100%

Tabel 3. Hasil Pengujian ANN 8 Input Untuk 4 Huruf dan 40 Training Set

Pengujian	Pertama	Ke-2	Ke-3
Jumlah Input	8	8	8
Jumlah Output	5	5	5
Jumlah Huruf	4	4	4
Training Set Perhuruf	40	40	40
Banyak Iterasi	5000	5000	5000
Jumlah Layer	2	2	3
Jumlah Node	8--2--5	8--8--5	8--4--4--5
Fungsi Aktivasi	Log Sigmoid - Linear	Log Sigmoid - Linear	Log Sig-Log Sig-Linear
Target Sesuai	158	160	160
Target Tidak Sesuai	2	0	0
Persentase Target	98.75%	100%	100%

Tabel 4.
Hasil Pengujian ANN 8 Input Untuk 16 Huruf

Pengujian	Pertama	Ke-2
Jumlah Input	8	8
Jumlah Output	5	5
Jumlah Huruf	16	16
Training Set Perhuruf	40	40
Banyak Iterasi	5000	20000
Jumlah Layer	2	2
Jumlah Node	8--2--5	8--2--5
Fungsi Aktivasi	Log Sigmoid - Linear	Log Sigmoid - Linear
Target Sesuai	40	40
Target Tidak Sesuai	600	600
Persentase Target	6.25%	6.25%

Tabel 5.
Hasil Pengujian ANN 3 Input Untuk 5 Huruf

Pengujian	Pertama	Ke-2	Ke-3
Jumlah Input	3	3	3
Jumlah Output	5	5	5
Jumlah Huruf	5	5	5
Training Set Perhuruf	10	10	10
Banyak Iterasi	5000	5000	5000
Jumlah Layer	2	2	3
Jumlah Node	3--2--5	3--8--5	3--4--4--5
Fungsi Aktivasi	Log Sigmoid - Linear	Log Sigmoid - Linear	Log Sig-Log Sig-Linear
Target Sesuai	26	47	50
Target Tidak Sesuai	24	3	0
Persentase Target	52%	94%	100%

Tabel 6.
Hasil Pengujian ANN 3 Input Untuk 10 Huruf

Pengujian	Pertama	Ke-2	Ke-3
Jumlah Input	3	3	3
Jumlah Output	5	5	5
Jumlah Huruf	10	10	10
Training Set Perhuruf	10	10	10
Banyak Iterasi	5000	5000	5000
Jumlah Layer	2	2	3
Jumlah Node	3--2--5	3--8--5	3--4--4--5
Fungsi Aktivasi	Log Sigmoid - Linear	Log Sigmoid - Linear	Log Sig-Log Sig-Linear
Target Sesuai	10	10	20
Target Tidak Sesuai	90	90	80
Persentase Target	10%	10%	20%

C. Pembahasan Pengujian

Pada pengujian ANN 8 input, input yang digunakan yaitu ptStart.x, ptStart.y, ptEnd.x, ptEnd.y, ptFar.x, ptFar.y, mc.x, dan mc.y. Nilai ptStart, ptEnd, dan ptFar diperoleh dari

convexity defects sedangkan mc diperoleh dari perhitungan pusat massa gambar menggunakan moments. Masing-masing huruf memiliki karakteristik nilai yang berbeda sehingga huruf-huruf ini dilatih menggunakan ANN dengan jumlah *training set* yang beragam perhurufnya.

Pada pengujian ANN 3 input, input yang digunakan yaitu jarak titik ptStart ke mass center, jarak titik ptFar ke mass center, dan luas dari kotak yang melingkupi kontur. Pada pengujian sebelumnya data yang diolah berupa koordinat-koordinat dari gambar tangan yang didapat sedangkan pada pengujian ini data yang diolah adalah garis sehingga memiliki kondisi yang berbeda saat pengambilan datanya.

V. KESIMPULAN

Pada pengujian ini diujikan suatu sistem yang dapat mengenali gestur tangan menggunakan kamera kinect. Penggunaan *depth sensor* dari kinect dapat dimanfaatkan untuk mendapatkan kontur tangan. Pengambilan kontur tangan membutuhkan metode lagi agar dapat dijadikan data tertentu sehingga data tersebut dapat dilatih dengan *artificial neural network*. Data yang dilatih ini yang menentukan tingkat keberhasilan dari sistem ini. Berdasarkan pengujian percobaan dengan jumlah input yang lebih banyak menghasilkan hasil yang lebih akurat dengan jumlah huruf yang hampir sama. Hasil yang dicapai sistem ini masih belum akurat karena proses pelatihan membutuhkan parameter-parameter yang lebih kompleks agar proses pengenalan dapat memberikan hasil yang lebih baik.

DAFTAR PUSTAKA

- [1] F. Liu, "Hand Gesture Recognition Using Kinect via Deterministic Learning," *Longyan Univ.*, 2017.
- [2] M. Y. Andrian, "Penerjemah Isyarat Indonesia Menggunakan Kamera Telepon Genggam Android," *Inst. Teknol. Sepuluh Nop.*, 2016.
- [3] A. A. S. Gunawan, "Pembelajaran Bahasa Isyarat Dengan Kinect dan Metode Tim Warping," *Univ. Binus*, 2013.
- [4] J. A, "Kinect for Windows SDK Programming Guide," pp. 8–11, 2012.
- [5] G. and A. K. Bradsk, "Learning OpenCV: Computer Vision with the OpenCVLibrary," *O'Reilly Media*, p. 2008, 115AD.
- [6] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Ontario: Prentice Hall PTR, 1998.
- [7] G. K, *An Introduction to Neural Networks*. London, 1997.