

# Skin Segmentation, Skull Segmentation, and Mesh Generation Tool of Medical Image

Eric Yogi Tjandra<sup>1</sup>, Liliana<sup>2</sup>

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

Email :eric.13292@gmail.com, lilian@peter.petra.ac.id

**ABSTRAK:**Perkembangan teknologi informasi dan komputer (TIK) telah merambah ke berbagai bidang ilmu yang ada di dunia saat ini, salah satunya bidang kedokteran. Saat ini telah ditemukan berbagai metode diagnose dan pengobatan yang dibutuhkan masyarakat berbasis TIK, seperti MRI dan CT Scanner. Baik MRI dan CT menghasilkan data berupa *volume image* yang berisi hasil *scanning* organ-organ internal, yang disebut juga sebagai *DICOM (Digital Imaging and Communications in Medicine) Image*. Untuk berbagai kebutuhan medis, pengolahan *DICOM image* ini sangat diperlukan.

Fokus dalam pembuatan skripsi ini adalah membuat sebuah fitur yang dapat membaca *volume image* dan dapat melakukan segmentasi kulit wajah dan tulang tengkorak manusia dalam waktu yang relatif singkat, serta melakukan *mesh generation* dari gambar DICOM yang telah diolah. Segmentasi kulit wajah dilakukan dengan proses *thresholding*, ekstraksi *largest connected component*, dan *holefilling* untuk mengisi lubang-lubang yang terbentuk. Sedangkan segmentasi tengkorak dilakukan dengan menghilangkan bagian tulang leher yang tersambung dengan tengkorak dan ekstraksi *largest connected component*. Setelah itu dilakukan proses *mesh generation* untuk menghasilkan objek 3D yang dapat diproses lebih lanjut. Proses *mesh generation* ini dilakukan dengan menggunakan algoritma *marching cubes*.

Hasil pengujian menunjukkan bahwa proses segmentasi kulit wajah dan segmentasi tulang tengkorak akan berjalan dengan baik bila tidak ada objek lain yang menempel pada wajah atau tengkorak. Proses segmentasi kulit wajah memakan waktu yang cukup signifikan, terutama diakibatkan oleh proses *holefilling*. Waktu yang dibutuhkan untuk proses *mesh generation* tergantung dari tingkat kompleksitas gambar. Kualitas hasil *mesh generation* dipengaruhi oleh *resolution reduction ratio*, *relaxation factor* dan *iteration of smoothing*.

Kata kunci:

Gambar Medis, Segmentasi, *Largest Connected Component*

**ABSTRACT :**The development of information and computer technology (ICT) has reached to various fields in the world, one of them is medical field. There are new methods of medication and diagnoses currently based on ICT, like MRI and CT Scanner. Both MRI and CT produces data as volume image which contains the scan results of internal organs, also known as DICOM (Digital Imaging and Communications in Medicine) Image. For various needs in the medical world, DICOM image processing is necessary.

This thesis aims to make a feature to load a volume image and do skin and skull segmentation in a short time, and also to do mesh generation from processed DICOM images. Skin segmentation is done by thresholding the image, extracting the largest connected component, and holefilling to fill the unnecessary holes. As for the skull segmentation, the process is done by removing the spines which is connected to the skull, and then extracting the largest connected component. Afterwards, mesh generation is done to produce the 3D objects from the processed images. This mesh generation process is done using the marching cubes algorithm.

The testing results show that the skin and skull segmentation process will work well when there are no other objects that are connected to the skin or the skull. Skin segmentation process takes a significant amount of time, primarily caused by the holefilling process. The time required for mesh regeneration depends on the complexities of the image. The mesh generation result's quality is affected by resolution reduction ratio, relaxation factor and iteration of smoothing.

