PERANCANGAN APLIKASI *IMAGE MORPHING* PADA CITRA DUA DIMENSI MENGGUNAKAN TEKNIK *FIELD MORPHING* DAN *DEFORMATION TECHNIQUES*

Abdurahman Tohari
10103153

Otoy_puak@plasa.com

Pembimbing I : Santi Novani, S.T.,M.T
Pembimbing II : Irfan Maliki, S.T

Fakultas Teknik dan Ilmu Komputer
Jurusan Teknik Informatika
Universitas Komputer Indonesia

Abstrak

Dalam pembuatan sebuah film, digunakan beberapa spesial efek untuk menghasilkan sebuah film yang lebih bagus. Salah satu spesial efek yang sering digunakan adalah morphing, yaitu suatu efek dimana suatu obyek diubah perlahan-lahan menjadi obyek lain. Untuk melakukan morphing, dapat digunakan berbagai metode. Metode yang paling sederhana adalah metode *cross dissolve* yaitu hanya menggunakan tingkat transparansi. Selain itu, ada metode *image morphing* yang hasilnya jauh lebih baik daripada *cross dissolve*, diantaranya adalah metode *Field Morphing* dan *Deformation Techniques*.

Tugas Akhir ini membahas tentang pembuatan efek morphing pada *image* dan video dengan menggunakan metode *field morphing* dan *deformation techniques*. Tujuan tugas akhir ini adalah untuk menunjukkan kelebihan dan kekurangan dari metode *Field Morphing* dan *Deformation Techniques*. Aplikasi dibuat dengan menggunakan bahasa pemrograman Borland delphi 7.0.

Hasil pengujian aplikasi ini, menunjukkan bahwa teknik *deformation techniques* membutuhkan waktu proses yang lebih sedikit daripada *field morphing*, tetapi hasil yang dihasilkan oleh metode *field morphing* lebih bagus dibandingkan dengan *deformation techniques*.

Kata kunci: Image morphing, field morphing, deformation techniques

1. PENDAHULUAN

1.1. Latar Belakang Masalah

Dengan berkembangnya *image processing,* manusia berlomba-lomba untuk menghasilkan sebuah *image* yang menarik dari *image-image* sederhana yang ada. Selain itu, sesuai dengan perkembangan waktu, *Image Processing* semakin menjadi kebutuhan oleh banyak orang terutama di bidang seni. Salah satu *image processing* yang saat ini sedang populer dan banyak di gunakan di dunia perfilman adalah *image morphing.* Proses *morphing* merupakan proses untuk membuat suatu efek dimana suatu obyek akan diubah perlahan-lahan menjadi obyek lain. Pada dasarnya *morphing* dilakukan dengan membuat gambar-gambar transisi di antara gambar asal dan gambar tujuan.

Secara tradisional efek *morphing* dilakukan dengan tidak efisien. Misalnya apabila ingin menghasilkan animasi perubahan dari wajah macan menjadi wajah serigala, maka digunakan gambar yang menunjukkan animasi perubahan yang terjadi diantara kedua gambar. Proses animasi yang terjadi digambar dalam banyak gambar sesuai dengan banyaknya *frame* yang diinginkan. Tetapi proses ini sangat sulit diterapkan dan juga hasil yang didapat jelek, karena gambar macan yang pertama digunakan adalah gambar asli dan pada saat diinginkan terjadinya animasi maka gambar asli itu akan digantikan dengan gambar buatan yang telah disiapkan.

Cara lain yang digunakan adalah dengan menampilkan foto macan dan foto serigala secara bergantian dan cepat. Karena kecepatan yang sangat tinggi, mata manusia secara tidak jelas atau kabur akan melihat sebagian dari wajah macan seperti wajah serigala. Hal ini terjadi karena mata manusia membutuhkan sekitar 1/30 detik untuk menangkap satu gambar. Tetapi cara ini membuat sakit mata karena hasil yang dihasilkan oleh setiap mata karena kecepatan ini berbeda-beda.

Cara-cara tradisional yang digunakan dan yang dicontohkan diatas sangatlah sulit, dan membutuhkan waktu dan tenaga yang lama serta

biaya yang tidak sedikit tentunya meskipun hasil yang dihasilkan belum tentu memuaskan. Untuk mengatasi hal tersebut, dengan berkembangnya *image processing* maka komputer akan membantu memecahkan masalah itu dengan beberapa teknik *morphing* yang ada diantaranya yaitu teknik *field morphing*, kelebihan dari teknik ini adalah memiliki garis feature dimana dengan garis ini komputer bisa dengan mudah mengetahui hubungan antara dua objek yang sama pada *image source* dan *image destination* yang terdapat pada image yang akan diporses, dan juga hasil dari metode ini juga terlihat lebih smooth (halus), sedangkan teknik *deformation image* memiliki kelebihan yaitu memakai metode *energy minimization* yang bisa mempercepat proses *morphing* tetapi kedua metode diatas sangat membutuhkan bantuan berupa garis feature yang dimasukan kedalam image yang akan diproses. Sehingga dalam hal ini, penulis mengambil judul **"PERANCANGAN APLIKASI IMAGE MORPHING PADA CITRA DUA DIMENSI MENGGUNAKAN *METODE FIELD MORPHING* DAN *DEFORMATION TECHNIQUES* "**

1.2. Identifikasi Masalah

Dari uraian latar belakang di atas dapat diidentifikasikan bahwa permasalahan adalah bagaimana merancang aplikasi animasi efek pada citra dua dimensi dengan menggunakan metode *image morphing*

1.3 Maksud dan Tujuan

1.3.1 Maksud

Maksud dari penulisan tugas akhir ini adalah untuk membuat program aplikasi image morphing pada citra dua dimensi.

