

## SISTEM APLIKASI PERHITUNGAN SUDUT PADA STRAIGHT LEG RAISE TEST MENGGUNAKAN MOTION CAPTURE KINECT DAN BAHASA C#

Histinawty, Tavipia Rumambi dan Matrisnya Hermita  
Universitas Gunadarma  
Jl. Margonda Raya No. 100, Depok, Jawa Barat 16424  
{hustina, tavipia, matrisnya}@staff.gunadarma.ac.id

### ABSTRAK

Pada beberapa orang sering mengalami nyeri pinggang bawah (*Low Back Pain*) yang disebabkan oleh aktivitas tubuh yang kurang baik. Nyeri pinggang bawah umumnya dapat diketahui lebih lanjut penyebabnya dengan *Straight Leg Raise Test (SLRT)* atau *Laseque Sign*. *SLRT* adalah pemeriksaan neurodinamik pada penderita untuk menilai iritasi akar saraf di daerah lumbosacral dengan cara berbaring pada tempat tidur dengan kedua tungkai diluruskan, kemudian mengangkat salah satu tungkai kaki sambil mempertahankan lutut tetap lurus. Pada orang normal, jika tidak merasakan nyeri dan dapat menahan hingga sudut kedua tungkai kaki sebesar  $70^\circ$ . Gerakan *SLRT* dapat direkam dengan teknologi *motion capture* Microsoft *Kinect Xbox 360*, karena kemampuan *Kinect* yang dapat melacak posisi sendi tubuh manusia. Untuk dapat mengukur besar sudut pada *SLRT* secara real time dan cepat maka dibangun suatu sistem aplikasi yang terintegrasi melalui *Kinect* dan dengan menggunakan Bahasa C# serta pendukung editor visual studio, XAML dan *Sqlite* agar dapat menampilkan besar sudut serta proses dari kalibrasi, perubahan citra terkalibrasi ke citra grayscale depth serta terlacaknya skeleton dan menyimpan dalam database. Aplikasi ini diharapkan dapat digunakan oleh praktisi medis terutama bidang rehabilitasi.

**Kata Kunci :** *Straight Leg Raise, Aplikasi, Kinect, Sudut*

### PENDAHULUAN

#### Latar Belakang.

Pada beberapa orang ada yang sering mengalami nyeri pinggang bawah (*Low Back Pain*). *Low Back Pain* merupakan salah satu gangguan tulang belakang yang disebabkan oleh aktivitas tubuh yang kurang baik [1]. Nyeri pinggang bawah umumnya dapat diketahui lebih lanjut penyebabnya dengan *Straight Leg Raise Test (SLRT)* atau *Laseque Sign*. *SLRT* adalah pemeriksaan neurodinamik untuk menilai iritasi akar saraf di daerah lumbosakral. Ini merupakan elemen integral dari pemeriksaan neurologis untuk pasien dengan nyeri punggung bawah dengan atau tanpa nyeri radikular. Nama lain yang kurang umum digunakan adalah tanda Lazarevic [2]. Menurut penelitian [3], pada *Laseque Sign* penderita berbaring pada tempat tidur atau meja pemeriksaan dokter dengan kedua tungkai diluruskan, kemudian mengangkat salah satu tungkai penderita sambil mempertahankan lutut tetap lurus. Pada orang normal tidak merasakan nyeri dan dapat menahan besar sudut kedua tungkai hingga besar sudut  $70^\circ$ . Jika penderita

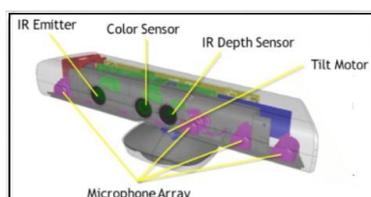
merasakan nyeri menjalar dari bokong hingga ke tungkai, maka sesuai dengan adanya *nerve sciatica* yaitu sebelum mencapai  $70^\circ$  dan dikatakan positif yang biasanya didapatkan pada penderita herniasi diskus pada tulang L5, S1 atau S2. Besar sudut memberikan indikasi ada keparahan iritasi akar saraf. Gerakan *SLRT* dapat direkam dengan teknologi *motion capture*. Salah satu *motion capture* yang digunakan adalah *Kinect Xbox 360* yang merupakan *Controller-Free Gaming* dan *Gaming Experience* yang dibuat oleh Microsoft. Dengan menggunakan *Kinect Xbox 360*, maka tubuh manusia akan terdeteksi sehingga dapat membuat informasi yang ingin disampaikan akan menjadi lebih baik karena objek dapat melihat sendiri bagaimana posisinya secara langsung. Gerakan *SLRT* tersebut serta sudut dapat dihitung dengan aplikasi yang menggunakan teknologi *motion capture Kinect*. Dengan kemampuan *Microsoft Kinect* yang dapat melacak posisi sendi tubuh manusia, dapat berguna sebagai alat untuk rehabilitasi stroke, baik dalam pengaturan klinis dan sebagai alat untuk membantu penderita

