

SISTEM PENERJEMAH SANDI *SEMAPHORE* MENGGUNAKAN SENSOR *KINECT* DENGAN PENGENALAN POLA DELAPAN TITIK

Ratna Aisuwarya^{1*}, Nadia Alfitri², Herry Wahyudi³

^{1,2,3}Jurusan Sistem Komputer Fakultas Teknologi Informasi Universitas Andalas

Jurusan Teknik Elektro Politeknik Negeri Padang

*Email : aisuwarya@fti.unand.ac.id

ABSTRAK

Salah satu bahasa non-verbal yang cukup terkenal yaitu sandi *Semaphore*. Namun tidak semua orang yang mengetahui sandi *Semaphore*, oleh karena itu diperlukan suatu sistem yang dapat menerjemahkan sandi *Semaphore* menjadi teks. Sistem ini menggunakan sensor *Kinect* yang digunakan untuk menangkap *image* objek. *Skeleton data* yang didapatkan dari *image* tersebut kemudian dimanfaatkan untuk mendapatkan koordinat *joint* (sendi) yang digunakan untuk membuat sebuah pola pengenalan sandi *Semaphore* menggunakan delapan titik. Pola ini dibuat dengan cara membuat delapan buah titik acuan disekitar tubuh objek yang membentuk lingkaran dengan titik pusat *joint* bahu tengah objek tersebut. Kemudian jika kedua tangan objek berada pada titik-titik acuan, maka program akan menerjemahkan menjadi huruf dengan cara mencocokkan posisi tangan objek sesuai dengan aturan sandi *semaphore*. Hasil terjemahan sandi *semaphore* dikirim ke arduino uno untuk ditampilkan pada led dot matriks 8x8.

Kata Kunci — *Kinect*, Penerjemah, *Semaphore*, *Skeletal Data*

ABSTRACT

One of the non-verbal language that is quite well known that semaphores code. But not everyone knows the semaphores code well, therefore a system that can translate semaphores code into text is needed. Kinect sensor is used to capture the image of the object. Then the image is processed to get skeleton data of the object that used to obtain the coordinates of the joints which are used to create a semaphores code recognition pattern using eight points. The pattern is made by making eight reference points around the objects that make up a circle with the center point is the middle-shoulder joint of the object. Then if the two objects are at hand reference points the program will translate into letters by matching the position of the object in accordance with the rules of the hands of the semaphore code. The translation of the semaphore code is sent to the Arduino Uno to be displayed on the LED dot matrix 8x8.

Keywords — *Kinect*, *Translator*, *Semaphore*, *Skeletal Data*

PENDAHULUAN

Sebagai makhluk sosial, manusia memiliki kebutuhan untuk berkomunikasi antar sesama manusia. Komunikasi yang biasa dilakukan menggunakan bahasa verbal, yaitu cara komunikasi dengan menggunakan kata – kata. Namun pada kondisi tertentu bahasa verbal tidak efektif digunakan sehingga digunakan bahasa non-verbal, yaitu komunikasi yang dilakukan tanpa kata-kata contohnya sandi sandi *Semaphore*. Sandi *Semaphore* merupakan sebuah bahasa isyarat yang menggunakan dua buah bendera untuk menyampaikan pesan.

Untuk membuat sebuah sistem yang dapat menerjemahkan sandi *Semaphore* tersebut, digunakan *Microsoft Kinect* sebagai sensor yang menangkap gerakan manusia. Data

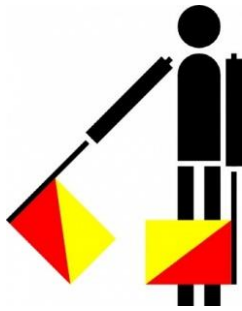
keluaran dari sensor *Kinect* tersebut kemudian diolah dan diterjemahkan oleh komputer dengan pengenalan pola delapan titik. Pengolahan data tersebut pada komputer menggunakan bahasa pemrograman Microsoft Visual C#. Hasil data penerjemahan oleh komputer kemudian dikirim ke Arduino Uno untuk kemudian ditampilkan pada LED dot matrik 8x8.