1.3.2 Tujuan

Tujuan yang ingin dicapai dari pembuatan aplikasi ini adalah:

1. Untuk menggabungkan dua buah gambar dalam proses image morphing.
2. Untuk menginterpolasi citra dua dimensi agar hasil terlihat lebih halus
3. Untuk menghasilkan suatu animasi *image morphing* dengan menggunakan metode *field morphing dan deformation techieques.*

1.3. Batasan Masalah

Dalam pembangunan aplikasi ini dibatasi beberapa hal yaitu :

1. Morphing yang dilakukan hanya akan sebatas pada citra dua dimensi.
2. Format *image* yang dipakai adalah Bitmap (.bmp) dan JPEG.
3. Dua image yang akan digunakan dalam proses image morphing tidak harus menggunakan format image yang sama dan ukuran yang sama.
4. Metode yang dipakai adalah metode field morphing dan deformation techniques.
5. Hasil dari aplikasi ini dapat disimpan dalam format AVI.
6. Desain aplikasi program ini menggunakan Borland Delphi 7.
7. *Format Video* yang dipakai adalah AVI.
8. Jumlah frame yang digunakan adalah: 15, 30, 60, 90, 120 dan 150.

2. LANDASAN TEORI

2.1. Grafika Komputer 2 Dimensi

Grafika Komputer 2 Dimensi merupakan gambar digital yang dihasilkan oleh komputer, khususnya yang berbentuk dua dimensi karena hanya mempunyai sumbu koordinat X dan Y saja. Gambaran grafika komputer 2D kira-kira adalah seperti sebuah kanvas kosong (pada komputer dianggap sebagai *frame buffer* yang berupa kumpulan *pixels*) berwarna putih (sebagai latar belakang warna, pada komputer latar belakang yang biasanya dipakai adalah berwarna hitam). Kanvas tersebut selanjutnya akan diberi gambar maupun warna sesuai dengan keinginan. Pada komputer, proses penggambaran dan pewarnaan sama dengan mengisi dengan bilangan tertentu sebagai petunjuk warna yang dipakai.

Grafika komputer 2 Dimensi dapat berupa grafik vektor, grafik raster maupun kombinasi antara keduanya. Grafik raster adalah penyimpanan gambar 2D pada komputer yang dilakukan secara apa adanya dalam bentuk *array of pixels* dan sering kali disebut sebagai file bitmap. Sedangkan garfik vektor merupakan proses penyimpanan gambar 2 Dimensi pada komputer yang dilakukan secara vektor, dimana data yang disimpan berupa posisi titik-titik yang kemudian dalam proses penggambarannya dihubungkan dengan garis, ataupun bangun sehingga membentuk sebuah gambar.

2.1.1. *Image Processing*

Image Processing adalah sebuah metode yang dipakai untuk memproses atau memanipulasi image dalam bentuk dua dimensi. Segala operasi untuk mempebaiki, menganalisa, atau mengubah suatu gambar disebut *image processing* (A. Usman, 2005, Pengolahan Citra Digital dan Teknik pemogramannya). Konsep dasar pemprosesan suatu objek yang menggunakan *image processing* diambil dari kemampuan indera penglihatan manusia yang selanjutnya dihubungkan dengan kemampuan otak manusia. Dalam sejarahnya, *image*

processing telah diaplikasikan dalam berbagai bentuk, dengan tingkat kesuksesan yang cukup besar. Seperti berbagai cabang ilmu lainnya, *image processing* pula gabungan berbagai cabang-cabang ilmu, diantaranya adalah optik, elektronik, matematika, fotografi, dan teknologi komputer. Beberapa faktor yang menyebabkan sistem *image processing* menjadi berkembang pesat pada saat ini, salah satu yang utama adalah dibutuhkannya suatu teknologi yang dapat bekerja secara mandiri, dalam artian teknologi tersebut dapat mengambil keputusan sendiri dari hasil pengolahan data sebelumnya. Selain itu juga tersedianya peralatan yang lebih *modern* untuk proses tampilan gambar yang menjadikan *image processing* berkembang pesat.

Pada umumnya, objektifitas dari *image processing* adalah mentransformasikan atau menganalisis suatu gambar sehingga informasi baru tentang gambar dapat dibuat dengan lebih jelas.

2.1.2 Sistem Warna RGB

Semua warna yang ada merupakan perpaduan dari 3(tiga) macam warna dasar yaitu: Merah(Red), Hijau(Green), Biru(Blue). Perpaduan ketiga warna dasar ini dipakai pada sistem warna RGB, apabila ketiga warna dasar dicampur, maka akan dihasilkan suatu warna tertentu, tergantung dari komposisi ketiga warna dasar tersebut.

Gambar pada sistem digital dapat diwakili denag format RGB untuk setiap titiknya, dimana setiap komponen warna diwakili dengan satu *byte.* Jadi untuk masing-masing komponen R, G, dan B mempunyai variasi dari 0 sampai 255. Total variasi yang dapat dihasilkan untuk sistem warna *digital* ini adalah 256 x 256 x 256 atau 16.777.216 jenis warna. Karena setiap komponen warna dieakili dengan satu *byte* atau delapan bit, maka total bit yang digunakan untuk merepresentasikan warna RGB adalah 8+8+8 atau 24 bit.

2.2 Teknik Morphing

Teknik morphing ini dimulai dengan cara-cara yang masih tradisional dan tidak memerlukan bantuan teknologi untuk membuatnya, tetapi hal itu dirasa belum cukup efektif dan sangat membutuhkan waktu yang relatif sangat banyak untuk menerapkannya. Selain membutuhkan waktu yang lama dalam proses pembuatannya, hasilnya juga masih tidak memuaskan.

Dengan bantuan komputer, efek morphing yang dihasilkan menjadi lebih bagus, dan waktu yang dibutuhkan relatif lebih cepat. Sejak tahun 1980an mulai banyak film-film yang menggunakan special effect dengan bantuan komputer. Contoh video yang menggunakan teknik image morphing adalah video klip Michael Jackson yang berjudul Black or White dan Film yang berjudul Willow dan Indiana Jones and Last Srusade. Teknik morphing menggunakan komputer, pada dasarnya tidak berbeda dengan teknik tradisional yang telah ada sebelumnya dengan cara menggambar. Hanya saja, gambar yang dihasilkan oleh komputer lebih realistik sehingga dapat memberikan kesan bahwa gambar tersebut nyata.

