

# REKONSTRUKSI OBYEK TIGA DIMENSI DARI GAMBAR DUA DIMENSI MENGGUNAKAN METODE *GENERALIZED VOXEL COLORING-LAYERED DEPTH IMAGE*

Rudy Adipranata<sup>1</sup>, Kartika Gunadi<sup>2</sup>, Victor Julian Lipesik

Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121 – 131, Surabaya 60236

E-mail : rudya@petra.ac.id<sup>1</sup>, kgunadi@petra.ac.id<sup>2</sup>

**ABSTRAK:** Pada penelitian ini dikembangkan sebuah perangkat lunak untuk merekonstruksi obyek tiga dimensi dari kumpulan gambar dua dimensi dengan menggunakan metode *generalized voxel coloring-layered depth image* (GVC-LDI). Metode GVC-LDI ini melakukan rekonstruksi dengan bantuan *link list LDI* guna mencari *voxel-voxel* yang merupakan bagian dari obyek tiga dimensi berdasarkan warna. Guna penentuan *voxel* tersebut dilakukan perhitungan dari *pixel-pixel* yang merupakan proyeksi dari sebuah *voxel*. Perhitungan dilakukan dengan menggunakan standar deviasi warna untuk menentukan apakah *pixel-pixel* yang bersesuaian mewakili lokasi obyek yang sama. Apabila standar deviasi warna lebih kecil dari *threshold* maka dapat dikatakan bahwa *voxel* tersebut termasuk bagian obyek. Proses ini dilakukan secara berulang untuk semua *voxel* hingga didapatkan *voxel-voxel* yang merupakan bagian dari obyek. *Voxel* tersebut kemudian digambar pada layar monitor sehingga diperoleh hasil berupa obyek tiga dimensi yang menyerupai obyek asli yang terdapat pada gambar dua dimensi. Pada penelitian ini juga dilakukan perbandingan antara metode GVC-LDI dengan metode *Generalized Voxel Coloring-Image Buffer* (GVC-IB) yang juga merupakan varian dari metode GVC. Pengembangan dari perangkat lunak ini adalah pembuatan aplikasi 3D *modeling* secara otomatis dan pembuatan aplikasi animasi 3D secara *real time*.

**Kata kunci:** rekonstruksi tiga dimensi, *generalized voxel coloring, layered depth image*.

**ABSTRACT:** The objective of this research is to develop software which capable to reconstruct 3D object from 2D images as references using *Generalized Voxel Coloring - Layered Depth Image* method (GVC-LDI). This method reconstruct 3D object using *LDI link list* as help to find voxels which correspond to the objects based on color. To find the voxels, we calculate the color standard deviation of the pixels which is projected from the object. If the standard deviation is smaller than the threshold, the voxel evaluated as a part of the object. The process repeated for each voxel until it gets all of the voxels which shape the object. The voxels can be drawn to screen to get the photorealistic 3D object that represent the 2D images. In this research, we also compare the result of GVC-LDI and *Generalized Voxel Coloring – Image Buffer* (GVC-IB) which is one of the GVC variant also. Future development of the software is automatic 3D modeling application and real time 3D animation application.

**Keywords:** 3D reconstruction, *generalized voxel coloring, layered depth image*.

## PENDAHULUAN

Hingga saat ini, rekonstruksi obyek tiga dimensi yang berasal dari obyek *real* merupakan bidang penelitian komputer grafik yang masih aktif. Hal ini dikarenakan banyak aplikasi yang membutuhkan rekonstruksi tersebut, seperti *virtual reality, 3D modelling* secara otomatis, *augmented reality*, dan lain-lain. Banyak metode yang telah dikembangkan untuk melakukan hal ini dimana secara garis besar metode tersebut dapat dikelompokkan menjadi dua bagian yaitu *image based rendering* dan *image based modeling*.

Perbedaan antara *image based rendering* dengan *image based modeling* [1] adalah dari proses yang dilakukan dimana pada *image based modeling* banyak

dilakukan perhitungan matematis untuk melakukan rekonstruksi dari obyek dua dimensi sehingga menjadi geometri primitif yang pada akhirnya menyusun sebuah obyek tiga dimensi. Sedangkan pada *image based rendering*, proses dilakukan dengan melakukan analisa dari gambar 2D untuk menentukan bagian mana yang merupakan obyek tiga dimensi sehingga hasil keluaran adalah suatu bentuk gambar sintesis berupa obyek tiga dimensi yang direkonstruksi. Salah satu analisa yang sering digunakan pada gambar 2D untuk membentuk obyek 3D adalah analisa warna.