Keyword:

Medical Image, Segmentation, Largest Connected Component

## 1. LATAR BELAKANG

Perkembangan teknologi informasi dan komputer (TIK) telah merambah ke berbagai bidang ilmu yang ada di dunia saat ini, salah satunya bidang kedokteran. Dengan menggunakan TIK, saat ini telah ditemukan berbagai metode diagnose dan pengobatan yang dibutuhkan masyarakat. Beberapa metode pemeriksaan yang menggunakan ilmu TIK adalah metode pemeriksaan organ internal yang dikenal dengan *CT Scan* dan *MRI*.

*CT* (atau *CAT, Computerized Axial Tomography*) adalah pemeriksaan organ internal dengan menggunakan radiasi sinar X, sedangkan *MRI (Magnetic Resonance Imaging)* memanfaatkan gelombang radio dan magnet untuk tujuan yang serupa. Baik MRI dan CT menghasilkan data berupa *volume image* yang berisi hasil *scanning* organ-organ internal, seperti tulang, otak, pembuluh darah, jaringan lapisan kulit, dan berbagai organ lainnya. *Volume image* tersebut biasanya disebut juga sebagai *DICOM (Digital Imaging dan Communications in Medicine) Image*. Untuk membaca file gambar tersebut, diperlukan aplikasi *DICOM viewer*, beberapa di antaranya yang dapat diunduh secara gratis adalah MicroDicom, RadiAnt Dicom Viewer, dan masih banyak lagi.

Untuk berbagai kebutuhan medis, pengolahan *DICOM image* ini sangat diperlukan. Sebagai contoh, untuk kebutuhan dokter bedah tulang yang harus membedah korban kecelakaan yang mengalami

patah tulang, maka data *volume image* perlu diolah agar dapat menampilkan bagian tulang saja dan dapat menghilangkan bagian-bagian lain yang tidak perlu untuk diperlihatkan. Hal ini tentunya akan mempermudah dokter bedah tersebut untuk lebih jelas melihat kondisi tulang sang korban.

Fokus dalam pembuatan skripsi ini adalah membuat sebuah fitur yang dapat membaca *volume image* dan dapat melakukan segmentasi kulit wajah dan tulang tengkorak manusia dalam waktu yang relatif singkat.

## 2. Landasan Teori

### 2.1 Thresholding

*Grey-level segmentation*, atau disebut juga *Thresholding* adalah proses konversi antara gambar *gray-level* dan *bilevel*. [4] *Thresholding* telah menjadi salah satu alat utama dalam melakukan segmentasi gambar dikarenakan sifatnya yang intuitif dan kemudahannya dalam implementasi. [2] Salah satu cara untuk mengekstrak objek dari latar belakangnya adalah dengan menentukan suatu *threshold T* yang dapat memisahkan keduanya. Setiap titik  $(x,y)$  yang sesuai dengan  $f(x,y) > T$  disebut sebagai *object point* (titik objek), sedangkan titik selain itu disebut *background point* (titik latar belakang).

Secara umum, sebuah gambar yang terkena *threshold* dapat didefinisikan sebagai:

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases} \quad (1)$$

Setiap pixel yang berlabel 1 dapat dikatakan sebagai objek, sedangkan pixel berlabel 0 dikatakan sebagai latar belakang.

### 2.2 Connected Component Extraction

Konektivitas antar pixel adalah konsep dasar di dalam konsep *digital image* [1]. Untuk menentukan apakah dua pixel terkoneksi, kedua pixel tersebut harus dicek apakah keduanya *neighbor* dan memiliki *gray level* yang memenuhi suatu kriteria kemiripan. Sebagai contoh, pada gambar biner dua pixel dikatakan terkoneksi jika keduanya bersebelahan dan memiliki nilai yang sama. Terdapat 2 jenis *adjacency* secara umum: *4-adjacency* dan *8-adjacency*. *4-adjacency* menerima pixel yang bersebelahan dengannya pada 4 arah: atas, bawah, kiri dan kanan; sedangkan *8-adjacency* menerima pixel yang bersebelahan dengannya pada 8 arah: kiri atas, atas, kanan atas, kanan, kanan bawah, bawah, kiri bawah, dan kiri.

