

SEGMENTASI 3D BERBASIS MODEL FASET MENGGUNAKAN METODE *REGION GROWING* TERMODIFIKASI

Priadhana Edi Kresnha
priadhana.edi@informatika.ftumj.ac.id
Universitas Muhammadiyah Jakarta

Gandjar Kiswanto
Teknik Mesin
Universitas Indonesia

Rahmat Widyanto
Ilmu Komputer
University Indonesia

ABSTRAK

Segmentasi adalah pembagian area sebuah gambar menjadi beberapa bagian berdasarkan kemiripan fitur, seperti warna, tekstur, intensitas piksel, dan lain sebagainya. Pada penelitian ini Segmentasi diimplementasikan pada sebuah model faset 3D menggunakan Metode Region Growing yang diadopsi dari penerapan metode pada gambar 2D. Segmentasi dilakukan berdasarkan kemiripan fitur yang terkandung dalam model faset 3D tersebut. Fitur 3D diidentifikasi dan diekstrak menggunakan beberapa metode, yaitu Angle Deficit / Guass-Bonnet Scheme dan Spherical Image. Ekstraksi Fitur dilakukan pada setiap verteks dari model faset. Verteks yang menyusun model faset 3D menjadi basis implementasi metode segmentasi menggantikan piksel pada gambar 2D. Hasil penelitian menunjukkan segmentasi Region Growing pada model faset memiliki tingkat akurasi cukup tinggi, merentang antara 70% hingga 100%, bergantung dari kehalusan (smoothness) dan kompleksitas model faset 3D.

Kata Kunci : *Model Faset 3D, Segmentasi, Region Growing*

I. Pendahuluan

Segmentasi merupakan salah satu metode yang paling banyak dikembangkan di dunia ilmu computer terutama bidang *computer vision* dan *image processing*. Fungsinya adalah untuk membagi sebuah citra menjadi beberapa bagian sehingga bagian yang ingin dianalisa (*region of interest*) bisa diekstrak dengan sempurna. Segmentasi banyak dilakukan untuk analisa berbagai jenis citra [7], seperti citra kedokteran (PET, MRI, CT-Scan), citra GIS / *remote sensing* (Radar, Landsat, IKONOS, QUICKBIRD), dan citra untuk keperluan penglihatan digital, seperti *computer vision* dan *visual navigation*.

Cara kerja segmentasi adalah dengan mengelompokkan piksel-piksel yang memiliki kemiripan dan berada pada region yang berdekatan. Penentuan kemiripan piksel didasarkan terutama oleh intensitasnya. Selain intensitas [1], kemiripan piksel bisa dilihat dari teksturnya maupun perubahan kontur pada citra. Masing-masing cara penentuan kemiripan piksel memiliki karakteristik sendiri-sendiri,

sehingga pendekatan segmentasinya pun berbeda.

Pada pengelompokan piksel berdasarkan intensitasnya, terdapat beberapa metode yang bisa digunakan [2,7,14], antara lain *thresholding*, *boundary detection*, *region-based*, dan *hybrid*.

Metode *thresholding* bekerja dengan mengelompokkan semua piksel berdasarkan rentang nilainya tanpa memandang lokasinya [1]. Nilai yang bisa dijadikan bahan pengukuran kemiripan dalam metode ini adalah intensitas piksel, grey level, atau RGB. Metode *thresholding* bisa menyebabkan bagian-bagian yang masuk dalam satu segmen lokasinya terpisah-pisah. Contoh penggunaan metode *thresholding* antara lain *k-means clustering* dan *SOM*.

Metode *boundary detection* bekerja dengan mengumpulkan garis tepi yang muncul pada suatu citra [14]. Segmen adalah wilayah yang dilingkupi oleh garis tepi. Contoh metode *boundary detection* antara lain *sobell*, *canny*.

Metode *region based* bekerja dengan mengumpulkan piksel-piksel yang memiliki

kemiripan nilai dan saling berdekatan (tidak dipisahkan dengan piksel lain dengan nilai berbeda) dimasukkan ke dalam suatu segmen [1]. Salah satu metode berbasis region adalah *region growing* yang menjadi pusat pembahasan pada penelitian ini.

Metode hybrid merupakan kombinasi dari metode *boundary detection* dan *region based* [1]. Tujuan metode *hybrid* adalah menggabungkan keunggulan dari kedua metode dan menghasilkan tingkat akurasi yang lebih tinggi. Kedua metode dikombinasikan melalui polling piksel, yaitu penentuan piksel dikelompokkan dalam suatu segmen didasarkan pada kemunculan terbanyak dari kedua metode tersebut.

Pada penelitian ini, metode segmentasi pada citra 2D disadur untuk diterapkan pada objek 3D. Berbeda dengan citra 2D, segmentasi objek 3D tidak didasarkan atas piksel, namun didasarkan atas verteks yang menyusun permukaan objek 3D tersebut. Pertimbangan pengelompokkan pada objek 3D pun bukan intensitas verteks atau nilai komponen z pada verteks tersebut, melainkan fitur yang diekstrak dari verteks penyusun objek 3D. Ekstraksi fitur verteks penyusun model faset sudah dilakukan pada penelitian [11]. Informasi fitur dari verteks penyusun model faset adalah hasil perhitungan kurvatur pada setiap verteks. Perhitungan kurvatur dilakukan dengan 2 metode, yaitu *gauss-bonnet scheme / angle deficit*, dan *spherical image*. Hasil dari kedua metode perhitungan kurvatur ini dikombinasikan menggunakan *if-then rules* dengan bilangan *threshold* untuk menentukan bentuk permukaan bidang pada sebuah verteks.

Model faset yang digunakan pada penelitian ini berbentuk *sculptured surface triangular meshes*. *Sculptured surface* merupakan bentuk 3D terbuka yang continue dan tidak membentuk siku pada permukaannya. *Triangular meshes* adalah objek 3D yang tersusun atas satuan geometri 2D sederhana berbentuk segitiga yang saling menyambung satu sama lainnya. Contoh dari *sculptured surface* yang juga *triangular meshes* yang digunakan pada penelitian ini bisa dilihat pada Gambar 11. Model faset ini disimpan dalam file berformat *stl* (*stereolithography*).