Pada *image based rendering*, mempunyai keuntungan mudahnya melakukan rekonstruksi suatu obyek yang bersifat *photorealistic*. Obyek *photorealistic* adalah sebuah obyek yang berasal dari obyek sebenarnya pada dunia nyata, jadi bukan obyek yang

merupakan hasil manipulasi atau kreasi dengan menggunakan perangkat lunak pengolah gambar. Sedangkan *image based modeling* untuk dapat melakukan rekonstruksi obyek *photorealistic* masih harus menggunakan metode lain seperti *texture mapping*.

Pada penelitian ini digunakan metode pada kelompok *image based rendering*, yaitu *Generalized Voxel Coloring-Layered Depth Image* yang dikemukakan oleh Greg Slabaugh [2]. Metode ini merupakan pengembangan dari metode *Voxel Coloring* yang dikemukakan oleh Seitz dan Dyer [3]. Metode *Generalized Voxel Coloring* (GVC) sendiri mempunyai dua varian yaitu *Image Buffer* dan *Layered Depth Image* [2]. Pada penelitian ini dilakukan implementasi varian *Layered Depth Image* dimana hal ini merupakan kelanjutan dari penelitian yang telah dilakukan sebelumnya yaitu implementasi varian *Image Buffer* [45]. Metode *GVC Layered Depth Image* mempunyai keuntungan mempunyai proses perbandingan yang lebih sedikit dibanding *Image Buffer*. Pada saat pengujian juga dilakukan perbandingan dengan menggunakan metode *Image Buffer* yang telah diimplementasikan pada penelitian yang terdahulu [4].

## KALIBRASI KAMERA

Kalibrasi kamera adalah salah satu langkah yang harus dilakukan dalam proses rekonstruksi 3D, dimana proses ini diperlukan untuk mendapatkan informasi parameter kamera yang digunakan untuk melakukan transformasi dari 3D (*world coordinate*) menuju ke 2D (*camera coordinate*). Ada beberapa metode yang sudah dikembangkan untuk melakukan proses kalibrasi kamera. Metode-metode ini dapat digolongkan ke dalam dua kategori, yaitu:

- *Photogrammetric Calibration*. Proses kalibrasi kamera dilakukan dengan mengamati obyek kalibrasi dimana geometri dalam ruang 3D telah diketahui dengan sangat tepat. Umumnya, metode ini dapat dilakukan dengan sangat efisien. Obyek kalibrasi biasanya terdiri dari dua atau tiga bidang yang terletak secara *orthogonal* satu dengan yang lainnya.
- *Self Calibration*. Metode-metode pada kategori ini tidak menggunakan obyek kalibrasi. Metode ini dilakukan dengan cara menggerakkan kamera pada pemandangan statis (*static scene*), yang dibatasi oleh parameter internal kamera dari perubahan letak kamera dengan menggunakan informasi yang terdapat pada gambar saja. Oleh karena itu, jika beberapa gambar diambil oleh kamera yang sama dengan parameter internal yang sama, hubungan antara gambar-gambar yang ada cukup untuk mendapatkan parameter internal dan eksternal yang ada.

Metode yang digunakan untuk melakukan proses rekonstruksi gambar 3D ini dikembangkan oleh Zhang [5]. Metode ini merupakan kombinasi dari *Photogrammetric Calibration* dan *Self Calibration*, karena menggunakan informasi metrik 2D dan menghasilkan informasi yang meliputi parameter intrinsik dan ekstrinsik yang akan digunakan dalam proses penghitungan letak benda dalam ruang 3D.

Parameter intrinsik ini terdiri dari empat unsur, yaitu nilai fokus kamera, yang merupakan jarak antara lensa kamera dengan bidang gambar, titik pusat proyeksi, yang merupakan lokasi titik tengah gambar dalam koordinat *pixel*, ukuran *pixel* efektif dan koefisien distorsi yang merupakan koefisien tingkat kelengkungan lensa yang meliputi radial dan tangensial distorsi.

Sedangkan parameter ekstrinsik ini terdiri dari dua buah matriks, yaitu matriks translasi dan matriks rotasi. Parameter ekstrinsik ini menggambarkan orientasi posisi dari kamera terhadap sistem koordinat sebenarnya dalam ruang 3D (*world coordinate*). Perhitungan untuk melakukan proyeksi dari 3D menuju 2D terdapat pada rumus (1) [5].

