

# IMPLEMENTASI SURFACES 3D SCANNER MENGUNAKAN METODE TRIANGULATION DAN TESSELATION UNTUK REVERSE ENGINEERING OBYEK SEDERHANA

Bolo Dwiartomo<sup>1)</sup>, Ayi Ruswandi<sup>2)</sup>, Kuncoro Nuryanto<sup>3)</sup>

Politeknik Manufaktur Negeri Bandung  
Jl Kanayakan No. 21 – Dago, Bandung - 40135  
Phone/Fax : 022. 250 0241 / 250 2649

<sup>1)</sup> bolo@polman-bandung.ac.id, <sup>2)</sup> aruswandi@polman-bandung.ac.id, <sup>3)</sup> nuryanto\_kuncoro@hotmail.com

## Abstrak

Dewasa ini industri manufaktur sudah mulai membutuhkan *3D scanner* untuk melakukan *quality control products* dan proses *reverse engineering*. Ada beberapa metode untuk melakukan *scanning*. Salah satunya adalah menggunakan laser garis dengan penangkap citra berupa kamera. Metode ini diharapkan dapat menggabungkan kemampuan *metrology* yang dimiliki oleh *CMM* yang memiliki kepresisian tinggi dan stereo vision yang memiliki kecepatan pengolah citra 3D cukup tinggi. Pada makalah ini, akan dituliskan hasil penelitian dengan melakukan implementasi *surfaces 3D scanner* dengan menggunakan metode *triangulation* dan *tesselation*. Prinsip *triangulation* digunakan sebagai metode pengukuran jarak pada *laser range finder*. Sebagai pembuatan *3D file* menggunakan metode *tesselation* untuk menghubungkan titik-titik yang berdekatan pada *point clouds*. *3D file* yang dibuat akan dapat dibaca oleh *engineering software* seperti : Solidworks, Catia, AutoCad dan sebagainya. Metode pergerakan *scanning* mengadaptasi sistem *scanning* pada mesin *CMM*. Proses *scanning* telah dapat menghasilkan objek 3D, tetapi pada sisi tertentu terutama pada bidang miring posisi pixel yang tertangkap kamera kerap berada pada posisi yang tidak dapat didefinisikan oleh software.

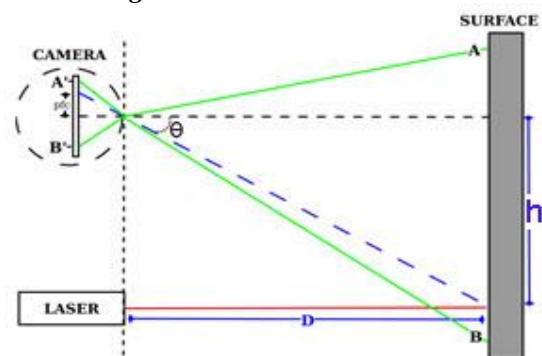
**Kata kunci:** *laser range finder, surfaces 3D scanner, metrology, rgb filter, 3D file, tesselation, triangulation.*

## 1. Pendahuluan

Dalam dunia manufaktur, *reverse engineering* dibutuhkan untuk membantu dalam melakukan *quality control product*. Produk yang sudah jadi akan di *scanning* kemudian data dari hasil *scanning* dibandingkan dengan desain produk dalam bentuk 3D. Dari hasil ini dapat diketahui penyimpangan antara produk jadi dengan desain.

*File 3D* yang dibentuk, berasal dari data koordinat berupa koordinat x, y, dan z. Data koordinat diperoleh melalui hasil *scanning* suatu objek. Metode untuk mendapatkan data koordinat sehingga menghasilkan data 3D menjadi kajian utama dalam penelitian ini. Metode yang digunakan adalah *triangulation* dan *tesselation*

## Prinsip Kerja Metode Triangulation Pada Laser Range Finder



Gambar 1. 1 Prinsip kerja perangkat sensor jarak [1]

Proyeksi *laser pointer* yang ditunjukkan dalam gambar 1.1 akan memberi titik cahaya ke suatu target yang berada dalam bidang pandang suatu kamera [1]. Sistem akan menjalankan algoritma untuk mencari titik merah yang paling tinggi pada bidang target yang akan dianggap sebagai

citra *laser pointer*. Titik *laser pointer* yang ditangkap akan mempunyai koordinat pada *image array* kamera yang diberi nama *pfc*. Posisi *pfc* ini nantinya akan dikonversi menjadi jarak.