Pada dasarnya pembuatan efek morphing dilakukan dengan membuat gambar-gambar peralihan diantara gambar asal dengan gambar tujuan berdasarkan kedua gambar tersebut. Dengan adanya gambar-gambar peralihan yang baru ini, maka perubahan yang terjadi dari gambar asal ke gambar tujuan akan terlihat lebih halus. Gambar-gambar peralihan inilah yang dibuat dengan menggunakan bantuan komputer. Dalam sebuah gambar teridiri dari banyak titik yang mempunyai nilai berbeda yang biasa disebut dengan pixel. Dalam proses pembuatan gambar-gambar peralihan oleh komputer titik-titik inilah yang akan mengalami perubahan. Perubahan yang terjadi dapat meliputi perubaha warna dan posisinya. Dengan adanya perubahan posisi dari titik-titik pada suatu gambar, maka akan mengakibatkan hilangnya beberapa titik pada gambar, sehingga gambar yang terbentuk nantinya akan terkesan berlubang, untuk mengatasi hal itu digunakan teknik interpolasi. Interpolasi ini digunakan untuk mengisi titik-titik yang hilang pada gambar dengan berpatokan pada titik-titik yang terisi yang ada di sekitarnya.

Ada beberapa algoritma yang dapat digunakan untuk membuat teknik morphing, di antaranya adalah cross dissolve, field morphing, mesh morphing, liquid morphing, dan deformation technique. Dimana masing-masing algoritma tersebut memiliki kelebihan dan kekurangan dalam hal waktu proses maupun morphing yang dihasilkan. Pada pembuatan efek morphing, terdapat suatu proses yang disebut dengan deformasi. deformasi merupakan suatu teknik yang digunakan untuk mengubah bentuk obyek pada gambar asal menjadi obyek peralihan yang menyerupai obyek pada gambar tujuan. Teknik deformasi inilah yang membedakan antara algoritma yang satu dengan algoritma yang lainnya.

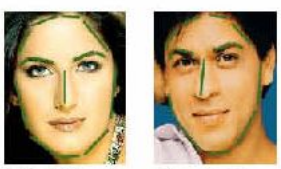
2.3. Proses Cross Dissolve

Proses cross dissolve dibutuhkan untuk memperhalus perubahan gambar
yang terjadi. Cross dissolve ini merupakan tenik image morphing yang paling sederhana dan mudah. Teknik cross dissolve ini merupakan dasar dari teknik-teknik morphing yang lain. Pada teknik cross dissolve yang dilakukan hanya menampilkan kedua gambar (gambar asal dan gambar tujuan) dengan mode transparan. Tingkat transparansi dari setiap gambar diatur dengan maksimum 100%, dan minimum 0%. Pada tingkat transparansi 100%, maka yang terlihat hanyalah gambar asal, sedangkan pada tingkat transparansi 0% yang terlihat hanyalah gambar tujuan. Selama jumlah frame, tingkat transparansi akan semakin berkurang dengan syarat jumlah tingkat transparansi gambar asal dijumlahkan dengan gambar tujuan adalah 100%. Dengan demikian akan terlihat gambar asal perlahan-lahan menghilang sambil gambar tujuan terlihat perlahan-lahan muncul.

2.4. Field Morphing

Field morphing merupakan salah satu teknik dalam image morphing yang menggunakan bantuan garis feature untuk setiap perubahan yang terjadi, sehingga animasi yang terjadi dengan teknik field morphing ini harus menerima masukan berupa dua buah gambar. Gambar pertama disebut sebagai image source dan gambar terakhir disebut sebagai image destination. Kegunaan dari garis Feature adalah untuk menunjukkan sebuah feature (bagian-bagian gambar yang membentuk obyek) yang sama antara daerah di gambar asal dengan daerah di gambar tujuan. Dengan adanya garis feature ini membuat komputer mengetahui hubungan antara dua obyek yang sama pada image source danimage destination. Sebuah garis feature di image source memiliki pasangannya di image destination.

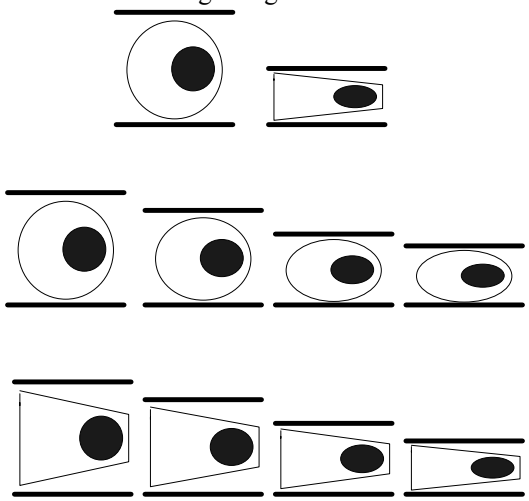
Dengan adanya garis feature, maka komputer dapat mengetahui hubungan antara dua object yang dianggap sama di gambar asal dan tujuan. Dalam teknik Field Morphing terdapat dua buah langkah. Langkah pertama adalah deformasi image source dan image destination. Langkah kedua adalah cross dissolve antara kedua gambar yang telah dideformasikan.



Gambar 2.16. Garis Feature Image Source dan Image Destination

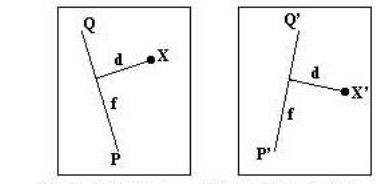
2.4.1. Deformasi Image

Pada gambar 2.16 ditunjukkan garis feature dan hubungannya pada image source dan image destination. Dengan bantuan garis feature tersebut diatas, alis mata dari image source akan diubah menjadi alis mata dari image destination. Perubahan yang ada ditentukan oleh jumlah frame yang ada. Diantara image
source dan image destination akan dilakukan interpolasi untuk mengisi animasi
frame yang dibutuhkan. Parameter yang digunakan untuk interpolasi ini adalah
frame saat ini dibagi dengan total frame.



Gambar 2.17. Deformasi Image Source dan Image Destination
Sumber: Agustinus Nalwan. (2001). Seri Aplikasi Pemrograman Movie & Special Effect2.

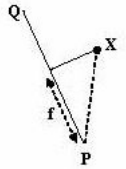
Pada gambar 2.17. ditunjukkan image source dan image destination yang diinginkan. Proses deformasi tidak hanya terjadi pada image source, melainkan juga pada image destination. Pada contoh gambar 2.17. image source diubah menjadi image destination dalam 4 frame. Perubahan itu diubah berdasarkan garis feature yang diberikan. Dari hasil keduanya, baru dilakukan proses cross dissolve untuk memperhalus perubahan yang terjadi.