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fx & 0 & Cx \\ 0 & fy & Cy \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

dimana (u,v) adalah koordinat 2D (*camera coordinate*), (X, Y, Z) adalah koordinat 3D (*world coordinate*). (fx, fy) adalah fokus kamera, (Cx, Cy) adalah koordinat pusat gambar. Sedangkan  $r_{11}$  hingga  $r_{33}$  adalah matriks rotasi kamera serta  $t_1$  hingga  $t_3$  adalah matriks translasi kamera. Matriks-matriks tersebut didapat dari proses kalibrasi kamera.

## GENERALIZED VOXEL COLORING

*Generalized Voxel Coloring* (GVC) adalah sebuah metode rekonstruksi gambar 3D dari beberapa gambar 2D. GVC menghasilkan rekonstruksi gambar 3D dengan menggunakan teori tentang *visibility* yang tepat dan penempatan kamera yang berubah-ubah.

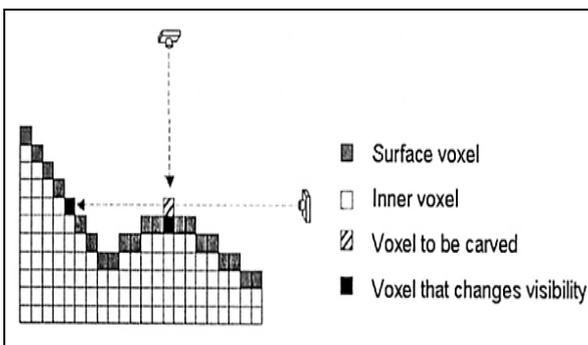
Pada GVC terdapat dua varian, yaitu GVC-IB (*Image Buffer*) dan GVC-LDI (*Layered Depth Image*). GVC-LDI adalah perbaikan dari metode GVC-IB yang merupakan algoritma dasar dari GVC. Perbedaan dari kedua metode ini terletak pada penggunaan memori dan waktu yang diperlukan untuk melakukan proses rekonstruksi. Pada penelitian ini digunakan metode GVC-LDI dimana nanti pada saat pengujian dibandingkan juga dengan GVC-IB yang telah diimplementasikan sebelumnya [4].

### Generalized Voxel Coloring–Layered Depth Image

Metode GVC bekerja dengan cara mengasumsikan awal sebuah ruang padat tiga dimensi yang tersusun atas *voxel-voxel*, dimana masing-masing *voxel* akan dievaluasi apakah dia merupakan bagian dari obyek atau bukan, dimana jika bukan merupakan bagian dari obyek akan dilakukan pemahatan terhadap *voxel* tersebut [2].

Pengevaluasian *voxel* dilakukan dengan memperhatikan *photo-consistency* dari *pixel* yang bersesuaian dengan *voxel* tersebut. *Photo-consistency* ini dihitung dengan menggunakan standar deviasi warna. Jadi untuk masing-masing *pixel* dari gambar-gambar 2D yang terproyeksi ke *voxel* yang sama dihitung apakah mempunyai standar deviasi lebih kecil daripada nilai *threshold*. Jika demikian, maka *voxel* tersebut merupakan bagian dari obyek, tetapi jika tidak, maka *voxel* tersebut akan dipahat. Proses evaluasi dilakukan mulai dari bagian permukaan ruang 3D menuju ke dalam sehingga pada akhirnya setelah tidak ada lagi *voxel* yang dapat dipahat, maka terbentuklah obyek 3D yang dimaksud [3].

Pada saat terjadi pemahatan, maka terjadi pula perubahan *visibility* dari *voxel* tersebut. Dimana *visibility* ini adalah informasi yang menyatakan apakah *voxel* terlihat dari sudut pandang mata orang yang melihat. Untuk memperlihatkan perubahan masalah *visibility*, dapat dilihat pada Gambar 1 [2].

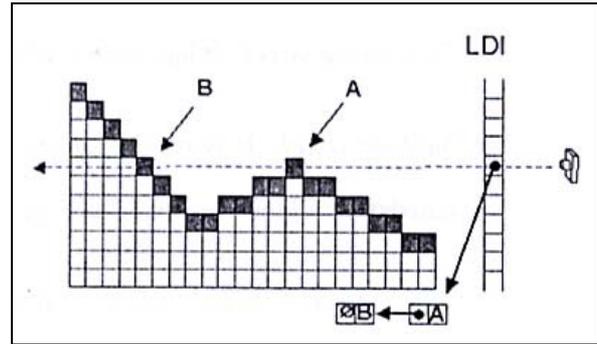


Gambar 1. *Voxel Visibility*

Jika terjadi perubahan *visibility* pada sebuah *voxel*, maka harus dihitung kembali *photo consistency* *voxel* tersebut.