Ekstraksi *Connected Components* berperan sangat penting dalam analisa gambar. Misalkan  $Y$  adalah *connected component* di dalam gambar. Adan asumsikan bahwa suatu titik  $p$  di dalam  $Y$ . Persamaan iteratif di bawah ini menyatakan semua elemen di dalam  $Y$ :

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \dots \quad (2)$$

dimana  $X_0 = p$ , dan  $B$  adalah *structuring element* yang diinginkan. Algoritma dijalankan hingga mencapai  $X_k = X_{k-1}$ , dan himpunan  $Y = X_k$ .

### 2.3 Marching Cubes

*Marching cubes* adalah algoritma yang dapat menkonstruksi *3D surface* dari sebuah *3D array* data. Model yang dihasilkan akan memiliki resolusi yang cukup tinggi. Algoritma ini diciptakan oleh Lorensen dan Cline (1987) dan berperan sangat penting dalam bidang *medical image* untuk visualisasi data MR dan CT secara efisien [3].

Algoritma *Marching Cubes* memanfaatkan pendekatan *divide and conquer* untuk mencari posisi *surface* dalam sebuah *logical cube* yang terbuat dari delapan pixel, masing-masing 4 pixel dari *slice* yang berdekatan. Algoritma ini akan mencari perpotongan antara *surface* dengan *cube*, kemudian berpindah ke *cube* yang berikutnya. Posisi perpotongan antara *surface* dengan *cube* inilah yang akan menentukan bentuk segitiga yang akan dibuat. Untuk mencari perpotongannya, apabila suatu vertex dalam *cube* memiliki nilai lebih besar atau sama dengan suatu *value*, maka vertex tersebut diberi nilai 1 dan dianggap berada di dalam *surface*. Sebaliknya akan diberi nilai 0 dan dianggap berada di luar *surface*. *Cube* akan berpotongan dengan *edge* dari *surface* dimana salah satu vertex dari *edge* tersebut berada di dalam *surface* dan vertex lainnya berada di luar.

Dengan demikian, akan ada  $2^8 = 256$  cara untuk terjadinya perpotongan. Dengan mempertimbangkan sifat-sifat simetris, komplemen dan rotasi, maka 256 cara tersebut dapat disederhanakan menjadi 14 kemungkinan saja (Gambar 2.5). Setelah itu, dibuatlah *table lookup* untuk keempatbelas kemungkinan tersebut.

Langkah selanjutnya adalah menghitung normal unit dari setiap vertex segitiga. Algoritma untuk *rendering* menggunakan normal ini untuk menghasilkan *Goraud-shaded image*. Arah dari vector *gradient normal*  $\vec{g}$  terhadap *surface* dinyatakan sebagai berikut:

$$\vec{g}(x,y,z) = \nabla f(x,y,z) \quad (2.3)$$

Untuk memperkirakan vector *gradient* pada *surface* yang diinginkan, pertama hitung vektor *gradient* pada vertex *cube* dan interpolasi secara linear *gradient* tersebut pada titik perpotongan. *Gradient* pada vertex *cube*  $(i,j,k)$  dihitung dengan

$$G_x(i,j,k) = \frac{D(i+1,j,k) - D(i-1,j,k)}{\Delta x} \quad (4)$$

$$G_y(i,j,k) = \frac{D(i,j+1,k) - D(i,j-1,k)}{\Delta y} \quad (5)$$

$$G_z(i,j,k) = \frac{D(i,j,k+1) - D(i,j,k-1)}{\Delta z} \quad (6)$$

dimana  $D(i,j,k)$  adalah kerapatan pixel  $(i,j)$  pada *slice* ke  $k$  dan  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$  adalah panjang dari sisi *cube*.

### 2.4 Laplacian Smoothing

Tujuan dari *surface smoothing* adalah menghilangkan noise dari mesh *surface* dengan kerusakan yang seminimal mungkin. [5] Dari sebuah mesh *surface*  $S$  yang memiliki noise, proses *smoothing* akan menghasilkan mesh *surface*  $S'$  yang halus dengan kualitas yang bagus dan semirip mungkin dengan aslinya tanpa menghilangkan fitur-fitur penting dari *surface* tersebut.

