

## MORPHING CITRA DENGAN BERBAGAI TEKNIK MORPHING

Kartika Gunadi, Resmana Lim, Stephan Yos

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra  
e-mail : kgunadi@petra.ac.id, resmana@petra.ac.id, m26400165@john.petra.ac.id

### Abstrak:

Dalam pembuatan sebuah film, ada banyak spesial efek yang ditambahkan ke dalam sebuah film untuk menghasilkan suatu film yang lebih bagus. Salah satu spesial efek yang sering digunakan adalah *morphing*, yaitu suatu efek dimana suatu objek diubah perlahan-lahan menjadi objek lain. Sebelum digunakannya komputer, efek ini dilakukan dengan cara tradisional yang sulit dan memakan waktu lama dalam pembuatannya dengan hasil yang kurang memuaskan, dengan menggunakan komputer selain waktu pembuatannya jauh lebih cepat, hasilnya pun jauh lebih bagus.

Tugas Akhir ini membahas tentang pembuatan efek *morphing* dengan menggunakan beberapa algoritma. Ada banyak algoritma yang dapat digunakan untuk membuat efek *morphing*, di antaranya adalah *cross dissolve*, *feature morph*, *mesh morph*, dan *selection morph*. Aplikasi yang dibuat diharapkan dapat menunjukkan kelebihan dan kekurangan dari masing-masing algoritma yang digunakan. Aplikasi dibuat dengan menggunakan bahasa pemrograman Borland Delphi 6.

Hasil pengujian aplikasi ini menunjukkan bahwa dari algoritma-algoritma di atas yang memiliki waktu proses tercepat adalah teknik *cross dissolve* tetapi memiliki hasil yang paling jelek. Sedangkan yang memiliki hasil yang bagus dan waktu proses yang tidak terlalu lama adalah teknik *mesh morph*.

Kata kunci: *Morphing* Citra, Animasi Komputer, Interpolasi, Pengolahan Citra Digital, Transformasi bentuk

### 1. Pendahuluan

*Morphing* adalah suatu efek dimana suatu objek diubah perlahan-lahan menjadi objek lain. Pada dasarnya *morphing* dilakukan dengan membuat gambar-gambar transisi di antara gambar asal dan gambar tujuan (sumber: <http://www-users.itlabs.umn.edu>). Efek *morphing* ini sering digunakan di dalam pembuatan film, contohnya dalam film *werewolf*, dimana manusia dapat berubah menjadi serigala.

Aplikasi yang dibuat bertujuan untuk mengetahui kelebihan dan kelemahan dari masing-masing teknik *morphing* yang digunakan dan dapat menghasilkan efek *morphing* antara 2 gambar yang diinputkan. Gambar yang digunakan adalah file dengan ekstensi \*.bmp. Algoritma yang digunakan dalam aplikasi ini adalah *cross dissolve*, *feature morphing*, *mesh morphing*, *selection morphing* dan *liquid morphing*. Pembuatan aplikasi ini menggunakan Borland Delphi 6 dengan sistem operasi berbasis *windows*.

### 2. Teori Penunjang

#### 2.1. Cross Dissolve

Teknik *morphing* ini merupakan teknik *morphing* yang paling sederhana dan merupakan dasar bagi teknik-teknik *morphing* yang lain. Pada teknik *cross dissolve* ini yang dilakukan hanyalah menampilkan kedua gambar (gambar asal dan gambar tujuan) dengan mode transparan. Adapun level transparansi dari setiap gambar dapat diatur. Pada awal animasi, level transparansi dari gambar asal diset maksimum (100%) sedangkan gambar tujuan diset minimum (0%). Level transparansi 100%

menyebabkan gambar hasil penggabungan antara kedua gambar menjadi hanya gambar asal. Kemudian selama proses animasi, perlahan-lahan level transparansi dari gambar asal semakin lama semakin dikurangi, sedangkan level transparansi gambar tujuan semakin ditambah, dengan syarat level transparansi gambar asal dan level transparansi gambar tujuan jika dijumlahkan hasilnya 100%. Dengan demikian akan terlihat gambar asal perlahan-lahan menghilang sambil gambar tujuan terlihat perlahan-lahan muncul.