stroke dalam latihan mereka di rumah [4]. Penelitian [5] menggunakan *Kinect* dengan aplikasi antarmuka untuk melakukan tes rehabilitasi fisik yang dapat mengukur kemampuan seseorang untuk berjalan dalam rentang 10 meter dan pengukuran rentang gerakan sendi leher, bahu siku, paha atas, lutut. *SLRT* dengan media *Kinect* ini diharapkan dapat membuat pengguna aplikasi terutama praktisi medis di klinik dapat mengetahui informasi besaran sudut dalam waktu nyata, cepat dan akurat sehingga dapat menunjang diagnosis nyeri pinggang bawah.

### Tinjauan Pustaka.

#### *Motion Capture Kinect.*

*Motion capture* (penangkapan gerak) merupakan suatu proses perekaman dan penterjemahan informasi gerakan dan lokasi subjek dari waktu ke waktu menjadi model digital [6]. *Motion capture* banyak diaplikasikan, diantaranya antara lain pada produksi animasi, analisa gerakan atau industry dan juga di militer. Salah satu *motion capture* ini adalah *Kinect*; perangkat input untuk mendeteksi gerakan yang diproduksi oleh *Microsoft*. *Kinect* menggunakan kamera sensor yang dapat menangkap gerakan pengguna, tanpa pengguna tidak perlu menyentuh secara langsung *controller game*. *Kinect* merupakan perangkat elektronik yang terdiri dari proyektor infrared, kamera, dan sebuah microchip untuk mendeteksi gerakan obyek dalam 3 dimensi.



Gambar 1. Komponen Kinect

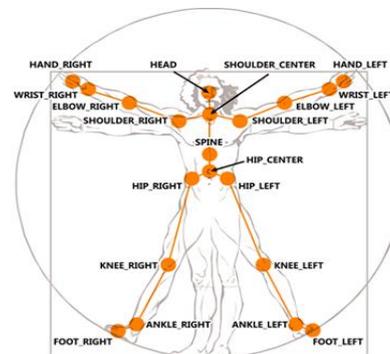
#### *Kinect XBOX 360*

Perangkat *motion capture Kinect XBOX 360* terdiri dari : [7]

1. Depth Sensing Sistem terdiri dari laser emitor IR dan kamera IR
2. RGB Camera
3. Motor Accelerometer dan Mikrofon

### Skeleton

*Kinect* memiliki kemampuan untuk melihat secara 3D melalui *depth sensor*. *Depth sensor* terdiri dari kamera inframerah, proyektor inframerah serta *skeletal tracking*. Objek manusia yang telah direkam oleh *Kinect*, maka *Skeletal Tracking* akan mengoptimalkan pelacakan objek yang menghadap *Kinect* untuk dijadikan kedalam bentuk *skeleton*. *Skeleton tracking* adalah pengolahan citra kedalam untuk mendirikan posisi sendi kerangka bentuk manusia. Setiap orang yang dideteksi akan menghasilkan satu set kerangka berisi 20 *control points* seperti pada gambar 2 dibawah ini [8].



Gambar 2. Skeleton Control Point

### Visual Studio

*Microsoft Visual Studio* merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi Windows, ataupun aplikasi Web. *Visual Studio* mencakup compiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). *Visual Studio* mendukung berbagai bahasa pemrograman dan memungkinkan kode editor dan debugger untuk mendukung hampir semua bahasa pemrograman. termasuk C # (via Visual C #).