Algoritma GVC-LDI menggunakan struktur data berupa link list untuk mencatat semua semua *surface voxel* yang terproyeksi pada *pixel P*. Hal ini ditunjukkan pada Gambar 3 [2].

Secara lengkap, algoritma GVC-LDI terdapat pada *pseudocode* Gambar 3 [2].



Gambar 2. Struktur *Layered Depth Image*

```

initialize SVL
render SVL to LDIs
for every voxel V
  carved(V) = false
copy SVL to CVSVL
while (CVSVL is not empty) {
  delete V from CVSVL
  compute vis(V)
  if (consist(vis(V)) = false) {
    carved(V) = true
    remove V from SVL
    for every pixel P in projection of V into all images {
      if (V is head of LDI(P))
        add next voxel on LDI(P) (if any) to CVSVL
      delete V from LDI(P)
    }
  }
  for every voxel N adjacent to V with N ∉ SVL {
    N_is_visible = false
    for every pixel P in projection of N to all images {
      add N to LDI(P)
      if (N is head of LDI(P))
        N_is_visible = true
    }
    add N to SVL
    if (N_is_visible)
      add N to CVSVL
  }
}
save voxel space

```

Gambar 3. *Pseudocode* GVC-LDI

### PERANCANGAN SISTEM

Secara garis besar, sistem perangkat lunak rekonstruksi gambar 2D menjadi gambar 3D ini terdiri dari empat bagian, yaitu:

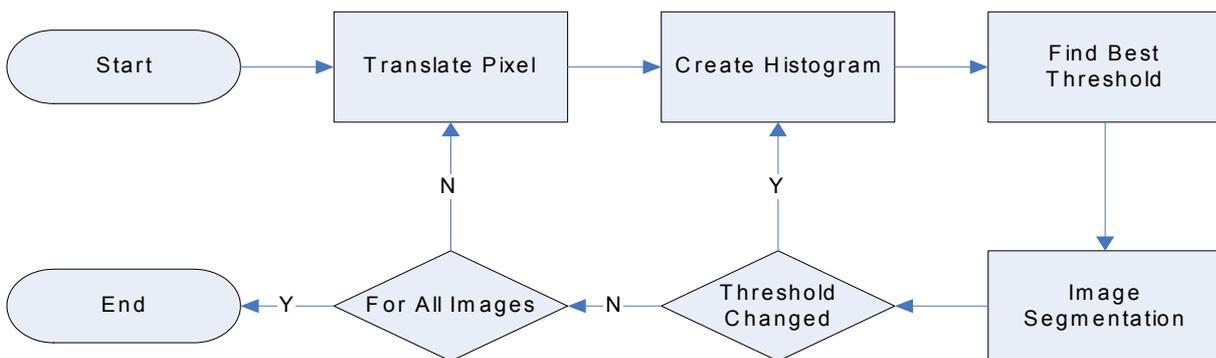
- *Input*  
Proses *input* terdiri dari proses pengambilan gambar *chessboard pattern* guna kalibrasi kamera dan gambar obyek dengan menggunakan kamera *digital*. Gambar yang didapatkan akan digunakan sebagai *input* pada proses selanjutnya.
- *Segmentation*  
Proses *segmentation* adalah proses untuk memisahkan obyek dengan lingkungan (latar belakang) sehingga memungkinkan sistem untuk hanya memproses obyek yang terdapat pada gambar.

- **Camera Calibration**  
 Proses *camera calibration* berfungsi untuk mencari parameter intrinsik kamera dan mencari matrik ekstrinsik tiap gambar yang nantinya digunakan untuk melakukan transformasi dari 3D (*world coordinate*) ke 2D (*camera coordinate*).
- **3D Reconstruction**  
*3D Reconstruction* adalah proses utama dari perangkat lunak, yaitu proses rekonstruksi obyek 3D dengan metode *generalized voxel coloring-layered depth image*.

