

# IMPLEMENTASI TEKNIK MESH MORPHING DAN SELECTION MORPHING PADA CITRA DIGITAL DENGAN DELPHI 7.0

Okie Dhian Pribadi, Eko Adi Sarwoko, dan Aris Sugiharto  
Fakultas Sains dan Matematika, Jurusan Matematika, Universitas Diponegoro

[oki.dhian.pribadi@gmail.com](mailto:oki.dhian.pribadi@gmail.com)

**ABSTRAK:** Dalam pembuatan sebuah film, ada banyak jenis spesial efek yang ditambahkan ke dalam sebuah film untuk menghasilkan suatu film yang berkualitas. Salah satu spesial efek yang sering digunakan adalah *morphing*, yaitu suatu efek dimana suatu obyek diubah perlahan-lahan menjadi obyek lain. Sebelum menggunakan 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 teknik *mesh morphing* dan *selection morphing*. Aplikasi yang dibuat diharapkan dapat menambah alternatif cara dalam pembuatan efek *morphing*, sehingga dapat menciptakan animasi yang lebih efektif dan efisien. Hasil pengujian aplikasi ini menunjukkan bahwa pada teknik *mesh morphing* sangat memperhitungkan *site point-site point* pada gambar asal dan gambar tujuan, karena akan mempengaruhi kualitas animasi yang dihasilkan. Halus tidaknya animasi bergantung pada peletakan *site point*. Jika hanya mempunyai 1 buah gambar, atau terlalu sulit memperoleh gambar tujuan yang sesuai, maka dapat menggunakan teknik *selection morphing* dalam pembuatan animasinya.

**Kata Kunci:** *Image Morphing, Mesh Morphing, Selection Morphing*, Pengolahan Citra Digital, Transformasi Bentuk.

## I. PENDAHULUAN

Salah satu *image processing* yang saat ini sedang populer dan banyak digunakan dalam dunia perfilman adalah *image morphing*, yaitu suatu efek yang mengubah suatu obyek secara perlahan-lahan menjadi obyek lain. Pada dasarnya *morphing* dilakukan dengan membuat gambar-gambar transisi diantara gambar asal dan gambar tujuan.

Secara tradisional efek *morphing* dilakukan dengan tidak efisien. Cara tradisional yang digunakan sangatlah sulit, dan membutuhkan waktu yang lama dan hasil yang dihasilkan belum tentu memuaskan. Untuk mengatasi hal tersebut dan dengan berkembangnya *image processing*, maka dengan komputer akan

membantu memecahkan masalah itu dengan beberapa teknik *morphing* yang ada. Di antaranya adalah *mesh morph* dan *selection morph* [1].

## II. HASIL DAN PEMBAHASAN

### 2.1. *Mesh Morphing*

Teknik ini sering juga disebut *Triangle Based Morphing*. Teknik ini mirip dengan *feature morphing*, tetapi memiliki waktu pemrosesan yang lebih cepat dan pemilihan garis *feature* yang lebih bagus [3]. Pada teknik ini menggunakan garis *feature* yang membentuk kurva tertutup berbentuk segitiga. Sama seperti pada *feature morphing*, dimana garis *feature* asal berpasangan dengan garis *feature* 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 inilah yang akan diinterpolasikan. Seiring dengan interpolasi ketiga garis 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 dideformasikan 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.

Jadi secara garis besar, langkah-langkah yang diperlukan untuk melakukan teknik *mesh morphing* ini adalah sebagai berikut:

- 1) Menginputkan *site point* gambar asal dan gambar tujuan.
- 2) Menggambar garis-garis *feature* yang membentuk kurva tertutup dan berbentuk segitiga dengan menggunakan teknik *delaunay triangulation*.
- 3) Melakukan deformasi gambar asal dan gambar tujuan berdasarkan garis-garis *feature* yang telah terbentuk.