LANDASAN TEORI

2.1 Sandi *Semaphore*

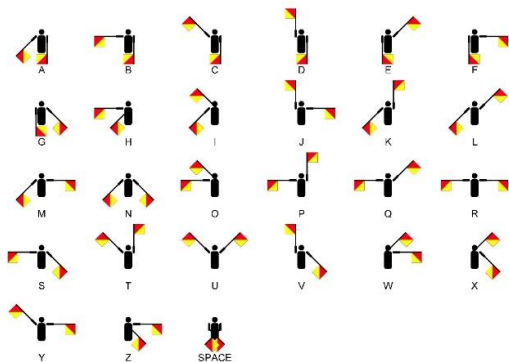
Sandi *Semaphore* adalah suatu cara untuk mengirim dan menerima berita dengan menggunakan 2 bendera, dimana masing – masing bendera tersebut berukuran 45 cm x 45 cm [1]. Sedangkan warna yang sering dipakai adalah merah dan kuning dengan warna merah

selalu berada dekat dengan tangkainya yang dapat dilihat seperti pada gambar 1.



Gbr. 1 Bendera Semaphore

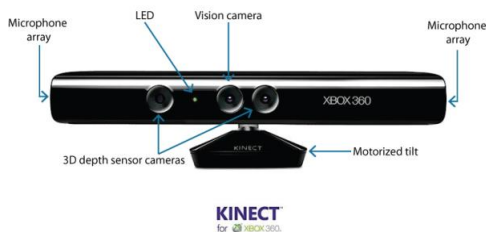
Sandi *Sempahore* yang digunakan dalam tulisan ini sebanyak 27 sandi yang terdiri dari 26 sandi alphabet dan 1 spasi seperti yang terlihat pada gambar 2.



Gbr. 2 Contoh Sandi *Semaphore*

2.2 Microsoft *Kinect*

Microsoft *Kinect for Xbox 360*, atau biasa disebut sebagai “*Kinect*”, merupakan sebuah sensor pada *console Xbox 360* yang memperkenalkan *motion gaming* sebagai fitur utamanya [2]. *Kinect* membuat pemain dapat berinteraksi dengan *Xbox 360* tanpa menggunakan *game controller*. Dengan menggunakan *Kinect* pemain dapat bermain *Xbox 360* cukup hanya dengan menggunakan gerakan tangan atau gerakan tubuh lainnya.



Gbr. 3 Microsoft *Kinect* for Xbox 360

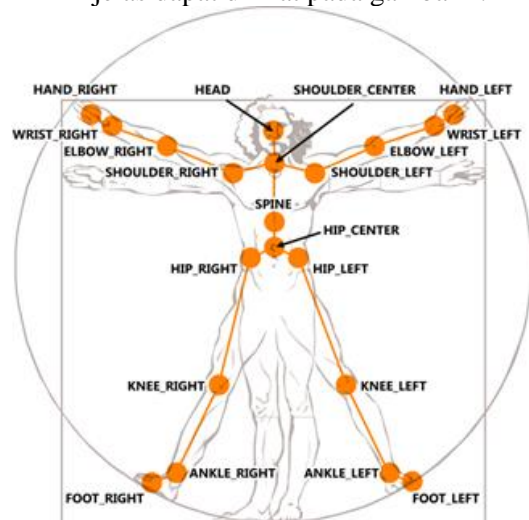
Kinect dilengkapi dengan kamera RGB, sensor kedalaman 3D, mikrofon multi-array

untuk menangkap dan mengenali suara, dan sebuah motor penggerak yang dapat dilihat pada gambar 3. Teknologi sensor kedalaman (*depth sensor*) *Kinect* merupakan sensor tiga dimensi untuk mengenali gerakan pemain [3]. Sensor ini dapat mengenali sampai enam orang sekaligus, namun hanya dua pemain berstatus aktif yang dapat dideteksi gerakannya oleh *Kinect*.

Depth sensor awalnya akan memetakan objek yang ditangkapnya berdasarkan jarak seperti gambar. Kemudian gambar tersebut akan dibandingkan dengan data hasil *training* (proses perekaman data) sebelumnya [4]. Data *training* tersebut dibuat oleh para peneliti dengan menggunakan 100.000 *frame* gambar objek manusia yang diambil dengan posisi yang berbeda – beda.

