

**ANALISA PERBANDINGAN METODE *KINECT GRAYSCALE* DAN
KINECT DEPTH IMAGE UNTUK DIAPLIKASIKAN DALAM *USER*
*INTERFACE***

Skripsi

untuk memenuhi sebagian persyaratan

mencapai derajat Sarjana S-1

Program Studi Teknik Informatika



disusun oleh

Fajria Antoni

08650050

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2014



PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/3207/2014

Skrripsi/Tugas Akhir dengan judul : Analisa Perbandingan Metode *Kinect Grayscale* dan *Kinect Depth Image* Untuk Diaplikasikan Dalam *User Interface*

Yang dipersiapkan dan disusun oleh :
Nama : Fajria Antoni
NIM : 08650050
Telah dimunaqasyahkan pada : Jum'at, 24 Oktober 2014
Nilai Munaqasyah : B
Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Aulia Faqih R, M.Kom
NIP. 19860306 201101 1 009

Penguji I

M. Mustakin, M.T
NIP. 19790331 200501 1 004

Penguji II

Agung Fatwanto, Ph.D
NIP. 19770103 200501 1 003

Yogyakarta, 29 Oktober 2014
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan



Prof. Drs. H. Akh. Minhaji, M.A, Ph.D
NIP. 19580919 198603 1 002



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Permohonan

Lamp : -

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

Assalamu 'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Fajria Antoni

NIM : 08650050

Judul Skripsi :

Analisis Perbandingan Metode *Kinect Grayscale* dan *Kinect Depth Image* untuk Diaplikasikan dalam *User Interface*

sudah dapat diajukan kembali kepada Program Studi Tekni Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Teknik Informatika

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu 'alaikum wr. wb.

Yogyakarta, 05 Oktober 2014

Pembimbing

Aulia Faqih Rifa'I, M.Kom

NIP: 19860306 201101 1 009

PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Fajria Antoni
Nim : 08650050
Program Studi : Teknik Informatika
Fakultas : Sains dan Teknologi

Menyatakan bahwa skripsi dengan judul **Analisis Perbandingan Metode *Kinect Grayscale* dan *Kinect Depth Image* untuk Diaplikasikan dalam *User Interface*** tidak terdapat pada karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu Perguruan Tinggi, dan sepengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 15 Oktober 2014

Yang Menyatakan,



Fajria Antoni
NIM : 08650050

KATA PENGANTAR

Bismillahirrahmanirrahim

Alhamdulillah, segala puji bagi Allah Subhanahu wa ta'ala atas limpahan rahmat, hidayah, serta bimbingan-Nya. Shalawat serta salam semoga tercurah kepada Nabi Muhammad Shallallohu 'alaihi wa sallam. Akhirnya penulis dapat menyelesaikan penelitian Tugas Akhir yang berjudul **“Analisis Perbandingan Metode Kinect Grayscale dan Kinect Depth Image untuk Diaplikasikan dalam User Interface”**. Oleh karena itu, dengan segala kerendahan hati pada kesempatan ini penulis mengucapkan banyak terima kasih kepada:

1. Prof. Drs. H. Akh. Minhaji, M.A., Ph.D selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
2. Bapak Agus Mulyanto, S.Si, M.Kom. selaku Ketua Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
3. Bapak Aulia Faqih , M.Kom, selaku pembimbing yang membimbing, mengarahkan, memberikan nasehat dan saran selama penyusunan skripsi.
4. Ayahanda Irawan dan Ibunda Siti Hasanah yang selalu memberikan dukungan dan semangat kepada penulis untuk menyelesaikan tulisannya.
6. Mas Emet ,Jan Faris Majd, Mas Andi Febriyanto, Mas Ceye yang telah banyak membantu dalam menyelesaikan penelitian penulis.

7. Seluruh teman-teman keluarga besar Program Studi Teknik Informatika, khususnya angkatan 2008 yang telah banyak sekali memberikan masukan, saran dan diskusi yang begitu berharga.

8. Teman – teman dan sahabat, Ratna Purnamasari, Feri Arman, Bayu Aswin, Mas Barok, Mbak Puput ,Mbak Ami terima kasih atas semangat dan selalu mengingatkan.

Penulis merasa masih banyak sekali kekurangan dan kelemahan dalam penelitian ini, oleh karena itu segala kritik dan saran senantiasa penulis harapkan dari para pembaca. Akhir kata, semoga penelitian ini dapat menjadi panduan serta referensi yang sangat berguna bagi pembaca dan dapat dimanfaatkan sebaik-baiknya.

15 Oktober 2014

HALAMAN PERSEMBAHAN

Ibunda Siti Hasanah



MOTTO

***BELAJAR MEMBUAT
HIDUP LEBIH BERMAKNA***

DAFTAR ISI

| | |
|--------------------------------------------------|------|
| HALAMAN JUDUL..... | i |
| HALAMAN PENGESAHAN SKRIPSI/TUGAS AKHIR..... | ii |
| SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR | iii |
| PERNYATAAN KEASLIAN SKRIPSI..... | iv |
| KATA PENGANTAR | v |
| HALAMAN PERSEMBAHAN | vii |
| MOTTO | viii |
| DAFTAR ISI..... | ix |
| DAFTAR GAMBAR | xii |
| DAFTAR TABEL..... | xiii |
| DAFTAR LAMPIRAN..... | xv |
| INTISARI..... | xvi |
| ABSTRACT..... | xvii |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 3 |
| 1.3 Pembatasan Masalah | 4 |
| 1.4 Tujuan Penelitian | 4 |
| 1.5 Manfaat Penelitian | 5 |
| 1.6 Keaslian Penelitian | 5 |
| BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI | 6 |
| 2.1 Tinjauan Pustaka..... | 6 |

| | |
|----------------------------------------------------------------------|-----------|
| 2.2 Landasan Teori | 7 |
| 2.2.1 Interaksi Manusia dan Komputer | 7 |
| 2.2.2 Triangulasi Kinect | 8 |
| 2.2.3 Citra Digital | 8 |
| 2.2.4 <i>Depth Image Grayscale</i> | 9 |
| 2.2.5 <i>Kinect Depth Image</i> | 9 |
| 2.2.6 Representasi Citra Digital..... | 10 |
| 2.3 <i>Kinect</i> | 12 |
| 2.3.1 Teknologi <i>Kinect</i> | 12 |
| BAB III METODE PENELITIAN..... | 16 |
| 3.1 Studi Pendahuluan | 16 |
| 3.2 Kebutuhan sistem..... | 16 |
| 3.3 Perancangan <i>Experiment</i> | 17 |
| 3.3.1 <i>Experiment</i> Pengenalan Bentuk <i>Geometri</i> | 19 |
| 3.3.2 <i>Experiment</i> Pengukuran Jarak | 24 |
| 3.3.3 <i>Experiment Tracking</i> | 25 |
| 3.3.4. <i>Experiment Resource Usage</i> | 27 |
| BAB IV HASIL PENELITIAN DAN PEMBAHASAN..... | 28 |
| 4.1 Hasil Penelitian | 28 |
| 4.1.1 Deskripsi Data Pengujian Pengenalan Pola <i>Geometri</i> | 28 |
| 4.1.2 Deskripsi Data Ekperiment Pengukuran Jarak | 30 |
| 4.1.3 Deskripsi Data Ekperiment <i>Tracking</i> | 31 |
| 4.1.4 Deskripsi Data Ekperiment <i>System usage</i> | 33 |

| | |
|------------------------------------------------------------|----|
| 4.2 Pembahasan dan Hasil | 36 |
| 4.2.1 Aspek Kemampuan Mengenali Pola <i>Geometri</i> | 36 |
| 4.2.2 Aspek Kemampuan Kengukur Jarak | 41 |
| 4.2.3 Aspek Kemampuan <i>Tracking</i> | 46 |
| 4.2.4 Aspek Tingkat Efisiensi | 48 |
| BAB V PENUTUP | 50 |
| 5.1 Kesimpulan | 50 |
| 5.2 Saran | 50 |
| DAFTAR PUSTAKA | 53 |
| LAMPIRAN | 55 |

DAFTAR GAMBAR

| | |
|----------------------------------------------------------------------------|----|
| Gambar 2.1 <i>Kinect</i> Tanpa Box..... | 13 |
| Gambar 2.2 Citra Infra Merah..... | 14 |
| Gambar 3.1 <i>Xbox Kinect</i> | 17 |
| Gambar 3.2 Pengaturan Ruang Pengujian | 18 |
| Gambar 3.3 Flow Chart Pengujian..... | 18 |
| Gambar 4.1 Aplikasi yang Berjalan | 34 |
| Gambar 4.2 <i>CPU Idle</i> | 35 |
| Gambar 4.3 <i>Memory Idle</i> | 35 |
| Gambar 4.4 Pengujian Kasus Pertama | 42 |
| Gambar 4.5 Pengujian Kasus Kedua..... | 42 |
| Gambar 4.6 Area Deteksi <i>Grayscale</i> 0 - 255 | 43 |
| Gambar 4.7 Area Deteksi dan Blank Objek..... | 43 |
| Gambar 4.8 Grafik Grafik Perbandingan Metode <i>Grayscale, Depth</i> | 45 |