Gambar 2.18. Deformasi Titik Terhadap Garis *Feature*

Dengan menggunakan teknik *field morphing,* perubahan pada setiap titik

harus dilakukan pencarian jarak terdekat antara titik dengan garis *feature*. Hal ini dapat dilakukan dengan mencari proyeksi vektor antara vektor PQ dan vektor PX.



Gambar 2.19. Pencarian proyeksi vektor PX terhadap PQ

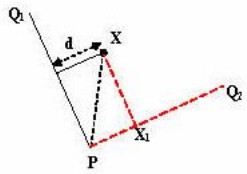
Pencarian proyeksi vektor PX terhadap titik PQ agar diddapat nilai dari f. Untuk mencari nilai dari f digunakan rumus :

$$f = \frac{\overrightarrow{PX} * .\overrightarrow{PQ}}{\left|\overrightarrow{PQ}\right|} \quad (2.1)$$

Dalam perhitungan nilai f yang kita butuhkan kisaran antara 0 dan1, yaitu dengan cara dilakukan pembagian dengan panjang vektor PQ.

$$f = \frac{\overrightarrow{PX} * \overrightarrow{PQ}}{\left|\overrightarrow{PQ}\right|^2} \quad (2.2)$$

Setelah didapat nilai f langkah selanjutnya mencari nilai d yang merupakan jarak tegak lurus antara titik dengan garis *feature*.



Gambar 2.20. Pencarian jarak d

Dengan vektor tegak lurus PQ maka nilai d didapat dengan cara mencari vektor proyeksi PX terhadap PQ.

$$PX_1 = \frac{\overrightarrow{PX} * \overrightarrow{PQ_2}}{\left|PQ\right|} \quad (2.3)$$

PQ_2 harus mempunyai panjang yang sama dengan PQ_1 agar nilai PX_1 sama dengan panjang d. Persamaan f dan d akan mentranslasikan koordinat x dan y ke bentuk koordinat baru, yaitu koordinat d dan f. Koordinat f dalam sistem koordinat baru dapat diidentifikasikan sebagai berikut :

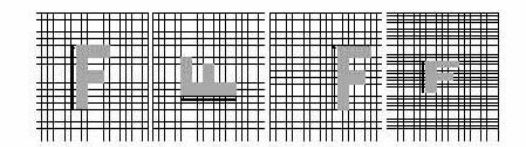
$$f \times \overrightarrow{P'Q'} \quad (2.4)$$

Sedangkan vector d dapat diidentifikasikan sebagai berikut :

$$\frac{\overrightarrow{\perp P^1Q^1}}{\overrightarrow{P^1Q^1}} \quad (2.5)$$

Jadi, titik asal X ditransformasikan ke titik tujuan X' dengan rumus :

$$X' = P' + f \times P'Q' + d \times \frac{\overrightarrow{\perp P'Q'}}{\left|\overrightarrow{P'Q'}\right|} \quad (2.6)$$



Gambar 2.21. Pengaruh Satu Garis Feature
Sumber:
http://www.hammerhead.com/thad/morph.html

Apabila garis feature hanya satu, maka X' dapat dicari dengan menggunakan rumus tersebut. Tetapi apabila garis feature yang mempengaruhi sebuah image lebih dari satu maka untuk mendapatkan nilai X' diperlukan pencarian bobot untuk setiap titik terhadap semua garis feature, semakin jauh jarak titik terhadap garis feature maka semakin kecil perubahan terhadap perubahan titik dan sebaliknya. Rumus pencarian bobot untuk setiap titik adalah sebagai berikut :

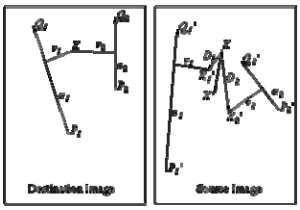
$$W = \left(\frac{1}{1 + dist}\right) \quad (2.7)$$

Dimana dist merupakan jarak terdekat antara titik dengan garis feature. Jarak terdekat yang berarti jika nilai f lebih dari 1 atau kurang dari 0, maka panjang dist tidak akan sama dengan nilai d, dikarenakan nilai d didapat dari perpanjangan dari garis feature. Sehingga apabila nilai $f < 0$ maka jarak terdekat antara titik dengan garis feature adalah jarak dari titik ke titik awal dari garis feature. Dan apabila nilai $f > 1$, maka jarak antara titik dengan garis feature adalah jarak dari titik ke titik akhir dari garis feature.

$$D_1 = X_1' - X \quad (2.8)$$
$$D_2 = X_2' - X$$

Akibat dari perhitungan bobot dan perubahan jarak dari setiap garis feature maka didapat rumus untuk perhitungan koordinat akhir yang dipengaruhi oleh lebih dari satu garis feature sebagai berikut :

$$X' = X + \frac{D_1W_1 + D_2W_2}{W_1 + W_2} \quad (2.9)$$



Gambar 2.22. Titik pada Image Source dan Image Destination yang Dipengaruhi oleh Lebih dari Satu Garis Feature

Rumus umum untuk mencari titik baru yang dipengaruhi oleh lebih dari satu garis feature adalah sebagai berikut :

$$X' = X + \frac{\sum_{i=1}^{n} D_i W_i}{\sum_{i=1}^{n} W_i}$$

(2.10)

2.4.2. Algoritma Field Morphing

Algoritma keseluruhan dari field morphing adalah :

1. Menentukan jumlah frame gambar yang ingin dihasilkan
2. Mengerjakan proses warping sebanyak n kali. Setiap kali, dua proses
3. warping dikerjakan, yaitu warping image source dan warping image destination.

Hasil warping image source dan image destination kemudian mengalami proses cross dissolve untuk setiap frame gambar.

2.5. Deformation Technique

Metode image morphing yang lain adalah deformation techniques. Metode image morphing deformation techniques menggunakan turunan dari C1 – continuous and one to one warp dari setiap titik yang ada pada dua image. Metode deformation techniques lebih sering disebut juga dengan metode energy minimization. Metode deformation technique menggunakan garis feature untuk menunjukkan hubungan antara image source dan image destination.

