

# Aplikasi untuk Mendeformasikan Objek Tanpa Background

Danny Setiawan Putra Robianto<sup>1</sup>, Liliana<sup>2</sup>, Kartika Gunadi<sup>3</sup>

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

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

Email: m26410061@john.petra.ac.id<sup>1</sup>, lilian@petra.ac.id<sup>2</sup>, kgunadi@petra.ac.id<sup>3</sup>

**ABSTRAK:** Animasi merupakan salah satu pengolahan citra digital yang membuat sebuah objek tampak bergerak. Animasi terdiri dari kumpulan gambar dengan urutan tertentu pada sebuah *timeline* yang mengalami perubahan sedikit demi sedikit. Pada teknik animasi klasik atau konvensional untuk membuat sebuah objek tampak bergerak dilakukan penggambaran ulang satu per satu dari gambar semula hingga menjadi gambar akhir yang diinginkan animator sehingga memerlukan waktu yang lama.

Skripsi ini bertujuan untuk membantu mempercepat proses pembuatan animasi dengan memanfaatkan teknik deformasi yang dapat menghasilkan file animasi. Metode deformasi objek gambar yang digunakan adalah *Moving Least Square* dengan *rigid transformation*. Selain objek gambar sebagai salah satu *input* diperlukan juga *control point* dan *line segment* untuk menentukan arah deformasi. Perhitungan tidak dilakukan pada semua *pixel* objek gambar namun berdasarkan *grid*. Proses dimulai dengan membentuk *grid* dan dilakukan proses deformasi berdasarkan *control point* dan *line segment*. Setelah mendapatkan *grid* yang baru, dilakukan *inverse projective mapping* untuk memetakan *pixel* gambar asli ke tujuan dan juga *bilinear interpolation* agar hasilnya menjadi *smooth*. Pembuatan aplikasi menggunakan bahasa pemrograman C# dengan Visual Studio 2012 sebagai IDE.

Hasil pengujian menunjukkan bahwa hasil deformasi dipengaruhi oleh ukuran dari *grid*, nilai *alpha*, dan peletakkan *control point* dan *line segment*. Kecepatan proses dan kualitas berbanding terbalik terhadap banyaknya titik yang dihitung. Proses deformasi dengan *control point* memiliki kompleksitas waktu  $O(n^3)$  sedangkan dengan *line segment*  $O(n^4)$ . Tidak ada setelan parameter yang pasti untuk dapat menghasilkan hasil yang sesuai. Pengembangan selanjutnya dapat dilakukan deformasi untuk objek 3D.

**Kata kunci:** Deformasi, *Moving Least Square*, *Rigid Transformation*

**ABSTRACT:** Animation is one of the digital image processing that makes an object appear to move. Animation consists of a collection of images with a specific sequence on a timeline that changes little by little. In classical or conventional animation techniques to make an object appear to move, it needs to be redrawn one by one from original images to be the animator's desired final image so it takes a long time.

This thesis aims to help accelerate the process of creating animation by using deformation techniques which can produce animation files. The used object image deformation method is the Moving Least Square with a rigid transformation. In addition to image objects as one of the required inputs, control points and line segments are also needed to determine the direction of deformation. The calculation is not done for all pixels, but based

on the pixel grid image object. The process begins with forming the grid and do the deformation based on control points and line segments. After getting a new grid, the inverse projective mapping is done to map the original image pixel to the destination and also bilinear interpolation so that the result becomes smooth. The application is made using the C # programming language with Visual Studio 2012 as the IDE.

The results show that the deformation results are influenced by the size of the grid, the value of alpha, and placed control points and line segments. Process speed and quality is inversely proportional to the number of points calculated. Deformation processes with control points has complexity  $O(n^3)$  while the line segment  $O(n^4)$ . There is no parameter settings that are sure to be able to produce appropriate results. For further development, deformation for 3D objects can be done.