Berdasarkan gambar 1.1, dengan menggunakan persamaan trigonometri dapat dihitung jarak (*D*) *laser pointer* dengan target yang dituju menjadi: [2]

$$D = \frac{h}{\tan \theta}$$

Nilai *h* (jarak kamera dengan *laser pointer*) bukan variabel melainkan suatu konstanta yang tidak boleh berubah. Parameter yang menjadi variabel yang dipengaruhi oleh jarak yaitu sudut  $\theta$ . Dari gambar 2.1 dapat dilihat bahwa perubahan sudut  $\theta$  akan sebanding dengan nilai *pfc*. Semakin besar *pfc* maka semakin besar pula sudut  $\theta$ . Oleh karena itu, dengan menggunakan nilai *rpc* sebagai faktor konversi dapat dituliskan persamaan sebagai berikut :

$$\theta = pfc * rpc$$

Dimana :

*pfc* = Jumlah *pixels* dari pusat *focal plane*  
*rpc* = *Radians per pixel pitch*

Untuk mendapatkan nilai *rpc* dapat menggunakan model persamaan linier untuk menemukan hubungan antara variable  $\theta$  dan *pfc*. Model ini didefinisikan dengan persamaan :

$$rpc = \frac{y_i}{z_i}$$

Nilai  $Y_i$  didapatkan dari sudut  $\theta$  aktual dikurangi dengan rata-rata jumlah nilai sudut  $\theta$  aktual dalam uji coba, sehingga dapat dituliskan sebagai berikut :

$$y_i = \theta_{aktual_i} - \frac{\sum_{i=1}^n \theta_{aktual_i}}{n}$$

Dimana

$$\theta_{aktual} = \{\theta_{aktual_1}, \theta_{aktual_2}, \dots, \theta_{aktual_n}\}$$

Sedangkan sudut  $\theta$  aktual diperoleh dari persamaan trigonometri, karena nilai *h* sudah diketahui dan jarak sebenarnya dengan titik *laser pointer* sudah diketahui, maka diperoleh persamaan :

$$\theta_{aktual_i} = \text{Arctan}\left(\frac{h}{D_{aktual}}\right)$$

Nilai  $Z_i$  diperoleh dari *pfc* dikurangi nilai rata-rata total *pfc* pada uji coba, sehingga persamaan dapat dituliskan sebagai berikut :

$$Z_i = pfc_i - \frac{\sum_{i=1}^n pfc_i}{n}$$

Dimana :  $pfc = \{pfc_1, pfc_2, \dots, pfc_n\}$

Nilai *pfc* belum tentu sebanding dengan nilai sudut  $\theta$ , karena adanya faktor deformasi lensa pada *webcamera*. Oleh sebab itu, perlu adanya parameter koreksi untuk melakukan koreksi dari faktor tersebut. Parameter koreksi itu biasa disebut sebagai *ro* (*radian offset*), sehingga persamaannya menjadi:

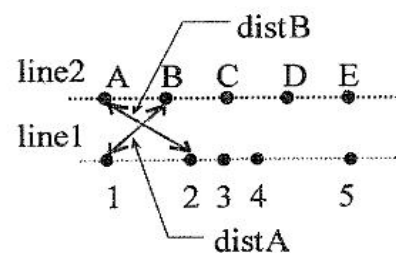
$$D = \frac{h}{\tan(pfc * rpc + ro)}$$

Nilai *ro* dapat diperoleh setelah nilai *pfc* dan *rpc* diperoleh. Cara mendapatkan nilai *ro* yaitu :

$$Ro_i = \theta_{aktual_i} - (pfc_i * rpc_i)$$

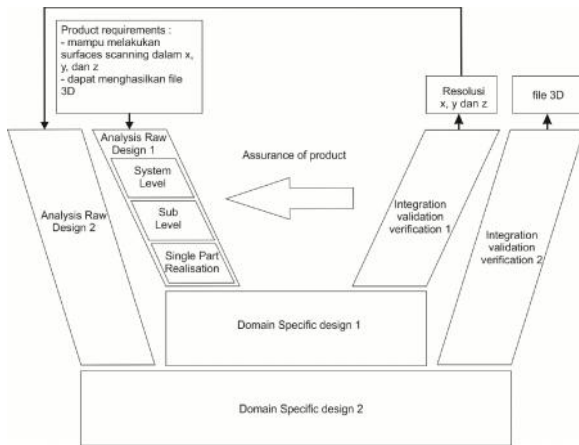
### Pembuatan 3D File Menggunakan Metode Tesselation

Pembuatan *3D file* membutuhkan 3 data koordinat atau *vertex* yang akan membentuk suatu *triangulation*. Metode pengolahan data koordinat menjadi *triangulation* ini dapat menggunakan algoritma *tesselation* dengan 2 data koordinat yang berdekatan [3]. Gambar 1.2 merupakan salah satu contoh data koordinat. Satu titik ini adalah pemodelan dari satu *vertex* yang mempunyai data koordinat (*x,y,z*). Misal titik A pada gambar 1.2 yang berdekatan adalah B, 1 dan 2. Jadi dengan menggunakan *tesselation*, *triangulation* yang dapat digunakan antara lain (*A,B,1*) , (*A,2,1*) , (*A,B,2*) atau (*A,1,2*). Dari keempat titik ini dicari mana titik yang paling berdekatan. Titik yang berdekatan yang lain selanjutnya dihubungkan seperti pada titik A.