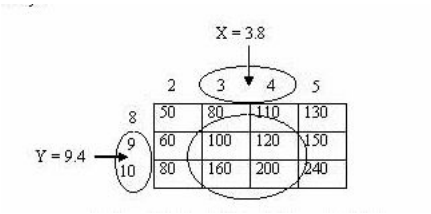
2.6. Tweening

Tweening lebih dikenal dengan teknik interpolasi. Melakukan interpolasi berarti menyisipkan sesuatu di antara dua bagian yang berbeda atau memperkirakan nilai dari suatu fungsi antara dua nilai yang telah diketahui (Sumber : *Morphing Magic* 1 st *edition).*

2.7. Interpolasi Bilinear

Operasi geometrik berhubungan dengan perubahan bentuk geometri citra, yaitu baik ukuran maupun orientasinya. Operasi geometri di antaranya meliputi pencerminan (flipping), rotasi/pemutaran (rotation), penskalaan (scaling/zooming), dan pembengkokan (warping). Operasi geometrik dilakukan dengan cara memetakan koordinat titik-titik citra asal ke koordinat titik-titik di citra hasil. Pemetaan dilakukan dengan menerapkan fungi interpolasi bilinear.

Pada metode interpolasi bilinear, maka titik-titik yang terdekat akan mempengaruhi hasil dari titik yang akan dicari. Makin dekat titik tetangga, maka makin besar bobotnya, dan sebaliknya makin jauh maka akan makin kecil bobotnya.



Gambar 2.25. Contoh Pixel Tetangga terdekat

Dari contoh Gambar 2.25, maka bobot ke arah horisontal untuk koordinat x = 3 dan x = 4 masing-masing adalah :

w(3) = 0.2 dan w(4) = 0.8

(2.11)

Koordinat x = 3.8 lebih dekat ke x = 4 dibandingkan dengan ke x = 3. Sementara bobot ke arah vertikal untuk koordinat y = 9 dan y = 10, masing-masing adalah :

w(9) = 0.6 dan w(10) = 0.4

(2.12)

Dari bobot vertikal dan bobot horisontal yang didapat, maka dapat dihitung bobot keseluruhan suatu titik.

w(3,9) = w(3) × w(9) = 0.2 × 0.6 = 0.12 (2.13)

w(3,10) = w(3) × w(10) = 0.2 × 0.4 = 0.08 (2.14)

w(4,9) = w(4) × w(9) = 0.8 × 0.6 = 0.48 (2.15)

w(4,10) = w(4) C w(10) = 0.8 × 0.4 = 0.32 (2.16)

Terlihat bahwa titik yang paling dekat dengan koordinat titik asal memiliki bobot yang paling besar (0.48), sedangkan titik terjauh memiliki bobot yang paling kecil (0.08). Total bobot untuk keempat titik tersebut adalah 1.

Setelah semua bobot pada keempat titik bertetangga tersebut diperoleh, maka perhitungan untuk mendapatkan sebuah titik hasil adalah :

Kå ´ = ∑w x Ki (2.17)

=w(3,9)xKi(3,9)+w(3,10)xKi(3,10)+w(4,9)xKi (4,9)+w(4,10)xKi(4,10) (2.18)

= 0.12 x100 + 0.08 x 160 + 0.48 x 120 + 0.32 x 200 (2.19)

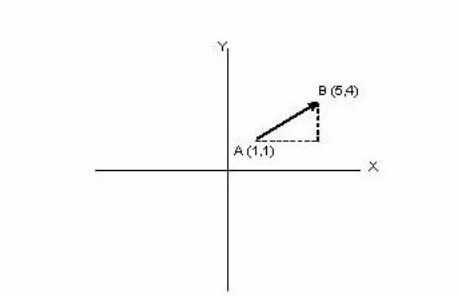
= 146.4 (2.20)

Karena nilai sebuah titik merupakan bilangan bulat, maka apabila hasil penjumlahan berbobot tersebut bilangan pecah, maka harus dilkukan pembulatan ke bilangan bulat terdekat. Kelebihan dari interpolasi bilinear adalah citra

hasilnya terlihat lebih halus (smooth) dibandingkan dengan hasil interpolasi tetangga terdekat, terutama jika dilakukan pembesaran atau zoom-in terhadap suatu citra.

2.8. Vektor

Vektor merupakan sebuah format untuk menyatakan sebuah perpindahan dari satu tempat ke tempat lainnya yang terdiri atas arah dan panjang. Vektor selalu mempunyai dua unsur, yaitu besar dan arah. Dalam dunia dua dimensi, Vektor memiliki dua buah komponen, yaitu x dan y. x merupakan komponen arah horisontal (sumbu X), sedangkan y merupakan komponen arah vertikal (sumbu Y). Dalam dunia tiga dimensi, vektor memiliki tiga komponen, yaitu x, y, dan z. Dalam penerapannya komponen x, y, dan z sering disebut sebagai i, j, k. Tambahan komponen z dalam dunia 3 dimensi merupakan sebuah komponen untuk mendeskripsikan arah kedalaman (sumbu Z).



Gambar 2.1. Vektor AB = 4x + 3y

Garis AB merupakan vektor dengan komponen 4x+ 3y. Vektor ini dapat diartikan sebagai perpindahan dari titik awal (titik A) ke titik akhir (titik B) Pada vektor AB karena dari titik A ke titik B diperlukan perpindahan koordinat sejauh 4 (koordinat X dari B dikurangi koordinat X dari A), maka komponen x berharga 4. Sedangkan pada sumbu Y dibutuhkan perpindahan sejauh 3 (Koordinat Y dari B dikurangi koordinat Y dari A), maka komponen y berharga 3.

3. ANALISIS DAN PERANCANGAN

Analisis sistem

Analisis sistem merupakan kegiatan mempelajari interaksi sistem yang terdiri atas pelaku proses dalam sistem prosedur dan data serta apa saja yang dibutuhkan dalam pembuatan aplikasi ini. Kegiatan ini memetakan suatu unit pada sistem pada model yang dapat mengidentifikasi seluruh aspek yang terlihat dalam sistem tersebut.

3.1.1 Analisis Masalah

Sesuai dengan hasil penelitian didapat adanya pembuatan animasi yang masih secara manual. Berdasarkan hal tersebut maka dibangun aplikasi image morphing untuk membantu membuat suatu animasi.