Makalah ini terbagi menjadi beberapa bab. Bab 2 berisi alur kerja sistem secara umum. Kemudian bab 3 memuat informasi mengenai kurvatur dan beberapa cara untuk menghitung kurvatur pada bidang 3D. Selanjutnya bab 4

membahas mengenai model faset 3D beserta media penyimpanannya berupa file *stl*. Metode *region growing* dijelaskan secara garis besar pada bab 5. Proses detail implementasi metode *region growing* dalam sistem dibahas pada bab 6. Hasil segmentasi beserta analisa dibahas pada bab 7. Pada bab 7 ini pula memuat beberapa gambar hasil segmentasi menggunakan metode *region growing*, dan keterkaitan presisi bilangan dengan hasil segmentasi. Terakhir bab 8 berisi kesimpulan dan saran, serta pengembangan penelitian lebih lanjut.

II. Alur Kerja Sistem

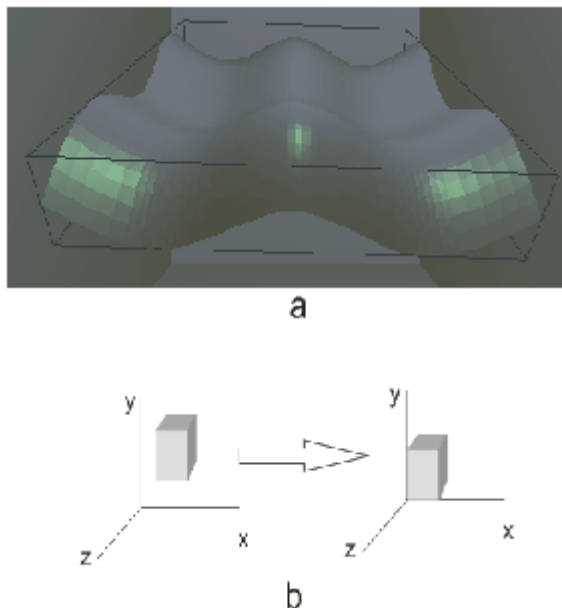
Alur kerja sistem secara garis dibagi ke dalam 4 subproses. Keempat subproses tersebut adalah pembacaan file *stl* dan pembentukan struktur data, penghitungan kurvatur setiap verteks penyusun permukaan model faset, penentuan bentuk permukaan pada setiap verteks berdasarkan estimasi kurvatur, pengelompokan / segmentasi bidang menggunakan metode *region growing*. Gambar 1 menjelaskan alur kerja system secara umum.

Struktur data yang digunakan dalam system adalah struktur data Lawson. Struktur data ini dikenal paling efisien dan kebutuhan memory yang digunakan tidak besar[9]. Dalam system ini terdapat 2 array utama penyusun struktur data, yaitu array verteks dan array segitiga. Array verteks berisi informasi posisi x-y-z verteks beserta beberapa atribut tambahan, seperti vector normal verteks, nilai kurvatur dan bentuk permukaan verteks. Sedangkan array segitiga berisi verteks-verteks penyusun segitiga dan vector normal segitiga tersebut.

Posisi model faset yang dibaca dari file *stl* biasanya tidak terletak pada posisi pusat (0,0,0). Posisi tersebut merupakan posisi terakhir ketika disimpan dari *software CAD (Computer Aided Design)*. Untuk mempermudah pencarian lokasi pada koordinat kartesius, maka posisi model faset disesuaikan dengan titik pusat (0,0,0). Caranya adalah dengan membentuk *bounding box*, dan memindahkan posisi *bounding box* dimana salah satu titik sudut *bounding box* berada pada titik pusat. Metode ini disebut normalisasi posisi. Ilustrasi normalisasi bisa dilihat pada Gambar 2.



Gambar 1. Alur kerja sistem

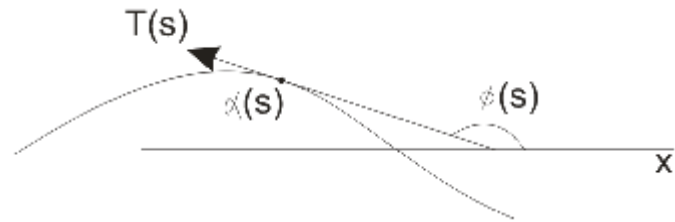


Gambar 2. Normalisasi posisi verteks, a) bounding box, b) memindahkan bounding box ke titik pusat (0,0,0)

III. Estimasi Kurvatur

Kurvatur adalah cara untuk mengetahui kelengkungan suatu kurva melalui metode kuantitatif. Secara logis, garis lurus memiliki nilai kurvatur nol, sedangkan lingkaran besar memiliki nilai kurvatur kecil. Semakin kecil lingkaran, nilai kurvatur semakin besar sebab kelengkungannya semakin tajam.

Parameter dari kurvatur adalah panjang garis kurva dan kecepatan perpindahan arah vector tangensial (*direction*). Vektor tangensial adalah vektor yang bersinggungan dengan garis kurva, dan sudut yang dibentuk antara x-axis dengan arah vector tangensial disebut sudut tangensial.



Gambar 3. Ilustrasi sebuah kurva

Gambar 3 mengilustrasikan sebuah kurva dengan beberapa komponen yang terlibat dalam perhitungan kurvatur. $T(s)$ adalah vector tangensial yang bersinggungan dengan kurva pada titik s , $\alpha(s)$ adalah nilai fungsi kurva pada titik s , dan $\Phi(s)$ adalah sudut yang dibentuk vector tangensial $T(s)$ terhadap sumbu x .