Gambar 1. 2 Algoritma tesselation dengan 2 data poin yang berdekatan[3]

## 2. Perancangan dan Implementasi Sistem



Gambar 2. 1 Metode perancangan berdasarkan VDI 2206

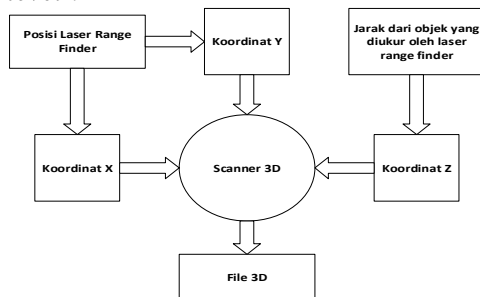
### 2.1. Product Requirement

*Product requirements* adalah kemampuan yang harus bisa dilakukan sistem. Sistem pada penelitian ini harus mampu melakukan *surfaces scanning* yang dapat mengambil data koordinat x, y dan z. Tidak hanya itu, sistem juga diharapkan agar dapat mengolah data koordinat menjadi *3D file*.

### 2.2. Analysis raw design 1

#### 2.2.1. Sistem Level

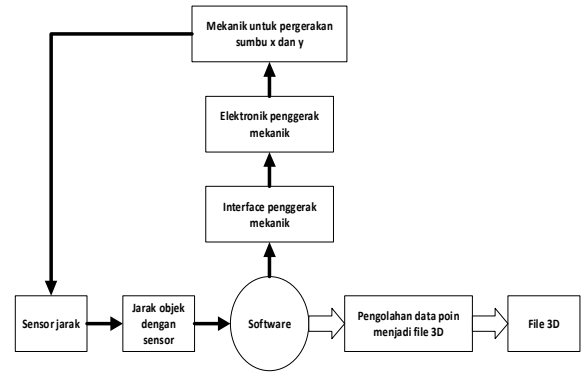
Rancangan dimulai dengan mendefinisikan tuntutan dari *surfaces scanner 3D* dan *output* yang akan dihasilkan. *Scanner 3D* membutuhkan data koordinat x, y dan z. Koordinat x dan y diperoleh dari posisi *laser range finder*. *Laser range finder* ini akan memberikan data koordinat berupa jarak yang nantinya akan menjadi koordinat z. Produk yang dihasilkan oleh *scanner 3D* ini adalah *3D file*. Berikut diagram dari *analysis raw design system level* :



Gambar 2. 2 Diagram analysis raw design sistem level

#### 2.2.2. Sub level

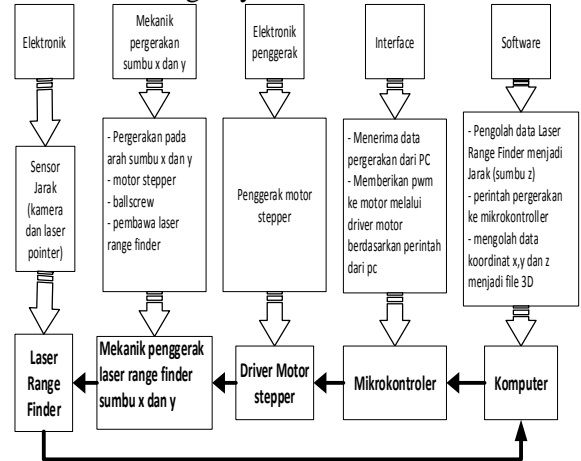
Tahap selanjutnya adalah *analysis raw design sub level* yang akan menjelaskan prinsip kerja dan bagaimana fungsi dari sistem *scanner*.



Gambar 2. 3 Analysis raw design sub level

### 2.2.3. Single Part Realisation

Tahap berikutnya *analysis raw design single part realisation* akan menentukan komponen apa saja yang akan digunakan berdasarkan tugasnya.