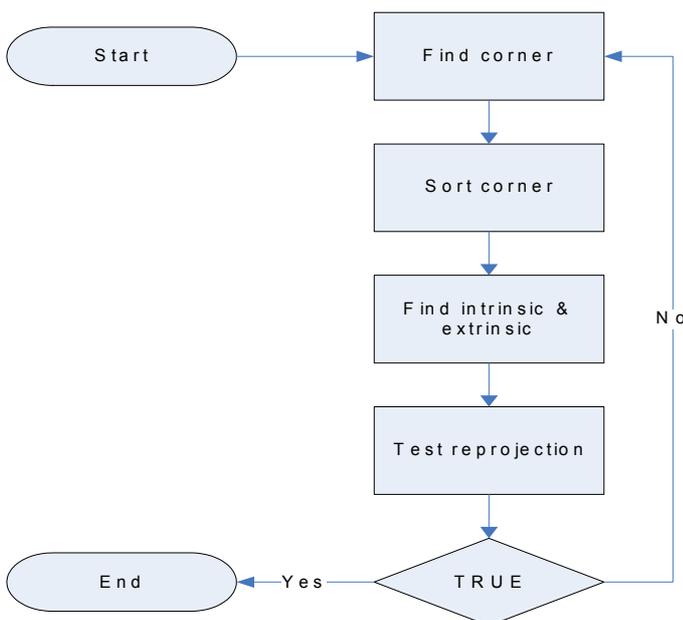
Input yang dibutuhkan oleh sistem terdiri dari dua yaitu input gambar 2D yang akan direkonstruksi serta input gambar *chessboard pattern*. Gambar *chessboard pattern* ini digunakan untuk proses kalibrasi kamera dimana titik-titik sudut pada *chessboard pattern* digunakan untuk inialisasi pemetaan antara koordinat 3D dan 2D yang telah diketahui.

Sebelum dilakukan proses rekonstruksi, input gambar 2D terlebih dahulu harus dilakukan segmentasi untuk memisahkan antara gambar obyek dengan latar belakang. Proses segmentasi ini dilakukan dengan menggunakan metode *iterative threshold*. Pada proses ini, tiap *pixel* akan dipetakan ke dalam sebuah *histogram* menurut nilai *grayscale*-nya. Kemudian, dicari nilai *threshold* sementara dan dilakukan proses segmentasi awal. Proses ini akan berulang selama nilai *threshold* yang baru masih berbeda jauh dari nilai *threshold* sebelumnya. Diagram alir proses segmentasi dapat dilihat pada Gambar 4.

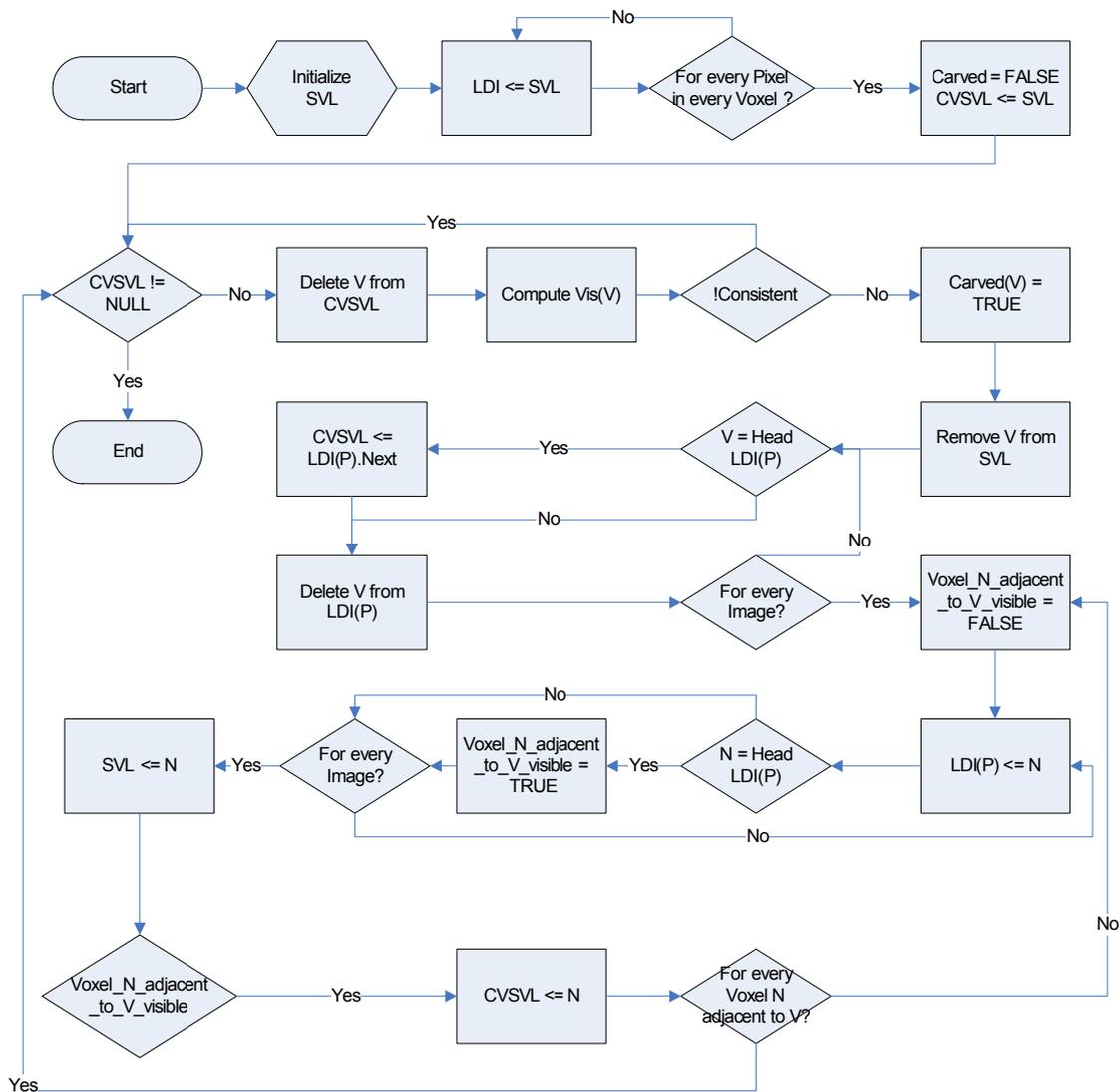
Setelah selesai proses segmentasi, dapat dilakukan proses kalibrasi kamera dimana diagram alir untuk proses kalibrasi kamera ini terdapat pada Gambar 5. Proses utama dari sistem adalah proses GVC-LDI itu sendiri dimana diagram alir untuk proses ini dapat dilihat pada Gambar 6.