3.1.1.1 Cara kerja Image Morphing

Langkah kerja Image morphing adalah sebagai berikut:

1 Ambil 2 buah image 2 dimensi.
2 Tentukan mana yang dijadikan image source dan mana yang image destination.
3 Bandingkan ukuran kedua image tersebut, jika tidak sama maka pilih salah satu yang akan dirubah ukurannya.
4 Cek format filenya.
5 Cek garis feature ada atau tidak?
6 Jika ada maka metode *field morphing* dan *deformation techniques* dapat diproses sesuai garis feature.
7 Jika tidak ada maka akan diproses dangan metode *cross dissolve*.

Tentukan mana yang akan dijadikan image source dan image destination setelah itu cek ukuran kedua image tersebut setelah ukuran keduanya sama, kemudian cek format filenya akan secara otomatis diubah ke format BMP, terakhir berikan garis feature ke kedua image tersebut maka proses akan dijalankan.

3.1.1.2 Analisis dengan menggunakan Metode *Field Morphing*

Metode *field morphing* merupakan metode yang digunakan untuk menhasilkan sebuah animasi antara image source dan image destination. Metode *field morphing* menggunakan garis feature untuk menentukan perubahan. Langkah yang dilakukan pada metode *field morphing* adalah sebagai berikut :

1. Mencari jarak dari titik ke garis feature, untuk mengetahui perpindahan yang terjadi pada garis feature yang baru.
2. Setelah diketahui perbedaan titik akhir ke titik awal, dicari bobot dari garis feature terhadap titik tersebut., semakin jauh jarak titik ke garis feature maka semakin sedikit pengaruh perpindahannya terhadap titik.
3. Setelah ditemukan bobot dan perpindahan titik terhadap setiap garis feature, dapat dicari titik akhir yang dipengaruhi oleh semua garis feature.
4. Perubahan ini tidak hanya pada image source melainkan pada image destination juga, setelah itu baru

dilakukan penggabungan sehingga hasilnya akan terlihat lebih halus.

Langkah pertama yang dilakukan pada metode field morphing adalah dengan mencari jarak dari titik ke garis feature. Hal ini dilakukan untuk mengetahui perpindahan yang terjadi pada garis feature yang baru. Setelah diketahui perbedaan dari titik akhir ke titik asal, maka perlu dicari bobot dari setiap garis feature terhadap titik tersebut. Semakin jauh jarak titik ke garis feature berarti semakin sedikit pengaruh perpindahannya terhadap titik. Setelah diketahui bobot dan perpindahan dari titik terhadap setiap garis feature maka dapat dicari titik akhir yang dipengaruhi oleh semua garis feature. Pada proses image morphing maka yang dilakukan perubahan tidak hanya pada image source saja, melainkan juga pada image destination setelah itu baru digabungkan, sehingga hasilnya akan terlihat lebih halus.

Gambar diagram alir (*flowchart*) dari metode *field morphing* dapat dilihat pada gambar 3.6.

3.1.1.3 Analisis dengan menggunakan Metode *Deformation Techniques*

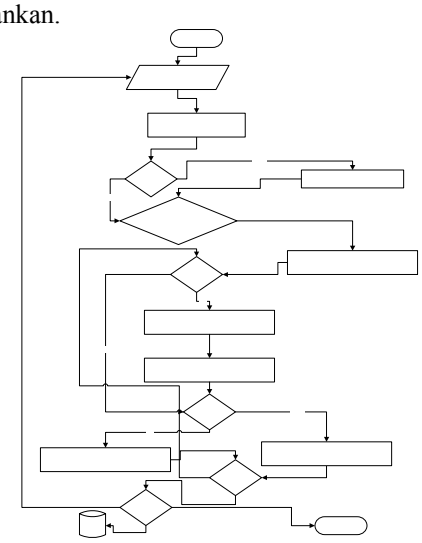
Metode *deformation techniques* merupakan metode yang digunakan untuk menghasilkan perubahan-perubahan yang diinginkan dari image source menjadi image destination seperti yang diinginkan oleh user. Proses *deformation techniques* menggunakan garis feature untuk menentukan hubungan antara image source menjadi image destination. Pada metode *deformation techniques* ini, cara yang dipakai untuk mendapatkan sebuah posisi pixel dari image source menjadi imgae destination lebih sederhana apabila dibandingkan dengan metode field morphing. Metode *deformation techniques* adalah sebagai berikut:

1. Mencari jarak dari titik ke titik awal da titik akhir dari garis feature.
2. Untuk mengetahui perpindahan pixel dari setiap frame sehingga terjadi suatu transisi yang halus.
3. Perpindahan dari tiap pixel dipengaruhi oloh garis-garis feature yang telah terbentuk.

Metode deformation techniques pertama kali akan mencari jarak dari titik ke titik awal dan titik akhir dari garis feature. Hal ini untuk mengetahui perpindahan pixel dari setiap frame sehingga terjadi suatu transisi yang halus. Perpindahan dari tiap pixel yang dipengaruhi oleh garis-garis feature yang telah terbentuk. Gambar diagram alir (*flowchart*) dari metode *deformation techniques* dapat dilihat pada gambar 3.7.

3.1.2 Perancangan Sistem

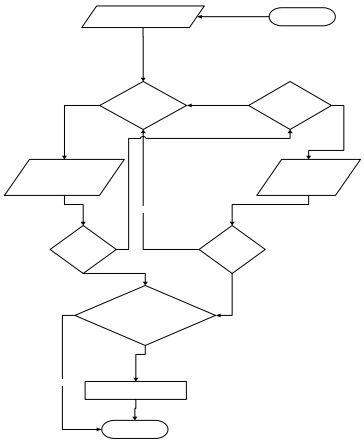
Dalam perancangan ini akan menggunakan flowchart sistem secara keseluruhan, yaitu bagaimana urutan jalan dari aplikasi morphing. Sistem aplikasi morphing mempunyai empat tahap utama yang harus dilakukan sebelum menampilkan hasil morphing dari dua buah image. Tahap-tahap yang harus dijalankan terlebih dahulu adalah membuka image source dan image destination, apabila image source dan image destination mempunyai ukuran yang berbeda maka ukuran harus disesuaikan dahulu dengan memilih image yang akan diubah ukurannya, kemudian cek format filenya, setelah itu memberikan hubungan antara image source dan image destination dengan memberikan garis feature atau point pada masing-masing image. Keempat proses ini harus dijalankan terlebih dahulu agar penempatan garis feature untuk animasi yang ingin dibentuk dapat dijalankan.