**Keywords:** *Deformation*, *Moving Least Square*, *Rigid Transformation*

## 1. LATAR BELAKANG

Pengolahan citra digital semakin banyak dimanfaatkan oleh orang awam untuk melakukan manipulasi terhadap *image* yang dimiliki. Manipulasi yang dapat dilakukan sangat bervariasi sesuai dengan kebutuhan, misalnya memperbaiki foto yang tampak gelap maka akan dilakukan pengolahan citra berupa gamma correction, brightness, dan contrast adjustment. Adanya kebutuhan tersebut mendukung munculnya beberapa research dan aplikasi-aplikasi komersil pengolah citra dengan berbagai fitur yang cukup lengkap.

Manipulasi citra sebagai vektor merupakan salah satu pemanfaatan pengolahan citra digital yang digunakan dalam dunia animasi untuk membuat sebuah objek tampak bergerak. Pergerakan yang terjadi didapatkan dari beberapa gambar yang ditampilkan secara bergantian dalam kurun waktu tertentu. Animasi yang baik membutuhkan beberapa gambar dengan objek yang sama dengan sedikit perubahan bentuk secara perlahan-lahan yang dirangkai pada sebuah *timeline* dengan *frame rate* tertentu. Tinggi rendahnya sebuah *frame rate* menentukan jumlah gambar yang akan digunakan dalam tiap detiknya. Semakin halus animasi maka akan dibutuhkan semakin banyak gambar yang akan berdampak pada waktu pembuatannya terutama jika harus menggambar satu per satu secara manual seperti pada *hand-drawn animation* yang merupakan teknik animasi klasik.

*Image deformation* merupakan salah satu teknik yang dapat dimanfaatkan untuk mengatasi banyaknya waktu yang diperlukan dalam pembuatan animasi secara klasik. Penggambaran satu per satu untuk menunjukkan pergerakan sebuah objek tidak lagi diperlukan. Teknik ini dilakukan dengan bantuan titik – titik atau

garis – garis sebagai patokan untuk mentransformasi objek yang diinginkan. Penerapan teknik *image deformation* sebaiknya dilakukan pada objek gambar yang tidak memiliki *background* karena proses transformasi yang dilakukan akan berdampak pada hampir keseluruhan bagian dari gambar sehingga dapat merusak *background*. Oleh karena itu, teknik ini sangatlah sesuai digunakan untuk objek gambar tanpa *background* seperti pada pembuatan animasi.

Teknik deformasi secara umum menggunakan bantuan *triangulation* dan *grid* dalam melakukan perhitungan matriks tranformasi karena perhitungan yang dilakukan bukan pada setiap *pixel* dari objek gambar. Penggunaan *triangulation* seperti pada riset yang dilakukan oleh Igarashi [3] dengan memanfaatkan *quadratic error metrics* menghasilkan objek gambar yang *rigid*. Namun, Schaefer [5] menemukan kelemahannya, yaitu ketika objek gambar yang memiliki tekstur berpola *uniform* dilakukan deformasi maka hasil yang diperoleh tidak *rigid* pada teksturnya. Oleh karena itu, pada riset yang dilakukan oleh Schaefer [5] digunakan *moving least square* dengan *rigid transformation* dan *grid* agar hasil yang diperoleh *rigid* pada bentuk dan teksturnya. Berdasarkan penjabaran – penjabaran diatas maka pada skripsi ini akan dibuat aplikasi untuk mendeformasikan sebuah objek gambar tanpa *background* yang lebih *rigid* menggunakan ide dari Schaefer [5].