Mekanisme cara kerja *Kinect* dalam menangkap dan mengenali pergerakan tubuh dari pemain adalah sebagai berikut [5][6] :

1. Posisi pemain harus menghadap sensor *Kinect* terlebih dahulu
2. Kemudian sensor *Kinect* menangkap titik – titik pada tubuh yang disebut dengan *joint*. Ada 20 titik pada tubuh yang dapat dideteksi oleh *Kinect* yaitu kepala, bahu tengah, bahu kanan, bahu kiri, siku kanan, siku kiri, pergelangan tangan kanan, pergelangan tangan kiri, tangan kanan, tangan kiri, tulang punggung, pinggul tengah, pinggul kanan, pinggul kiri, lutut kanan, lutut kiri, pergelangan kaki kanan, pergelangan kaki kiri, kaki kanan dan kaki kiri. Untuk posisi lebih jelas dapat dilihat pada gambar 4.



Gbr. 4 *Joint* Pada Tubuh Manusia

2.3 Pengenalan Pola Menggunakan Lingkaran Delapan Titik

Dengan memanfaatkan kemampuan sensor *Kinect* yang dapat mendeteksi tubuh manusia beserta sendi – sendinya, dapat dibuat sebuah metode yang dapat mengenali pola Sandi *Semaphore* tersebut. Dalam proses pembuatan pola penerjemah Sandi *Semaphore* ini diperlukan koordinat beberapa *joint* manusia yang menjadi objek. *Joint – joint* tersebut adalah :

1. *Joint ShoulderCenter* (bahu tengah)

Bahu tengah dijadikan sebagai pusat dari delapan lingkaran yang berada disekitar objek

2. *Joint RightHand* (tangan kanan)

Posisi tangan kanan diperlukan dalam pencocokan pola sandi *Semaphore*

3. *Joint LeftHand* (tangan kiri)

Posisi tangan kiri diperlukan dalam pencocokan pola sandi *Semaphore*

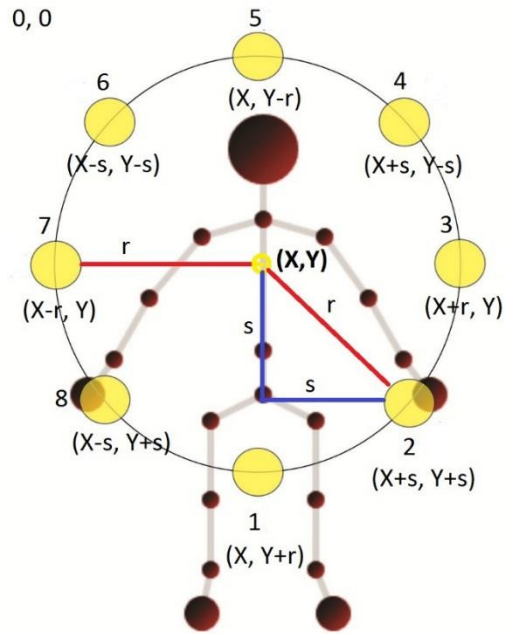
4. *Joint Spine* (punggung)

Posisi *joint ShoulderCenter* terhadap sumbu Y dikurangi posisi *joint Spine* terhadap sumbu Y dikali dua, digunakan sebagai jari – jari antara pusat pola (*joint ShoulderCenter*) dengan delapan lingkaran

Dari koordinat *joint – joint* tersebut proses selanjutnya yaitu membuat delapan titik acuan pola seperti pada gambar 5 untuk menerjemahkan sandi *Semaphore*. Setelah mendapatkan titik pusat dengan koordinat (X,

Y), jari - jari ($r = \sqrt{\frac{r^2}{2}}$) dan s yaitu $\sqrt{\frac{r^2}{2}}$, maka delapan titik acuan sandi *Semaphore* dapat diposisikan. Kedelapan titik acuan diberi nama “Satu”, ”Dua”, ”Tiga”, ”Empat”, ”Lima”, ”Enam”, ”Tujuh”, ”Delapan” dengan posisi sebagai berikut :