Gambar 3.1 *Flowchart* Sistem Field Morphing Secara Umum

3.1.2.1 Proses Membuka Image Source dan Image Destination

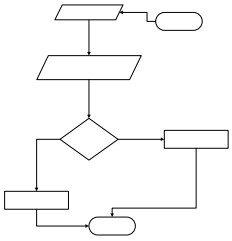
Pada bagian ini terdapat 2 proses utama yang dilakukan oleh sistem, yaitu proses membuka image dan menampilkannya ke layar, serta mengecek ukuran antara imagesource dan image destination apakah sudah sesuai, apabila ukuran image source dan image destination tidak sesuai maka dilakukan proses interpolasi untuk menyesuaikan ukuran keduanya. Apabila salah satu image belum terbuka maka pengecekan ukuran tidak akan dilakukan. Flowchart untuk membuka image source dan image destination dapat dilihat pada Gambar 3.2.



Gambar 3.2. *Flowchart* untuk Membuka *Image Source* dan *Image Destination*

3.1.2.2. Proses Ubah Ukuran Image

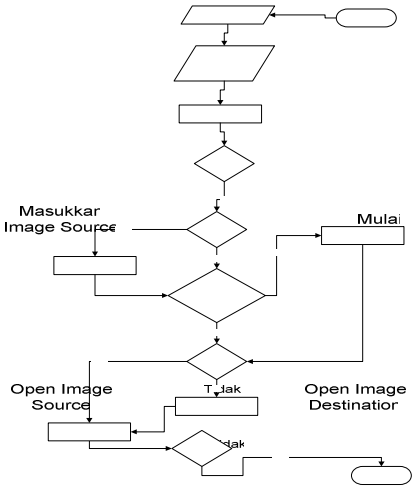
Proses morphing akan dapat berjalan apabila ukuran image source dan image destination sama. Hal ini dikarenakan jumlah pixel yang berbeda apabila ukuran image berbeda. Pada saat mengubah ukuran image, user diberi kebebasan untuk memilih ukuran image mana yang ukurannya ingin diubah. Setelah memilih, maka program akan mengecek apakah format file dari kedua image sudah ber-extension bitmap (*.bmp). Flowchart sistem untuk mengubah ukuran image, dapat dilihat pada Gambar 3.3.



Gambar 3.3. *Flowchart* untuk Mengubah Ukuran *Image*

3.1.2.3. Proses *Field Morphing*

Pada bagian ini, proses dibagi menjadi 2 bagian penting agar sebuah image source dan sebuah image destination dapat diproses menggunakan metode field morphing. Flowchart sistem untuk metode field morphing dapat dilihat pada Gambar 3.4.

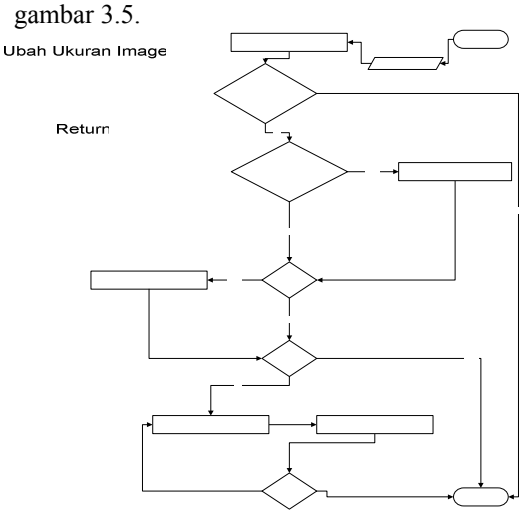


Gambar 3.4 *Flowchart* Proses *Field Morphing*

Pada field morphing dibutuhkan adanya pemasukan garis feature, tetapi apabila dalam gambar tidak terdapat garis feature sama sekali berarti image tersebut akan diubah sama dengan hanya menggunakan teknik cross dissolve.

3.1.2.4. Proses *Deformation Techniques*

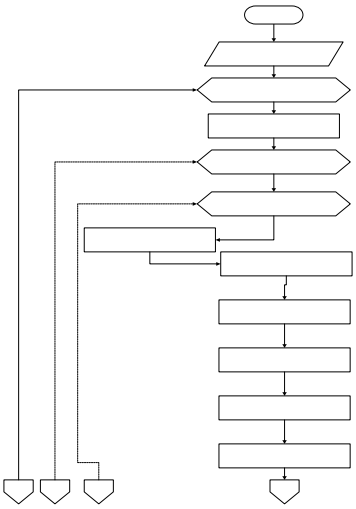
Pada metode *deformation techniques,* terdapat 2 buah proses utama yang harus dijalankan berdasarkan garis *feature* yang diberikan pada *image source* dan *image destination. Flowchart* sistem pada metode *deformation techniques* dapat dilihat pada gambar 3.5.



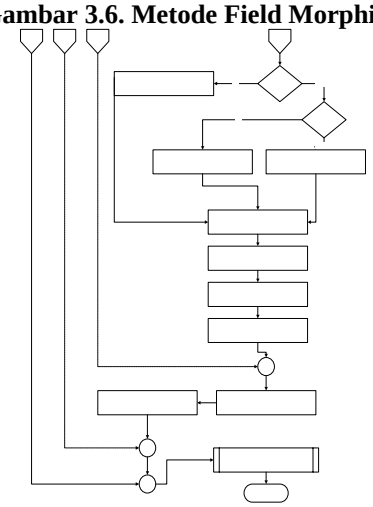
Gambar 3.5 *flowchart* proses *deformation techniques*

Diagram alir (*flowchart*) metode *field morphing*

Gambar 3.6 ini merupakan proses diagram alir dari metode *field morphing*.