Algoritma *cross dissolve* secara garis besar dapat ditulis sebagai berikut.

```
For frame= 0 to jumlah_frame do
begin
R= frame/jumlah_frame
For x= 0 to image_width do
For y= 0 to image_height do
gmb_hsl[x,y]=(1-R)*gmb_asal[x,y]
+ R*gmb_tuj[x,y]
end
```

Looping For yang terluar digunakan untuk melakukan *morphing* dari frame nomor 0 hingga jumlah frame. Nilai R berfungsi sebagai transparansi level dari gambar tujuan. Nilai R didapat dari membagi frame saat ini dengan jumlah frame sehingga didapat range mulai 0 (0%) hingga 1 (100%). Variabel gambar\_hasil menunjukkan hasil gambar pada frame saat ini, gambar\_asal menunjukkan data gambar asal dan gambar\_tujuan menunjukkan data gambar tujuan.

## 2.2. Feature Morph

Pada teknik *cross dissolve*, tidak memperhatikan keberadaan sebuah objek di gambar asal dan gambar tujuan, seluruh gambar diasumsikan sebagai objek. Teknik *feature morphing* diciptakan untuk mengatasi masalah ini. Pada teknik ini, sebuah objek akan di-morph dari posisi, bentuk dan warna pada gambar asal perlahan-lahan menjadi posisi, bentuk dan warna pada gambar tujuan (Agustinus Nalwan: 2001, 57-111). Sebagai gambaran, mata yang berada pada posisi yang berlainan antara gambar asal dan tujuan akan berubah dan berpindah perlahan-lahan. Jika pada *cross dissolve*, pada pertengahan *morphing* akan terlihat 2 pasang mata secara samar-samar.

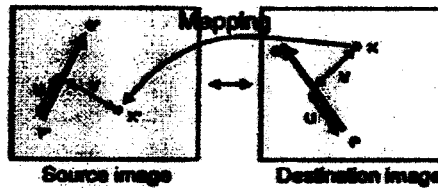
Pada teknik *feature morphing* ini digunakan garis-garis untuk menandai objek-objek dari gambar yang disebut dengan garis-garis *feature*. Sebuah garis *feature* pada gambar asal memiliki pasangannya pada gambar tujuan. Daerah di sekitar garis *feature* gambar asal ini menunjukkan sebuah *feature*/objek yang sama dengan daerah di sekitar garis *feature* pada gambar tujuan. Sehingga komputer mengetahui hubungan antara dua objek yang sama di gambar asal dan tujuan.

Dalam teknik *feature morphing* terdapat dua buah langkah. Langkah pertama ialah *deformation* antara gambar asal dan gambar tujuan dan langkah kedua ialah *cross dissolve* antara kedua gambar yang telah di-*deformation*.

### 2.2.1. Deformation (Deformasi)

Pada langkah ini, kedua buah gambar dideformasikan menjadi dua buah gambar yang sesuai dengan frame saat ini berdasarkan garis-garis *feature* yang terdapat ada gambar asal dan gambar tujuan.

Objek di gambar asal dideformasikan mengikuti perubahan dari garis-garis *feature*. Garis-garis ini berubah mulai dari bentuknya di gambar asal menjadi bentuknya di gambar tujuan. Perubahan ini ditentukan oleh frame mana yang sedang diproses. Selain mendeformasikan gambar asal, juga mendeformasikan gambar tujuan sesuai dengan bentuk garis-garis *feature* pada frame saat ini.



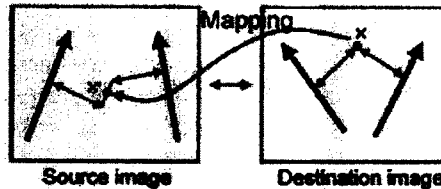
Gambar 1. Posisi Sebuah Titik Relatif terhadap Sebuah Garis Feature  
Sumber: <http://faculty.petra.ac.id/resmana/private/pcd/projects/morphing/>

Untuk melakukan proses deformasi, langkah pertama yang harus dilakukan adalah mencari koordinat dari masing-masing titik pada gambar terhadap garis *feature* yang ada. Koordinat axis *u* searah dengan garis *feature*, sedangkan koordinat axis *v* tegak lurus terhadap garis *feature*. Titik pusat dari koordinat ini ada pada titik awal dari garis *feature*.