### Bahasa Pemrograman C#

C# merupakan sebuah bahasa pemrograman yang berorientasi objek sebagai bagian dari inisiatif kerangka .NET Framework, framework inilah yang nanti digunakan untuk mengkompilasi dan

menjalankan kode C#. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain) dengan beberapa penyederhanaan.

### XAML

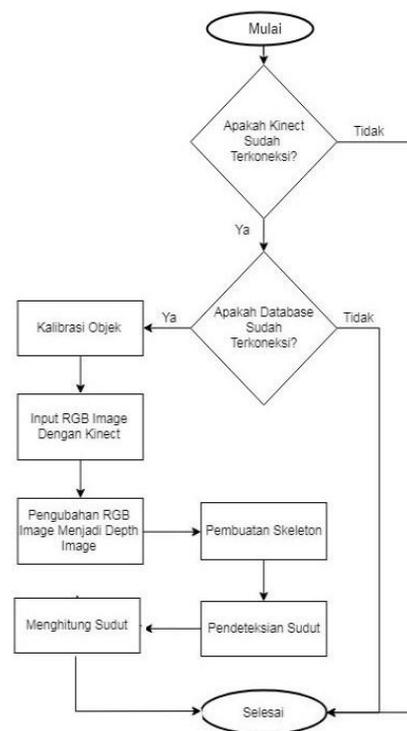
XAML adalah singkatan dari Extensible Application Markup Language. XAML bahasa markup deklaratif yang diaplikasikan pada model pemrograman *.NET Framework* dengan tujuan untuk menyederhanakan pembuatan *user interface*. XML bertujuan untuk menyimpan dan membawa data. XAML dapat menampilkan elemen *user interface (UI)*, memisahkan pendefinisian logika *runtime UI* dengan menggunakan file *code* yang berbeda yang kemudian digabungkan melalui sebuah partial class.

### SQLite

SQLite merupakan sistem manajemen database relasional yang dibangun dalam sebuah library bahasa pemrograman C#. SQLite mudah dikelola dan sangat cepat jika berjalan pada lingkup yang sama sehingga tidak ada sumber daya tambahan, misalnya seperti jaringan ketika menjalankan query maupun mengambil data serta SQLite tidak membutuhkan konfigurasi banyak. Selain itu juga SQLite dapat diintegrasikan dengan API (*Application Programming Interface*).

### METODE PENELITIAN

Tahapan metodologi sistem aplikasi pengukuran besaran sudut antara 2 tungkai kaki pada saat *SLRT* dapat dijelaskan dalam diagram alur pada gambar 3 dibawah ini :



Gambar 3. Tahapan metodologi perhitungan sudut SLRT

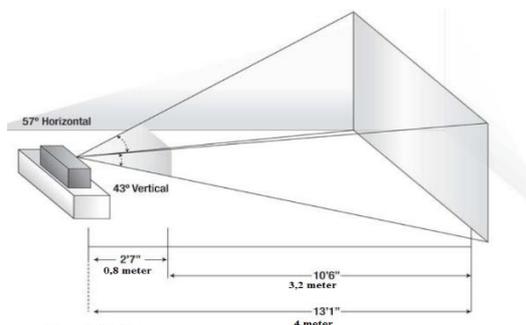
### Persiapan Sistem

Untuk dapat memulai membuat sistem aplikasi, tahap pertama yang dilakukan adalah mempersiapkan sistem. Persiapan diawali dengan menghubungkan kamera Kinect dengan PC. Setelah menghubungkan Kinect dengan PC, selanjutnya hubungkan Kinect dengan stop kontak agar Kinect dapat berjalan. Instal *Microsoft Kinect SDK* untuk dapat mengoperasikan Kinect di PC. Sampai tahap ini maka kamera Kinect sudah terhubung dan dapat digunakan.

### Kalibrasi Objek

Kalibrasi objek pada Kinect dilakukan untuk mengenali objek serta memperoleh nilai cm/pixel dari objek yang berada di depan sensor kamera Kinect. Nilai ini digunakan untuk memperoleh nilai aktual dari objek 3D ketika dilakukan proses digitalisasi. Kalibrasi ini bertujuan untuk mengenali fisik objek dan benda-benda di lingkungan sekitar objek. Pada proses kalibrasi ini merupakan tahap penyesuaian antara posisi kamera dan posisi tubuh pasien dalam pose telentang sedemikian sehingga

dapat terdeteksi sesuai dengan posisi standar yang ditentukan. Hasil akuisisi citra terkalibrasi diubah kedalam citra kedalaman (*grayscale depth*). Gambar 4 menjelaskan jangkauan sensor Kinect.