## 2. Moving Least Square Deformation

*Moving Least Square Deformation* adalah sebuah metode untuk mendeformasikan sebuah gambar dengan merekonstruksi fungsi kontinyu dari suatu himpunan titik sampel yang terorganisir melalui perhitungan *weight least square* terhadap wilayah sekitar dimana nilai rekonstruksi itu diminta. Dalam penerapannya himpunan titik sampel tersebut merupakan himpunan titik kontrol  $p$  dan  $q$  sebagai kumpulan titik kontrol deformasi dari titik kontrol  $p$  [5]. Persamaan fungsi deformasi  $f$  harus memenuhi 3 properti seperti yang diuraikan pada pengenalan penggunaan *Moving Least Squares* [4]. Persamaan umum *Moving Least Square* adalah

$$\sum_i w_i |L_v(p_i) - q_i|^2 \quad (1)$$

dimana  $L_v$  merupakan *affline transformation* pada setiap titik  $v$  pada gambar,  $p_i$  dan  $q_i$  merupakan baris vektor, dan  $w_i$  merupakan *weights*. substitusikan dengan matriks dari *rigid transformation* sehingga didapatkan fungsi deformasi  $f$  sebagai *rigid deformation*. Penerapannya fungsi *rigid deformation* dapat dilakukan dengan bantuan titik kontrol dan segmen garis.

### 2.1 Rigid transformation using control point

Penelitian yang telah dilakukan Igarashi [3] telah menunjukkan bahwa untuk mendapatkan bentuk yang realistis maka deformasi harus dilakukan sekaku mungkin. Dalam upaya memperoleh hasil yang kaku ditemukan adanya kendala, yaitu *non-linear constraint*  $M^T M = I$ . Horn menunjukkan bahwa tranformasi *rigid* yang optimal didapatkan melalui *eigenvalues* dan *eigenvector* dari sebuah kovarian matriks yang melibatkan  $p_i$  dan  $q_i$  dapat menyelesaikan permasalahan tersebut [2]. Schaefer [5] membuktikan bahwa *rigid deformations* berkaitan dengan *similarity deformation* dengan mendapatkan persamaan *rigid deformation*  $f_r(v)$  yang didapatkan melalui normalisasi vektor  $\bar{f}_r(v)$  yang dikalikan dengan suatu nilai skalar  $v - p$ . dan ditranslasikan oleh  $q$ .

$$f_r(v) = |v - p| \frac{\bar{f}_r(v)}{|\bar{f}_r(v)|} + q. \quad (2)$$

dimana  $\bar{f}_r(v)$  merupakan vektor yang didapatkan dari persamaan 3

$$\bar{f}_r(v) = \sum_i \hat{q}_i A_i \quad (3)$$

## 2.2 Rigid transformation using line segment

Metode *Moving Least Square* yang berdasarkan titik-titik kontrol digeneralisasikan untuk kurva sembarang pada bidang. Pertama, diasumsikan bahwa  $p_i(t)$  merupakan kurva kontrol ke  $i^{th}$  dan  $q_i(t)$  merupakan kurva deformasi yang berkorespondensi dengan  $p_i(t)$ . Persamaan kuadratik 1 digeneralisasikan dengan mengintegrasikan setiap kurva kontrol  $p_i(t)$  dimana  $t \in [0,1]$  sehingga membentuk persamaan 4

$$\sum_i \int_0^1 w_i(t) |p_i(t)M + T - q_i(t)|^2 \quad (4)$$

Vektor rotasi  $\bar{f}_r(v)$  yang digunakan adalah

$$\bar{f}_r(v) = \sum_j (\hat{c}_j \hat{d}_j) A_j \quad (5)$$

## 3. INVERSE PROJECTIVE MAPPING

*Projective mapping* adalah sebuah metode untuk dapat memetakan setiap titik dari *source* ke *destination* melalui *matrix mapping*. Sebuah *projective mapping* memiliki 8 derajat kebebasan yang dapat ditentukan dari 4 koordinat ujung sumber dan tujuan segiempat [1]. Jika diberikan korepondensi pemetaan  $(u_k, v_k)$  ke  $(x_k, y_k)$  untuk vektor berurutan maka  $k = 0, 1, 2, 3$ . Semua koordinat diasumsikan sebagai bilangan real.