DAFTAR TABEL

| | |
|-----------------------------------------------------------------------------|----|
| Tabel 3.1 <i>Experiment</i> Pengenalan Bentuk <i>Geometri</i> | 19 |
| Tabel 3.2 <i>Batasan pengamatan visual pola persegi</i> | 21 |
| Tabel 3.3 <i>Batasan pengamatan visual pola bintang</i> | 22 |
| Tabel 3.4 <i>Experiment Resource Usage</i> | 23 |
| Tabel 3.5 <i>Experiment</i> Pengukuran Jarak | 24 |
| Tabel 3.6 <i>Experiment Tracking</i> | 25 |
| Tabel 3.7 <i>Experiment Resource Usage</i> | 27 |
| Tabel 4.1 Data Hasil Pengujian Pola Lingkaran Metode <i>Grayscale</i> | 28 |
| Tabel 4.2 Data Hasil Pengujian Pola Persegi Metode <i>Grayscale</i> | 29 |
| Tabel 4.3 Data Hasil Pengujian Pola Bintang Metode <i>Grayscale</i> | 29 |
| Tabel 4.4 Data Hasil Pengujian Pola Lingkaran Metode <i>Depth</i> | 29 |
| Tabel 4.5 Data Hasil Pengujian Pola Persegi Metode <i>Depth</i> | 29 |
| Tabel 4.6 Data Hasil Pengujian Pola Bintang Metode <i>Depth</i> | 30 |
| Tabel 4.7 Data Pengujian Jarak <i>Grayscale</i> | 30 |
| Tabel 4.8 Data Pengujian Jarak <i>Depth</i> | 30 |
| Tabel 4.9 Data Pengujian <i>Tracking Depth</i> | 31 |
| Tabel 4.10 Perbandingan Tingkat Efisiensi <i>Grayscale, Depth</i> | 36 |
| Tabel 4.11 Data Hasil Pengujian Pola Lingkaran Jarak 60cm..... | 37 |
| Tabel 4.12 Data Hasil Pengujian Pola Lingkaran Jarak 90cm..... | 37 |
| Tabel 4.13 Data Hasil Pengujian Pola Lingkaran Jarak 120cm..... | 38 |
| Tabel 4.14 Data Hasil Pengujian Pola Persegi Jarak 60cm | 38 |
| Tabel 4.15 Data Hasil Pengujian Pola Persegi Jarak 90cm | 39 |

| | |
|-------------------------------------------------------------------|----|
| Tabel 4.16 Data Hasil Pengujian Pola Persegi Jarak 120cm | 39 |
| Tabel 4.17 Data Hasil Pengujian Pola Bintang Jarak 60cm | 40 |
| Tabel 4.18 Data Hasil Pengujian Pola Bintang Jarak 90cm | 40 |
| Tabel 4.19 Data Hasil Pengujian Pola bintang Jarak 120cm | 40 |
| Tabel 4.20 Hasil Perhitungan Dengan Metode <i>Grayscale</i> | 45 |
| Tabel 4.21 Hasil Perhitungan Dengan Metode <i>Depth</i> | 45 |
| Tabel 4.22 Hasil Pengujian <i>Tracking</i> | 48 |
| Tabel 4.23 Hasil Pengujian <i>Usage</i> | 48 |



DAFTAR LAMPIRAN

| | |
|---------------------------------------------|----|
| <i>Coding</i> | 55 |
| Data Pengujian Pola <i>Grayscale</i> | 63 |
| Data Pengujian Pola <i>Depth</i> | 64 |
| Data Pengujian Jarak <i>Grayscale</i> | 66 |
| Data Pengujian Jarak <i>Depth</i> | 69 |
| Data Pengujian <i>Usage Grayscale</i> | 75 |
| Data Pengujian <i>Usage Depth</i> | 78 |
| <i>Curriculum Vitae</i> | 81 |

**Analisis Perbandingan Metode *Kinect Grayscale* dan *Kinect Depth Image*
Untuk Diaplikasikan Dalam *User Interface***

Fajria Antoni

NIM 08650050

Intisari

Tujuan dari penelitian ini adalah untuk memanfaatkan mengetahui *metode* mana yang lebih tepat dalam aplikasi *user interface*.

Metode pengumpulan data yang digunakan adalah studi literatur. Pengumpulan data dengan metode studi literatur dilakukan dengan cara membaca buku, jurnal ilmiah, makalah, dan menggunakan *media internet* sebagai sumber informasi.

Berdasarkan penelitian yang dilakukan dapat disimpulkan bahwa dari kedua metode memiliki kelebihan dan kekurangan. Metode grayscale baik untuk pemrosesan cepat dan pemrosesan image. Sedangkan metode depth baik untuk aplikasi yang membutuhkan tingkat ketelitian tinggi dan memerlukan resource yang tinggi.

Kata Kunci : *kinect, grayscale, depth image, user interface*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini *interface* yang paling sering digunakan dalam interaksi manusia dengan komputer adalah *mouse*. *Mouse* pada awalnya masih berbentuk *trackball* berukuran sangat besar yang ditemukan oleh *Tom Cranston* dan *Fred Longstaff* seorang angkatan laut Kanada pada tahun 1952 dan menggunakan sebuah bola *bowling*. Perkembangan *mouse* sangat pesat hingga pada tahun 1963 ditemukan metode baru *mouse* pengganti metode *trackball* yang ditemukan oleh Douglas Engelbart pada tahun 1964. Namun *mouse* sendiri masih banyak sekali kekurangan dalam pengembangannya, contohnya adalah penggunaan *mouse* yang kurang efektif dalam aktifitas yang lama contohnya *design* dan *digital painting*. Penggunaan *mouse* dalam waktu lama *menyebabkan repetitive stress injuries (RSI)* yang berakibat beberapa penyakit dan kelainan pada tangan. (Franck Dernoncourt, 2012).

Perkembangan teknologi khususnya komputer dan elektronik memicu perkembangan alat input yang lebih natural. Douglas Engelbart (1964) menciptakan *prototype mouse* yang digunakan pertama kali pada *GUI Windows*. Pada tahun 1968 telah diciptakan *touch screen* sebagai alat pengendali lalu lintas udara oleh E.A. Johnson. Pada tahun 2006 *Nintendo Wii* sebagai *evolutioner controler game* yang sebelumnya *joystick* menjadi *sistem remote* yang dapat mendeteksi gerakan. Pada 4 November 2010 *Xbox Kinect* oleh *Primesense* yg

didesain untuk *console Xbox* yang langsung terjual 10 juta unit dalam bulan pertama. (Greg Borestein,2012)

Kinect adalah sebuah *Depth camera* . Berbeda dengan *camera* biasa. *Kinect* berfungsi menangkap jarak dari suatu objek pada suatu ruangan. *Kinect* menggunakan sinar inframerah untuk membuat *depth image*. Sinar *IR* di tembakan kepada suatu objek dan pantulannya di tangkap kembali oleh *camera IR* metode ini di sebut sebagai *triangulasi stereocam*. Hasil dari *camera IR* adalah gambar berupa *grayscale*. *Grayscale* adalah gambar yang memiliki tingkatan warna. Tingkatan warna inilah yang menunjukkan seberapa jauh objek terhadap kamera. Semakin putih maka objek itu semakin dekat, dan begitu pula sebaliknya. (Greg Borestein,2012)

Beberapa *device* yang memiliki basis *stereocam* seperti *kinect* contohnya adalah *PS 3 move*, *intel perceptual computing*. *PS 3 move* adalah *device* yang memiliki sistem kerja kombinasi antara *stereocam* dan *acelerometer*. *intel perceptual computing* memiliki sistem kerja sama dengan *kinect*, yaitu *stereocam*, yang membedakan adalah *intel perceptual computing* terfokus pada *gesture* jari dan wajah, sedangkan *kinect* dapat mendeteksi seluruh badan dan dapat mencakup 6 pengguna.