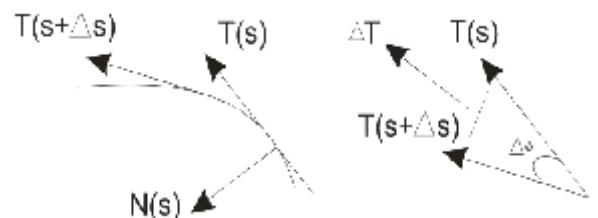
Kurvatur k dari $\alpha(s)$ adalah laju perubahan sudut tangensial pada titik s terhadap panjang kurva hitung [3]. Jika diubah dalam bentuk persamaan, nilai kurvatur k adalah sebagai berikut,

$$k = \frac{d\Phi}{ds} \tag{1}$$

Kurvatur juga bisa dihitung dari perubahan vector tangensial pada titik s terhadap panjang kurva hitung [3], atau jika diubah dalam bentuk persamaan menjadi sebagai berikut,

$$k = \frac{dT}{ds} \tag{2}$$

Ilustrasi perhitungan kurvatur dapat dilihat pada Gambar 4.



Gambar 4. Perhitungan kurvatur

Untuk mempermudah perhitungan kurvatur 2D, parameter perhitungan bisa direduksi hanya jari-jari kurva pada posisi s . Berikut adalah derivasi dari perhitungan kurvatur [3].

$$\alpha(s) = r \left(\cos \frac{s}{r}, \sin \frac{s}{r} \right)$$

$$T(s) = \alpha'(s) = r \left(-\sin \frac{s}{r}, \cos \frac{s}{r} \right)$$

$$N = \left(-\cos \frac{s}{r}, -\sin \frac{s}{r} \right)$$

$$T'(s) = \alpha''(s) = \frac{1}{r} \left(-\cos \frac{s}{r}, -\sin \frac{s}{r} \right) = \frac{1}{r} N$$

(3)

Dimana $\alpha(s)$ adalah nilai kurva pada posisi s , vector tangensial $T(s)$ adalah vektor singgung kurva pada posisi s , merupakan turunan dari $\alpha(s)$. Dan $T'(s)$ adalah perubahan T pada jarak yang sangat kecil, atau disebut kurvatur. Dari persamaan (3), bisa didapat bahwa kurvatur berbanding terbalik dengan jari-jari sebagaimana yang ditampilkan pada persamaan (4).

$$k = \frac{1}{r}$$

(4)

Persamaan (1,2,4) berlaku untuk penghitungan kurvatur pada bidang 2D. Untuk bidang 3D perhitungan kurvatur di suatu titik s harus dilakukan terhadap semua arah kurva yang melalui titik s . Jumlah kurva ini tidak berhingga, untuk itu, diperlukan pendekatan untuk menghitung kurvatur di ruang 3D. Metode pendekatan yang diambil pada penelitian ini adalah *gauss-bonnet scheme / angle deficit* dan *spherical image*. Kombinasi dari hasil estimasi kedua metode kurvatur dijadikan sebagai acuan dalam menentukan bentuk permukaan bidang, dimana bentuk permukaan ini merupakan karakteristik yang dimiliki verteks untuk proses segmentasi.



Gambar 5. Perhitungan kurvatur pada titik s melibatkan semua kurva pada bidang 3D yang melalui titik s

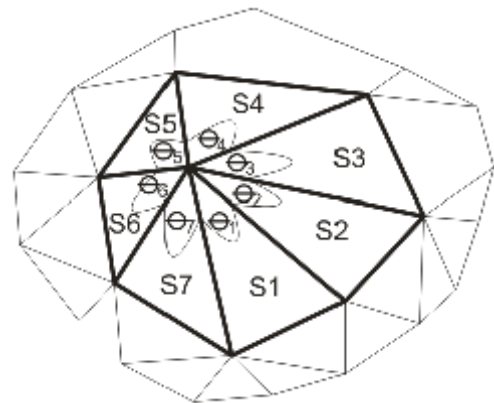
A. Gauss-Bonnet Scheme

Metode gauss-bonnet memanfaatkan selisih dari keliling lingkaran utuh (2Π) dengan total sudut yang mengelilingi sebuah verteks. Oleh karena itu metode ini sering disebut sebagai *angle deficit*. Berikut adalah persamaan dari *gauss-bonnet scheme* [16],

$$K = \frac{2\pi - \sum \Theta_{i,i+1}}{\frac{1}{3} \sum S_{i,i+1}}$$

(5)

Dimana K adalah global curvature, Θ adalah sudut yang dibentuk segitiga yang mengelilingi titik perhitungan kurvatur, dan S adalah luas segitiga yang mengelilingi titik perhitungan kurvatur.



Gambar 6. Parameter Gauss-Bonnet berupa sudut dan luas segitiga

Kecenderungan nilai kurvatur yang dihasilkan oleh metode ini adalah positif jika bentuk permukaan cembung atau cekung, nol jika bentuk permukaan datar, dan negative jika permukaan membentuk sadel (*saddle point*). Berikut adalah tabel estimasi kurvatur beserta kecenderungan bentuknya.

Tabel 1. Nilai kurvatur Gauss-Bonnet dan bentuk permukaan bidang

Nilai Kurvatur	Bentuk permukaan
$Abs(k) < t$	Datar
$k < -t$	Sadel
$k > t$	Cembung / cekung

B. Spherical Image

Metode *Spherical Image* bekerja dengan memproyeksikan segitiga-segitiga yang mengitari verteks focus kurvatur melalui vector normal dari setiap verteks penyusun segitiga [15]. Total luas segitiga proyeksi akan

dibandingkan dengan total luas segitiga asli penyusun permukaan model faset. Berikut adalah persamaan kurvatur berdasarkan metode *Spherical Image*,

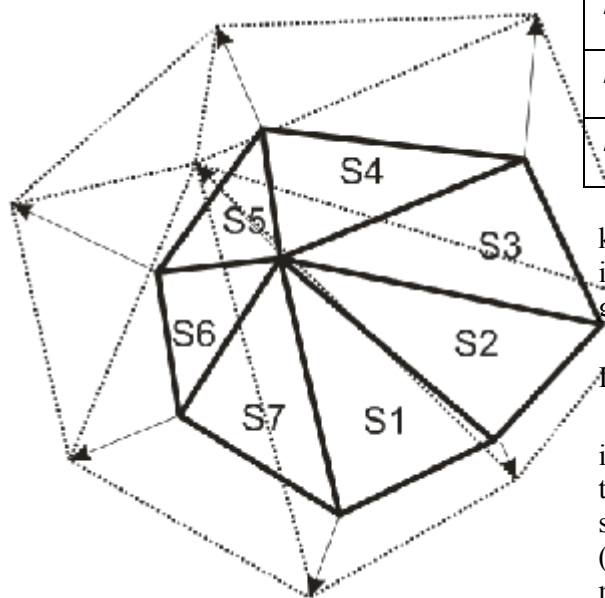
$$K = \frac{\sum L_{n_i n_o n_{i+1}}}{\sum L_{P_i P_o P_{i+1}}} \tag{6}$$

Dimana K adalah kurvatur L_n adalah luas bidang proyeksi dan L_p adalah luas bidang asli.