Jika korespondensi vektor adalah sebagai berikut:

$$\begin{array}{cccc} x & y & u & v \\ x_0 & y_0 & 0 & 0 \\ x_1 & y_1 & 1 & 0 \\ x_2 & y_2 & 1 & 1 \\ x_3 & y_3 & 0 & 1 \end{array}$$

maka didapatkan

$$\begin{aligned} c &= x_0 \\ a + c - g x_1 &= x_1 \\ a + b + c - g x_2 + h x_2 &= x_2 \\ b + c - h x_3 &= x_3 \\ f &= y_0 \\ d + f - g y_1 &= y_1 \\ d + e + f - g y_2 + h y_2 &= y_2 \\ e + f - h y_3 &= y_3 \end{aligned} \quad (6)$$

Inverse *projective mapping* dapat digunakan untuk memetakan *quadrilateral* menjadi pesergi. Hasil 8 koofisien  $a, b, c, d, e, f, g, h$  pada persamaan 6 dimanfaatkan untuk menemukan *inverse* dari *mapping* yang diinginkan dan dilakukan *adjoint* untuk menghitung *mapping quadrilateral* menjadi persegi melalui persamaan berikut:

$$(u' v' q) = (x' y' w) \begin{pmatrix} ei - fh & fg - di & dh - eg \\ ch - bi & ai - cg & bg - ah \\ bf - ce & cd - af & ae - bd \end{pmatrix} \quad (7)$$

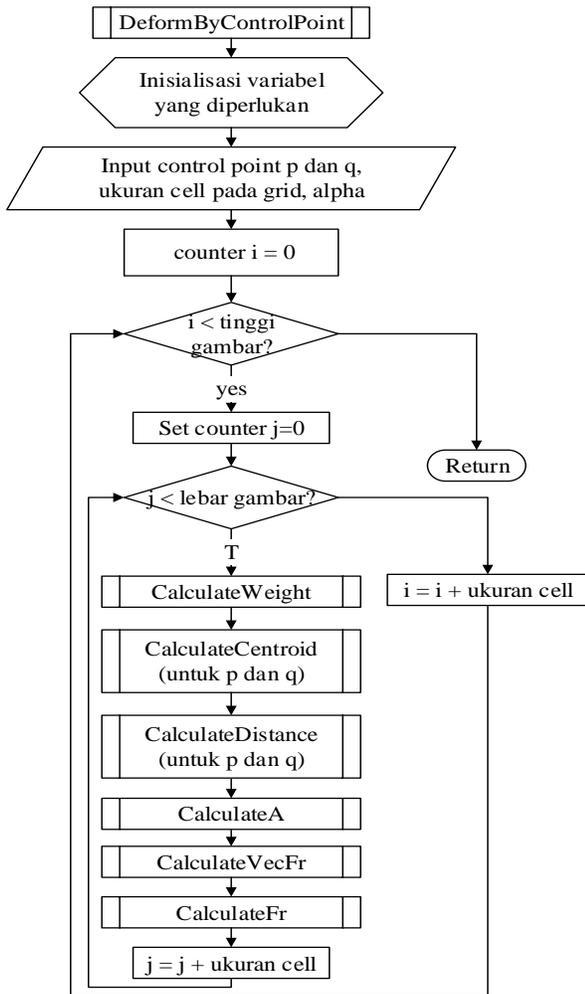
## 4. DESAIN SISTEM

### 4.1 Control Point

Proses ini adalah proses utama untuk melakukan deformasi pada gambar melalui perhitungan matematis yang dilakukan pada setiap titik perpotongan pada *grid* dengan bantuan *control point*. Proses deformasi diawali dengan meminta *input* dari *user* berupa *control*

point dan ukuran tiap cell pada grid. Terdapat 2 jenis control point yang di-input-kan oleh user, yaitu control point awal ( $p$ ) dan control point akhir ( $q$ ) yang menunjukkan arah dari deformasi. Kemudian dilakukan setting nilai  $\alpha$  dan counter untuk proses berulang perhitungan deformasi terhadap setiap titik ujung dari cell pada grid.