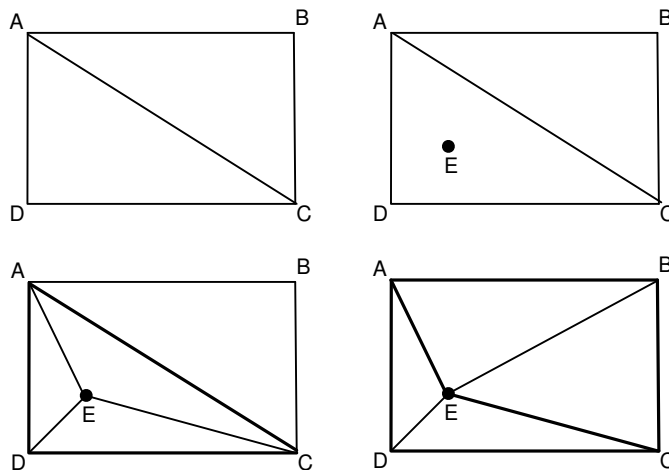
- 4) Melakukan proses *cross dissolve* terhadap kedua gambar hasil deformasi tadi.

## 2.2. *Delaunay Triangulation*

Syarat sebuah *triangle network* disebut *delaunay triangle network* adalah jika segitiga-segitiga yang dihasilkan adalah segitiga yang hampir mendekati segitiga sama sisi [5].

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

Salah satu algoritma untuk merekonstruksi *delaunay triangle network* adalah *Insertion Algorithm*. Secara garis besar, langkah-langkah dari *Insert Algorithm* dapat dilihat pada gambar 2.1.



**Gambar 2.1.** Langkah-Langkah dari *Insert Algorithm* [5].

Penjelasan *Insert Algorithm* pada gambar 2.1. adalah sebagai berikut. Pertama-tama dimulai dengan meletakkan dua buah segitiga di layar. Kemudian *site point* dimasukkan satu per satu. Setiap kali *site point* dimasukkan ke dalam *network*, misalnya *site point* ini adalah titik E, maka dilakukan proses sebagai berikut :

- 1) Cari segitiga dimana *site point* itu berada, yaitu di segitiga ACD.

- 2) Selanjutnya segitiga ACD dipecah menjadi AED, DEC, dan AEC.
- 3) Kemudian 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 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 di atas, sisi yang dimasukkan adalah AD, DC, CA (digambar dengan garis tebal). Sisi AD tidak mewakili 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*.
- 4) Setelah sisi CA dimasukkan ke dalam *array temp*, kemudian 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 + sudut AEC lebih besar dari sudut EAB + sudut ECB. Jadi diagonal EB lebih bagus dari pada diagonal CA, dan diagonal ACEB perlu ditukar. Pertukaran diagonal tersebut meliputi proses sebagai berikut :
  - a. Hapus segitiga AEC dan ABC.
  - b. Ciptakan segitiga AEB dan EBC.
  - c. Masukkan sisi-sisi luar segitiga AEB dan EBC ini kedalam *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*.
- 5) Selanjutnya karena sisi CA telah selesai diperiksa maka hapus sisi ini dari *array temp*.

- 6) Berikutnya ambil sisi selanjutnya yang akan diproses dalam *array temp*.
- 7) Proses ini akan terus dilakukan sampai sisi dalam *array temp* telah habis. Setelah *array temp* kosong, ulangi lagi seluruh proses di atas dengan memasukkan *site point* yang baru.

### **2.3. Selection Morphing**

Teknik *selection morphing* yaitu proses *morphing* yang dilakukan hanya pada bagian-bagian tertentu saja [3]. Misalnya membuat film dengan peran utamanya adalah seekor kucing. Kemudian seumpama kucing ini disuruh melakukan suatu adegan menangis, senyum, atau mengedipkan sebelah mata pada saat-saat tertentu. Merupakan hal yang mustahil untuk melatih kucing supaya melakukan adegan-adegan di atas. Dengan menggunakan teknik ini maka dapat diciptakan animasi atau gambar dari kucing yang sedang mengedipkan sebelah mata hanya dengan bermodalkan sebuah foto kucing tersebut. Pada teknik ini proses yang dilakukan hanyalah melakukan deformasi pada gambar yang di-*input*-kan.

Pada teknik *feature morphing* dan *mesh morphing*, langkah pertama yang dilakukan adalah proses deformasi dari gambar asal ke *frame* saat ini, juga deformasi dari gambar tujuan ke *frame* saat ini. Baru langkah selanjutnya adalah *cross dissolve* antara kedua gambar hasil deformasi. Jika *frame* saat ini adalah *frame* terakhir, maka gambar asal akan dideformasikan hampir menyerupai gambar kedua. Jika proses *morphing* tersebut dilakukan antar gambar kucing dengan wajah normal ke gambar tujuan kucing mengedipkan sebelah matanya, maka gambar asal yang dideformasikan tadi telah menyerupai kucing yang mengedipkan sebelah matanya, karena sebelah matanya ditarik hingga menuju ke pose mengedipkan sebelah mata.