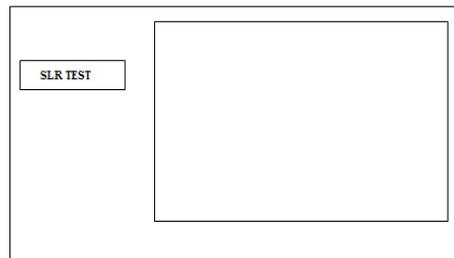
Gambar 4. Jangkauan sensor Kinect

Tahapan-tahapan setelah kalibrasi adalah input citra *RGB*, perubahan citra *RGB* menjadi citra *Grayscale Depth*, pelacakan atau pembuatan Skeleton, pendeteksian dan perhitungan sudut.

Jika semua tahapan dari penginputan citra *RGB*, dan perubahan citra menjadi *Grayscale Depth* berhasil, dan pelacakan skeleton terbentuk, maka dapat ditentukan besar sudut dari dua buah garis *skeleton* yang saling bertemu. Garis yang terbentuk dari hasil *tracking skeleton* akan di asumsikan sebagai vektor, dimana representasi data *skeleton* adalah tiga dimensi (X, Y, Z). Yang nantinya hasil perhitungan rumus *arcustangen* akan menghasilkan nilai *radian*. Nilai *radian* tersebut diubah menjadi *degrees* (derajat) dengan menggunakan rumus *degrees*. Sudut yang akan dideteksi adalah sudut yang terbentuk dari dua buah garis *skeleton* yang saling bertemu yaitu garis yang terbentuk dari titik sendi *hip centre* dengan titik sendi *knee right*, dan garis yang terbentuk dari titik sendi *hip centre* dengan titik sendi *knee left*.

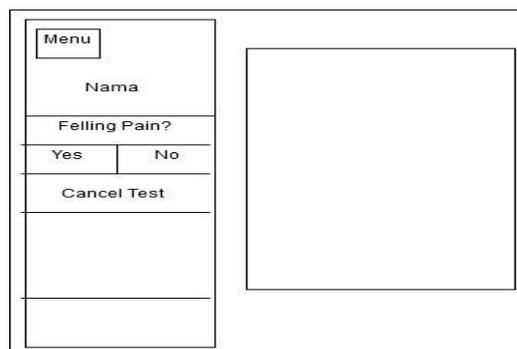
#### Perancangan User Interface Aplikasi

Perancangan tampilan yang akan diterapkan sebagai *user interface* aplikasi yang akan dibuat. Berikut adalah tampilan dari menu utama. Pada button **SLR Test** itu adalah memulai test, kemudian pada rancangan tampilan ini juga dapat dikeluarkan hasil test atau daftar yang sudah melakukan test.



Gambar 5. Tampilan utama SLRT

Pada rancangan tampilan berikut adalah untuk melihat proses test yaitu view citra *Grayscale depth* pada sebelah kiri dan view citra asli dan hasil skeleton pada sebelah kanan. Pada saat gerakan pengangkatan salah satu kaki keatas (*SLRT*) ditanyakan informasi mengenai ada tidak nya rasa nyeri. Jika ada maka button *Yes* diklik, sebaliknya jika tidak ada nyeri button *No* diklik kotak *Feeling Pain*.



Gambar 6. Tampilan View SLRT

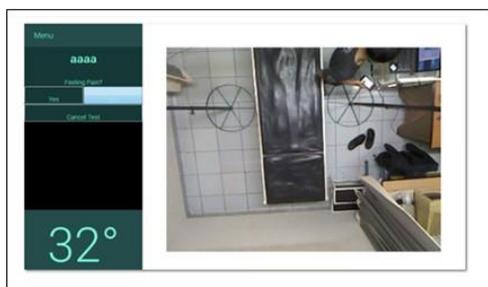
#### HASIL DAN PEMBAHASAN

Pertama-tama untuk dapat menjalankan aplikasi, hubungkan *Kinect* yang sudah terpasang sesuai dengan posisi *SLRT* yaitu diatas tempat tidur yang ada pada ruang pemeriksaan pasien pada umumnya dengan ketinggian sekitar 2 meter. Hal terpenting disini agar kalibrasi berhasil adalah penempatan *Kinect* serta posisi objek manusia harus ada dalam jangkauan *Kinect* dan objek dapat ter-*capture* seperti pada Gambar 4. Pada tahapan ini ketika aplikasi dijalankan maka akan muncul tampilan aplikasi yang terdapat satu kotak berisi *SLR Test*.