Dasar dari proses deformasi adalah posisi (*u,v*) sebuah titik terhadap sebuah garis *feature* selalu relatif sama. Bila garis *feature* mengalami transformasi seperti rotasi, dilatasi atau translasi, maka titik tersebut juga mengalami hal yang sama dan tetap memiliki posisi yang relatif sama. Bila garis *feature* mengalami rotasi atau translasi, koordinat titik terhadap suatu garis *feature* tidak berubah, tetapi bila garis *feature* mengalami dilatasi maka koordinat titik tersebut ikut berubah sesuai dengan faktor pengali pada dilatasi yang terjadi.

Untuk proses deformasi dengan banyak garis *feature*, maka sebuah titik pada gambar akan dipengaruhi oleh semua garis *feature* yang ada. Seberapa besar kontribusi (*R*) sebuah garis *feature* terhadap suatu titik dipengaruhi oleh jarak titik terhadap garis *feature* (*d*) dan panjang garis *feature* tersebut. Semakin dekat jarak titik terhadap garis *feature* semakin besar kontribusi garis *feature* tersebut begitu pula bila semakin panjang garis *feature* tersebut maka semakin besar pula kontribusi garis *feature* tersebut.

$$R = \frac{\text{Panjang feature}}{(1 + d)} \quad (2.3)$$



Gambar 2. Sebuah Titik Dipengaruhi oleh 2 Garis Feature

Sumber: <http://faculty.petra.ac.id/resmana/private/pcd/projects/morphing/>

Koordinat posisi titik yang baru didapat dari menjumlahkan posisi–posisi baru yang dihasilkan dari setiap garis *feature* yang ada. Namun, sebelum dijumlahkan masing–masing dikalikan terlebih dahulu dengan sebuah harga yang merupakan nilai kontribusi dari garis *feature* tersebut terhadap titik tersebut. Kemudian hasil penjumlahan dibagi dengan jumlah kontribusi yang ada.

Pembuatan tiap frame pada teknik *feature morphing* dilakukan dengan melakukan deformasi pada gambar asal dan deformasi pada gambar tujuan. Misalnya pada pembuatan frame ke-10 dari 20 frame, maka gambar asal dideformasikan menjadi suatu gambar sementara (A) dengan berpatokan pada garis *feature* yang ada pada gambar asal dan garis *feature* sementara yang didapat dari interpolasi antara garis *feature* asal dan tujuan sesuai dengan nomor frame saat ini. Begitu pula gambar tujuan dideformasikan menjadi suatu gambar sementara (B) dengan berpatokan pada garis *feature* sementara tadi dan garis *feature* yang ada pada gambar tujuan. Kemudian gambar A dan B digabungkan dengan menggunakan teknik *cross dissolve*.

### 2.3. Mesh Morph

Teknik *morphing* ini juga sering disebut *Triangle Based Morphing*. Teknik *morphing* ini mirip dengan *feature morphing*, tetapi memiliki waktu proses yang lebih cepat dan pemilihan *feature* yang lebih bagus. Pada teknik ini menggunakan *feature-feature* yang membentuk kurva tertutup berbentuk segitiga. Sama seperti pada *feature morphing*, dimana garis *feature* asal berpasangan dengan garis *feature* tujuan. Disini sebuah segitiga asal berpasangan dengan sebuah segitiga tujuan.