Proses perhitungan terbagi menjadi beberapa proses, yaitu proses menghitung  $weight$  suatu titik terhadap control point, menghitung titik pusat dari semua control point  $p$  dan  $q$ , menghitung selisih setiap control point terhadap pusatnya, menghitung matriks  $A$ , dan menghitung nilai vektor  $\vec{f}_r$ , serta menghitung nilai skalar  $f_r$ . Masing-masing fungsi tersebut secara berurutan dialiaskan menjadi *CalculateWeight*, *CalculateCentroid*, *CalculateDistance*, *CalculateA*, *CalculateVecFr*, dan *CalculateFr*. Flowchart dari proses *DeformByControlPoint* dapat dilihat pada Gambar 1.

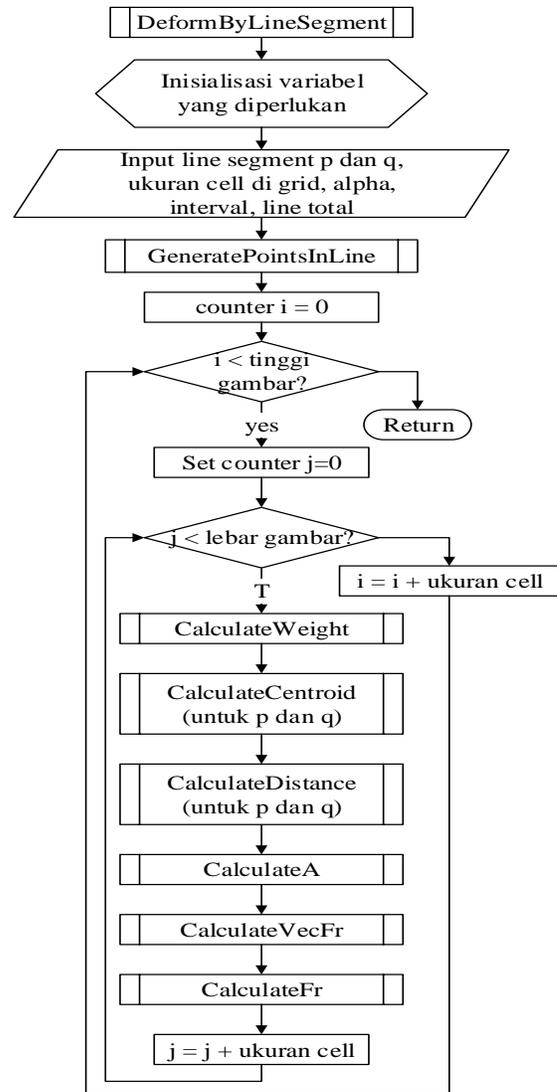


Gambar 1. Diagram alir perhitungan dengan control point

## 4.2 Line Segment

Proses ini adalah proses utama untuk melakukan deformasi pada gambar melalui perhitungan matematis yang dilakukan pada setiap titik perpotongan pada grid dengan bantuan line segment. Proses deformasi diawali dengan meminta input dari user berupa line segment dan ukuran tiap cell pada grid. Terdapat 2 jenis line

segment yang di-input-kan oleh user, yaitu line segment awal ( $p$ ) dan line segment akhir ( $q$ ) yang menunjukkan arah dari deformasi. Kemudian dilakukan penyetelan nilai  $\alpha$  dan counter untuk proses berulang perhitungan deformasi terhadap setiap titik ujung dari cell pada grid.