Adapun vektor normal setiap verteks didapat melalui rata-rata vector normal segitiga dengan mempertimbangkan luas segitiga yang berkontribusi terhadap verteks tersebut. Pencarian vektor normal dengan cara ini disebut *resultan normal berbobot*, dengan persamaannya adalah sebagai berikut,

$$N_p = \frac{\sum_{i=1}^n N_i * L_i}{\sum_{i=1}^n L_i} \tag{7}$$

Dimana N_p adalah vektor normal yang ingin dicari, N_i adalah vector normal dari segitiga i. Setiap vector normal N terdiri dari 3 komponen nilai, yaitu nilai axis-x, nilai axis-y, dan nilai axis-z.



Gambar 7. *Spherical Image* membuat bidang proyeksi menggunakan vector normal verteks

Kecenderungan nilai kurvatur yang dihasilkan oleh metode *spherical image* adalah >1 jika bentuk permukaan cembung atau cekung, <1 jika bentuk permukaan cekung, dan 1 jika permukaan datar atau membentuk sadel (*saddle point*). Berikut adalah tabel estimasi kurvatur beserta kecenderungan bentuknya untuk metode *spherical image*.

Tabel 2. Nilai kurvatur *Spherical Image* dan bentuk permukaan bidang

Nilai Kurvatur	Bentuk permukaan
$k > 1+t$	Cembung
$K < 1-t$	Cekung
$Abs(k-1) < t$	Datar / sadel

C. Karakteristik permukaan bidang

Karakteristik permukaan bidang bisa diestimasi dengan sempurna dengan menggabungkan kedua metode perhitungan kurvatur. Penggabungan ini menggunakan *if-then rules* dan *threshold*. Berikut adalah tabel aturan *if-then rules* untuk menggabungkan kedua metode estimasi kurvatur tersebut.

Tabel 3. Penggabungan *Gauss-Bonnet & Spherical Image*

Gauss-Bonnet	Spherical Image	Bentuk permukaan
$abs(k_{gb}) < t$	$abs(k_s - 1) < t$	Datar
$k_{gb} < -t$	$abs(k_s - 1) < t$	Sadel
$k_{gb} > t$	$k_s > 1+t$	Cembung
$k_{gb} > t$	$k_s < 1-t$	Cekung

Semua verteks harus menyimpan karakteristik yang dimilikinya untuk proses identifikasi segmen pada metode *region growing*.

IV. Model Faset 3D

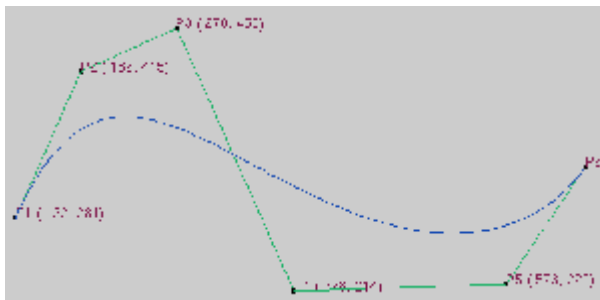
Model 3D yang digunakan pada penelitian ini adalah model faset 3D. Selain model faset terdapat model-model lain yang bisa digunakan, seperti model solid geometri, model *NURBS (Non-uniform Rational B-Spline Surface)*, dan model parametric. Model solid adalah model pejal yang disediakan oleh system CAD (*Computer Aided Design*) sesuai dengan spesifikasi software CAD yang bersangkutan. Model solid yang digenerate di satu software

CAD hanya bisa digunakan di software tersebut. Model solid bukan standar umum yang bisa digunakan di semua sistem berbasis desain 3D.

Model NURBS (*Non-uniform Rational B-Spline Surface*) adalah model 3D dimana surface model tersusun berdasarkan *basis function* dan *control point* [12,17]. Bentuk umum dari model NURBS adalah sebagai berikut,

$$C(u) = \frac{\sum_{i=1}^k \frac{N_{i,n} w_i}{\sum_{j=1}^k N_{j,n} w_j} P_i}{\sum_{i=1}^k N_{i,n} w_i} \tag{8}$$

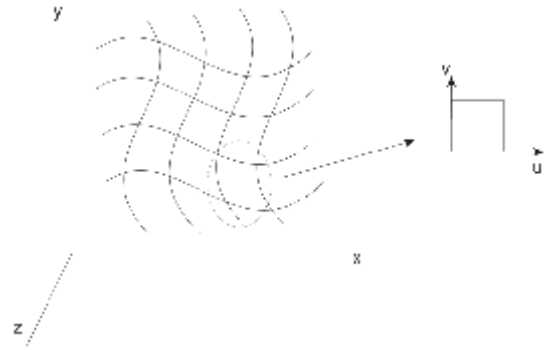
Dimana k adalah jumlah control point, P_i adalah control point (terdiri dari 3 komponen, yaitu x,y,dan z), w_i adalah bobot dari masing-masing control point, dan N adalah basis function.