Jadi dengan memanfaatkan deformasi gambar asal tanpa *cross dissolve* dengan deformasi gambar akhir sudah diperoleh animasi dari kucing yang perlahan-lahan akan mengedipkan sebelah matanya. Teknik deformasi yang digunakan di sini sama dengan teknik deformasi pada *mesh morphing*.

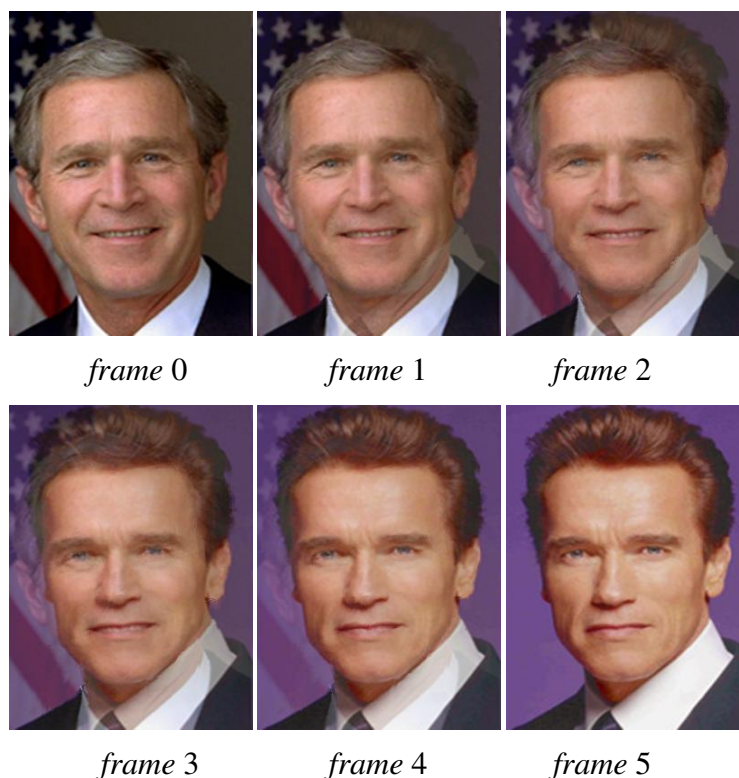
## 2.4. Hasil Pengujian

Pengujian untuk program *image morphing* ini dilakukan untuk membuktikan bahwa aplikasi Tugas Akhir ini berguna dalam pembuatan suatu animasi atau film dan juga untuk memperlihatkan setiap proses yang dilakukan dalam membuat suatu *image morphing*.

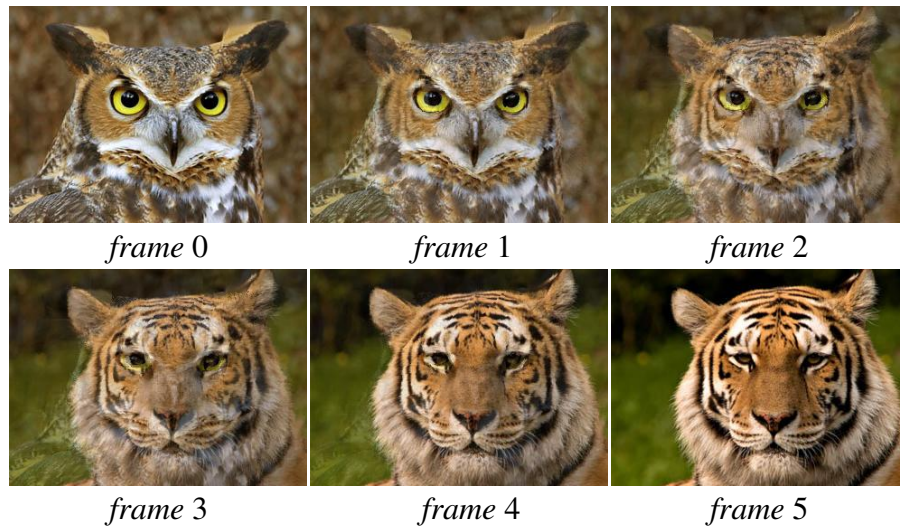
Pengujian yang akan dilakukan pada program ini adalah menilai ketepatan gambar yang dihasilkan oleh proses *morphing*. Pengujian ini dilakukan dengan metode pengukuran secara kualitatif (*visual inspection*). Metode pengukuran secara kualitatif ini dilakukan karena tidak ada perhitungan yang dapat dilakukan untuk mengukur ketepatan suatu gambar. Ketepatan gambar yang dimaksud adalah tidak adanya gambar-gambar aneh atau gambar yang tidak diinginkan pada *frame-frame* tertentu pada animasi yang dihasilkan dari suatu proses *morphing*.