Gambar 2. 4 Diagram analysis raw design single part

Berdasarkan diagram tersebut didapatkan beberapa komponen yang dibutuhkan dalam perancangan penelitian ini. Berikut komponen yang akan digunakan :

#### a. Komputer

Komputer bertugas sebagai pengendali sistem, dari pergerakan x, y dan pengambilan data jarak dengan menggunakan *web camera* dan *laser pointer*. Program pengendali dibuat menggunakan Visual C++ 2012. Visual C++ dapat menggunakan *OpenCv library* dan *CvBlob Library*. *Library* ini adalah *library* yang dapat membantu dalam melakukan pengolahan citra.

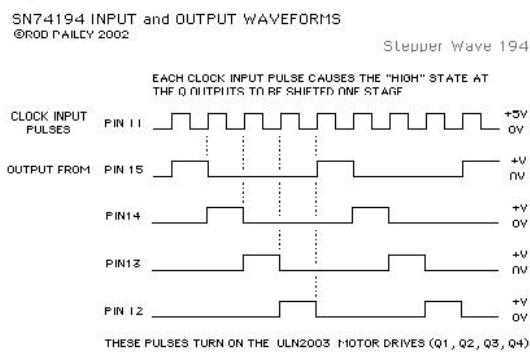
#### b. Mikrokontroler

Perangkat ini sebagai penghubung antara komputer dengan mekanik. Putaran motor X dan motor Y diatur oleh komputer dengan bantuan mikrokontroler. Spesifikasi mikrokontroler yang digunakan dapat menggunakan *Attiny 2313 minimum system*

dengan menambahkan modul komunikasi *rs232*. Namun, pada penelitian ini menggunakan modul *ATmega 8535 minimum system* yang sudah mempunyai modul komunikasi *rs232* untuk komunikasi dengan komputer. Modul ini digunakan karena sudah banyak dijual di toko-toko elektronik sehingga tidak perlu melakukan perakitan kembali. Mikrokontroler ini dapat diprogram menggunakan *software* Atmel Studio yang merupakan *software* buatan Atmel untuk memprogram mikrokontroler seperti Atmega.

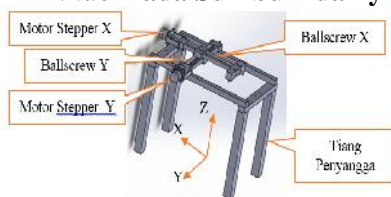
#### c. Stepper Motor Driver

Penggerak motor *stepper* membutuhkan rangkaian *driver* yang fungsinya untuk memberikan catu daya ke motor *stepper*. *Driver* tidak hanya mengeluarkan tegangan, namun tegangan yang dikeluarkan juga harus dalam bentuk pulsa karena motor *stepper* bergerak *step by step* sesuai dengan pulsa yang diberikan. Berikut pulsa *driver* pada motor *stepper unipolar*:



Gambar 2. 5Pulsa driver stepper motor[4]

#### d. Mekanik Pergerakan Laser range Finder Pada Sumbu x dan y



Gambar 3. 6Alat Scanner pada sumbu x dan y

Mekanik ini terdiri dari 2 *ballscrew*. Satu *ballscrew* untuk pergerakan arah sumbu x dan satunya lagi digunakan sebagai pergerakan sumbu y. Setiap *ballscrew* digerakkan oleh satu motor *stepper*. Tujuan rancangan penggerak motor *stepper* ini untuk menyederhanakan sistem karena banyaknya rotasi pada motor *stepper* berbanding lurus dengan banyaknya pulsa input yang diberikan. Oleh karena itu, pada motor *stepper* tidak diperlukan *encoder* yang berfungsi sebagai sensor penghitung jumlah putaran.

Dimensi obyek yang dapat diukur pada bidang x dan y adalah 250 x 250 mm. Panjang obyek yang dapat diukur pada bidang z akan ditentukan setelah mendapatkan resolusi dari *laser range finder*. Resolusi ini juga akan menentukan ketinggian dari tiang penyangga mekanik. Tiang penyangga akan dirancang pada *analysis raw design 2*.

#### e. Perangkat Sensor Laser Range Finder

Perangkat sensor *Laser Range Finder* yang digunakan terdiri dari *web camera* dan *laser pointer*. Sensor ini digunakan untuk pengukuran jarak antara objek dengan kamera. Jarak antara *laser* dan kamera adalah 50 mm. Berikut spesifikasi dari komponen yang digunakan untuk membuat *laser range finder*:

##### 1. Web camera

*Web camera* sebagai penangkap citra *laser pointer*. Citra ini kemudian diolah oleh komputer. Spesifikasi dari *web camera* :

- Model : logitech c160
- Ukuran Image Array : 640 x 480

##### 2. Laser Pointer

- *Laser (Light Amplification by Stimulated Emission of Radiation)* merupakan mekanisme suatu alat yang memancarkan radiasi elektromagnetik, biasanya dalam bentuk cahaya yang tidak dapat dilihat maupun dapat dilihat dengan mata normal, melalui proses pancaran terstimulasi. *Laser pointer* ini digunakan sebagai citra yang akan ditangkap *webcamera*. Objek yang terkena *laser pointer* adalah daerah yang diukur jaraknya.