Gambar 4. Diagram Alir Proses Segmentasi



Gambar 5. Diagram Alir Proses Kalibrasi Kamera



Gambar 6. Diagram Alir Generalized Voxel Coloring – Layered Depth Image

IMPLEMENTASI DAN HASIL PENGUJIAN

Aplikasi ini dikembangkan menggunakan bahasa pemrograman Microsoft Visual C++ 6.0. Untuk proses kalibrasi kamera digunakan fungsi-fungsi yang terdapat pada library OpenCV [6], dan untuk menampilkan hasil rekonstruksi ke layar monitor digunakan fungsi-fungsi yang terdapat pada library OpenGL [7].

Pada saat pengujian digunakan parameter-parameter seperti terdapat pada Tabel 1.

Tabel 1. Parameter Pengujian

Ukuran <i>chessboard pattern</i> dan ukuran setiap kotak	5 x 7 kotak, 10 mm <sup>2</sup>
Jumlah gambar <i>input</i> dan perubahan sudut	12 gambar, 30°
Ukuran gambar <i>input</i>	640 x 480 <i>pixel</i>
Ukuran <i>voxel</i>	1 mm <sup>3</sup>
Nilai <i>threshold</i> pembeding	25, 75, 125, 175

Pada pengujian ini digunakan 12 gambar dimana 3 gambar terlihat pada Gambar 7.



Gambar 7. Tiga Gambar dari 12 Gambar Input

Proses rekonstruksi dilakukan pada volume 40 mm x 60 mm x 80 mm. Hasil proses rekonstruksi dengan gambar input serta parameter yang ditunjukkan pada Tabel 1 dapat dilihat pada Gambar 8.

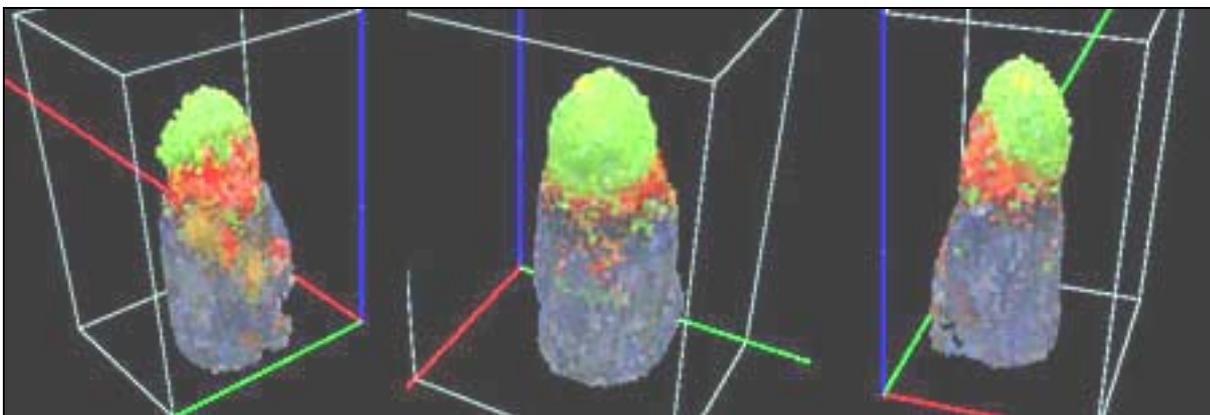
Guna mendapatkan hasil yang baik perlu dilakukan percobaan rekonstruksi dengan nilai *threshold* yang berbeda-beda. Gambar 9 adalah gambar hasil pengaruh variasi nilai *threshold* pada rekonstruksi 3D.