### 2.4.1. Mesh Morphing

Pada pengujian di-*input*-kan jumlah *frame* sebanyak 5 buah. Hasil subyektif kinerja *mesh morphing* yaitu dengan mengamati gambar hasil *morphing* secara visual yang diperlihatkan pada gambar 2.2. dan gambar 2.3.



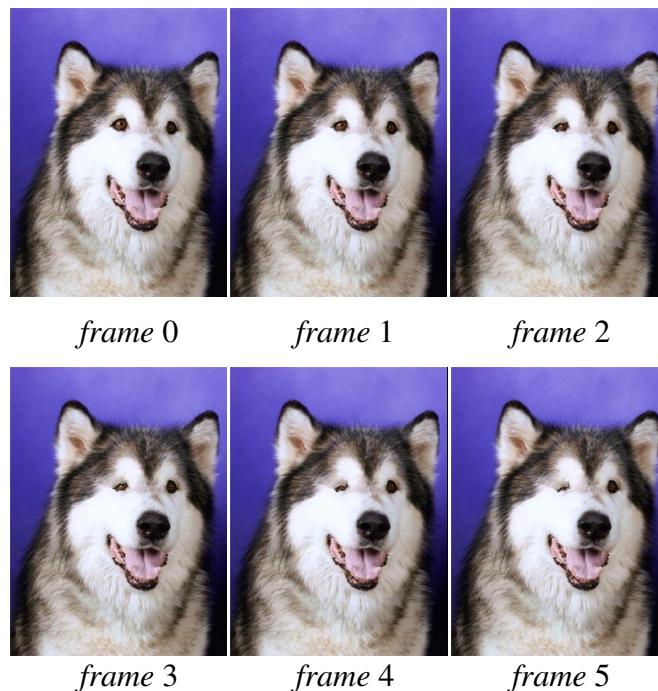
**Gambar 2.2.** Hasil uji *mesh morphing* dengan gambar berukuran 270 x 360.



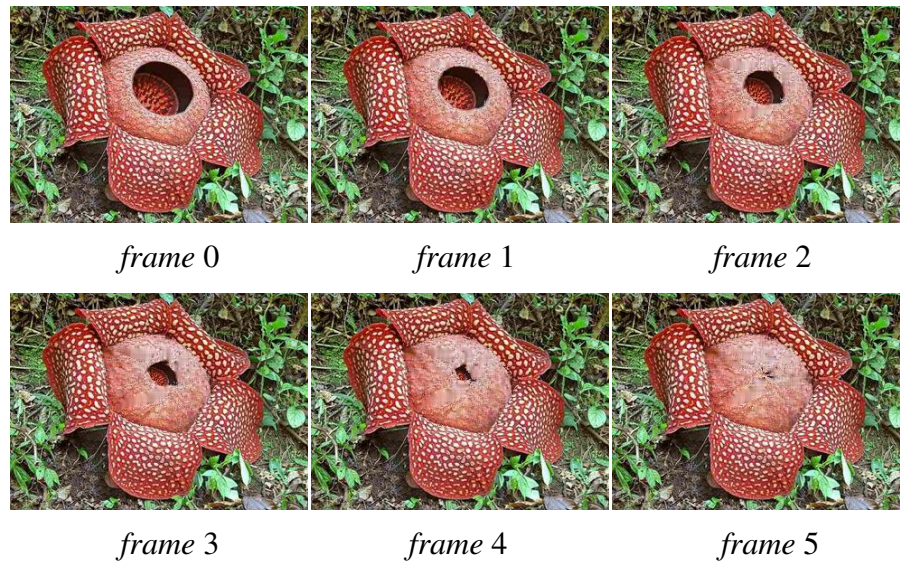
**Gambar 2.3.** Hasil uji *mesh morphing* dengan gambar berukuran 330 x 240.