Gambar 2. 7Laser range finder

#### 2.3.Domain Specific Design 1

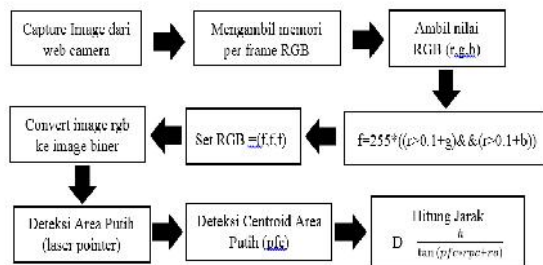
Tujuan dari tahap ini adalah melakukan *modelling* dalam bentuk *flowchart*. Produk yang akan dihasilkan pada V-Model pertama yaitu resolusi dari setiap koordinat x, y dan z. Resolusi tersebut akan menjadi resolusi untuk membuat *3D file* pada V-Model ke 2. Tahap ini akan dibagi dua sistem untuk memudahkan dalam mendapatkan resolusi yaitu:



2.3.1. Perintah Pergerakan Motor Stepper (Resolusi Koordinat x dan y)

Motor stepper sebagai alat penggerak mekanik. Terdapat 2 motor stepper yang dapat menggerakkan sumbu x dan y. Setiap motor stepper terdapat driver motor yang dikendalikan oleh satu mikrokontroler. Mikrokontroler ini akan memberikan pulsa ke motor stepper melalui driver motor berdasarkan perintah komputer [4]. Komputer memberikan perintah ke motor stepper dalam koordinat sumbu x dan y untuk berputar Cw maupun Ccw.

2.3.2. Pengolahan Citra Laser Pointer Menjadi Jarak (Resolusi Koordinat Z)



Gambar 3. 8 Proses pengolahan citra laser menjadi jarak

Program Visual C++ mengambil data array RGB dari web camera. Nilai RGB pada citra diatur kembali dengan menggunakan formula penyaring warna RGB. Dari hasil ini akan didapatkan citra berwarna hitam dan putih. Warna putih merepresentasikan titik laserpointer. Dengan bantuan CvBlob untuk mendeteksi area putih, akan didapatkan posisi sekaligus centroid dari area putih tersebut. Program akan mencari jarak vertikal image array dari centroid area putih. Jarak tersebut dihitung menggunakan rumus triangulation. Resolusi dan kalibrasi nilai rpc dan ro yang didapatkan dari perhitungan melalui metode triangulation dapat dilihat pada analisa dan pembahasan



Gambar 3. 9 Ilustrasi scanning pixel untuk tiap garis

2.4. Integration Validation Verification 1

Berdasarkan domain specific design terdapat 2 sistem. Oleh karena itu, pada bagian ini sistem tetap akan dibagi dua tetapi harus dapat menghasilkan resolusi x, y dan z. Resolusi x dan y berasal dari pergerakan ballscrew yang digerakkan motor stepper. Resolusi z didapatkan dari laser range finder.

2.5. Analysis raw design 2

Dari hasil V-Model pertama didapatkan nilai resolusi dari tiap koordinat x, y dan z. Nilai resolusi pada x dan y akan menjadi faktor pengali dari setiap koordinat x dan y. Sedangkan untuk koordinat akan langsung didapatkan nilainya. Pada tahap ini tidak melakukan perubahan bagian elektronik. Pada bagian mekanik dirancang tiang penyangga ballscrew dengan ketinggian yang berasal dari resolusi sumbu z. Untuk software pada komputer terjadi penambahan untuk melakukan pergerakan seperti pada CMM, mengambil data koordinat dari setiap pergerakan dan membuat 3Dfile.

2.6. Domain Specific Design 2

Pada tahap ini melakukan modelling untuk pergerakan mekanik dalam pengambilan koordinat xyz dan pengolahan koordinat x, y, dan z menjadi STL file. Cara membuat file ini dapat dilihat pada Bab II dan flowchart dapat dilihat pada lampiran F.