Dalam pembuatan tiap frame pada *mesh morphing*, segitiga asal diinterpolasikan menjadi segitiga tujuan dengan menganggap sebuah segitiga terdiri atas tiga buah garis *feature*, ketiga garis *feature* inilah yang kita interpolasikan. Seiring dengan interpolasi ketiga garis *feature* tersebut, dilakukan proses deformasi terhadap titik-titik dalam segitiga tersebut dengan cara yang sama seperti *feature morphing*. Bedanya disini sebuah titik dalam segitiga hanya dipengaruhi oleh ketiga garis *feature* pembentuk segitiga tersebut. Sehingga masing-masing titik hanya dedeformasikan oleh tiga buah garis dimana garis-garis tersebut mewakili segitiga dimana titik tersebut berada.

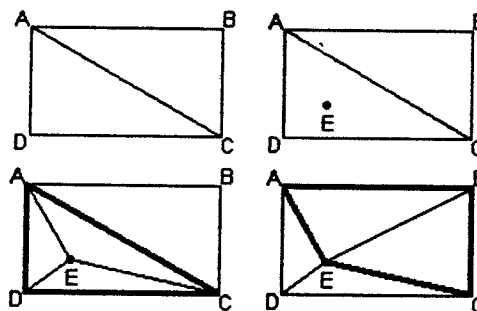
Dalam peletakan garis-garis *feature* pada gambar, dipergunakan teknik *delaunay triangulation*, sehingga yang perlu diinputkan hanyalah titik-titik pembentuk segitiga-segitiga tersebut yang disebut *site point* kemudian Secara otomatis algoritma tersebut akan menghubungkan titik-titik tersebut menjadi segitiga-segitiga.

#### 2.3.1. Delaunay Triangulation

Syarat sebuah *triangle network* disebut sebagai *delaunay triangle network* ialah jika segitiga-segitiga yang dihasilkan adalah segitiga yang hampir mendekati segitiga sama sisi.

Semakin gendut sebuah segitiga, semakin besar area dibawah segitiga tersebut sehingga area dari sebuah *feature* yang dikelilingi oleh segitiga ini menjadi lebih besar. Semakin sedikit dan gendut jumlah segitiga yang digunakan untuk memecah sebuah kurva yang mengelilingi *feature*, semakin kecil distorsi yang dihasilkan pada saat melakukan proses deformasi pada area didalam kurva tersebut.

Salah satu algoritma untuk merekonstruksi *delaunay triangle network* adalah *Insertion Algorithm*. Algoritma ini adalah sebagai berikut. Pertama-tama dimulai dengan meletakkan dua buah segitiga di layar.



Gambar 3. Langkah-Langkah dari *Insertion Algorithm*  
Sumber: Agustinus Nalwan. (2001). Seri Aplikasi  
Pemrograman Movie & Special Effect 2.

Kemudian *site point* dimasukkan satu per satu. Setiap kali sebuah *site point* dimasukkan kedalam network, misalnya *site point* ini adalah titik E, dilakukan proses sebagai berikut :

- Cari segitiga dimana *site point* itu berada. Yaitu segitiga ADC.
- Selanjutnya segitiga ADC dipecah menjadi AED, DEC dan AEC.
- Langkah selanjutnya adalah merekonstruksi *triangle network* untuk mendapatkan segitiga-segitiga sama sisi. Caranya dengan memeriksa setiap kurva segi empat yang ada dalam *triangle network*, apakah diagonal dari segi empat tersebut dapat ditukar sehingga kedua segitiga didalamnya menjadi lebih optimal. Pemeriksaan ini dimulai dari sekitar ketiga segitiga yang baru. Untuk itu sisi-sisi luar ketiga segitiga yang baru, dimasukkan ke dalam sebuah array dengan nama temp. Adapun sisi-sisi ini mewakili sebuah kurva segi empat yang akan diperiksa. Untuk situasi diatas,

sisi yang dimasukkan adalah AD, DC, CA (digambar dengan garis tebal). Sisi AD tidak mewakili kurva segi empat karena hanya ada satu segitiga yang memiliki sisi AD yaitu segitiga AED, demikian pula dengan sisi DC. Sedangkan sisi CA mewakili sebuah segi empat yaitu AECB. Oleh karena sisi AD dan DC tidak mewakili segi empat maka kedua sisi ini tidak jadi dimasukkan ke dalam array temp.