Gambar 2. Diagram alir perhitungan dengan line segment

Pada proses perhitungan deformasi menggunakan line segment terdapat adanya satuan waktu  $t$  yang digunakan untuk menghitung beberapa titik pada setiap line segment yang mempengaruhi nilai weight sehingga mempengaruhi juga nilai centroid serta matriks  $A$ . Sebelum melakukan proses perhitungan dilakukan terlebih dahulu generate titik-titik pada setiap garis sesuai dengan intervalnya. Beberapa proses perhitungan deformasi line segment, yaitu proses untuk menghitung weight suatu titik terhadap line segment, menghitung titik pusat dari semua line segment  $p$  dan  $q$ , menghitung selisih setiap line segment terhadap pusatnya, menghitung matriks  $A$ , dan menghitung nilai vektor  $\vec{f}_r$ , serta menghitung nilai skalar  $f_r$ . Masing-masing fungsi tersebut secara berurutan dialiaskan menjadi *CalculateWeight*, *CalculateCentroid*, *CalculateDistance*, *CalculateA*,

*CalculateVecFr*, dan *CalculateFr*. Flowchart dari proses *DeformByLineSegment* dapat dilihat pada Gambar 2.

## 5. PENGUJIAN

Pengujian pertama dilakukan untuk mengetahui kompleksitas proses deformasi berdasarkan waktu dalam notasi Big O baik pada *worse case* maupun pada *average case*. Berikut merupakan tabel hasil pengujian Big O dengan variasi terhadap beberapa aspek.

**Tabel 1. Tabel hasil pengujian Big O terhadap beberapa aspek**

No	Aspek	Big O ( <i>worse case</i> )	Big O ( <i>average case</i> )
1	Jumlah titik yang dihitung berdasarkan luas gambar	$O(n^3)$	$O(cn)$
2	Jumlah <i>control point</i>	$O(n^3)$	$O(cn^2)$
3	Jumlah <i>line segment</i>	$O(n^4)$	$O(cn^3)$

Berdasarkan Tabel 1 kompleksitas deformasi menggunakan *control point* bersifat *cubic* dikarenakan adanya perulangan sebanyak luas gambar dimana setiap *pixel* yang akan dihitung dikalikan dengan setiap *control point*. Deformasi menggunakan *line segment* memiliki kompleksitas yang melebihi dari deformasi menggunakan *control point* karena pada setiap *line segment* terbentuk titik-titik oleh interval yang akan dihitung dalam proses deformasi. Semakin tinggi orde pangkat pada Big O maka menunjukkan bahwa semakin kompleks proses yang dilakukan. Selain itu, didapatkan bahwa aspek jumlah titik yang dihitung dapat menurunkan tingkat kompleksitas hingga menjadi linear yang dikalikan dengan sebuah konstanta *c* sebagai jumlah titik pada gambar. Jumlah *control point* dan *line segment* pada *average case* dapat menurunkan 1 orde pangkat untuk kompleksitasnya.

Pengujian kedua dilakukan untuk mengetahui pengaruh ukuran *cell* pada *grid* terhadap kecepatan deformasi. Metode deformasi yang digunakan beracuan pada *control point* dan *line segment* dengan parameter yang tetap untuk nilai  $\alpha = 2$  dan interval  $t = 0.1$  (*line segment*). Berikut tabel hasil pengujian dengan varian ukuran *cell* yang berbeda-beda.

Tabel 2 menunjukkan bahwa ukuran *cell* pada *grid* mempengaruhi jumlah titik yang akan dihitung dalam deformasi sehingga menyebabkan waktu yang diperlukan semakin banyak. Semakin besar ukuran *cell* maka semakin sedikit titik yang diperhitungkan dan sebaliknya, semakin kecil ukuran *cell* maka akan semakin banyak titik yang diperhitungkan.

Pengujian ketiga adalah pengaruh ukuran *cell* pada *grid* dengan parameter nilai  $\alpha = 2$  dan interval = 0.1 (untuk *line segment*) terhadap kualitas hasil proses deformasi.