2.6.1. Proses Pengambilan Koordinat x, y dan z

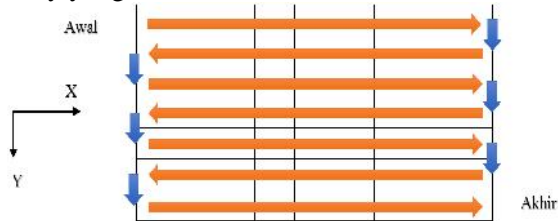
Pengendali utama alat ini adalah komputer. Komputer akan melakukan capture image melalui kamera kemudian mengolahnya sebagai data jarak. Pergerakan alat scanning pada sumbu x dan y akan ditentukan oleh komputer dengan bantuan mikrokontroler menggunakan komunikasi RS 232.



Gambar 3. 10 Gambaran umum sistem kerja

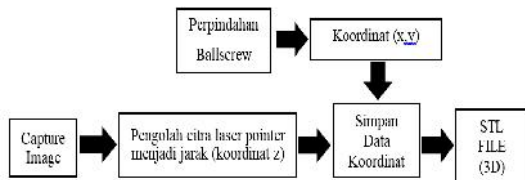
Untuk menggerakkan alat ke arah x dan y ini mempergunakan ballscrew yang digerakkan oleh motor stepper. Pola pergerakan scanning dapat dilihat pada gambar 3.11

mengikuti pola pergerakan pada mesin CMM. Pergerakan awal adalah arah x sampai x akhir kemudian dilanjutkan pergerakan satu unit arah y. *Scanning* di lanjutkan dalam arah x akhir sampai x awal. Proses akan berulang sampai x dan y yang sudah ditentukan.



Gambar 2. 11 Pola pergerakan scanning dilihat dari sumbu z

2.6.2. Proses Pengolahan Data Koordinat x, y dan z



Gambar 2. 12 Proses pengolahan data koordinat

Pada penelitian ini diperlukan koordinat x, y dan z. Koordinat tersebut digunakan untuk pembuatan 3D file. Koordinat x dan y berasal dari posisi *laser range finder* yang digerakkan oleh *ballscrew* pada mekanik. Hasil pengolahan citra *laser pointer* menjadi jarak akan disimpan sebagai koordinat z. Setelah semua koordinat diperoleh program akan melakukan konversi data koordinat menjadi *STL file*. File ini adalah 3D file yang dapat dibaca oleh 3D software seperti Autocad, Solidworks, dan sebagainya.

2.7. Integration Validation Verification 2

Pada tahap ini yang mengalami perubahan adalah program pada Visual C++ dengan menambahkan program untuk membuat 3D file sesuai dengan flowchart pada *Domain Specific Design 2*.

3. Pengujian Dan Analisa

3.1. Kalibrasi Laser Range Finder

Berikut tabel hasil kalibrasi *laser range finder* menggunakan h = 50 mm. Pada tabel tersebut didapatkan nilai *rpc* = 0.00147 dan *ro* = 0.033574 sehingga perhitungan nilai jarak adalah

$$D = \frac{50}{\tan(pfc * 0.00147 + 0.033574)}$$

No.	pfc	D aktual (mm)	θ aktual	Zi	rpc	y	ro
1	47	485	0.10273	-37.1	0.001467	-0.05444	0.033761
2	52	450	0.110657	-32.1	0.001449	-0.04651	0.035307
3	59	416	0.119618	-25.1	0.001496	-0.03755	0.031347
4	79	331	0.149924	-5.1	0.001421	-0.00725	0.03766
5	91	296	0.167339	6.9	0.001474	0.010168	0.033241
6	107	259	0.190704	22.9	0.001464	0.033533	0.034022
7	124	228	0.215881	39.9	0.001471	0.05871	0.033425
8	143	201	0.243808	58.9	0.001471	0.086636	0.033468
9	51	461	0.108038	-33.1	0.001484	-0.04913	0.032333
10	88	304	0.163014	3.9	0.001498	0.005843	0.031175
Total	841	Total	1.571713	Total	0.014696	Total	0.335739
Mean	84.1	Mean	0.157171	Mean	0.00147	Mean	0.033574

3.2. Resolusi Sumbu Z

Resolusi adalah nilai perubahan terkecil yang dapat dirasakan oleh alat ukur. Berdasarkan perhitungan *rpc* dan *ro*, selanjutnya dilakukan perhitungan untuk membandingkan antara resolusi terhadap jarak dan *pfc*. Resolusi ini adalah selisih antara D hitung yang nilai *pfc*-nya berdekatan, misalnya :