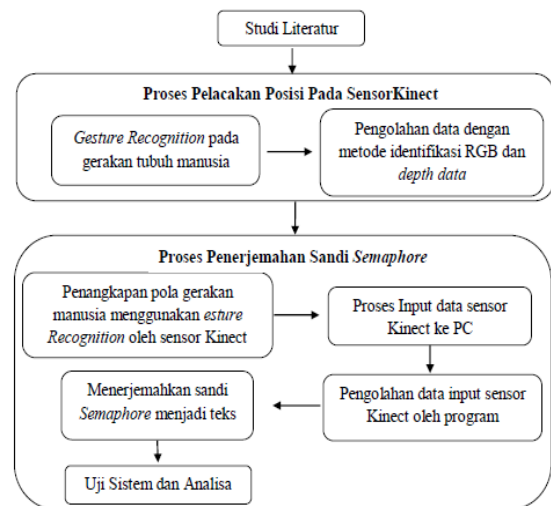
- a. Titik Satu berada pada koordinat (X, Y+r).
- b. Titik Dua berada pada koordinat (X+s, Y+s).
- c. Titik Tiga berada pada koordinat (X+r, Y).
- d. Titik Empat berada pada koordinat (X+s, Y-s).
- e. Titik Lima berada pada koordinat (X, Y-r).
- f. Titik Enam berada pada koordinat ((X-s, Y-s).
- g. Titik Tujuh berada pada koordinat (X-r, Y).
- h. Titik Delapan berada pada koordinat (X-s, Y+s).



Gbr. 5 Pola Penerjemah Sandi *Semaphore* Menggunakan Delapan Titik

METODOLOGI PENELITIAN

Mengacu pada pengklasifikasian metodologi penelitian oleh Zhang, Huang, dan Yu[7], penelitian ini menggunakan metode *Experiment*. Yaitu, pengklasifikasian suatu objek berdasarkan ciri, perancangan dan pengembangan infrastruktur, analisis dan pengukuran kinerja alat, dan penerapan dan optimasi algoritma. Pada gambar 6 berikut dijabarkan tahapan-tahapan yang dilakukan dalam keseluruhan perancangan sistem penerjemah sandi *Semaphore*.



Gbr. 6. Disain Penelitian

1.1. Proses Penerjemahan Sandi Semaphore

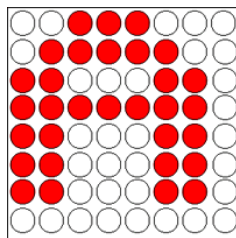
Setelah dapat menentukan titik pusat serta titik acuan sandi Semaphore, maka proses penerjemahan sandi dapat dilakukan. Dalam proses pencocokan ini dibuat toleransi pembacaan jarak antara *joint* tangan *user* dan titik acuan sebesar *t*. Tabel 1 menampilkan detail pencocokan koordinat untuk setiap huruf dari A sampai Z.

3.2 Proses Tampilan Pada Led Dot Matriks 8x8

Untuk menampilkan sebuah karakter pada led dot matrik menggunakan arduino sebelumnya diinisialisasi pin – pin modul MAX7219 yaitu pin DIN, CS, CLK yang pada penelitian ini akan di hubungkan ke Port 8, 9 dan 10.

```
int Max7219_pinCLK = 10;
int Max7219_pinCS = 9;
int Max7219_pinDIN = 8;
```

Setelah pin terinisialisasi selanjutnya yaitu menentukan led yang hidup dan mati sehingga tersusun membentuk sebuah karakter yang diinginkan.



Gbr. 7 Contoh Tampilan Huruf A Pada Led Dot Matriks 8x8

Tabel 1

Rumus Pencocokan Koordinat Sandi Semaphore

Pada modul MAX7219 led dihidupkan baris perbaris. Untuk menampilkan huruf A pada led dot matriks 8x8 seperti gambar 6 memerlukan input seperti berikut :

- Baris pertama : 0011 1000 b atau 48h
- Baris kedua : 0111 1100 b atau 7Ch
- Baris ketiga : 1100 0110 b atau C6h
- Baris keempat : 1111 1110 b atau FEh
- Baris kelima : 1100 0110 b atau C6h
- Baris keenam : 1100 0110 b atau C6h