Kinect mempunyai dua metode dalam mendapatkan data *depth*, yaitu dengan analisis warna *grayscale* dan analisis *depth image*. Data *depth* adalah nilai

hasil dari proses kedua analisis. Data depth adalah data yang berisi nilai jarak objek pada setiap pixelnya. Analisis *grayscale* adalah metode untuk memperoleh data jarak benda yang berada di depan *kinect* dengan cara melakukan pemrosesan citra. Semakin terang suatu *pixel* dalam *image* tersebut menandakan semakin dekat pula terhadap *kinect*. Pada analisis *depth image* untuk memperoleh data jarak benda yang berada di depan *kinect* dengan cara menghitung *travel time* cahaya infrared yang dipancarkan oleh *kinect* yang kemudian disimpan dalam data *array* (Greg Borestein,2012)

Dua metode ini dapat digunakan untuk aplikasi *interface* namun perlu diketahui aspek-aspek yang dibutuhkan dalam aplikasi *interface* yaitu berupa kemampuan pengenalan objek geometri tracking objek pengukuran jarak, dan tingkan efisiensi. Penggunaan dan pemilihan metode ini sangat fital dalam pembuatan software, karena kesalahan pada pemilihan metode berakibat system yang berat, dan tidak akurat. Sebelumnya belum pernah ada penelitian yang membahas hal tersebut, oleh karena itu perlu diadakan eksperimen untuk dapat membandingkan kedua metode tersebut .

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dalam penelitian ini dapat diambil rumusan permasalahan : Bagaimana menganalisis perbandingan metode *grayscale* dan metode *depth image* untuk diaplikasikan sebagai *user interface* ?

1.3 Pembatasan Masalah

Pada penelitian ini, diberikan batasan masalah sebagai berikut:

1. Menggunakan *Kinect for Xbox* yang digunakan pada sistem operasi *windows 7 32bit* sebagai sensor 3D
2. Menggunakan resolusi *800*600 pixel 30 fps* pada *RGB cam* dan *11bit depth level* pada *depth cam*.
3. Menggunakan *library Kinect for Windows SDK, Simple OpenNI ,NITE*
4. Tempat pengujian yang memiliki sudut yang dapat menangkap seluruh objek .
5. Menggunakan aplikasi *processing* .
6. Meneliti dua metode *grayscale* dan *depth*
7. Meneliti kemampuan pengenalan objek geometri,tracking objek ,pengukuran jarak, dan tingkat efisiensi
8. Penelitian ini difokuskan pada analisis perbandingan dua metode dan menggunakan pengujian pada sumbu x, y dan z yang diaplikasikan sebagai *user interface*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membandingkan metode *grayscale* dan metode *depth image* pada *kinect* sehingga dapat diketahui metode yang lebih tepat untuk pembuatan aplikasi.

1.5 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan manfaat:

1. Dapat mengetahui metode yang tepat pada *kinect* dalam pengoptimalan fungsi *user interface*.
2. Mengetahui karakteristik setiap metode beserta kekurangan dan kelebihan jika diaplikasikan pada pembuatan aplikasi.

1.6 Keaslian Penelitian

Penelitian yang menggunakan *kinect* sebagai deteksi tubuh pernah dilakukan sebelumnya menggunakan beberapa metode dan beberapa objek tubuh yang digunakan. Penelitian ini difokuskan pada analisis perbandingan metode grayscale dan depth dengan menggunakan pengujian pada sumbu x, y dan z yang diaplikasikan sebagai *user interface*.

BAB V

PENUTUP

5.1 KESIMPULAN

Berdasarkan penelitian yang dilakukan dapat disimpulkan bahwa dari kedua metode memiliki kelebihan dan kekurangan.

1. Metode *grayscale* lebih baik pada pengujian pola persegi sehingga cocok digunakan untuk mendeteksi objek berupa garis lurus.
2. Metode *depth* lebih baik pada pengujian pola bintang sehingga cocok digunakan untuk mendeteksi objek berupa sudut.
3. Pada pengujian pola lingkaran kedua metode sama baiknya.
4. Metode *grayscale* lebih akurat mengukur jarak untuk penggunaan jarak dekat
5. Metode *depth* lebih akurat mengukur jarak untuk penggunaan jarak menengah dan jarak jauh
6. Metode *depth* memiliki tingkat akurasi yang lebih baik dalam *tracking*.
7. Metode *grayscale* lebih stabil dalam *tracking* benda bergerak maupun diam.

5.2 SARAN

Penelitian ini masih dapat dikembangkan lebih lanjut , adapun beberapa perkembangan selanjutnya adalah:

1. Memperbaiki proses pengukuran pada dunia nyata.

2. Menggunakan alat peraga dan alat pengukur yang lebih baik dan teliti.
3. Menggunakan ruangan yang lebih baik penataan dan luas yang memadai.



LAMPIRAN 1.CODING

pengujian jarak depth

```
import SimpleOpenNI.*;
SimpleOpenNI kinect;

void setup(){
  size(640,480);
  kinect= new SimpleOpenNI(this);
  kinect.enableDepth();
}

void draw(){
  kinect.update();
  PImage depthImage=kinect.depthImage();
  image(depthImage,0,0);
}

void mousePressed(){

  int[]depthValues=kinect.depthMap();
  int clickPosition=mouseX + (mouseY*640);
  int millimeters=depthValues[clickPosition];
  float inches=millimeters/25.4;
  println("mm:"+millimeters+"in:"+inches);
}
```


Pengujian jarak grayscale

```
import SimpleOpenNI.*;
SimpleOpenNI kinect;

void setup(){
kinect=new SimpleOpenNI(this);
kinect.enableDepth();
size(640,480);
}

void draw(){
kinect.update();
PImage depthImage=kinect.depthImage();
image(depthImage,0,0);
}

void mousePressed(){
color c= get(mouseX,mouseY);
println("r:"+red(c)+"g:"+green(c)+"b:" +blue(c));
}
```

Pola depth

```
/* -----  
* SimpleOpenNI DepthMap3d Test  
* -----  
* Processing Wrapper for the OpenNI/Kinect 2 library  
* http://code.google.com/p/simple-openni  
* -----  
* prog: Max Rheiner / Interaction Design / ZhdK / http://iad.zhdK.ch/  
* date: 12/12/2012 (m/d/y)  
* -----  
*/  
  
import SimpleOpenNI.*;  
  
SimpleOpenNI context;  
float zoomF = 0.3f;  
float rotX = radians(180); // by default rotate the hole scene 180deg around the x-  
axis,  
// the data from openni comes upside down  
float rotY = radians(0);  
  
void setup()  
{  
  size(1024,768,P3D);  
  
  context = new SimpleOpenNI(this);  
  if(context.isInit() == false)  
  {  
    println("Can't init SimpleOpenNI, maybe the camera is not connected!");  
    exit();  
    return;  
  }  
  
  // disable mirror  
  context.setMirror(false);  
  
  // enable depthMap generation  
  context.enableDepth();  
  
  stroke(255,255,255);  
  smooth();  
  perspective(radians(45),  
             float(width)/float(height),
```

```

        10,150000);
    }

void draw()
{
    // update the cam
    context.update();

    background(0,0,0);

    translate(width/2, height/2, 0);
    rotateX(rotX);
    rotateY(rotY);
    scale(zoomF);

    int[] depthMap = context.depthMap();
    int steps = 3; // to speed up the drawing, draw every third point
    int index;
    PVector realWorldPoint;

    translate(0,0,-1000); // set the rotation center of the scene 1000 infront of the camera

    stroke(255);

    PVector[] realWorldMap = context.depthMapRealWorld();

    // draw pointcloud
    beginShape(POINTS);
    for(int y=0;y < context.depthHeight();y+=steps)
    {
        for(int x=0;x < context.depthWidth();x+=steps)
        {
            index = x + y * context.depthWidth();
            if(depthMap[index] > 0)
            {
                // draw the projected point
                // realWorldPoint = context.depthMapRealWorld()[index];
                realWorldPoint = realWorldMap[index];
                vertex(realWorldPoint.x,realWorldPoint.y,realWorldPoint.z); // make realworld z
                // negative, in the 3d drawing coordsystem +z points in the direction of the eye
            }
            //println("x: " + x + " y: " + y);
        }
    }
}

```

```

endShape();

// draw the kinect cam
// context.drawCamFrustum();
}

void keyPressed()
{
  switch(key)
  {
    case ' ':
      context.setMirror(!context.mirror());
      break;
  }

  switch(keyCode)
  {
    case LEFT:
      rotY += 0.1f;
      break;
    case RIGHT:
      // zoom out
      rotY -= 0.1f;
      break;
    case UP:
      if(keyEvent.isShiftDown())
        zoomF += 0.02f;
      else
        rotX += 0.1f;
      break;
    case DOWN:
      if(keyEvent.isShiftDown())
      {
        zoomF -= 0.02f;
        if(zoomF < 0.01)
          zoomF = 0.01;
      }
      else
        rotX -= 0.1f;
      break;
  }
}