Nilai *pfc* = 1

$$D1 = \frac{50}{\tan(1 * 0.00147 + 0.033574)} = 1426 \text{ mm}$$

Nilai *pfc* = 2

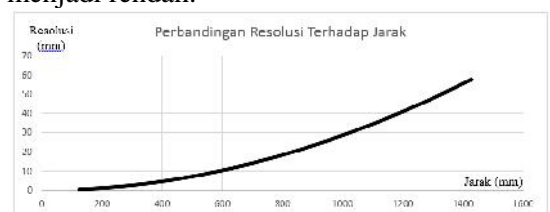
$$D2 = \frac{50}{\tan(2 * 0.00147 + 0.033574)} = 1369 \text{ mm}$$

Selisih antara D1 dengan D2 ini akan disebut sebagai resolusi. Jadi resolusi antara pada *pfc* = 1 dan *pfc* = 2 adalah 57 mm. Nilai 57 mm ini akan dijadikan resolusi pada *pfc* ke 1 pada kurva untuk mempermudah penjelasan. Berikut kurva yang terbentuk :



Gambar 3. 1 Perbandingan resolusi terhadap pfc

Gambar 3.1 menunjukkan bahwa semakin besar *pfc* maka resolusinya akan semakin tinggi hampir mendekati 0. Berbeda dengan *pfc*, gambar 3.2 menunjukkan perbandingan antara resolusi dengan jarak. Kurva tersebut menunjukkan bahwa semakin besar jarak maka akan mengakibatkan resolusi menjadi rendah.



Gambar 3. 2 Perbandingan resolusi terhadap jarak

3.3. Kalibrasi dan Resolusi Sumbu X

Pengukuran dilakukan dalam 10 unit perpindahan. Satu unit perpindahan nantinya akan menjadi perpindahan satu koordinat. Pada tabel di bawah ini pengukuran dilakukan dengan menggunakan jangka sorong dengan ketelitian 0.02 milimeter.

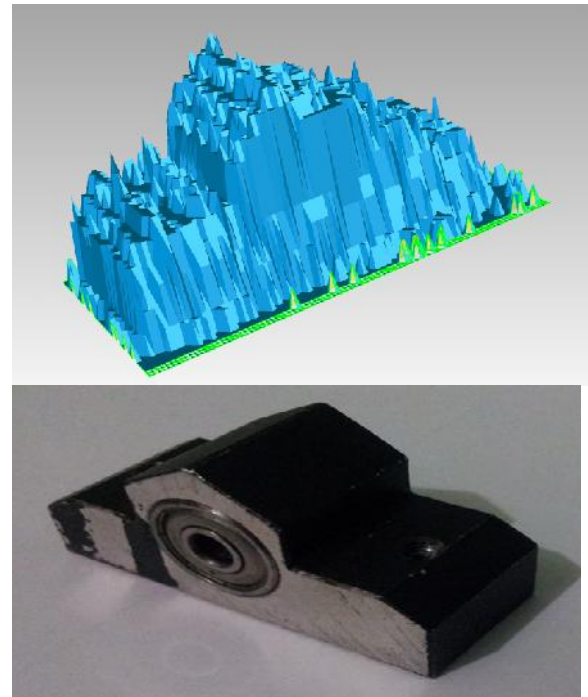
No.	Jarak (10 unit)	Jarak (1 unit)	S	S <sup>2</sup>
1	5.98	0.598	-0.0069	0.0000476100
2	6.12	0.612	0.0071	0.0000504100
3	5.96	0.596	-0.0089	0.0000792100
4	6.12	0.612	0.0071	0.0000504100
5	5.8	0.58	-0.0249	0.0006200100
6	5.98	0.598	-0.0069	0.0000476100
7	5.9	0.59	-0.0149	0.0002220100
8	6.14	0.614	0.0091	0.0000828100
9	6.1	0.61	0.0051	0.0000260100
10	6.1	0.61	0.0051	0.0000260100
11	6.12	0.612	0.0071	0.0000504100
12	6.06	0.606	0.0011	0.0000012100
13	6.12	0.612	0.0071	0.0000504100
14	5.9	0.59	-0.0149	0.0002220100
15	6.16	0.616	0.0111	0.0001232100
16	6.06	0.606	0.0011	0.0000012100
17	6.08	0.608	0.0031	0.0000096100
18	6.14	0.614	0.0091	0.0000828100
19	6.02	0.602	-0.0029	0.0000084100
20	6.12	0.612	0.0071	0.0000504100
<b>Total</b>	<b>120.98</b>	<b>12.098</b>	<b>Total S<sup>2</sup></b>	<b>0.0018518000</b>
<b>Mean</b>	<b>6.049</b>	<b>0.6049</b>	<b>(Total S<sup>2</sup>)/(n-1)</b>	<b>0.0000974632</b>
			<b>Standar Deviasi</b>	<b>0.009872343</b>

3.4. Kalibrasi dan Resolusi Sumbu Y

Kalibrasi sumbu y juga dengan menggunakan 10 unit koordinat seperti pada kalibrasi sumbu x.