Hasil rekonstruksi yang didapat dengan menggunakan GVC-LDI dibandingkan pula dengan hasil rekonstruksi dengan menggunakan metode GVC-IB yang telah dilakukan pada penelitian sebelumnya [4]. Hasil perbandingan dapat dilihat pada Tabel 2.

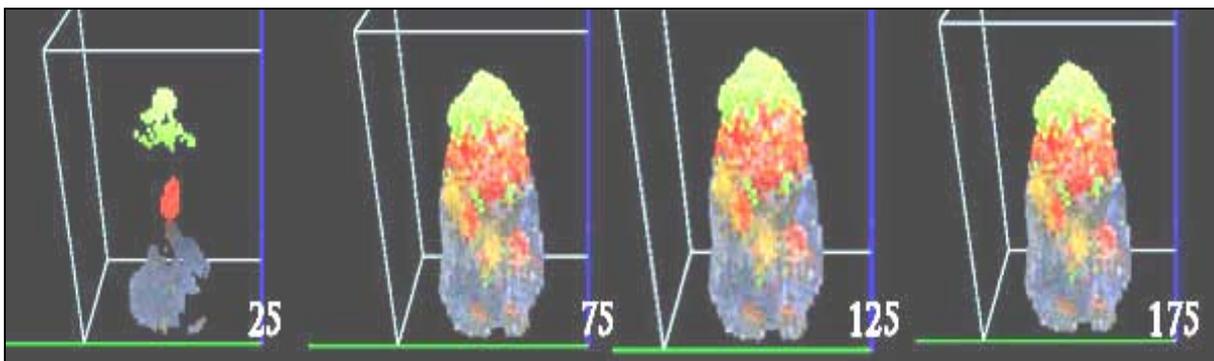
Tabel 2. Perbandingan Metode GVC-IB dan GVC-LDI

Metode	GVC-IB				GVC-LDI			
Threshold	10	30	75	100	10	30	75	100
Jumlah proses evaluasi (ribuan)	26,02	47,3	75,4	92,9	3,96	3,87	3,81	3,80
Jumlah voxel hasil	1687	3539	5509	7497	1687	1851	1969	1987
Waktu Proses (dalam 0,3 detik)	1,1	1,4	1,6	158	171	173	178	
Jumlah Memory (dalam Mb)	54,7	56,5	54,5	54,3	73,1	72,8	72,8	71,7

Dari hasil pengujian pada Tabel 2, terlihat bahwa GVC-LDI unggul dalam hal jumlah proses evaluasi



Gambar 8. Hasil Rekonstruksi

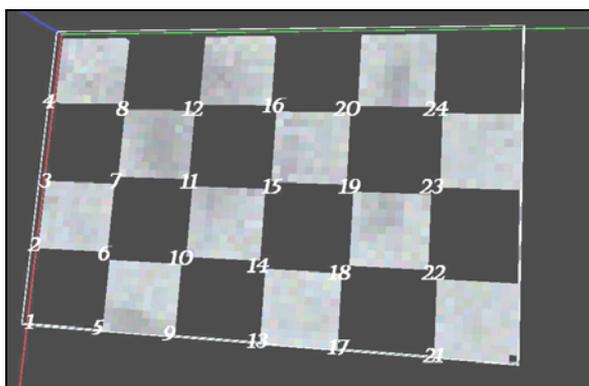


Gambar 9. Pengujian Terhadap Nilai *Threshold*

(semakin sedikit jumlah proses evaluasi, semakin baik metode tersebut). Voxel yang dihasilkan oleh GVC-LDI mempunyai jumlah yang lebih sedikit dibanding GVC-IB terutama untuk *threshold* yang besar, hal ini menunjukkan banyak voxel yang tidak diperlukan dapat tereliminasi walaupun *threshold* yang digunakan besar. Tetapi kelemahan GVC-LDI adalah pada waktu proses yang sangat lama serta jumlah memori yang dibutuhkan juga lebih besar dari memori yang dibutuhkan oleh GVC-IB.