Gambar 8. Contoh Kurva NURBS

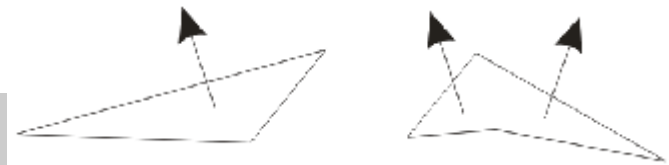
Model parametric adalah model 3D yang direpresentasikan menggunakan sebuah persamaan berparameter. Persamaan ini umumnya berbentuk polinomial, dimana setiap faset dari model penyusun masing-masing memiliki persamaan polinomial dengan parameter yang berbeda.

$$\begin{aligned} x(u, v) &= a_x u^2 + b_x uv + c_x v^2 + d_x u + e_x v + f_x \\ y(u, v) &= a_y u^2 + b_y uv + c_y v^2 + d_y u + e_y v + f_y \\ z(u, v) &= a_z u^2 + b_z uv + c_z v^2 + d_z u + e_z v + f_z \end{aligned} \tag{9}$$

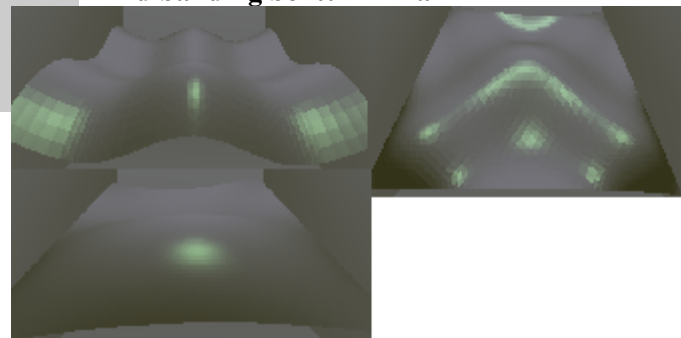


Gambar 9. Model Parametric

Model faset sendiri adalah model 3D yang tersusun atas sejumlah besar model 2D sederhana. Model 2D penyusun umumnya berbentuk segitiga, karena segitiga memiliki konsistensi vektor normal dari ketiga sisi dan sudutnya. Karena tersusun dari kumpulan segitiga, maka model faset ini sering disebut dengan *triangular meshes*.



Gambar 10. Model 2D penyusun model faset, segitiga memiliki konsistensi vector normal dibanding bentuk 2D lain



Gambar 11. Beberapa contoh model faset yang digunakan dalam penelitian ini

Model faset dipilih dalam penelitian ini karena strukturnya yang sederhana dan mudah diolah dalam CAM (*Computer Aided Manufacturing*), terutama dalam menentukan titik potong antara sebuah bidang datar 3D dengan model faset. Model faset adalah model standar sistem berbasis desain 3D, sehingga menggunakan software apa saja model ini bisa di-load dengan mudah.

Media penyimpanan digital model faset adalah file stl (*stereolithography*)[18]. Dalam file ini terdapat beberapa keyword yang mendefinisikan model 3D. Ada 2 jenis file stl,

yaitu file stl berbasis biner (*binary stl file*) dan file stl berbasis text (*text based stl file*). File stl biner memuat informasi berupa jumlah model faset, posisi-posisi verteks beserta vector normal segitiga dalam bentuk biner yang tidak terbaca manusia. Sedangkan file stl bentuk text mudah dibaca manusia. Secara ukuran file stl biner lebih disukai karena jauh lebih kecil dibanding file stl text. Berikut adalah contoh isi file stl text.

```
SOLID
  FACET NORMAL -3.27480E-01 -9.40168E-02 +9.40168E-01
  OUTER LOOP
    VERIEK +0.00000E+00 +1.56250E+00 -9.04374E-02
    VERIEK +0.00000E+00 +0.00000E+00 -9.99999E-02
    VERIEK +5.58941E-01 +6.74513E-01 -9.73806E-02
  ENDOUR
ENDFACET
FACET NORMAL -1.90432E-01 -2.14385E-01 +9.58005E-01
  OUTER LOOP
    VERIEK +1.54271E+00 +0.00000E+00 -9.69339E-02
    VERIEK +5.58941E-01 +6.74513E-01 -9.73806E-02
    VERIEK +0.00000E+00 +0.00000E+00 -9.99999E-02
  ENDOUR
ENDFACET
ENDSOLID
```

V. Region Growing

Region growing adalah salah satu metode segmentasi berbasis region (*region based segmentation*). Segmentasi ini berjalan berdasarkan nilai intrinsik suatu piksel dan posisi piksel. Piksel-piksel berdekatan (bertetangga) yang memiliki kemiripan nilai (di bawah batas ambang / *threshold*) akan dikelompokkan dalam segmen yang sama.

Berikut adalah dasar matematis dari metode region growing[14]. Proses region growing dimulai dari sebuah segmen A_1 yang berisi 1 piksel pada suatu citra, dan setelah berjalan, bermunculan segmen-segmen lain yaitu $A_1, A_2, A_1, \dots, A_n$. Misalkan T adalah kumpulan piksel-piksel yang belum terlokasikan ke salah satu diantara n segmen tersebut, maka T bisa ditulis dalam bentuk symbol matematika sebagai berikut,

$$T = \left\{ x \notin \bigcup_{i=1}^n A_i \wedge \exists k : N(x) \cap A_k \neq \emptyset \right\}_y \quad (10)$$

Dimana $N(x)$ adalah piksel-piksel yang bersebelahan langsung dengan x .

Selanjutnya dilakukan perhitungan selisih antara piksel x dengan rata-rata segmen lain

dimana segmen lain tersebut salah satu anggotanya adalah piksel tetangga dari x ,

$$\delta(x, A_i) = |g(x) - \text{mean}_{y \in A_i} [g(y)]| \quad (11)$$

Dimana $g(x)$ melambangkan nilai citra pada piksel x , dan i adalah index dari region dimana $N(x)$ beririsan dengan A_i .

Selanjutnya Proses *growing* dilakukan dengan memilih selisih x dengan rata-rata segmen tetangga yang memiliki nilai terkecil. Pilihan x terkecil selanjutnya ditulis sebagai z .

$$\delta(z, A_j) = \min_{x \in T, k \in [1, n]} \{ \delta(x, A_k) \} \quad (12)$$

Jika nilai $\delta(z, A_i)$ lebih kecil dari *threshold*, maka piksel z dimasukkan ke dalam segmen A_i . Namun jika tidak, maka pilih salah satu region A dimana region A tersebut memiliki kesamaan substansial dengan piksel z .

$$A = \arg \min_{A_k} \{ \delta[z, A_k] \} \quad (13)$$

Jika kedua kondisi tersebut tidak memenuhi, dan piksel z sangat berbeda dengan kondisi segmen-segmen yang berada di sekitarnya, maka region baru A_{n+1} harus dibentuk dengan anggota awalnya adalah piksel z .