```

Pola grayscale

```
import SimpleOpenNI.*;
SimpleOpenNI kinect;
void setup() {
  // double the width to display two images side by side
  size(640*2, 480);
  kinect = new SimpleOpenNI(this);
  kinect.enableDepth();
  kinect.enableRGB(); }

void draw() {
  kinect.update();
  PImage depthImage = kinect.depthImage();
  PImage rgbImage = kinect.rgbImage();
  image(depthImage, 0, 0); image(rgbImage, 640, 0); }
```

Tracking grayscale

```
import SimpleOpenNI.*;
SimpleOpenNI kinect;

void setup(){
  size(640,480);

  kinect= new SimpleOpenNI(this);
  kinect.enableDepth();
  kinect.enableRGB();
}

void draw(){
  kinect.update();
  PImage depthImage=kinect.depthImage();

  image(depthImage,0,0);
  int brightestX= 0;
  int brightestY= 0;
  float brightestValue=0;

  depthImage.loadPixels();
  int index=0;
  for(int y=0; y < depthImage.height; y++){

    for(int x=0;x<depthImage.width;x++){
      int pixelValue=depthImage.pixels[index];
      float pixelBrightness=brightness(pixelValue);
      if (pixelBrightness > brightestValue){
        brightestValue = pixelBrightness;
      }
      brightestX = x;
      brightestY = y;
    }
    index++;
  }
  fill(255,204,0,128);
  ellipse(brightestX, brightestY,200,200);
}
```

Tracking depth

```
import SimpleOpenNI.*;
SimpleOpenNI kinect;
int closestValue;
int closestX;
int closestY;
void setup(){
  size(640,480);
  kinect=new SimpleOpenNI(this);
  kinect.enableDepth();
}

void draw(){
  closestValue=8000;
  PImage depthImage=kinect.depthImage();


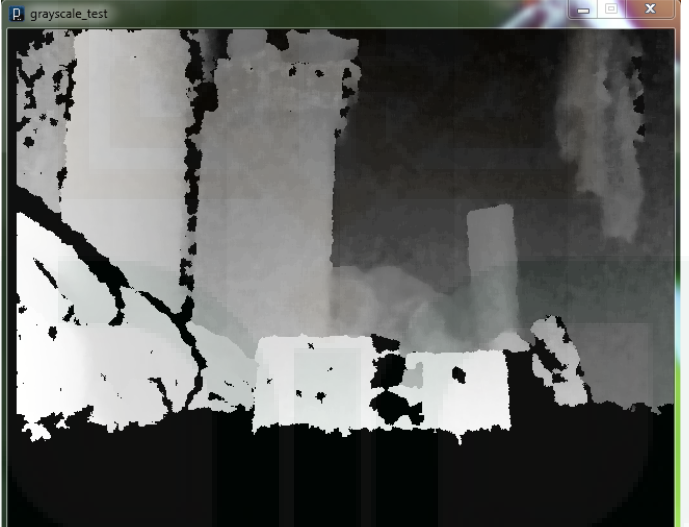
  image(depthImage,0,0);
  kinect.update();

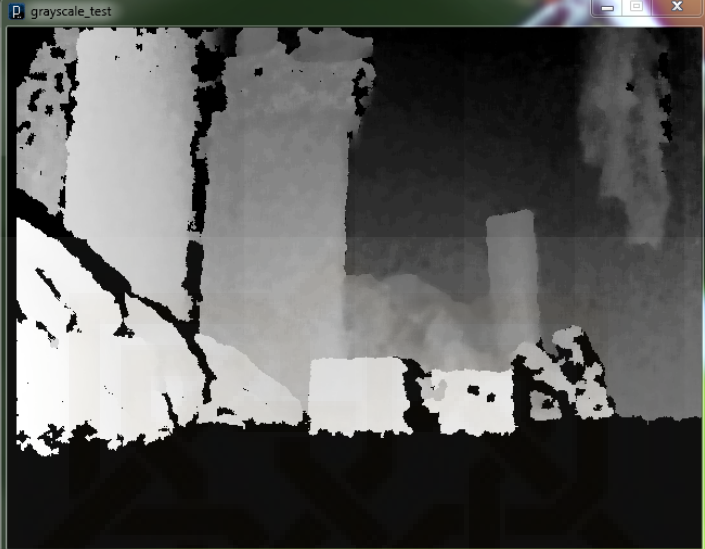
  int[]depthValues=kinect.depthMap();
  for(int y=0 ; y<480; y++){
    for(int x=0 ;x<460;x++){

      int i=x+y*640;
      int currentDepthValue=depthValues[i];
      if(currentDepthValue>0 && currentDepthValue<closestValue){
        closestValue=currentDepthValue;
        closestX=x;
        closestY=y;
      }
    }
  }
  fill(255,0,0);

  ellipse(closestX,closestY,25,25);
}
```

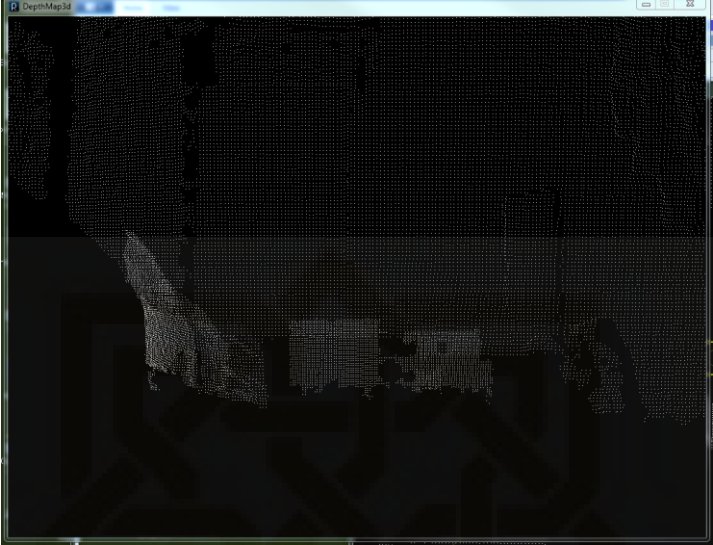
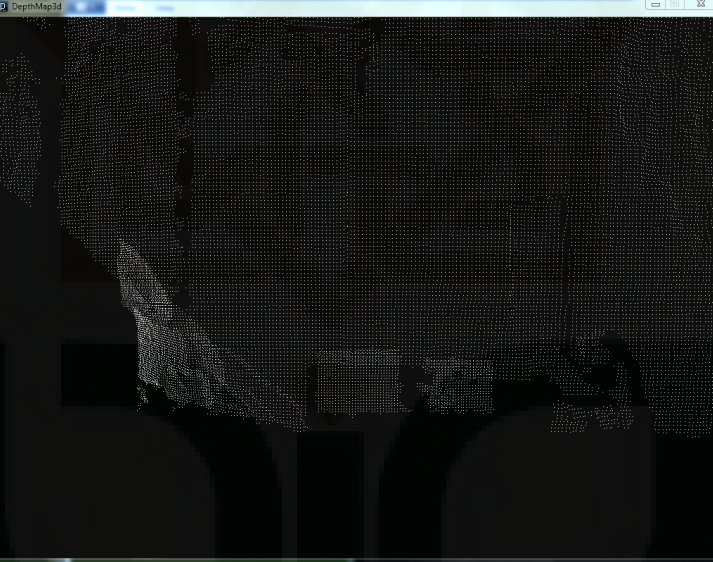
LAMPIRAN 2.DATA PENGUJIAN POLA *GRAYSCALE*

| citra | Jarak (cm) |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
|  A grayscale image showing a building facade with a prominent horizontal band of windows. The image is displayed in a window titled 'grayscale_test'. The image is somewhat blurry, indicating a closer distance. | 60 |
|  A grayscale image showing the same building facade as above, but at a greater distance of 90 cm. The image is sharper and more detailed, showing individual window panes and architectural features. It is also displayed in a window titled 'grayscale_test'. | 90 |