Baris ketujuh : 1100 0110 b atau C6h

Huru f	Posisi <i>Joint</i> Tangan Kanan (X1, Y1)	Posisi <i>Joint</i> Tangan Kiri (X2, Y2)
A	$(X-s)-t \leq X1 \leq (X-s)+t$ $(Y+s)-t \leq Y1 \leq (Y+s)+t$	$(X-t) \leq X2 \leq (X+t)$ $((Y+r)-t) \leq Y2 \leq ((Y+r)+t)$
B	$(X-r)-t \leq X1 \leq (X-r)+t$ $Y-t \leq Y1 \leq Y+t$	$(X-t) \leq X2 \leq (X+t)$ $((Y+r)-t) \leq Y2 \leq ((Y+r)+t)$
C	$(X-s)+t \leq X1 \leq (X-s)+t$ $(Y-s)-t \leq Y1 \leq (Y-s)+t$	$(X-t) \leq X2 \leq (X+t)$ $((Y+r)-t) \leq Y2 \leq ((Y+r)+t)$
D	$X-t \leq X1 \leq X+t$ $(Y-r)-t \leq Y1 \leq (Y-r)+t$	$(X-t) \leq X2 \leq (X+t)$ $((Y+r)-t) \leq Y2 \leq ((Y+r)+t)$
E	$(X-t) \leq X1 \leq (X+t)$ $((Y+r)-t) \leq Y1 \leq ((Y+r)+t)$	$(X+s)-t \leq X2 \leq (X+s)+t$ $(Y-s)-t \leq Y2 \leq (Y-s)+t$
F	$(X-t) \leq X1 \leq (X+t)$ $((Y+r)-t) \leq Y1 \leq ((Y+r)+t)$	$(X+r)-t \leq X2 \leq (X+r)+t$ $Y-t \leq Y2 \leq Y+t$
G	$(X-t) \leq X1 \leq (X+t)$ $((Y+r)-t) \leq Y1 \leq ((Y+r)+t)$	$(X+s)-t \leq X2 \leq (X+s)+t$ $(Y+s)-t \leq Y2 \leq (Y+s)+t$
H	$(X-r)-t \leq X1 \leq (X-r)+t$ $Y-t \leq Y1 \leq Y+t$	$(X-s)-t \leq X2 \leq (X-s)+t$ $(Y+s)-t \leq Y2 \leq (Y+s)+t$
I	$(X-s)-t \leq X1 \leq (X-s)+t$ $(Y+s)-t \leq Y1 \leq (Y+s)+t$	$(X-s)-t \leq X2 \leq (X-s)+t$ $(Y-s)-t \leq Y2 \leq (Y-s)+t$
J	$X-t \leq X1 \leq X+t$ $(Y-r)-t \leq Y1 \leq (Y-r)+t$	$(X+r)-t \leq X2 \leq (X+r)+t$ $Y-t \leq Y2 \leq Y+t$
K	$(X-s)-t \leq X1 \leq (X-s)+t$ $(Y+s)-t \leq Y1 \leq (Y+s)+t$	$X-t \leq X2 \leq X+t$ $(Y-r)-t \leq Y2 \leq (Y-r)+t$
L	$(X-s)-t \leq X1 \leq (X-s)+t$ $(Y+s)-t \leq Y1 \leq (Y+s)+t$	$(X+s)-t \leq X2 \leq (X+s)+t$ $(Y-s)-t \leq Y2 \leq (Y-s)+t$
M	$(X-s)+t \leq X1 \leq (X-s)+t$ $(Y+s)-t \leq Y1 \leq (Y+s)+t$	$(X+r)-t \leq X2 \leq (X+r)+t$ $Y-t \leq Y2 \leq Y+t$
N	$(X-s)-t \leq X1 \leq (X-s)+t$ $(Y+s)-t \leq Y1 \leq (Y+s)+t$	$(X+s)-t \leq X2 \leq (X+s)+t$ $(Y+s)-t \leq Y2 \leq (Y+s)+t$
O	$(X-r)-t \leq X1 \leq (X-r)+t$ $Y-t \leq Y1 \leq Y+t$	$(X-s)-t \leq X2 \leq (X-s)+t$ $(Y-s)-t \leq Y2 \leq (Y-s)+t$
P	$(X-r)-t \leq X1 \leq (X-r)+t$ $Y-t \leq Y1 \leq Y+t$	$X-t \leq X2 \leq X+t$ $(Y-r)-t \leq Y2 \leq (Y-r)+t$
Q	$(X-r)-t \leq X1 \leq (X-r)+t$ $Y-t \leq Y1 \leq Y+t$	$(X+s)-t \leq X2 \leq (X+s)+t$ $(Y-s)-t \leq Y2 \leq (Y-s)+t$
R	$(X-r)-t \leq X1 \leq (X-r)+t$ $Y-t \leq Y1 \leq Y+t$	$(X+r)-t \leq X2 \leq (X+r)+t$ $Y-t \leq Y2 \leq Y+t$
S	$(X-r)-t \leq X1 \leq (X-r)+t$ $Y-t \leq Y1 \leq Y+t$	$(X+s)-t \leq X2 \leq (X+s)+t$ $(Y+s)-t \leq Y2 \leq (Y+s)+t$
T	$(X-s)-t \leq X1 \leq (X-s)+t$ $(Y-s)-t \leq Y1 \leq (Y-s)+t$	$X-t \leq X2 \leq X+t$ $(Y-r)-t \leq Y2 \leq (Y-r)+t$
U	$(X-s)-t \leq X1 \leq (X-s)+t$ $(Y-s)-t \leq Y1 \leq (Y-s)+t$	$(X+s)-t \leq X2 \leq (X+s)+t$ $(Y-s)-t \leq Y2 \leq (Y-s)+t$
V	$X-t \leq X1 \leq X+t$ $(Y-r)-t \leq Y1 \leq (Y-r)+t$	$(X+s)-t \leq X2 \leq (X+s)+t$ $(Y+s)-t \leq Y2 \leq (Y+s)+t$
W	$(X+s)+t \leq X1 \leq (X+s)+t$ $(Y-s)-t \leq Y1 \leq (Y-s)+t$	$(X+r)-t \leq X2 \leq (X+r)+t$ $Y-t \leq Y2 \leq Y+t$
X	$(X+s)+t \leq X1 \leq (X+s)+t$ $(Y-s)-t \leq Y1 \leq (Y-s)+t$	$(X+s)-t \leq X2 \leq (X+s)+t$ $(Y+s)-t \leq Y2 \leq (Y+s)+t$
Y	$(X-s)-t \leq X1 \leq (X-s)+t$ $(Y-s)-t \leq Y1 \leq (Y-s)+t$	$(X+r)-t \leq X2 \leq (X+r)+t$ $Y-t \leq Y2 \leq Y+t$
Z	$(X+s)-t \leq X1 \leq (X+s)+t$ $(Y+s)-t \leq Y1 \leq (Y+s)+t$	$(X+r)-t \leq X2 \leq (X+r)+t$ $Y-t \leq Y2 \leq Y+t$