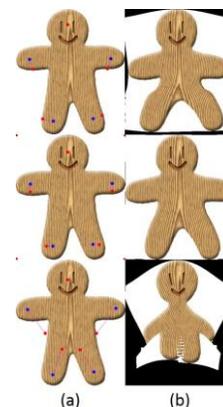
Berdasarkan Tabel 3 didapatkan bahwa bahwa ukuran *cell* pada *grid* mempengaruhi hasil deformasi yang ditandai dengan adanya patahan pada *texture* gambar. Bentuk yang dihasilkan telah sesuai dengan arah pergeseran *control point* atau *line segment* dan menghasilkan bentuk yang *rigid*. Hal ini dipengaruhi oleh pemetaan proyeksi yang dihitung berdasarkan pada setiap *skew cell*. Semakin luas *cell* yang digunakan maka hasil proyeksi tidak akan sesuai dari 1 *skew cell* dengan *skew cell* yang lain. Selain itu, terdapat perbedaan pada struktur objek karena semakin kecil ukuran *cell* pada *grid* maka akan semakin banyak *skew cell* yang terbentuk sehingga akan semakin detail area yang dipengaruhi

pada saat pemetaan ulang. Sebaliknya, jika semakin besar ukuran *cell* pada *grid* maka akan semakin sedikit *skew cell* yang terbentuk sehingga akan semakin tidak detail area yang dipengaruhi pada saat pemetaan ulang.

**Tabel 2. Pengujian terhadap pengaruh ukuran *cell* pada *grid* dengan kecepatan proses deformasi**

Ukuran <i>cell</i> ( <i>pixel</i> )	Jumlah titik <i>grid</i>	Waktu yang diperlukan	Keterangan
10×10	531	00:00:00.7389157	<i>control point</i>
20×20	146	00:00:00.5651105	<i>control point</i>
30×30	66	00:00:00.5362412	<i>control point</i>
40×40	38	00:00:00.4862641	<i>control point</i>
50×50	27	00:00:00.4656845	<i>control point</i>
10×10	1068	00:00:05.5991046	<i>line segment</i>
20×20	275	00:00:02.1412544	<i>line segment</i>
30×30	128	00:00:01.4996123	<i>line segment</i>
40×40	79	00:00:01.3064361	<i>line segment</i>
50×50	56	00:00:01.2274242	<i>line segment</i>

Selanjutnya adalah pengujian peletakan *control point* terhadap hasil deformasi dapat dilihat pada Gambar 4.

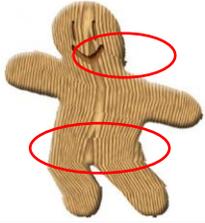


**Gambar 3. (a) Gambar dengan *control point* (b) Hasil deformasi**

Berdasarkan Gambar 3, dapat dilihat bahwa gambar (a) pertama ketika dilakukan deformasi maka *background* akan ikut tertarik. Hal ini berbeda dengan gambar (a) ke 2 karena ditambahkan *control point* pada ujung-ujung gambar. Pergeseran *control point* yang terlalu ekstrim seperti pada gambar (a) ke 3 dapat menyebabkan *foldback* (lipatan) sehingga hasil deformasinya tidak memiliki warna yang sesuai seperti pada gambar (b) ke 3. Adanya warna-warna yang tidak sesuai ini disebabkan oleh kegagalan pemetaan proyeksi sebab urutan pada ujung *skew cell* sudah tidak lagi searah dengan jarum jam atau dapat dikatakan telah terjadi *swap*.

Selanjutnya, pengujian dilakukan terhadap nilai  $\alpha$  untuk mengetahui pengaruhnya terhadap hasil deformasi dengan *control point* dan *line segment*.

**Tabel 3. Pengujian terhadap pengaruh ukuran *cell* pada grid dengan kecepatan proses deformasi**

Gambar	Keterangan
	Acuan: <i>control point</i> Ukuran <i>cell</i> : 10 × 10 <i>pixel</i> Hasil: <i>smooth</i>
	Acuan: <i>control point</i> Ukuran <i>cell</i> : 30 × 30 <i>pixel</i> Hasil: ada patahan
	Acuan: <i>control point</i> Ukuran <i>cell</i> : 50 × 50 <i>pixel</i> Hasil: ada patahan
	Acuan: <i>line segment</i> Ukuran <i>cell</i> : 10 × 10 <i>pixel</i> Hasil: <i>smooth</i>
	Acuan: <i>line segment</i> Ukuran <i>cell</i> : 30 × 30 <i>pixel</i> Hasil: ada patahan
	Acuan: <i>line segment</i> Ukuran <i>cell</i> : 50 × 50 <i>pixel</i> Hasil: ada patahan