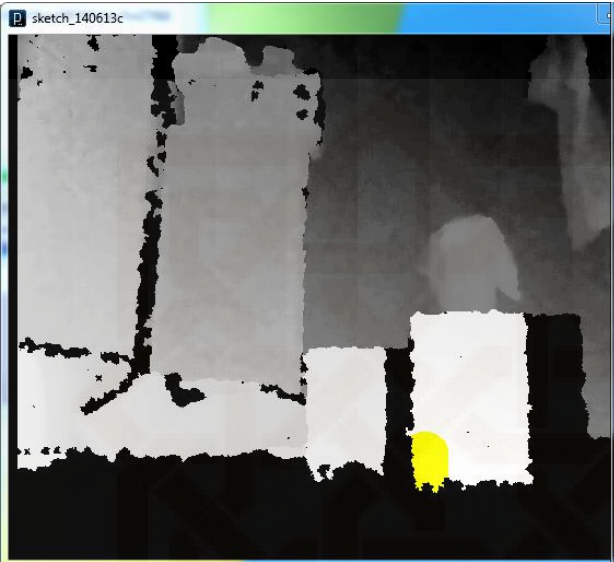
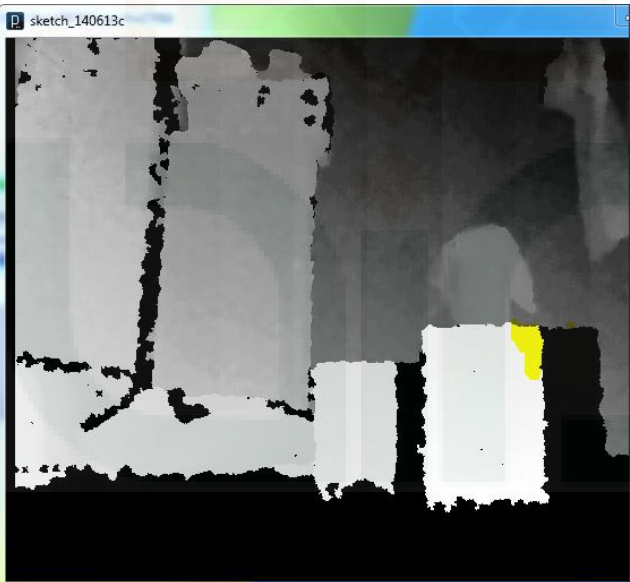
| citra | Jarak (cm) |
|------------------------------------------------------------------------------------|------------|
|  | 120 |


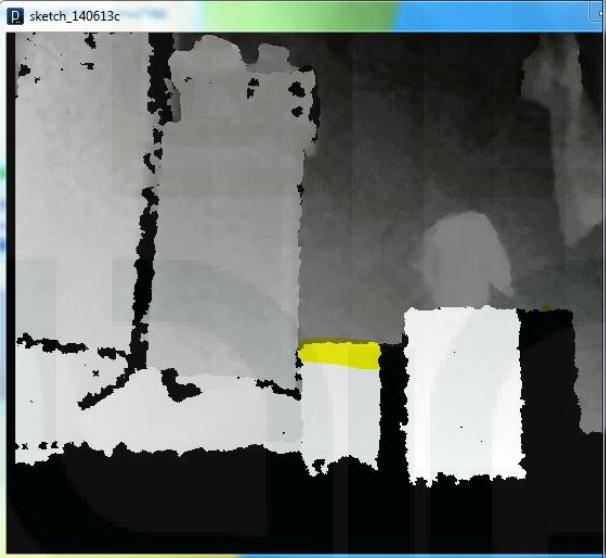
LAMPIRAN 3. DATA PENGUJIAN POLA *DEPTH*



| citra | Jarak (cm) |
|--------------------------------------------------------------------------------------|------------|
|  | 60 |

| citra | Jarak (cm) |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
|  A screenshot of a DepthMap3d application window showing a 3D point cloud of a scene. The scene includes a table and chairs. The point cloud is rendered in grayscale, with the table and chairs appearing as distinct shapes against a dark background. The window title is "DepthMap3d". | 90 |
|  A screenshot of a DepthMap3d application window showing a 3D point cloud of the same scene as above. The point cloud is rendered in grayscale. The scene includes a table and chairs. The window title is "DepthMap3d". | 120 |

LAMPIRAN 4. DATA PENGUJIAN JARAK *GRAYS*CALE

| Citra | Jarak | Nilai kecerahan |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----------------|
|  A grayscale image of a test chart with various patterns and shapes. A small yellow circle is visible on the right side of the chart. The image is displayed in a window titled 'sketch_140613c'. | 60cm | 254 |
|  A grayscale image of a test chart, similar to the one above, but with a slightly different appearance due to the increased distance. A small yellow circle is visible on the right side of the chart. The image is displayed in a window titled 'sketch_140613c'. | 61 cm | 237 |

| Citra | Jarak | Nilai kecerahan |
|--------------------------------------------------------------------------------------|-------|-----------------|
|  | 90 cm | 233 |
|  | 91 cm | 225 |

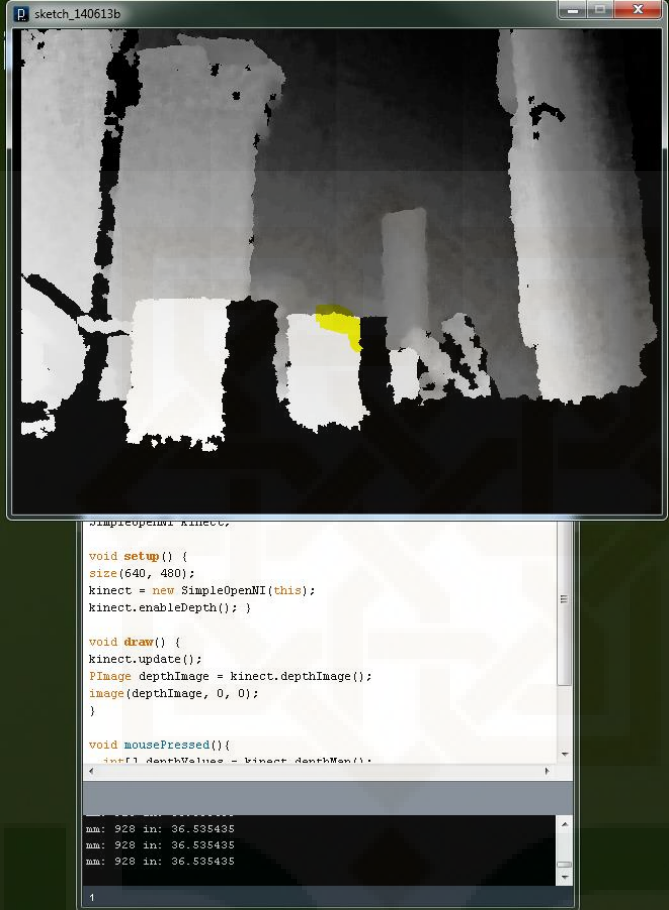
| Citra | Jarak | Nilai kecerahan |
|-------------------------------------------------------------------------------------|--------|-----------------|
|  | 180 cm | 210 |
|  | 181 cm | 211 |

LAMPIRAN 5. DATA PENGUJIAN JARAK *DEPTH*

| Citra | Jarak (cm) | Nilai <i>array</i> (mm) |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|-------------------------|
|  <p>The screenshot shows a P5.js sketch window titled 'sketch_140613b'. The main canvas displays a grayscale depth image of a scene, likely a table with various objects. A yellow rectangular highlight is visible on the table surface. Below the canvas, the code editor shows the following code:</p> <pre> SimpleOpenNI Kinect; void setup() { size(640, 480); Kinect = new SimpleOpenNI(this); Kinect.enableDepth(); } void draw() { Kinect.update(); PImage depthImage = Kinect.depthImage(); image(depthImage, 0, 0); } void mousePressed(){ int depthValue = Kinect.depthMap(); } </pre> <p>The console output at the bottom shows three lines of data: 'mm: 604 in: 23.779528'.</p> | 60cm | 604 |

| Citra | Jarak (cm) | Nilai array (mm) |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------------|
|  <p>The screenshot shows a Java sketch window titled "sketch_140613b". The main window displays a grayscale depth image from a Kinect sensor. A yellow rectangular highlight is visible on a table in the foreground. Below the image, the code is as follows:</p> <pre> import openni.*; void setup() { size(640, 480); kinect = new SimpleOpenNI(this); kinect.enableDepth(); } void draw() { kinect.update(); PImage depthImage = kinect.depthImage(); image(depthImage, 0, 0); } void mousePressed(){ int[] depthValues = kinect.depthMap(); } </pre> <p>The console output at the bottom shows three lines of data:</p> <pre> mm: 628 in: 24.72441 mm: 628 in: 24.72441 mm: 628 in: 24.72441 </pre> | 61 cm | 610 |