Baris kedelapan : 0000 0000 b
atau 00h

Berikut merupakan bilangan deretan heksadesimal yang digunakan pada tulisan ini untuk menampilkan huruf “A” sampai “Z” pada led dot matriks 8x8 :

HASIL DAN PEMBAHASAN

4.1 Pengujian Kinect

Untuk pengujian ini akan digunakan program sederhana yang memberikan *output* apakah tangan kanan lebih tinggi dari tangan kanan atau sebaliknya tangan kanan lebih rendah dari tangan kiri. Gambar 8 dan 9 adalah tampilan program pengujian untuk menguji kemampuan Kinect mengenali tangan kanan dan tangan kiri serta koordinat tangan terhadap sumbu-Y.



Gbr. 8 Tampilan Pengujian Tangan Kiri



Gbr. 9 Tampilan Pengujian tangan Kanan

4.2 Pengujian Toleransi Pencocokan Koordinat

Pada saat objek bersikap membuat sebuah sandi *Semaphore*, posisi tangan objek tersebut sangat sulit untuk sesuai dengan titik acuan sandi secara sempurna. Oleh karena itu untuk membuat program penerjemah berfungsi

dengan baik harus memberikan toleransi pencocokan agar program dapat menerjemahkan sandi walaupun posisi objek tidak sempurna.

Besar nilai toleransi bukan merupakan nilai tetap. Hal ini dikarenakan jarak antara sensor Kinect dan objek yang tidak tetap, semakin jauh objek dari sensor Kinect maka semakin kecil jari – jari antara titik tengah tubuh dan titik acuan penerjemahan. Sehingga jika nilai toleransi diberikan nilai tetap maka akan terjadi kesalahan pada penerjemahan seperti pembacaan ganda.

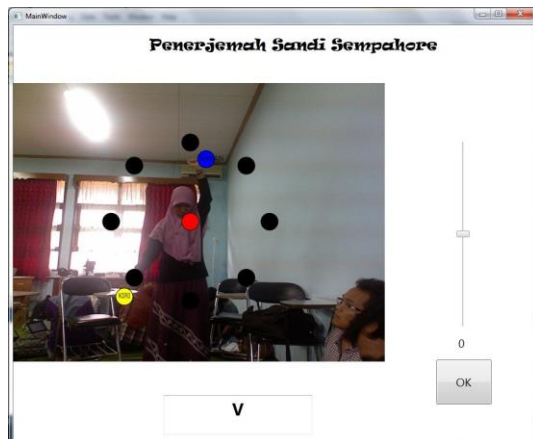
Oleh karena itu perlu dilakukan pengujian penerjemahan terhadap beberapa orang dengan toleransi berbeda. Pengujian hanya dilakukan untuk 4 huruf yaitu huruf “A”, “Q”, “V”, “Y” dimana 4 huruf tersebut mewakili kedelapan titik acuan secara berbeda. Nilai jari – jari antara titik pusat tubuh dan titik acuan penerjemahan (r) akan menjadi variabel dalam perhitungan toleransi. Pengujian akan dilakukan dengan lima toleransi yang berbeda untuk tiap objek.