#### 2.4.2. *Selection Morphing*

Pada *selection morphing* ini cukup menggunakan 1 gambar saja, dikarenakan gambar tujuan sama dengan gambar asal. Pengujian di-*input*-kan jumlah *frame* sebanyak 5 buah. Hasil subyektif kinerja *selection morphing* yaitu dengan mengamati gambar hasil *morphing* secara visual yang diperlihatkan pada gambar 2.4. dan gambar 2.5.



**Gambar 2.4.** Hasil uji *selection morphing* dengan gambar berukuran 270 x 360.



**Gambar 2.5.** Hasil uji *selection morphing* dengan gambar berukuran 330 x 240.

### III. KESIMPULAN

Setelah melakukan pengujian pada aplikasi yang sudah dibuat dengan menggunakan bahasa pemrograman *Delphi 7.0*, dapat disimpulkan bahwa :

1. Pada pengujian *morphing* dengan teknik *mesh morphing*, terlihat bahwa teknik ini sangat memperhitungkan *site point-site point* pada gambar asal dan tujuan.
2. Baik itu pada *mesh morphing* maupun *selection morphing*, *user* harus pintar dalam meletakkan *site point-site point*, sehingga animasi *morphing* yang dihasilkan terlihat halus.
3. Jika hanya mempunyai 1 buah gambar, atau terlalu susah untuk mendapatkan gambar tujuan yang sesuai, maka lebih baik menggunakan teknik *selection morphing* dalam pembuatan animasi *morphing*-nya.

### IV. UCAPAN TERIMA KASIH

Pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Drs. Solichin Zaki, M.Kom selaku Ketua Jurusan Matematika FSM UNDIP.



2. Bambang Irawanto, S.Si, M.Si selaku Ketua Program Studi Matematika Jurusan Matematika FSM UNDIP.
3. Drs. Eko Adi Sarwoko, M.Kom dan Aris Sugiharto, S.Si, M.Kom selaku dosen pembimbing I dan pembimbing II yang telah meluangkan waktu memberikan bimbingannya hingga selesainya Tugas Akhir ini.
4. Triastuti Wuryandari, S.Si, M.Si selaku dosen wali yang telah mengarahkan dan mendampingi penulis selama menjadi mahasiswa Jurusan Matematika FSM UNDIP.
5. Semua pihak yang telah memberikan dukungan serta bantuan kepada penulis dalam menyelesaikan penyusunan Tugas Akhir ini.

## V. DAFTAR PUSTAKA

- [1] Abednego, Luciana dan Saputro, Nico. 2004. *Implementasi Teknik Feature Morphing Pada Citra Dua Dimensi*.  
<http://home.unpar.ac.id/~integral/Volume%209/Integral%209%20No.%201/Morphing%20Dua%20Dimensi.pdf>. Diakses terakhir pada tanggal 24 Agustus 2012.
- [2] Fadlisyah, Taufiq, Zulfikar, Fauzan. 2008. *Pengolahan Citra Menggunakan Delphi*. Yogyakarta: Graha Ilmu.
- [3] Gunadi, Kartika dkk. 2004. *Morphing Citra dengan Berbagai Teknik Morphing*, Universitas Gunadarma Jakarta.  
<http://ejournal.gunadarma.ac.id/files/B04.pdf>. Diakses terakhir pada tanggal 16 Juli 2012.
- [4] Munir, Rinaldi. 2004. *PENGOLAHAN CITRA DIGITAL dengan Pendekatan Algoritmik*. Bandung: Informatika.
- [5] Nalwan, Agustinus. 2001. *Seri Aplikasi Pemrograman Movie dan Special Effect 2*. Jakarta: Elex Media Komputindo.
- [6] Wahana Komputer. 2003. *Panduan Praktis Pemrograman Borland Delphi 7.0*. Yogyakarta: Andi.