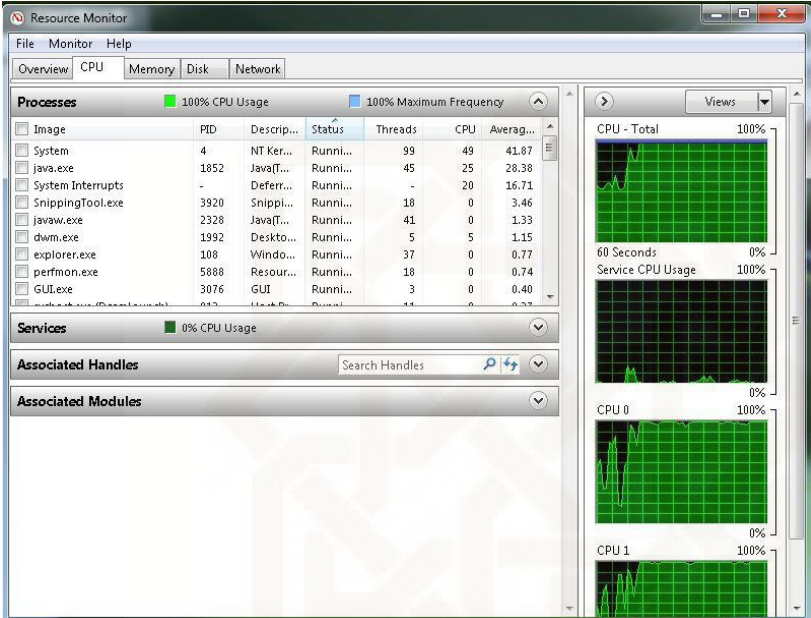
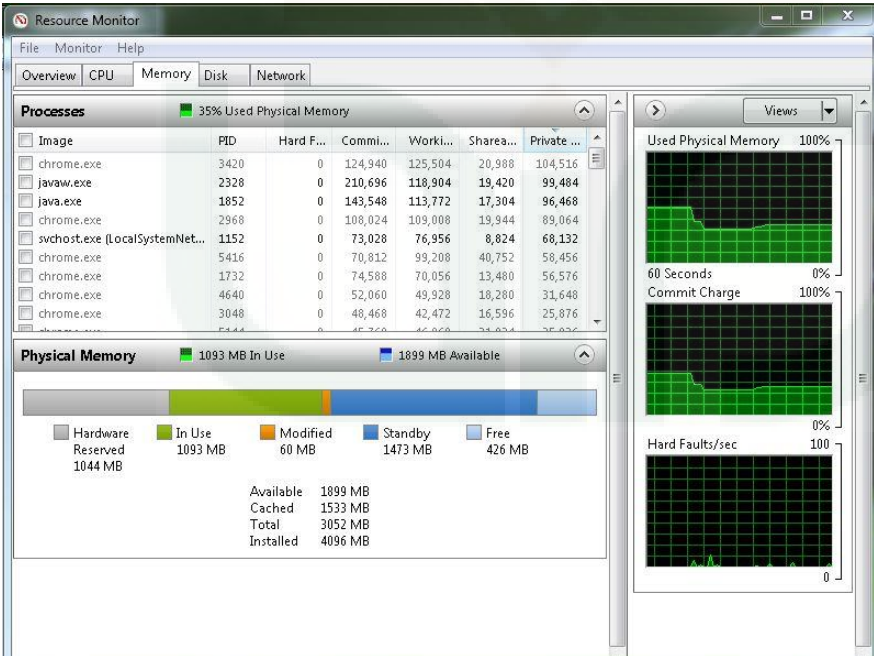
| Citra | Jarak (cm) | Nilai array (mm) |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------------|
|  <pre> SimpleOpenNI kinect; void setup() { size(640, 480); kinect = new SimpleOpenNI(this); kinect.enableDepth(); } void draw() { kinect.update(); PImage depthImage = kinect.depthImage(); image(depthImage, 0, 0); } void mousePressed(){ int[] depthValues = kinect.depthMap(); } // Press any key to stop // Press any key to stop // Press any key to stop </pre> <p> na: 902 in: 35.51181 na: 902 in: 35.51181 na: 902 in: 35.51181 </p> | 90 cm | 902 |

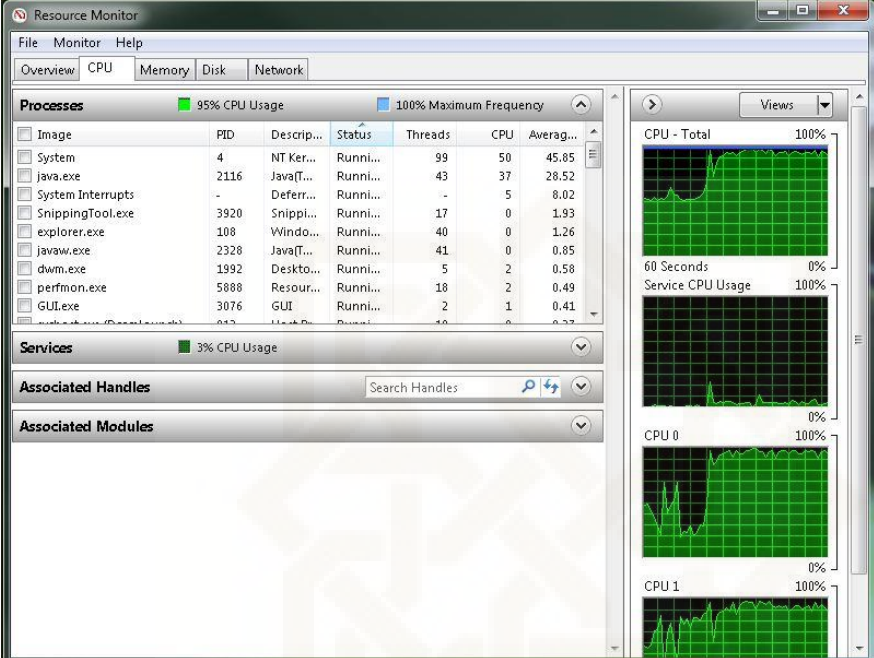
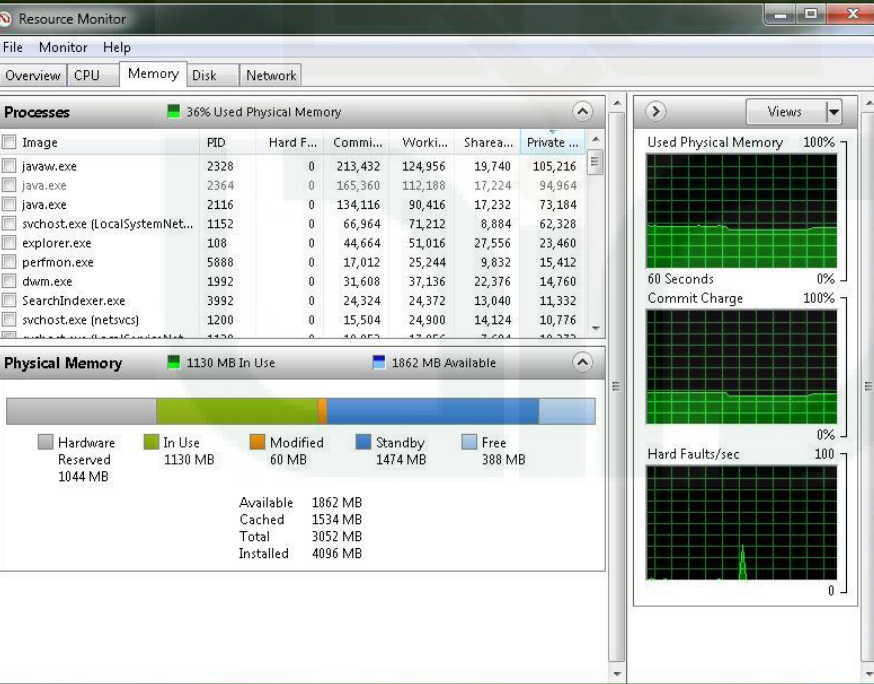
| Citra | Jarak (cm) | Nilai array (mm) |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------------|
|  <pre> sketch_140613b SimpleOpenNI Kinect, void setup() { size(640, 480); Kinect = new SimpleOpenNI(this); Kinect.enableDepth(); } void draw() { Kinect.update(); PImage depthImage = Kinect.depthImage(); image(depthImage, 0, 0); } void mousePressed(){ int[] depthValues = Kinect.depthMap(); } mm: 928 in: 36.535435 mm: 928 in: 36.535435 mm: 928 in: 36.535435 1 </pre> | 91 cm | 928 |

| Citra | Jarak (cm) | Nilai array (mm) |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------------|
|  <pre> SimpleOpenNI kinect; void setup() { size(640, 480); kinect = new SimpleOpenNI(this); kinect.enableDepth(); } void draw() { kinect.update(); PImage depthImage = kinect.depthImage(); image(depthImage, 0, 0); } void mousePressed(){ int[] depthValues = kinect.depthMap(); </pre> <p> mm: 1800 in: 70.86614 mm: 1800 in: 70.86614 mm: 1800 in: 70.86614 </p> | 180 cm | 1800 |

| Citra | Jarak (cm) | Nilai array (mm) |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------------|
|  <pre> void setup() { size(640, 480); kinect = new SimpleOpenNI(this); kinect.enableDepth(); } void draw() { kinect.update(); PImage depthImage = kinect.depthImage(); image(depthImage, 0, 0); } void mousePressed(){ int[] depthValues = kinect.depthMap(); } </pre> <p>mm: 1809 in: 71.220474 mm: 1809 in: 71.220474 mm: 1809 in: 71.220474</p> | 181 cm | 1809 |