**Gambar 7.** Tampilan Awal Aplikasi

Setelah menekan kotak SLR Test, maka akan muncul tampilan siap uji SLRT dan menghitung sudut. Dengan muncul kotak *skeleton view*, *depth view*, dan *RGB view* yang masih kosong meskipun sudah terkalibrasi karena objek lainnya sudah ada dalam jangkauan Kinect. Jika objek manusia sudah masuk dalam jangkauan Kinect, kotak *skeleton view*, *depth view*, dan *RGB view* dan besaran sudut antara kedua kaki akan muncul. Jadi pada tahap ini memastikan Kinect sudah terkalibrasi dan jangkauan Kinect yaitu dengan jarak pandang  $57^\circ$  horisontal, vertikal  $43^\circ$  dengan  $70^\circ$  diagonal, dan jangkauan operasi adalah antara 0,8 meter dan 4 meter. Jarak yang dibutuhkan minimum yang digunakan untuk menangkap objek pada Kinect kurang lebih sekitar 3,2 meter. Tinggi objek manusia yang dapat ditangkap oleh kamera Kinect kurang lebih 160 cm seperti pada gambar 4 [9].



**Gambar 8.** Tampilan Siap Test

Untuk dapat memulai mengukur sudut pada saat SLRT, jika semua kotak *depth view*, *skeleton view* dan *RGB view* sudah muncul, maka aplikasi akan

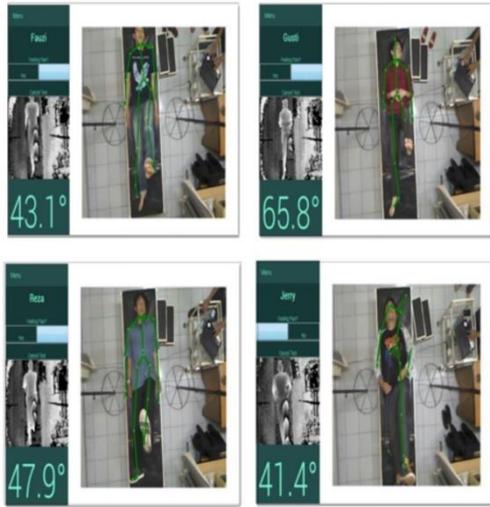
melakukan pengukuran sudut saat SLRT secara otomatis sesuai dengan sudut yang terbentuk antara 2 kaki pada gerakan SLRT, yang hasilnya akan ditampilkan pada aplikasi.

*Pseudocode* program untuk mengukur sudut gerakan SLRT dengan potongan pseudocode berikut :

```
namespace AutoSLR
using Microsoft.Kinect;
using System.Windows.Media;
using System.Windows.Media.Media3D;
public class Angle
public double GetAngle(Skeleton
skeleton)
Vector3D HipCenter = new
Vector3D(skeleton.Joints[JointType.Hip
Center].Position.X,
skeleton.Joints[JointType.HipCenter].P
osition.Y,
skeleton.Joints[JointType.HipCenter].P
osition.Z);
Vector3D KneeRight = new
Vector3D(skeleton.Joints[JointType.Kne
eRight].Position.X,
skeleton.Joints[JointType.KneeRight].P
osition.Y,
skeleton.Joints[JointType.KneeRight].P
osition.Z);
Vector3D KneeLeft = new
Vector3D(skeleton.Joints[JointType.Kne
eLeft].Position.X,
skeleton.Joints[JointType.KneeLeft].Pos
ition.Y,
skeleton.Joints[JointType.KneeLeft].Pos
ition.Z);
// Convert Point To Vector
Vector3D vectorA = KneeRight -
HipCenter;
Vector3D vectorB = KneeLeft -
HipCenter;
// Convert 3d Vector to 2d
// In here two vector axis(x, y)
replaced by y,z axis in
vectorA and vectorB
// And Do Angle Calculation
double angle =
Math.Atan2(vectorB.Y, vectorB.Z) -
Math.Atan2(vectorA.Y, vectorA.Z);
// Convert Radius to Angle in Degree
angle = angle * (180 / Math.PI);
```