Untuk sebuah *surface* parametric  $S: F = F(u,v)$ ,  $(u,v) \in \Omega$ , permasalahan dari *global surface smoothing* dapat dijabarkan sebagai berikut:

$$\text{Min}_s(S) \int_{\Omega} E(S) \quad (7)$$

dimana energi yang terminimalisasi harus mengandung dua unsur: satu untuk mengukur kualitas *surface* dan satu untuk mengukur tingkat ketepatan data yang dibandingkan dengan aslinya:

$$E(S') = \alpha \int_{\Omega} \Psi(S') du dv + \beta \int_{\Omega} (S' - S)^2 du dv \quad (8)$$

dimana  $\alpha$  dan  $\beta$  adalah berat dari kedua unsur tersebut. Model minimalisasi untuk *surface smoothing* di atas merupakan generalisasi dari total variational model dari Rudin-Osher-Fatemi yang terkenal dalam bidang literature image denoising.

Smoothing energy yang dipakai untuk smoothing bisa merupakan membrane energy:

$$\int_{\Omega} \Psi(S') dudv = \int_{\Omega} (Fu^2 + Fv^2) dudv \quad (9)$$

atau thin-plate energy:

$$\int_{\Omega} \Psi(S) dudv = \int_{\Omega} (Fuu^2 + 2Fuv^2 + Fvv^2) dudv \quad (10)$$

Bila  $S = (V, E, F)$  adalah mesh segitiga, operator Laplacian akan dapat diperkirakan secara linier pada setiap vertex sebagai berikut:

$$L(v_i) = \sum_{j \in i^*} (v_j - v_i), \quad (11)$$

dimana  $i^*$  adalah himpunan indeks vertex dari neighbor dari vertex  $v_i$ , dan  $w_{ij}$  adalah weight dari garis  $(i, j)$  yang berkorespondensi pada  $v_i$  dengan  $\sum_{j \in i^*} w_{ij} = 1$ .

Algoritma Laplacian ini cukup sederhana. Ide dasarnya adalah posisi dari vertex  $v_i$  diganti dengan hasil rata-rata dari posisi setiap vertex yang bersebelahan dengannya. Setiap vertex secara bertahap digeser sesuai dengan arah Laplacian. Pada beberapa kasus dengan mesh yang memiliki hubungan antar vertex yang tidak wajar, hal ini akan menyebabkan berbagai masalah seperti distorsi geometris, konvergensi yang lambat pada mesh berukuran besar, ketidakstabilan numerik dan lain-lain.

### 3. DESAIN SISTEM

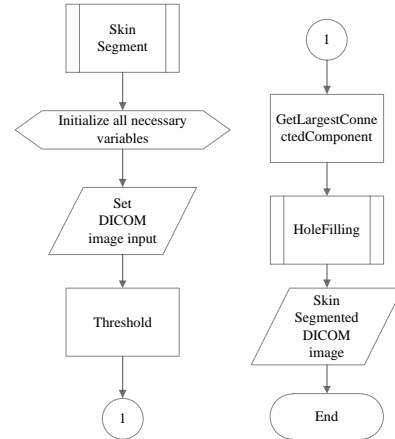
#### 3.1 Skin Segmentation

Permasalahan dalam segmentasi kulit wajah adalah bagaimana menyeleksi bagian kulit di dalam gambar, menghilangkan bagian yang tidak diperlukan dan menutup bagian-bagian rongga pada wajah (Misal, lubang hidung, rongga tenggorokan, dan sebagainya). Hasil yang diharapkan dari proses ini adalah dapat menampilkan kulit wajah tanpa ada bagian-bagian lain yang tidak perlu.

Secara umum, untuk mendapatkan bagian kulit cukup dengan *threshold* gambar dengan batas bawah yang rendah. Hal ini dikarenakan di dalam gambar, bagian kulit memiliki intensitas yang cukup rendah. Namun, kekurangan dalam penyelesaian ini adalah cara ini tidak menghapus bagian rongga.