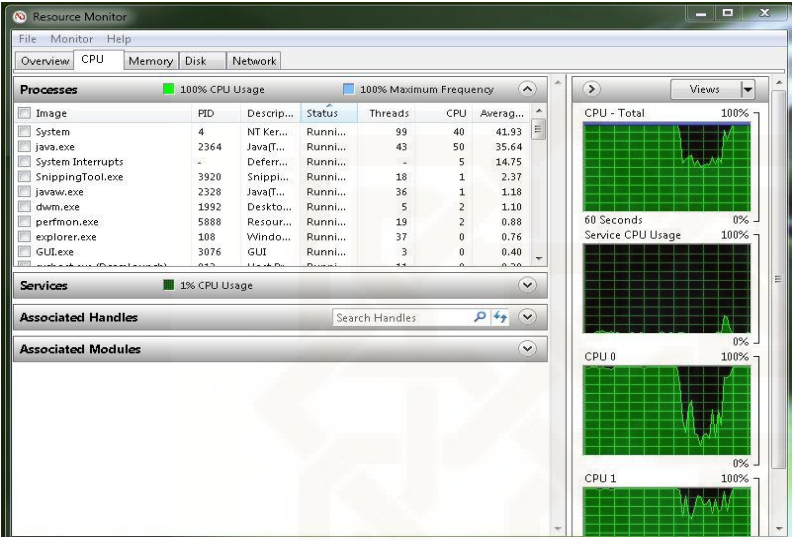
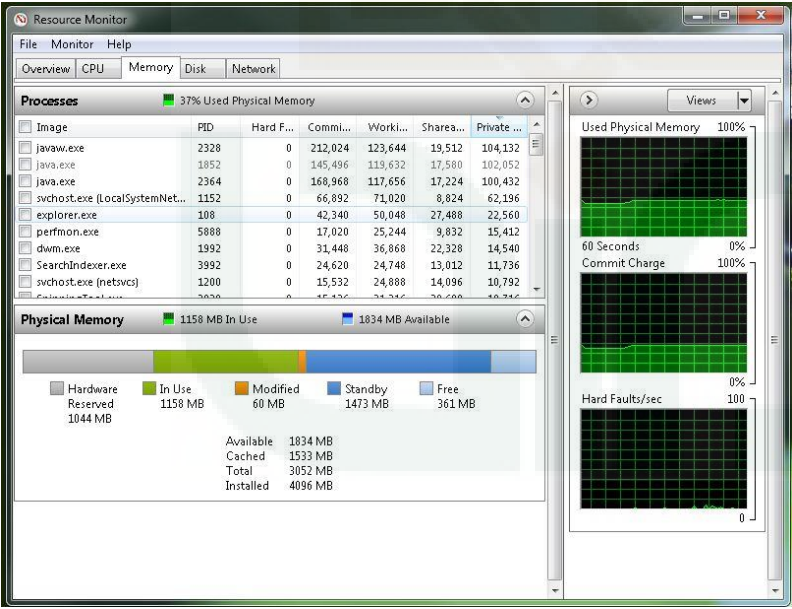
LAMPIRAN 6.DATA PENGUJIAN USAGE GRAYSCALE

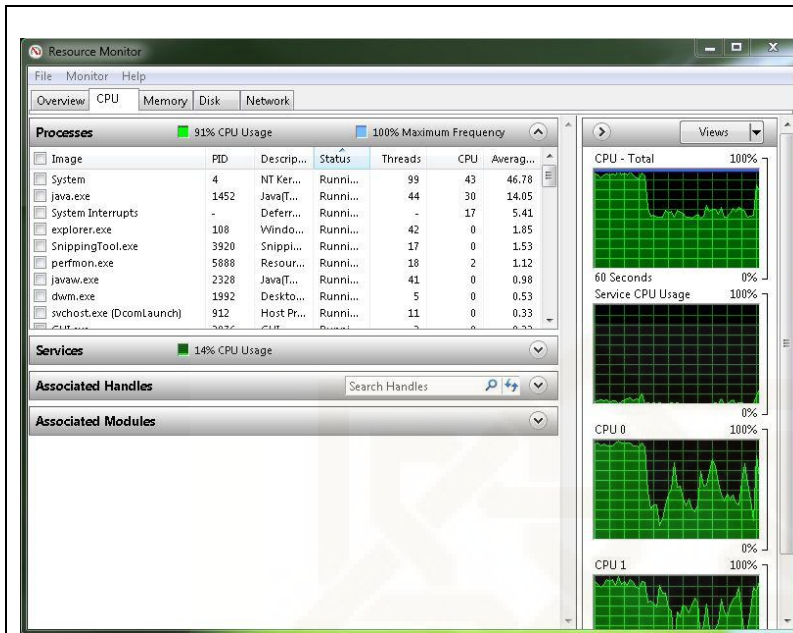
| Citra | Pengujian | Hasil |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------------|
|  <p>The screenshot shows the Windows Resource Monitor application with the CPU tab selected. The 'Processes' list shows several Java-based applications (java.exe, javaw.exe) and other system processes. The 'CPU' section at the top indicates '100% CPU Usage'. On the right side, there are four graphs: 'CPU - Total' (100%), 'Service CPU Usage' (100%), 'CPU 0' (100%), and 'CPU 1' (100%).</p> | Pola | 100% <i>CPU</i> |
|  <p>The screenshot shows the Windows Resource Monitor application with the Memory tab selected. The 'Processes' list shows multiple instances of chrome.exe and java.exe. The 'Physical Memory' section at the top indicates '1093 MB In Use' and '1899 MB Available'. Below this is a bar chart showing memory usage breakdown: Hardware Reserved (1044 MB), In Use (1093 MB), Modified (60 MB), Standby (1473 MB), and Free (426 MB). On the right side, there are three graphs: 'Used Physical Memory' (100%), 'Commit Charge' (100%), and 'Hard Faults/sec' (0).</p> | Pola | 1093MB RAM |

| Citra | Pengujian | Hasil |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|----------------------|
|  <p>The screenshot shows the Windows Resource Monitor application with the CPU tab selected. The 'Processes' section shows a list of running processes with columns for PID, Description, Status, Threads, CPU, and Average CPU. The 'CPU' usage is indicated as 95%. On the right, there are four graphs: 'CPU - Total' (100%), 'Service CPU Usage' (100%), 'CPU 0' (100%), and 'CPU 1' (100%).</p> | Jarak | 100% <i>CPU</i> |
|  <p>The screenshot shows the Windows Resource Monitor application with the Memory tab selected. The 'Physical Memory' section shows a bar chart and text indicating '1130 MB In Use' and '1862 MB Available'. Below this, a detailed breakdown is provided: Hardware Reserved (1044 MB), In Use (1130 MB), Modified (60 MB), Standby (1474 MB), and Free (388 MB). The 'Processes' section shows a list of processes with columns for PID, Hard F..., Commi..., Worki..., Share..., and Private... On the right, there are three graphs: 'Used Physical Memory' (100%), 'Commit Charge' (100%), and 'Hard Faults/sec' (0).</p> | Jarak | 1130MB <i>RAM</i> |

| Citra | Pengujian | Hasil |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|---------------|
| <p>The screenshot shows the Windows Resource Monitor with the CPU tab selected. The 'Processes' section displays a list of running processes, including System, System Interrupts, java.exe, SnippingTool.exe, explorer.exe, dwm.exe, javaw.exe, GUI.exe, and svchost.exe. The CPU usage is at 100%. On the right side, there are four performance graphs: 'CPU - Total' at 100%, 'Service CPU Usage' at 100%, 'CPU 0' at 100%, and 'CPU 1' at 100%.</p> | Tracking | 100% CPU |
| <p>The screenshot shows the Windows Resource Monitor with the Memory tab selected. The 'Physical Memory' section shows a bar chart and summary: 37% Used Physical Memory, 1142 MB In Use, and 1850 MB Available. A detailed breakdown shows: Hardware Reserved (1044 MB), In Use (1142 MB), Modified (60 MB), Standby (1477 MB), and Free (373 MB). On the right side, there are three performance graphs: 'Used Physical Memory' at 100%, 'Commit Charge' at 100%, and 'Hard Faults/sec' at 0.</p> | Tracking | 1142MB RAM |