Berdasarkan pengujian yang telah dilakukan, diperoleh tingkat keberhasilan tertinggi pada toleransi $t=0,5r$. Namun nilai toleransi sebesar $0,5r$ ini tidak cocok digunakan karena memungkinkan terjadinya kesalahan pada saat pencocokan pola. Hal ini disebabkan adanya irisan antara dua buah titik acuan yang bersebelahan. Sebagai contoh pada sandi *Semaphore* huruf “A” akan terjadi kemungkinan salah penterjemahan menjadi huruf “B” atau huruf “H”.

Oleh karena itu berdasarkan pengujian yang telah dilakukan diperoleh bahwa rata – rata keberhasilan penterjemahan paling tinggi selain pada $0,5r$ yaitu pada toleransi $t=0,4 \times r$. Oleh karena itu pada program akan ditetapkan toleransi sebesar $0,4r$.

4.3 Implementasi Program

Sistem yang dibangun membutuhkan dua buah program yaitu program pada Laptop menggunakan bahasa C# untuk menerjemahkan sandi Semaphore dari gerakan manusia yang ditangkap oleh sensor Kinect dan program pada Arduino untuk menampilkan hasil terjemahan sandi Semaphore dalam bentuk huruf pada led dot matriks. Gambar 10 merupakan tampilan program penerjemah sandi *Semaphore* ketika menerjemahkan huruf “V”:



Gbr. 10 Tampilan Program Ketika Menerjemahkan Sandi Semaphore

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan dan pengujian yang telah dilakukan pada sistem yang telah dibuat, maka dapat disimpulkan bahwa :

1. *Skeletal Tracking* sensor Kinect dapat diterapkan pada penerjemah sandi Semaphore
2. Untuk dapat menerjemahkan sandi *Semaphore* dengan baik diperlukan toleransi pembacaan koordinat sebesar 40% dari jari-jari pola lingkaran delapan titik
3. *Sistem* penerjemah sandi Semaphore menggunakan sensor Kinect dapat bekerja dengan tingkat keberhasilan sebesar 70,19%

5.2 Saran

Untuk penelitian dan pengembangan sistem ini selanjutnya, dapat disarankan beberapa hal sebagai berikut :

1. Posisi joint yang ditampilkan seharusnya dapat sama persis sesuai dengan anggota

tubuh saat ditampilkan pada layar computer.

2. Membuat rumus yang lebih akurat agar pada saat pembacaan lebih akurat, terutama ketika mengharuskan objek menyilangkan tangan yang membuat penerjemahan sering tidak berhasil
3. Menggunakan media penampil lain yang lebih mudah dilihat dan menarik, karena led dot matriks yang digunakan sekarang masih tergolong kecil.

DAFTAR PUSTAKA

- , Bagian bagian sensor Kinect, <http://www.xbox.com/en-GB/kinect/>, Mei 2103
- , Trik Mudah Kuasai Semaphore. pramukanet.org, Diakses 18 Juli 2013.
- , Model motion gaming, <http://www.microsoft.com/en-us/kinectforwindows/>, Mei 2013
- Jean, Jared St. 2013. *Kinect Hacks*. O'Reilly Media Inc. United States of America
- Webb, Jarrett., and Ashley, James. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress.
- Du, Heng. To., dan TszHang. *Hand Gesture Recognition Using Kinect*. Boston University Department of Electrical and Computer Engineering, 8 Saint Mary's Street, Boston, MA 02215
- Tang, Matthew. 2011. *Recognizing Hand Gestures with Microsoft's Kinect*. Stanford University, Department of Electrical Engineering
- Zhang, S., Huang, L. H., dan Yu, D. H., 2006, *An Analysis of Information Systems Research in Chinese Mainland*, Communications of the AIS, vol. 17, pp 785800.