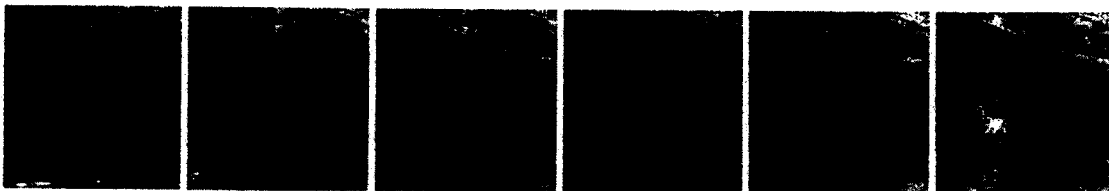
- Setelah sisi CA dimasukkan ke dalam array temp, dilakukan pemeriksaan kurva segi empat yang diwakili oleh sisi CA, yaitu segi empat AECB. Diagonal dari kurva AECB saat ini adalah CA. Pertama-tama dihitung besar sudut EAB + sudut ECB dan sudut ABC + sudut AEC. Ternyata sudut ABC + AEC lebih besar dari sudut EAB + ECB. Jadi diagonal EB lebih bagus daripada diagonal CA, dan diagonal dari AECB perlu ditukar. Pertukaran diagonal tersebut meliputi proses sebagai berikut:
  - Hapus segitiga AEC dan ABC.
  - Ciptakan segitiga AEB dan EBC.
  - Masukkan sisi-sisi luar segitiga AEB dan EBC ini ke dalam array temp. Yaitu sisi AB, BC, AE dan EC. Sisi EB tidak perlu dimasukkan karena merupakan diagonal sisi yang baru dibentuk. Sisi AB dan BC tidak mewakili segi empat, maka kedua sisi ini tidak jadi dimasukkan ke dalam array temp.
- Selanjutnya karena sisi CA telah selesai diperiksa maka hapus sisi ini dari array temp.
- Berikutnya ambil sisi selanjutnya yang akan diproses dalam array temp.
- Proses ini terus dilakukan sampai sisi dalam array temp habis. Setelah array temp kosong, ulangi lagi seluruh proses diatas dengan memasukkan *site point* yang baru.

#### 2.4. Selection Morph

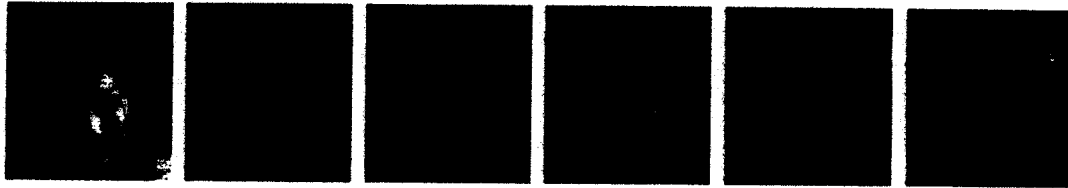
Teknik *selection morph* yaitu proses *morph* yang dilakukan hanya pada bagian-bagian tertentu saja. Misalnya bila hendak membuat film dengan peran utama kucing, seumpama kucing ini melakukan suatu adegan menangis, senyum pada saat-saat tertentu. Merupakan hal yang mustahil untuk melatih kucing supaya senyum atau menangis. Dengan menggunakan teknik ini dapat diciptakan animasi atau gambar dari kucing yang sedang tersenyum hanya dengan bermodalkan sebuah foto kucing tersebut. Pada teknik *selection morph* ini proses yang dilakukan hanyalah melakukan deformasi pada gambar yang diinputkan.

Pada teknik *feature morphing* dan *mesh morphing*, langkah pertama yang selalu dilakukan adalah proses deformasi dari gambar asal ke frame saat ini, juga deformasi dari gambar tujuan ke frame saat ini. Baru langkah selanjutnya dilakukan *cross dissolve* antara kedua gambar hasil deformasi tersebut. Jika frame saat ini adalah frame terakhir, maka gambar asal akan dideformasikan hampir menyerupai gambar kedua. Jika proses *morph* tersebut dilakukan antara gambar kucing dengan wajah normal ke gambar tujuan kucing tersenyum, maka gambar asal yang dideformasikan tadi telah menyerupai kucing tersenyum karena bibirnya ditarik hingga menuju ke pose tersenyum. Jadi dengan hanya memanfaatkan deformasi gambar asal tanpa *cross dissolve* dengan deformasi gambar akhir sudah diperoleh animasi dari kucing yang perlahan-lahan tersenyum. Teknik deformasi yang digunakan disini sama dengan teknik deformasi pada *mesh morph*.