**Tabel 4. Tabel hasil pengujian pengaruh nilai *alpha* dengan hasil deformasi**

No	Gambar	
1	<i>Alpha 0.5</i>	<i>Alpha 2</i>
	<i>Alpha 3.5</i>	<i>Alpha 5</i>
	Keterangan: parameter ukuran <i>cell</i> = 10, deformasi dengan bantuan <i>control point</i> .	
	2	<i>Alpha 0.5</i>
<i>Alpha 3.5</i>		<i>Alpha 5</i>
Keterangan: parameter ukuran <i>cell</i> = 10, interval = 0.1, deformasi dengan bantuan <i>line segment</i> .		

Tabel 4 menunjukkan bahwa semakin besar nilai *alpha* maka perpindahan titik-titik yang terdeformasi akan semakin menjauhi posisi awalnya karena berpengaruh pada nilai *weight*. Semakin besar nilai *alpha* maka nilai *weight* akan menjadi semakin kecil dan sebaliknya, semakin kecil nilai *alpha* maka nilai *weight* akan semakin besar. Perpindahan yang terlalu jauh dapat menyebabkan *foldback* (lipatan) yang merupakan kelemahan dari deformasi 2D karena tidak memiliki informasi warna belakang objek.

## 6. Kesimpulan

Berdasarkan hasil pengujian dapat disimpulkan beberapa hal sebagai berikut:

- Kompleksitas waktu proses deformasi dengan *control point* adalah  $O(n^3)$  sedangkan dengan *line segment* adalah  $O(n^4)$ .
- Ukuran *cell* pada grid menentukan titik-titik yang dihitung sehingga berpengaruh pada kecepatan dan kualitas pada hasil deformasi yang diukur berdasarkan tekstur dan bentuk. Semakin besar ukuran cell cenderung akan menghasilkan output yang tidak smooth (adanya patahan pada tekstur dan atau bentuk) tetapi prosesnya akan semakin cepat dan sebaliknya, semakin kecil ukuran cell cenderung akan menghasilkan output yang smooth tetapi prosesnya akan semakin lambat.
- Tinggi-rendahnya nilai alpha mempengaruhi perpindahan titik-titik yang dihitung.
- Nilai interval t pada deformasi dengan *line segment* sangat mempengaruhi kecepatan deformasi namun hasilnya tidak ada perubahan yang signifikan.
- Peletakan control point dan line segment mempengaruhi perpindahan titik-titik yang berpengaruh pada output.
- Tidak ada parameter yang pasti untuk memperoleh hasil yang maksimal karena bergantung juga pada kompleksitas objek yang dideformasikan.

## 7. REFERENSI

- [1] Heckbert, Paul S. (1989). *Fundamental Texture Mapping and Image Warping*. Berkeley: The University of California, Department of Electrical Engineering and Computer Science, pp. 17-21.
- [2] Horn, B. (1987). *Closed-form solution of absolute orientation using unit quaternions*. Journal of the Optical Society of America A 4, 4 (April), 629-642.
- [3] Igarashi, T., Moscovich T., Hughes, J. F. (2005). As-Rigid-As-Possible Shape Manipulation. ACM Trans. Vol. 24, 3, 1134-1141.
- [4] Levin, David. (1998). *Mathematics of Computation: The Approximation Power of Moving Least-Squares*. ASM. Vol 67, 224, 1517-1531.
- [5] Schaefer, S., McPhail, T., Warren, J. (2006). *Image Deformation Using Moving Least Square*. ACM Trans. Graph. Vol. 25, 3.