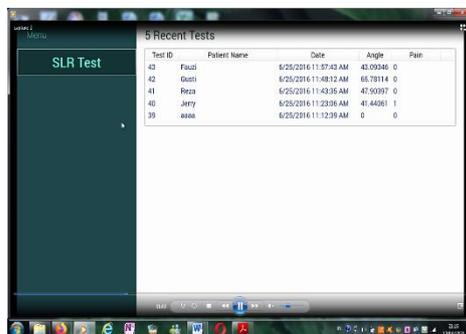
return angle;

Pada gambar 9 dibawah ini hasil pengujian sistem aplikasi perhitungan sudut SLRT.



Gambar 9. Hasil Ujicoba aplikasi SLRT

Setelah berhasil mendeteksi sudut dan mengukur sudutnya, maka akan terlihat dalam daftar data pada tampilan. Pengguna dapat menyimpan hasilnya kedalam database seperti pada gambar 11 dibawah ini:



Gambar 10. Tampilan akhir aplikasi SLRT

## PENUTUP

Aplikasi pengukuran sudut gerakan SLRT menggunakan Kinect XBOX 360 merupakan aplikasi *motion capture* secara *real time* serta integrasi antara editor Visual Studio, GUI XAML, database Sqlite dan koding C#. Aplikasi akan merekam gerakan SLRT dan mendeteksi sudut gerakan tersebut setelah skeleton terlacak, selanjutnya sudut yang terbentuk dari skeleton terlacak akan diproses dengan

perhitungan sudut. Output yang ditampilkan berupa tampilan *RGB view*, *Grayscale depth view*, dan *Skeleton view*, serta sudut yang terekam dan daftar pengguna yang melakukan gerakan SLRT. Aplikasi ini dapat dimanfaatkan oleh praktisi medis terutama bidang rehabilitasi agar dapat memperoleh pengukuran sudut gerakan SLRT lebih akurat, efektif dan efisien. Aplikasi ini akan memberikan hasil tampilan tergantung dari hasil *capture* dari *Kinect*.

## DAFTAR PUSTAKA

- [1] Maher C, Underwood M, Buchbinder R. "Non-specific low back pain". *Lancet*. 2017 Feb 18;389(10070):736-747. doi: 10.1016/S0140-6736(16)30970-9. Epub 2016 Oct 11. PMID: 27745712.
- [2] M Das J, Nadi M. "Lasegue Sign". [Updated 2021 Mar 31]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2021 Jan- Available from: <https://www.ncbi.nlm.nih.gov/books/NBK545299/>
- [3] Bahar, A., Wuysang, D., "Sistem Neuropsikiatri Pemeriksaan Neurologik", Departemen Neurologi Fakultas Kedokteran Universitas Hasanuddin, Makassar. 2015
- [4] K. LaBelle, "Evaluation Of Kinect Joint Tracking For Clinical And In-Home Stroke Rehabilitation Tools", Indiana, 2011
- [5] N. Kitsunezaki, E. Adachi, T. Masuda, J. Mizusawa., "KINECT Applications for The Physical Rehabilitation"., 2013
- [6] Guerra-Filho G., "Optical motion capture: theory and implementation". 2005, *Journal of Theoretical and Applied Informatics*, vol. 12, no. 2, pp. 61-89
- [7] Cruz.,L, Lucio,D., Velho.L, "Kinect and RGBD Images: Challenges and Applications". IMPA - VISGRAF Lab., 2012
- [8] Wibowo, A.T., Yudaningtyas, Erni., Sunaryo, S., "Teknologi natural User Interface menggunakan Kinect Sebagai Pemicu Kerja Perangkat Keras Berbasis Fuzzy Interface System".

Jurnal EECCIS. Volume 7, No. 1,  
<http://jurnaleeccis.ub.ac.id/index.php/eccis/article/viewFile/194/167>, Juni 2013.

- [9] Webb,J., Ashley,J., “*Beginning Kinect Programming with the Microsoft Kinect SDK*”, Apress Publishing Pte Ltd. , 2012