LAMPIRAN 7. DATA PENGUJIAN USAGE DEPTH

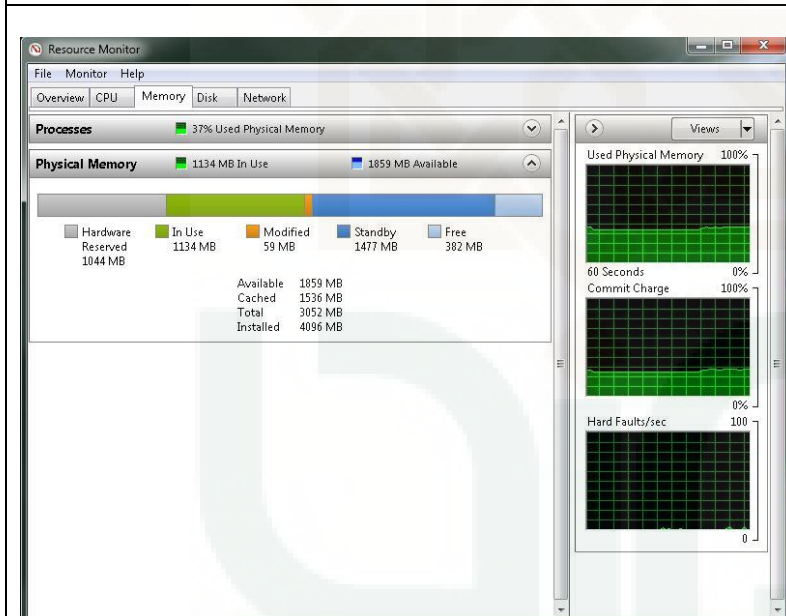
| Citra | Pengujian | Hasil | | | | | | | | | | | | | | | | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------------|--------|---------|----------|-------|---------|---------|------|--------|-----------|---------|--------|---------|-------|---------|-----------|---------|------|----------------------|
|  <p>The screenshot shows the Windows Resource Monitor window with the 'CPU' tab selected. The 'Processes' list shows 'System' with 99 threads and 41.93% CPU usage, and 'java.exe' with 43 threads and 35.64% CPU usage. The right-hand side displays four graphs: 'CPU - Total' at 100%, 'Service CPU Usage' at 100%, 'CPU 0' at 100%, and 'CPU 1' at 100%.</p> | Pola | 100% <i>CPU</i> | | | | | | | | | | | | | | | | | | |
|  <p>The screenshot shows the Windows Resource Monitor window with the 'Memory' tab selected. The 'Processes' list shows 'javaw.exe' with 2328 instances and 104,132 MB of private memory. The 'Physical Memory' section shows a bar chart with 1158 MB in use and 1834 MB available. Below the bar chart, a summary table is displayed:</p> <table border="1" data-bbox="244 1489 774 1646"> <tr> <td>Hardware Reserved</td> <td>1044 MB</td> </tr> <tr> <td>In Use</td> <td>1158 MB</td> </tr> <tr> <td>Modified</td> <td>60 MB</td> </tr> <tr> <td>Standby</td> <td>1473 MB</td> </tr> <tr> <td>Free</td> <td>361 MB</td> </tr> <tr> <td>Available</td> <td>1834 MB</td> </tr> <tr> <td>Cached</td> <td>1533 MB</td> </tr> <tr> <td>Total</td> <td>3052 MB</td> </tr> <tr> <td>Installed</td> <td>4096 MB</td> </tr> </table> <p>The right-hand side displays three graphs: 'Used Physical Memory' at 100%, 'Commit Charge' at 100%, and 'Hard Faults/sec' at 0.</p> | Hardware Reserved | 1044 MB | In Use | 1158 MB | Modified | 60 MB | Standby | 1473 MB | Free | 361 MB | Available | 1834 MB | Cached | 1533 MB | Total | 3052 MB | Installed | 4096 MB | Pola | 1158MB <i>RAM</i> |
| Hardware Reserved | 1044 MB | | | | | | | | | | | | | | | | | | | |
| In Use | 1158 MB | | | | | | | | | | | | | | | | | | | |
| Modified | 60 MB | | | | | | | | | | | | | | | | | | | |
| Standby | 1473 MB | | | | | | | | | | | | | | | | | | | |
| Free | 361 MB | | | | | | | | | | | | | | | | | | | |
| Available | 1834 MB | | | | | | | | | | | | | | | | | | | |
| Cached | 1533 MB | | | | | | | | | | | | | | | | | | | |
| Total | 3052 MB | | | | | | | | | | | | | | | | | | | |
| Installed | 4096 MB | | | | | | | | | | | | | | | | | | | |



Jarak

92%

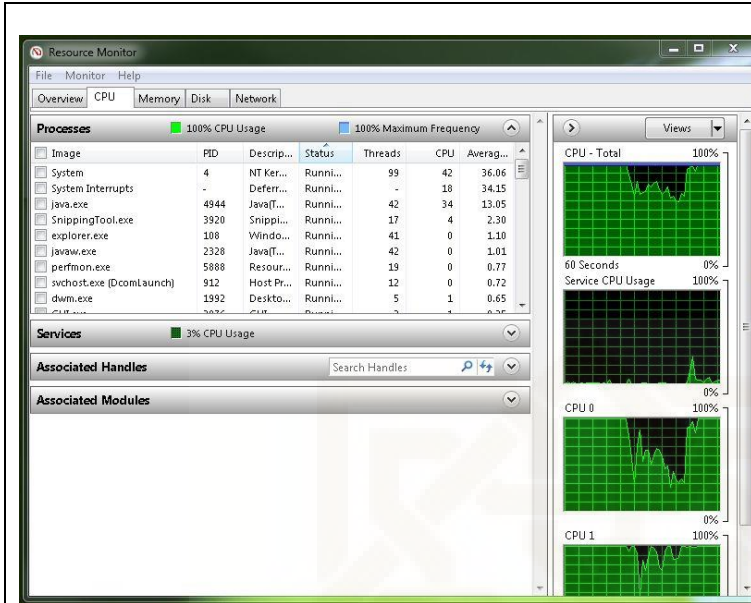
CPU



Jarak

1134MB

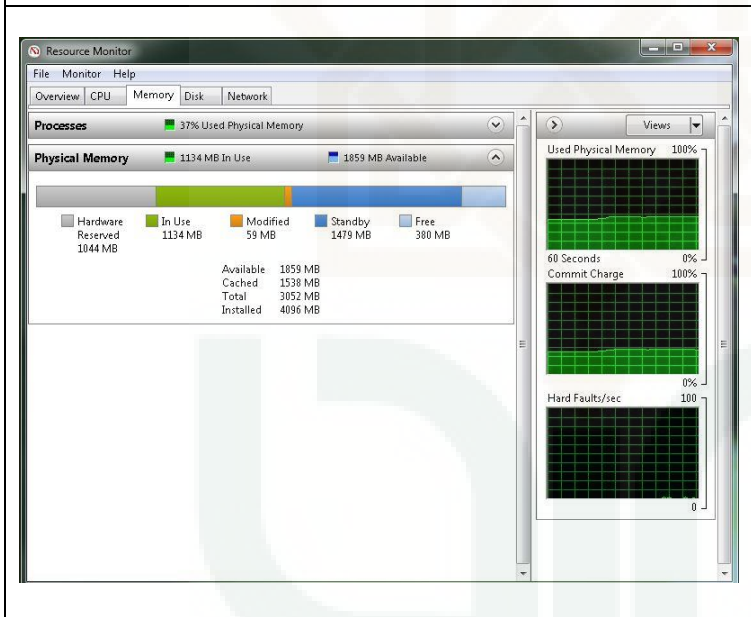
RAM



Tracking

100%

CPU



Tracking

1134MB

RAM

CURRICULUM VITAE

Nama : Fajria Antoni
Tempat, Tanggal Lahir : Bantul, 25 Oktober 1989
Jenis Kelamin : Laki-laki
Nama Ayah / Pekerjaan : Irawan, S.T / Swasta
Nama Ibu / Pekerjaan : Siti Hasanah, M.Pd . BI / Pegawai Negri
Alamat : Pondongan 13/02 Jagalan Banguntapan Bantul
Yogyakarta
No. Hp : 089 851 850 41
Email : fajriaantoni@gmail.com

Riwayat Pendidikan :

1996-2002 : SD Muhammadiyah Bodon, Yogyakarta
2002-2005 : SMP Negeri 1, Banguntapan
2005-2008 : SMA Negeri 10, Yogyakarta
2008-2013 : Program Studi Teknik Informatika, Fakultas
Sains Dan Teknologi, Universitas Islam Negeri
Sunan Kalijaga Yogyakarta