Langkah-langkah pada *Skin segmentation* terdiri dari *thresholding*, *largest connected component*, dan *holefilling*. Semua pixel yang berintensitas di bawah batas tertentu terkena *threshold* menjadi berintensitas tinggi. Dengan asumsi bahwa setelah di-*threshold* bagian wajah adalah bagian yang terbesar di dalam gambar, algoritma *largest connected component* akan menghilangkan *noise* atau bagian-bagian lain yang bukan termasuk wajah. Kemudian untuk menghilangkan rongga-rongga tubuh di dalam gambar, dilakukan proses *holefilling*. Proses *holefilling* dalam pembuatan aplikasi ini menggunakan penerapan dari algoritma *floodfill* pada umumnya. Namun penerapan *floodfill* ini digunakan untuk menandai bagian-bagian yang tidak termasuk objek atau dapat dikatakan "di luar (*outside*)".

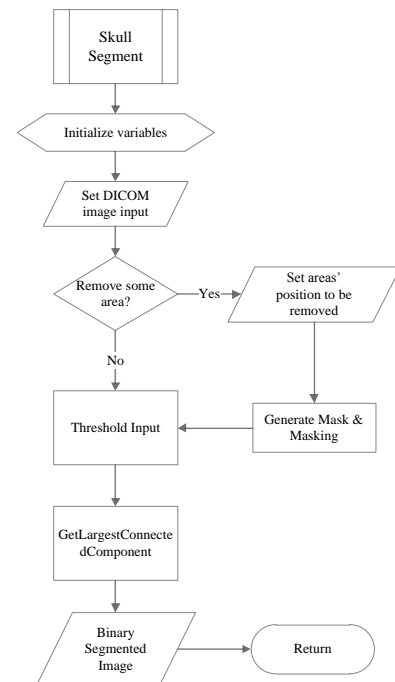
Pada setiap  $z$ , dilakukan proses *markOutside* untuk melakukan penandaan apakah suatu pixel termasuk di luar atau di dalam objek. Penandaan ini disimpan di dalam suatu *array*. Setelah itu, dimulai proses pengecekan. Bila suatu pixel ditandai sebagai *inside* dan masih berwarna gelap, maka pixel tersebut akan diubah nilainya menjadi tinggi. Diagram alir *skin segmentation* dapat dilihat di Gambar 1.



Gambar 1. Diagram alir *skin segmentation*

#### 3.2 Skull Segmentation

Permasalahan dalam segmentasi tengkorak adalah bagaimana menyeleksi bagian tengkorak dan menghapus bagian lain yang tidak diperlukan. Karakteristik utama dari tengkorak yang dapat dilihat pada gambar DICOM adalah tingkat intensitasnya yang cukup tinggi, sehingga metode *thresholding* dapat menyelesaikan sebagian besar masalah di dalam proses ini.



Gambar 2. Diagram alir *Skull Segmentation*

*Skull segmentation* dilakukan dengan *thresholding* dan *largest connected component*. *Thresholding* akan menyeleksi hanya bagian-bagian yang memiliki intensitas tinggi. Hal ini dilakukan sebab bagian tengkorak memiliki intensitas tinggi. Namun bagian-bagian lain yang juga memiliki intensitas tinggi (misal: pembuluh darah, dan lain-lain) juga akan ikut terseleksi.

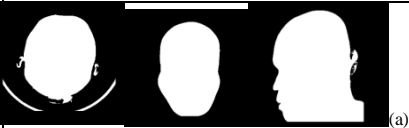
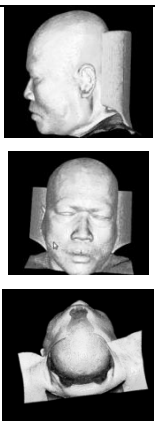
Pada dasarnya, tulang tengkorak selalu terhubung dengan tulang belakang, sementara hasil yang diminta hanya tulang tengkorak saja. Karena itu sebelum melakukan algoritma *largest connected*

component, adalah user perlu melakukan pemisahan antara tengkorak dengan tulang belakang. Hal ini dapat dilakukan cukup dengan memilih area pada slice tertentu yang ingin dihapus. Setelah itu, largest connected component secara otomatis akan menghilangkan bagian tulang belakang. Diagram alir skull segmentation dapat dilihat di Gambar 2.