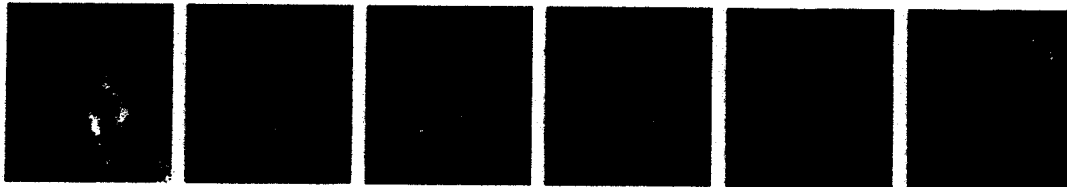
### 3. Hasil Dan Pengujian



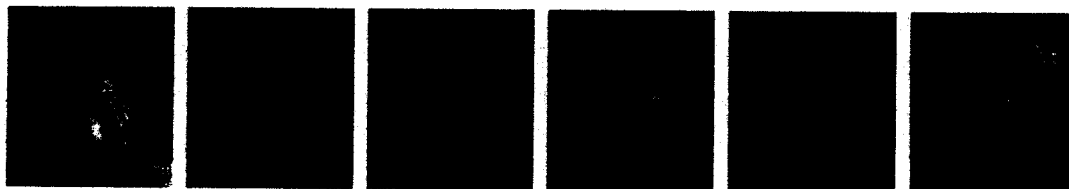
Gambar 6. Gambar Hasil Selection Morph



Gambar 7. Gambar Hasil Cross Dissolve



Gambar 8. Gambar Hasil Feature Morph



Gambar 9. Gambar Hasil Mesh Morph

Tabel 1. Tabel Penilaian Pengujian Kualitatif secara Subjektif

Uji Kualitatif	Score					Uji Kualitatif
Buruk	1	2	3	4	5	Baik

Tabel 2. Tabel Hasil Pengujian Kualitatif secara Subjektif

Responden ke-	Cross Dissolve	Feature Morph	Mesh Morph
1	2	3	4
2	1	3	3
3	3	4	5
4	2	3	4
5	1	2	3
<b>Rata-rata</b>	<b>1.8</b>	<b>3</b>	<b>3.8</b>

Tabel 3. Tabel Hasil Pengujian Waktu Proses Cross Dissolve

Pengujian ke-	Ukuran gambar	Jumlah Frame	Waktu Proses
1	224 x 150	80	0.55 detik
2	350 x 300	100	1.70 detik
3	400 x 400	80	1.87 detik
4	500 x 500	100	3.68 detik

Tabel 4. Tabel Hasil Pengujian Waktu Proses Feature Morph

Pengujian ke-	Ukuran gambar	Jumlah Frame	Jumlah Feature	Waktu Proses
1	175 x 136	50	15	0 menit 53.66 detik
2	224 x 150	30	20	1 menit 05.20 detik
3	300 x 300	50	10	2 menit 06.93 detik
4	300 x 300	50	18	4 menit 15.46 detik

Tabel 5. Tabel Hasil Pengujian Waktu Proses Mesh Morph

Pengujian ke-	Ukuran gambar	Jumlah Frame	Jumlah Site Point	Jumlah Feature	Waktu Proses
1	175 x 136	50	18	47	0 menit 32.62 detik
2	224 x 150	30	35	98	0 menit 58.33 detik
3	300 x 300	50	16	41	1 menit 48.92 detik
4	300 x 300	50	23	62	2 menit 46.59 detik