D. Region Growing Model Faset 3D

Model faset 3D terbentuk dari kumpulan faset yang saling berhubungan, dan masing-masing faset terbentuk dari verteks dan sisi. Di sini bisa disimpulkan bahwa model faset tersusun atas verteks yang saling bertetangga. Proses pembentukan *region growing* adalah dengan memilih salah satu verteks sebagai inisiasi verteks, dimasukkan ke sebuah region baru, kemudian mencari verteks tetangganya yang memiliki kesamaan karakteristik dengan verteks inisiasi tersebut.

Berbeda dengan region growing pada citra, karakteristik verteks yang menjadi dasar pengelompokkan bukanlah nilai bilangan decimal, namun lebih ke bentuk permukaan bidang pada verteks tersebut.

Hal ini menyebabkan munculnya segitiga-segitiga yang verteks-verteksnya masuk ke lebih dari 1 segmen. Segitiga ini menjadi boundary segmen (Gambar 12).

Umumnya proses segmentasi tidak sempurna karena munculnya banyak segmen-segmen yang anggotanya terlalu sedikit, sehingga model yang besar terbagi atas banyak segmen. Untuk itu harus ada proses tambahan yang merupakan modifikasi dari proses *region growing* standar, yaitu menyatukan segmen dimana segmen yang anggotanya lebih sedikit disatukan dengan segmen tetangga yang anggotanya lebih banyak. Hal ini terus dilakukan hingga semua segmen memiliki anggota berjumlah lebih dari *threshold* yang ditentukan.

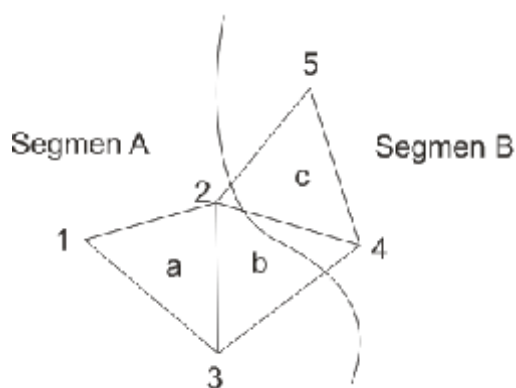
Dua segmen disebut bertetangga jika terdapat verteks dalam kedua segmen tersebut, dimana verteks-verteks tersebut saling bertetangga. Verteks tersebut saling bertetangga jika terdapat minimal sebuah segitiga T yang memiliki anggota kedua verteks tersebut. Misalkan terdapat $N(S_i, S_j)$ dimana S_i, S_j saling bertetangga, maka symbol matematika berikut berlaku,

$$N(S_i, S_j) = \{x \in S_i \wedge y \in S_j : x, y \in T\} \quad (14)$$

Dimana x, y adalah piksel, kemudian S_i, S_j adalah 2 segmen yang saling bertetangga, dan T adalah segitiga.

Definisi ini juga bisa dilihat dari sudut pandang segitiga. Dua buah segmen disebut bertetangga jika terdapat minimal sebuah segitiga yang masuk ke dalam sebuah segmen tersebut.

$$T(S_i, S_j) = \{T \in S_i \wedge T \in S_j\} \quad (15)$$



Gambar 12. Garis pemisah segmen

Pada Gambar 12, Segmen A dan Segmen B bertetangga disebabkan karena terdapat segitiga b dan c yang merupakan anggota kedua segmen

tersebut. Selain itu, terdapat pula verteks 2 dan 3 yang masuk ke Segmen A, bertetangga dengan verteks 4 dan 5 yang masuk ke Segmen B.

VI. Implementasi Region Growing

Sebelum proses *region growing* dimulai, setiap verteks harus sudah dikelompokkan ke dalam salah bentuk terlebih dahulu. Berbeda dengan *region growing* pada citra, pertimbangan dimasukkannya sebuah verteks ke segmen tertentu bukan didasarkan atas nilai fitur dengan *threshold*, namun didasarkan atas karakteristiknya, yang direpresentasikan dalam bentuk permukaan bidang. Berikut adalah proses yang terjadi pada metode *region growing* 3D.

1. Kumpulkan verteks yang sudah diketahui karakteristiknya dalam sebuah list.
2. Ambil sebuah verteks dari list yang belum masuk ke segmen manapun, buat segmen baru. Segmen ini memiliki karakteristik sama dengan verteks tersebut.
3. Cari semua verteks tetangga yang memiliki karakteristik sama, masukkan ke segmen tersebut.
4. Cari tetangga dari verteks-verteks yang sudah dimasukkan dalam segmen tersebut untuk diperiksa dan dimasukkan ke segmen yang sama.
5. Kembali ke langkah 3 hingga tidak ditemukan verteks tetangga dari segmen yang sudah terbentuk yang memiliki karakteristik yang sama dengan segmen yang sedang diperiksa.
6. Kembali ke langkah 2 hingga tidak ada verteks yang belum dikelompokkan ke satu segmen.
7. Tetapkan ukuran minimum segmen (berdasarkan jumlah verteks yang dikandung segmen tersebut).
8. Jika segmen berada di bawah ukuran minimum, gabungkan segmen tersebut dengan tetangganya, dan ikutkan karakteristiknya dengan segmen yang memiliki ukuran terbesar.
9. Ulangi langkah 7 hingga tidak ada lagi segmen dengan ukuran di bawah standar minimum.
10. Periksa kembali segmen yang terbentuk, gabungkan segmen bertetangga yang memiliki kesamaan karakteristik.