#### 4. PENGUJIAN

Pengujian dilakukan pada performa dari penerapan algoritma pada aplikasi. Pengujian aplikasi dibagi menjadi pengujian secara kualitas dan pengujian secara kuantitas. Pengujian secara kualitas adalah pengujian kualitas dari hasil proses algoritma Skin Segmentation dan Skull Segmentation. Pengujian secara kuantitas pengujian running time untuk Skin Segmentation, Skull Segmentation, dan Mesh Generation.

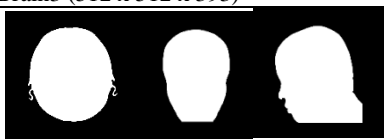

**Tabel 1 Hasil Skin Segmentation pada file Brain2**

1	Nama File:	Brain2 (512 x 512 x 381)
2	Hasil Skin Segmentation: (pada slice tengah-tengah)	 (a) Transversal (b) Coronal (c) Sagittal
3	Hasil Mesh Generation:	
4	Keterangan	Tidak ada lubang, terdapat objek: bantalan


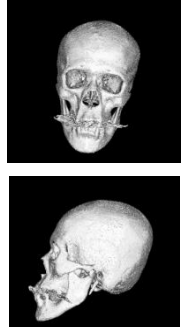
Pada Tabel 1 dapat dilihat hasil dari skin segmentation berhasil untuk menghilangkan lubang-lubang dari kepala. Lubang hidung, tenggorokan dan lainnya tertutup. Selain itu juga tidak terdapat noise. Namun masih terdapat bantalan di belakang kepala yang ikut tersegmentasi. Hal ini terjadi dikarenakan pada gambar aslinya, bantalan tersebut tersambung dengan kulit wajah. Oleh karena itu, algoritma yang digunakan tidak dapat menghapus objek-objek tersebut.

Pada Tabel 2 juga menunjukkan hasil skin segmentation dengan hasil yang lebih baik. Tidak ada objek asing yang ikut tersegmentasi pada gambar.


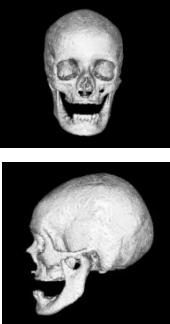
**Tabel 2 Hasil Skin Segmentation pada file Brain3**

1	Nama File:	Brain3 (512 x 512 x 393)
2	Hasil Skin Segmentation: (pada slice tengah-tengah)	 (a) Transversal (b) Coronal (c) Sagittal
3	Hasil Mesh Generation:	
4	Keterangan	Tidak ada lubang

**Tabel 3 Tabel Pengujian Hasil Skull Segmentation pada File Brain2**

1	Nama File:	Brain2 (512 x 512 x 381)
2	Hasil Skull Segmentation: (pada slice tengah-tengah)	 (a) Transversal (b) Coronal (c) Sagittal
3	Hasil Mesh Generation:	

**Tabel 4 Tabel Pengujian Hasil *Skull Segmentation* pada *File Head***

1	Nama File:	Head (512 x 512 x 234)
2	Hasil <i>Skull Segmentation</i> : (pada <i>slice</i> tengah-tengah)	 (a) Transversal (b) Coronal (c) Sagittal
3	Hasil Mesh Generation:	

Tabel 3 dan 4 menunjukkan hasil *skull segmentation*. Kualitas gambar yang dihasilkan cukup bagus, seluruh tengkorak dapat tersegmentasi dan tulang belakang dapat dihapus dengan baik. Namun pada tabel 3, ada bagian artifak pada bagian mulut yang mengalami sedikit kerusakan. Kerusakan tersebut merupakan bagian dari gambar asli yang tidak dapat diatasi oleh algoritma yang digunakan. Sedangkan pada tabel 4 dapat dilihat bahwa bentuk tengkorak dapat didapatkan dengan sempurna.