Selain melakukan pengujian terhadap hasil rekonstruksi secara visual, dilakukan pula pengujian terhadap ketepatan posisi yang dihasilkan. Pada pengujian ini digunakan *pattern* sederhana untuk mengetahui sejauh mana selisih posisi antara obyek asli dengan obyek hasil rekonstruksi. Untuk itu dilakukan pengukuran pada beberapa titik, yang diambil pada titik-titik yang mudah dicari posisinya (dalam koordinat) dan dibandingkan antara benda asli terhadap hasil rekonstruksi. Gambar *pattern* yang digunakan untuk pengujian ini dapat dilihat pada Gambar 10.

Hasil perbandingan dapat dilihat pada Tabel 3 berikut.



Gambar 10. Obyek *Chessboard Pattern* Guna Pengujian Ketepatan

Tabel 3. Perbandingan Ketepatan Posisi

Titik	PosX Asli	PosY Asli	PosX Hasil	PosY Hasil	Titik	PosX Asli	PosY Asli	PosX Hasil	PosY Hasil
1	0	0	0	0	13	0	30	0	30
2	10	0	10	0	14	10	30	10	30
3	20	0	20	0	15	20	30	20	30
4	30	0	30	0	16	30	30	30	30
5	0	10	0	10	17	0	40	0	40
6	10	10	10	10	18	10	40	10	40
7	20	10	20	10	19	20	40	20	40
8	30	10	30	10	20	30	40	30	40
9	0	20	0	20	21	0	50	0	50
10	10	20	10	20	22	10	50	10	50
11	20	20	20	20	23	20	50	20	50
12	30	20	30	20	24	30	50	30	50

Dari hasil pengujian pada Tabel 3 terlihat bahwa posisi yang dihasilkan pada proses rekonstruksi sama dengan posisi pada gambar asli.

## KESIMPULAN

Berdasarkan percobaan yang telah dilakukan maka dapat disimpulkan beberapa hal, antara lain:

- Nilai *threshold* pada proses rekonstruksi sangat mempengaruhi obyek 3D hasil rekonstruksi. Nilai *threshold* yang terlalu kecil akan membuat adanya *voxel* dari obyek yang seharusnya *valid* namun ikut terpahat, sedangkan nilai *threshold* yang terlalu besar akan menyebabkan adanya *voxel* dari lingkungan (*background*) yang dianggap *valid* sehingga ikut ditampilkan. Perlu dilakukan proses rekonstruksi secara berulang-ulang untuk menentukan nilai *threshold* yang cocok (tepat) sehingga menghasilkan obyek rekonstruksi 3D yang paling baik. Nilai *threshold* yang baik berada antara 50 sampai 100.
- Dari pengujian perbandingan antara metode GVC-LDI dengan GVC-IB, didapat bahwa proses rekonstruksi dengan menggunakan metode GVC-LDI mempunyai algoritma iterasi yang lebih efisien dibanding GVC-IB sehingga proses evaluasi yang dilakukan lebih sedikit dan menghasilkan *voxel* yang lebih sedikit pula walaupun dengan menggunakan *threshold* besar.

Metode GVC-LDI mempunyai kekurangan dibanding metode GVC-IB yaitu membutuhkan memori yang sangat besar saat menjalankan proses serta membutuhkan waktu proses yang lebih lama.

## DAFTAR PUSTAKA

1. Izquierdo, E., and J. R. Ohm. *Image-based rendering and 3D modeling: A Complete Framework*. Signal Processing: Image Communication 15, 2000, pp. 817–858.
2. Slabaugh, Gregory G., *Novel Volumetric Scene Reconstruction Methods for New View Synthesis*. USA: Georgia Institute of Technology, 2002.
3. Seitz, Steven M., and Charles R. Dyer, *Photorealistic Scene Reconstruction by Voxel Coloring*. Proc. Computer Vision and Pattern Recognition Conf, 1997, pp.1067-1073.
4. Adipranata, Rudy, Resmana Lim, dan Anton Setiawan, *Rekonstruksi Obyek 3D dari Gambar 2D Dengan Metode Generalized Voxel Coloring*. Prosiding Seminar Nasional KOMMIT, Jakarta, 2006.
5. Zhang, Zhengyou, *A Flexible New Technique for Camera Calibration*. PDF, 2002.

6. Open CV Documentation by Intel Research. <<http://sourceforge.net/projects/opencv/>>, akses: 3 Maret 2005.
7. Wright, Richard S., and Benjamin Lipchak, *OpenGL Superbible (3rd ed)*. Indianapolis: Sams Publishing, 2005.