11. Ulangi langkah 10 hingga tidak ada perubahan segmen.
12. Tentukan batas-batas segmen. Batas segmen diketahui dengan mencari sisi yang hanya dimiliki oleh satu segitiga saja pada suatu segmen. Sisi ini yang menjadi segmen.

$$B_i = \left\{ \bigcup_{j=1}^n e_j : e_j \in T_a \wedge e_j \in T_b \wedge T_a \in S_i \wedge T_b \in S_k \wedge S_i \neq S_k \right\} \quad (16)$$

VII. Hasil dan Analisa

Model faset yang digunakan untuk ujicoba segmentasi bisa dilihat pada Gambar 13. Model-model tersebut mewakili beberapa kombinasi dari berbagai bentuk permukaan. Hasil segmentasi terhadap model uji bisa dilihat pada Gambar 14.

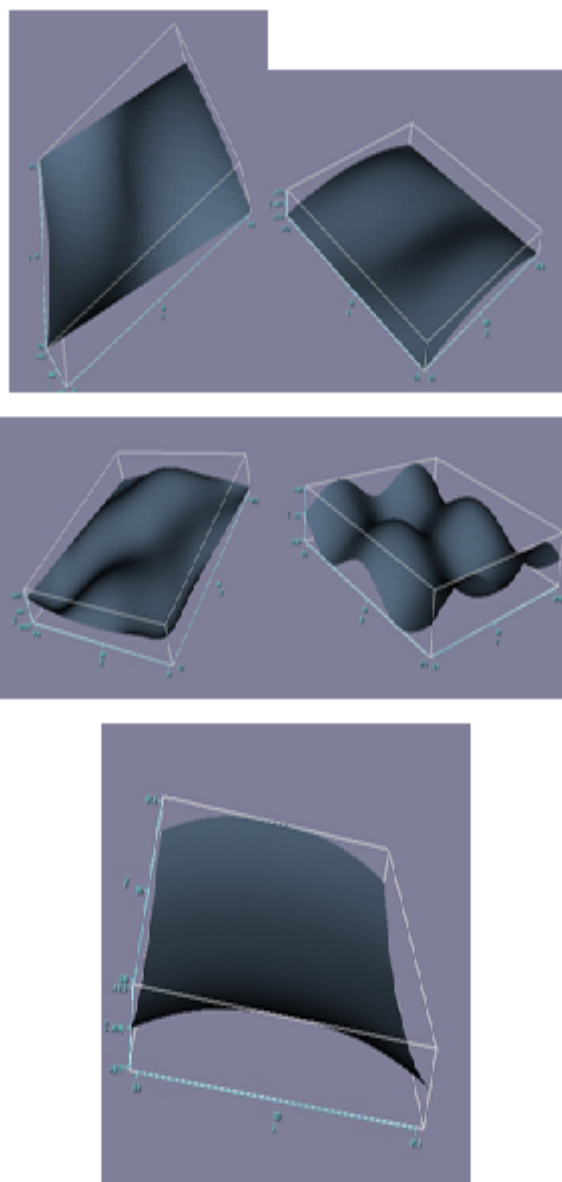
Pada Gambar 14a untuk model berbentuk cembung-cekung, hasil segmentasi masih kurang sempurna, dengan tingkat akurasi sebesar 72%. Pada bentuk tersebut bentuk cembung hanya terdeteksi sedikit, begitu pula dengan bentuk cekung. Namun bagian datarnya sudah teridentifikasi dengan baik. Pada Gambar 14b model berbentuk cembung-cembung. Tingkat akurasi lebih baik dari model pertama, yaitu 80%. Bentuk cembung sudah terlihat polanya, begitu pula bentuk sadel dan datar yang mengelilingi bentuk cembung tersebut. Pada Gambar 14c dan 14d, semua bentuk cekung dan cembung sudah terdeteksi sempurna, begitu pula bagian sadel dan datar. Akurasi pada Gambar tersebut mencapai 95%. Pada Gambar 14e, keseluruhan Gambar berbentuk sadel, dan system bisa mendeteksi hanya ada 1 segmen dan segmen tersebut berbentuk sadel. Disini akurasi system adalah 100%.

Tingkat kehalusan model (*threshold*) berbeda-beda. Model pada Gambar 14a memiliki tingkat kehalusan tinggi dibanding model lainnya, dalam artian *threshold* segitiga penyusunnya sangat kecil. Kecilnya *threshold* menyulitkan perhitungan sebab bilangan floating point memiliki keterbatasan menyimpan dan memproses bilangan yang sangat kecil.

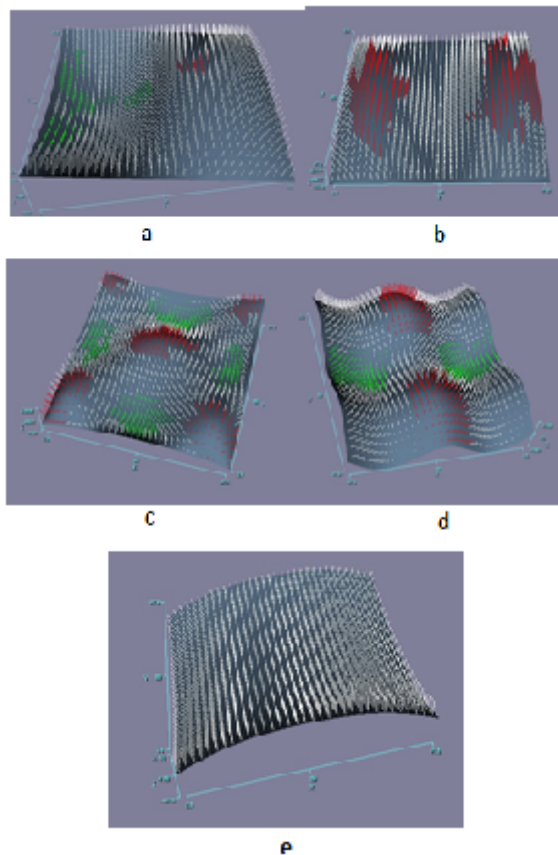
Karena pertimbangan kecepatan, tipe data yang digunakan adalah floating point 32 bit. Nilai presisi untuk bilangan kecil (antara 0 hingga 1) hanya mencapai 2^{-126} . Nilai presisi

tersebut akan semakin besar jika angka depan koma menjauh dari 1.