**Tabel 5 Pengujian Waktu *Running Time Skin Segmentation***

No	Nama File	Skin Segmentation				Total Time (secs)
		Threshold Value	LCC Time (secs)	Hole Filling Z (secs)	Hole Filling X (secs)	
1	Brain2	-184	36.67	34.25	44.85	<b>115.77</b>
2	Brain3	-199	34.1	32.48	43.02	<b>109.6</b>
3	Brain4	-355	29.91	36.04	47.21	<b>113.16</b>
4	Brain5	-128	33.82	29.68	38.93	<b>102.43</b>
5	FCL88E	-255	66.39	61.2	79.99	<b>207.58</b>
6	Head	-184	28.55	28.55	37.31	<b>94.41</b>

Pada tabel 5 dapat dilihat bahwa *Running time* dari algoritma untuk *skin segmentation* berbanding lurus dengan ukuran gambar. Semakin besar ukuran gambar yang diproses, semakin besar waktu yang dibutuhkan untuk algoritma. Rata-rata waktu yang digunakan oleh algoritma ini untuk ukuran gambar yang digunakan pada sampel-sampel ini adalah 123.85 detik, atau kurang lebih 2 menit. Waktu yang didapatkan ini sampai saat ini masih dapat diterima, mengingat belum ada algoritma lain yang dapat melakukan proses ini dengan waktu yang lebih cepat dan hasil yang cukup baik. Tetapi pada praktiknya, diperlukan waktu yang jauh lebih cepat agar sistem benar-benar dapat digunakan.

Salah satu faktor yang menyebabkan lambatnya *running time* dari algoritma ini adalah pada proses *holefilling*. Pada gambar, setiap

*pixel*akan diakses setidaknya dua kali. Sekali untuk pengecekan dan sekali lagi untuk mengubah nilai *pixel*.

**Tabel 6 Pengujian Waktu *Running Time Skull Segmentation***

No	Nama File	Skull Segmentation		
		Area Removed	Threshold Value	Time (secs)
1	Brain2	X: 83-144 Y: 73-106 Z: 159	105	<b>26.02</b>
2	Brain3	X: 84-138 Y: 76-107 Z: 150	126	<b>29.95</b>
3	Brain4	X: 92-148 Y: 92-119 Z: 180	274	<b>23.4</b>
4	Brain5	X: 70-152 Y: 66-125 Z: 160	128	<b>24.1</b>
5	FCL88E	X: 144-231 Y: 182-236 Z: 156	232	<b>46.31</b>
6	Head	X: 213-285 Y: 225-289 Z: 151	128	<b>13.93</b>

Dapat dilihat dari Tabel 6 bahwa rata-rata *running time* dari *skull segmentation* adalah sekitar 25 detik. Waktu yang diperoleh ini tergolong cukup cepat dan dapat diterima, dengan kualitas hasil yang terbilang cukup memuaskan.


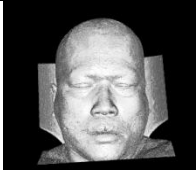



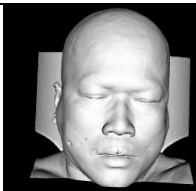

**Tabel 7 Pengujian Waktu *Running Time Mesh Generation***

No	Nama File	Skin Segmentation		Skull Segmentation	
		Mesh generating (secs)	Smoothing and Decimation (secs)	Mesh Generating (secs)	Smoothing and Decimation (secs)
1	Brain2	21.48	58.21	31.39	85.98
2	Brain3	17.54	46.43	33.11	91.64
3	Brain4	15.45	40.47	37.4	97.77
4	Brain5	15.82	44.04	32.3	91.3
5	FCL88E	24.22	57.59	25.94	58.34
6	Head	5.69	13.2	7.85	18.3