No.	10 unit	1 unit	S	S <sup>2</sup>
1	6.02	0.602	0.0031	0.0000096100
2	6.08	0.608	-0.0029	0.0000084100
3	6	0.6	0.0051	0.0000260100
4	6.02	0.602	0.0031	0.0000096100
5	6.02	0.602	0.0031	0.0000096100
6	6.02	0.602	0.0031	0.0000096100
7	6.02	0.602	0.0031	0.0000096100
8	6.08	0.608	-0.0029	0.0000084100
9	5.94	0.594	0.0111	0.0001232100
10	6.1	0.61	-0.0049	0.0000240100
11	6.08	0.608	-0.0029	0.0000084100
12	6.1	0.61	-0.0049	0.0000240100
13	6.02	0.602	0.0031	0.0000096100
14	6.08	0.608	-0.0029	0.0000084100
15	6.1	0.61	-0.0049	0.0000240100
16	6.1	0.61	-0.0049	0.0000240100
17	6.08	0.608	-0.0029	0.0000084100
18	6.02	0.602	0.0031	0.0000096100
19	6.08	0.608	-0.0029	0.0000084100
20	6.06	0.606	-0.0009	0.0000008100
<b>Total</b>	<b>121.02</b>	<b>12.102</b>	<b>Total S<sup>2</sup></b>	<b>0.0003638000</b>
<b>Mean</b>	<b>6.051</b>	<b>0.6051</b>	<b>(Total S<sup>2</sup>)/(n-1)</b>	<b>0.0000191474</b>
			<b>Standar Deviasi</b>	<b>0.004375771</b>

3.5. Pengujian Rekonstruksi 3D



Gambar 3. 3 Hasil scanning dan benda aktual

4. Kesimpulan

- Penelitian ini hanya dapat melakukan *scanning* obyek pada bagian permukaan atas objek (*surfaces*).
- Variasi error laser range finder pada penelitian ini adalah  $\pm 2.417$  mm.
- Bentuk permukaan 3D dapat membentuk permukaan aktual obyek secara kasar.
- Rekonstruksi 3D hasil *scanning* belum dapat digunakan untuk pengukuran karena variasi error yang relatif besar dan bentuk 3D dengan bentuk aktual masih banyak penyimpangan.
- 3D file yang dibuat dapat dibaca oleh *engineering software* seperti: Solidworks, Catia, Autocad, Geomagic Studio dan sebagainya.

Saran

- Gunakanlah kamera dengan *image array* yang lebih besar sehingga bisa didapatkan ketelitian yang tinggi dalam pengukuran.
- Perlunya penambahan *metering* seperti pada kamera digital untuk pengolahan citra sehingga apabila terjadi perubahan intensitas cahaya, obyek yang diukur masih dapat terukur.
- Pada penelitian ini citra *laser pointer* yang ditangkap kamera diolah menggunakan algoritma titik berat (*centroid*) sehingga

apabila pada bidang tertentu misalnya bidang miring, citra laser tidak dapat ditangkap secara sempurna oleh kamera pada sudut tertentu. Hal ini menyebabkan *error* dalam pengukuran.

- d. Perlunya pengolahan data koordinat yang masih kasar dengan menggunakan metode lain seperti interpolasi, rata-rata dan sebagainya.
- e. Penggunaan system infinite triangulation approach untuk mengkompensasi kontur permukaan [5].

### Daftar Pustaka

- [1] C. E. Portugal-Zambrano and J. P. Mena-Chalco, "Robust Range Finder through a Laser Pointer and a Webcam," *Elsevier Science B.V*, 2011.
- [2] S. N. Utama, A. Muslim and O. Setyawati, "Analisis Perhitungan Laser Range finder Menggunakan Persamaan Geometri Pada Sistem Keamanan Ruang," *Jurnal EECCIS*, vol. 6, pp. 150-154, Desember 2012.
- [3] I. Tyvaert, P. G. Fadel and P. E. Rohaud, "A Methodology to Create STL Files from Data Point Clouds Generated with a Coordinate Measuring Machine," *Universite de Technologie de Troyes*, pp. 47-58.
- [4] linux-cae, "Stepper Motor Driver," <http://www.linux-consulting.com/>, 2004. [Online]. Available: <http://www.linux-cae.net/Projects/StepperMotor/TimerButton.rpaisley/Stepper.html>.
- [5] Danny Shacham, 'Lasers Take On Blades', SME Magazine Aerospace and Defense, 2012.