Tabel 7. Tabel Hasil Pengujian Waktu Selection Morph

Pengujian ke-	Ukuran gambar	Jumlah Frame	Jumlah Site Point	Jumlah Feature	Waktu Proses
1	175 x 136	50	18	47	09.45 detik
2	200 x 200	30	18	47	09.89 detik
3	224 x 150	50	24	65	20.87 detik
4	300 x 300	50	19	50	35.04 detik

#### 4. Kesimpulan

Dari keseluruhan sistem aplikasi yang telah dibuat dengan menggunakan beberapa algoritma, dapat disimpulkan beberapa hal dibawah ini:

- Teknik *morphing* dengan *cross dissolve* adalah teknik yang paling sederhana, waktu proses cepat, tetapi tidak memperhatikan hubungan antar objek yang sama dari gambar asal dan gambar tujuan. Hal ini mengakibatkan animasi yang dihasilkan menjadi janggal. Seperti munculnya dua pasang mata pada saat animasi.
- Teknik *morphing* dengan *feature morph* jauh lebih bagus dari teknik *cross dissolve*, karena dengan teknik ini objek yang sama antara kedua gambar diperhitungkan dalam proses *morphing*. Teknik ini masih memiliki kelemahan pada pemilihan *feature*. Pemilihan sebuah *feature* dengan garis kurang begitu efisien. Radius pengaruh sebuah garis *feature* terhadap suatu titik tidak dapat dikontrol oleh user. Kelemahan kedua ialah waktu proses yang dibutuhkan untuk melakukan *morphing* sangatlah lama.
- Teknik *morphing* dengan *mesh morphing* memiliki kelebihan di waktu proses jika dibandingkan dengan *feature morph*, karena pada *mesh morphing* setiap titik pada gambar hanya dipengaruhi oleh tiga garis *feature* yang mengelilinginya pada saat proses deformasi. Kualitas animasi yang dihasilkan dengan cenderung lebih baik dari pada *feature morph*. Kelemahan dari teknik *mesh morph* dibandingkan dengan *feature morph* yaitu deformasi dilakukan lokal di dalam setiap segitiga. Sebuah area yang berada dalam dua buah segitiga akan dideformasikan lebih kasar daripada teknik *feature morph*.
- Teknik *morphing* dengan *selection morph* memungkinkan untuk melakukan proses *morph* hanya pada bagian-bagian tertentu dari gambar. Teknik ini juga memungkinkan untuk mengedit suatu gambar menjadi gambar yang dikehendaki. Sebenarnya pada teknik ini yang dilakukan hanyalah proses deformasi seperti yang terjadi pada *mesh morph*.
- Ketepatan gambar pada animasi yang dihasilkan dari suatu proses *morph* sangat bergantung pada input dari user. Khususnya untuk teknik *feature*, *mesh* dan *selection morph*. Jika peletakkan garis

*feature* atau titik site point tidak tepat maka animasi yang dihasilkan pun akan terkesan janggal dan aneh.

## 5. Daftar Pustaka

- [1] Borland. Borland Delphi 6.0 Help File. 2001.
- [2] Antony Pranata. *Pemrograman Borland Delphi Edisi 3*. Yogyakarta: Andi. 2001.
- [3] Agustinus Nalwan. *Seri Aplikasi Pemrograman Movie & Special Effect*. Jakarta: PT Elex Media Komputiondo. 2001.
- [4] Thaddeus Beier. "Feature-Based Image Metamorphosis". 1992.  
<<http://www.hammerhead.com/thad/thad.html>>
- [5] Image Morphing. <<http://faculty.petra.ac.id/resmana/private/pod/projects/morphing/>>
- [6] Toronto University. "Today: Texture Mapping & Morphing". 14 November 2002.  
<<http://www.cs.toronto.edu/~kyros/courses/320/Lectures/lecture-17.pdf>>.
- [7] (6 June 2003). 2D Image Morphing Algorithms.  
<http://davis.wpi.edu/~matt/courses/morph/2d.htm>
- [8] Renate Schaaf. Components, "Utilities, Sample Applications and Info for Borland Delphi".  
<<http://www.xydot.com/delphi/media.htm>>