Dapat dilihat pada Tabel 7 bahwa *running time mesh generation* untuk data yang telah diproses *skin segmentation* lebih cepat daripada data yang telah diproses *skull segmentation*. Perbedaan kedua data tersebut adalah tingkat kerumitan dari masing-masing data. Data dari *skin segmentation* relatif lebih sederhana daripada data dari *skull segmentation*, sehingga jumlah vertex yang terbentuk pada data dari *skin segmentation* lebih sedikit daripada jumlah jumlah vertex yang terbentuk pada data dari *skull segmentation*. Oleh karena itulah *running time mesh*

generation untuk data yang telah diproses *skin segmentation* lebih cepat daripada data yang telah diproses *skull segmentation*.

**Tabel 8** Tabel Perbandingan Hasil *Mesh Generation*

No	Gambar Hasil	Keterangan parameter
1		Ratio : 0.10 Relaxation Factor : 0.10 Iteration : 1
2		Ratio : 0.30 Relaxation Factor : 0.10 Iteration : 1
3		Ratio : 0.70 Relaxation Factor : 0.10 Iteration : 1
4		Ratio : 0.30 Relaxation Factor : 0.40 Iteration : 1
5		Ratio : 0.30 Relaxation Factor : 0.70 Iteration : 1
6		Ratio : 0.30 Relaxation Factor : 0.70 Iteration : 3
7		Ratio : 0.30 Relaxation Factor : 0.70 Iteration : 5

Dari Tabel 8, dapat dilihat perbandingan kualitas hasil *mesh generation* dengan berbagai parameter. Dari percobaan 1, 2 dan 3 dapat dilihat bahwa semakin besar *resolution reduction ratio*, *mesh* yang dihasilkan semakin halus. Hal ini dikarenakan semakin besar *ratio*, ukuran *mesh* yang dihasilkan akan semakin besar dan jumlah vertex akan semakin banyak. Dari percobaan 2, 4 dan 5 dapat pula dilihat pengaruh *relaxation factor* pada proses *smoothing*. Semakin besar angka *relaxation factor*, semakin halus pula kualitas *mesh* yang dihasilkan. Dari percobaan 5, 6, dan 7 juga dapat dilihat pengaruh jumlah iterasi pada proses *smoothing*. Semakin banyak iterasi yang dilakukan, semakin baik pula kualitas *mesh*.

## 5. Kesimpulan

Berdasarkan hasil pengujian dapat disimpulkan beberapa hal sebagai berikut:

- Rangkaian algoritma yang digunakan untuk *skin segmentation* (*Largest Connected Component* dan *Hole Filling*) berhasil dengan baik bila tidak ada objek lain yang menempel langsung pada kulit wajah dalam gambar.
- Proses *skin segmentation* memakan waktu yang cukup signifikan, terutama disebabkan oleh proses *Hole Filling*.
- Rangkaian algoritma yang digunakan untuk *skull segmentation* berhasil dengan baik bila tidak ada objek lain yang menempel langsung pada tulang tengkorak dalam gambar.
- Semakin kompleks objek, proses *mesh generation*-nya akan memakan waktu semakin lama.
- Kualitas hasil *mesh generation* dipengaruhi oleh *resolution reduction ratio*, *relaxation factor* dan *iteration of smoothing*.

## 6. REFERENSI

- [1] Gonzales, R.C., Woods, R.E. (2004). *Digital Image Processing* (2nd ed.). Upper Saddle River: Prentice-Hall.
- [2] Hussain, Z. (1991). *Digital Image Processing Practical Applications of Parallel Processing Techniques*. Singapore: Ellis Herwood Limited.
- [3] Lorensen, W.E., Cline, H.E. (1987). *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*. In *Computer Graphics Vol. 21, 4*, 163-169.
- [4] Parker, J.R., (1997). *Algorithms for Image Processing and Computer Vision*. Wiley Computer Publishing.
- [5] Zhongping, J., Ligang, L., Guojin, W. (2005). *A Global Laplacian Smoothing Approach with Feature Preservation*. Ninth International Conference on Computer Aided Design and Computer Graphics 2005.