Disini terlihat bahwa semakin halus model, nilai *threshold* harus semakin kecil untuk meningkatkan presisi hasil segmentasi. Model pada Gambar 14a memiliki tingkat kehalusan yang tinggi, sehingga dengan nilai *threshold* yang sama, akurasi segmentasi tidak lebih baik dari yang lain. Namun kecilnya nilai *threshold* ada batasnya karena computer memiliki keterbatasan dalam menyimpan bilangan riil. Ketika sebuah bilangan sudah melebihi kemampuan tipe data untuk menyimpan nilai, *threshold* yang semakin kecil malah merusak hasil segmentasi.



Gambar 13. Model yang diujikan dalam penelitian



Gambar 14. Hasil segmentasi

VIII. Kesimpulan

Pada penelitian ini segmentasi model faset 3D menggunakan *region growing* telah diimplementasikan dan diujicobakan dengan sempurna. Proses segmentasi *region growing* harus dimodifikasi agar bisa menerima input berbentuk model 3D. Pada citra 2D metode *region growing* mengumpulkan piksel ke dalam 1 segmen dengan memeriksa intensitas tetangganya secara rekursif. Piksel tetangga adalah piksel yang berjarak 1 dengan piksel yang lain. Sedangkan pada model 3D *region growing* dilakukan dengan memeriksa kesamaan fitur berupa bentuk model antara satu verteks dengan verteks lain yang bertetangga (dihubungkan langsung dengan sebuah edge).

Hasil ujicoba menunjukkan tingkat akurasi segmentasi *region growing* cukup baik, berkisar antara 70% hingga 100%. Tingkat akurasi sangat dipengaruhi oleh kehalusan gambar dan bilangan presisi yang digunakan.

Untuk penelitian selanjutnya, segmentasi *region growing* perlu dikombinasikan dengan metode lain, seperti *convex hull* atau *dbscan* untuk menutupi kekurangan metode *region growing*.

PERNYATAAN

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada ketua jurusan teknik informatika dan dekan fakultas teknik atas kesempatan dan bantuan yang diberikan untuk berpartisipasi dalam seminar nasional teknologi 2014 yang diadakan oleh FTUMJ (<http://www.ftumj.ac.id>) di Jakarta.

References

- [1] Adams, R., Bischof, L. "Seeded Region Growing", IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL.16, No.6, June 1994.
- [2] Grenier, T., Muller, C., R., Costes, N., Janier, M., Gimenez, G. "3D Robust Adaptive Region Growing for segmenting [18F]fluoride ion PET images", Nuclear Science Symposium Conference Record, Vol.5, Oct 29-Nov 1 2006.
- [3] Jia, Y., B, "Curvature", Computer Science 477/577 Notes, Oct 9 2014, <http://www.cs.iastate.edu/~cs577/handouts/curvature.pdf>. [4 November 2014]
- [4] Kiswanto, G. "Pengembangan dan Pembuatan Sistem CAM (Computer Aided Manufacturing) yang Handal Berbasis Model Faset 3D untuk Pemesinan Multi-axis dengan Optimasi Orientasi Pahat dan Segmentasi Area dan Arah Pemesinan", Laporan Kemajuan RUT XII Tahap II, 2005.
- [5] Kiswanto, G., Kresnha, P. E. "Triangles Bucketing and Running Bucket Simulation for Possible Gouging Area Verification in CAM-system Based on Faceted Models", Proceeding of National Conference on Technology In Simulation TechnoSim 2006, University of Gadjah Mada, Yogyakarta, September 2006.
- [6] Kiswanto, G., Kresnha, P., E. "Curvature Evaluation of Faceted Models for Optimal Milling Direction. The 1st APCOMS, Agustus 2007.
- [7] Kamdi, S., Krishna, R.,K. "Image Segmentation and Region Growing Algorithm", IJTEE Vol.2, Issue 1, 2013
- [8] Kiswanto, G., Rahmat, W., Kresnha, P., E. "Bucketing. Strategies for Efficient Triangle Detection in 3D Faceted Model CAM System", University of Indonesia, Journal of Informatics and Computer Science, 2009.

- [9] Kresnha, P., E. "Pengembangan Sistem CAM Berbasis ModelFaset 3D untuk Pemesinan 5 Axis", Laporan Kerja Praktik, 2007.
- [10] Kresnha, P., E. "Perbandingan Metode Estimasi Kurvatur dalam Menentukan Bentuk Permukaan Bidang Pada Model Faset 3D", Laporan Tugas Akhir, 2007.
- [11] Kresnha, P., E, Widyanto, R., Kiswanto, G. "Intelligent Fuzzy 3D Feature Identification Based on Faceted Model", Proceeding of The 2nd International Conference on IT Application and Management : Managing Global Uncertainties, pages 112-121, University of Indonesia, Jakarta, Juni 2009.
- [12] Lavoie, P. "An introduction to NURBS", January 20, 1999, <http://hompi.sogang.ac.kr/gsmc/digital/gallery-data/nurbsintro.pdf>. [4 November 2014].
- [13] Lefebvre, P., P., Lauwers, B. "STL Model Segmentation for Multi-Axis Machining Operations Planning", 2004.
- [14] Lin, Z., Jin, J., Talbot, H. "Unseeded region growing for 3D image segmentation", Visualisation 2000, Pan-Sydney Workshop on Visual Information Processing, Conferences in Research and Practice in Information Technology, Vol. 2, December 2000.
- [15] Meek, D., Walton, D. "On Surface Normal and Gaussian Curvature Approximations Given Data Sampled from A Smooth Surface", Computer Aided Geometric Design, (17):521-543, 2000.
- [16] Surazhsky, T., Magid, E., Soldea, O., Elber, G., Rivlin, E. "Comparison of Gaussian and Mean Curvatures Estimation Methods on Triangular Meshes". In IEEE Transaction on Robotics and Automation, volume 1, pages 1021-1026, September 2003.
- [17] Verth, J., V., Bishop, L., M. "Essential Mathematics for Games and Interactive Applications 2nd Edition", Morgan Kaufman, Burlington,USA, 2008.
- [18] Wah, W., H. "Introduction to STL Format", June 1999, http://download.novedge.com/Brands/FPS/Documents/Introduction_To_STL_File_Format.pdf. [4 November 2014].