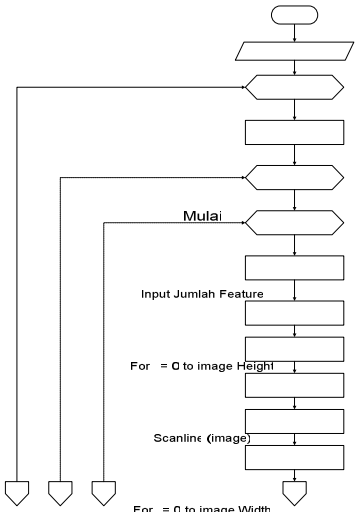
Gambar 3.6. Metode Field Morphing



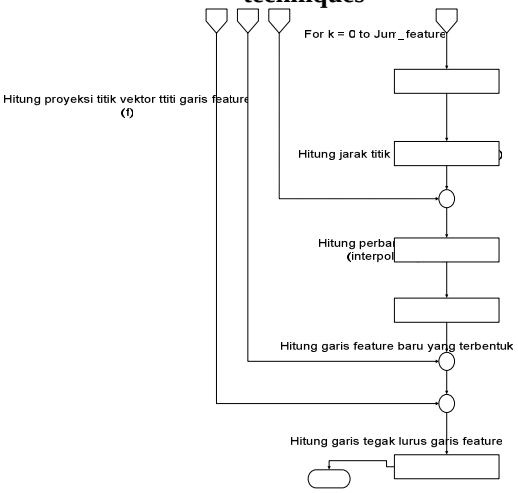
Gambar 3.6. Metode Field Morphing (sambungan)

Diagram alir (*flowchart*) metode *deformation techniques*

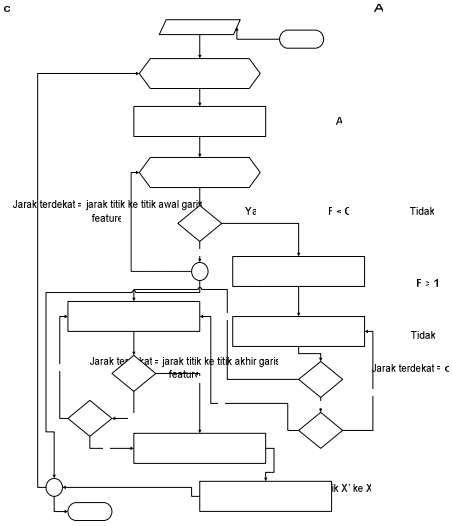
Gambar 3.6 ini merupakan proses diagram alir dari metode *deformation techniques*



Gambar 3.7. Metode deformation techniques



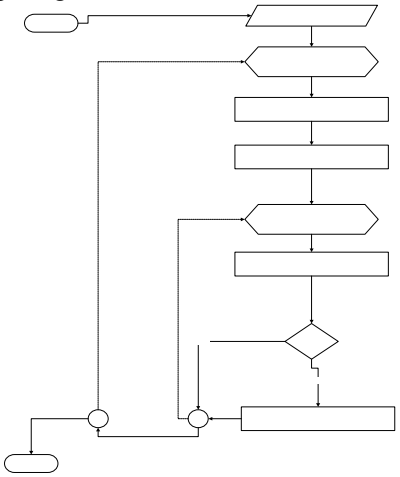
Gambar 3.7. Metode deformation techniques (sambungan)



Gambar 3.8. Interpolasi Image Hasil

3.1.2.6. Proses Cross Dissolve

Proses ini merupakan proses terakhir dari image morphing. Proses ini menghitung perbandingan dari image source dan image destination yang akan tampil sehingga terlihat perubahan dari image source menjadi image destination. Flowchart proses cross dissolve dapat dilihat pada gambar 3.9



Gambar 3.9. Flowchart Proses Cross Disslove

3.1.3 Analisis Kebutuhan *Non Fungsional*

Analisis kebutuhan *non fungsional* dilakukan untuk menghasilkan spesifikasi kebutuhan *non fungsional*. Spesifikasi kebutuhan *non fungsional* adalah spesifikasi yang rinci tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan.

3.1.3.1 Analisis Perangkat Keras

Komputer yang akan digunakan untuk menjalankan sistem dengan spesifikasi sebagai berikut:

Spesifikasi komputer yang diusulkan:

1. *Processor* AMD Athlon (tm) XP 2100,
2. RAM 256 MB,
3. *Hard Disk* 40 GB,
4. *CD ROM Drive*,
5. *Monitor* 15',
6. *Keyboard* dan *mouse* (Perangkat keras tambahan),

3.1.3.2 Analisis Perangkat Lunak

Software yang digunakan untuk menjalankan aplikasi image morphing ini, yaitu dengan menggunakan:

a. Sistem operasi Selesa yang digunakan *Windows XP*
b. Delphi 7 untuk menjalankan aplikasi
c. *MSU Video Quality Measurement Tool*

3.1.4 Struktur Data

Dalam aplikasi ini, digunakan beberapa *record, procedure* dan juga *function* untuk memudahkan pembuatan dan pengecekannya. *Record procedure* dan *function* yang ada dalam aplikasi ini terdiri dari berbagai macam dan fungsi yang berlainan.

3.1.4.1 *Record*

Aplikasi morphing ini menggunakan beberapa *record, procedure* dan *function. Record* yang ada digunakan untuk menyimpan data dari suatau objek sehingga penggunaan variabelnya dapat lebih efektif. *Record* yang ada dalam aplikasi ini adalah :

1. *Record TgrsFeature*

```
Tgrs Feature = record
        x_1,y_1:real;
        x_2,y_2 :real;
        jrk :real;
end;
```

Record TgrsFeature digunakan untuk menyimpan data yang berhubungan dengan garis *feature.* Data yang harus disimpan berhubungan dengan garis *feature* adalah x1, y1, x2, y2 dan jrk. Data x1 dan y1 merupakan data koordinat dari awal garis *feature*, x2 dan y2 merupakan data koordinat dari akhir garis *feature,* sedangkan jrk merupakan data yang menyimpan hasil dari panjang garis *feature*. Data x1, y1, x2 dan y2 akan diinputkan oleh *user,* sedangkan data jrk akna dihitung secara otomatis oleh program.

Penggunaan *record* ini akan mempermudah perhitungan yang menggunakan garis *feature,* karena pemakaian variabel akan lebih terstrutur. Hal ini dikarenakan penggunaan garis *feature* yang lebih dari satu dari setiap proses.

2. *Record TPixel*

```
TPixel = record
        RGB :byte;
end;
array pixel=array [0,0] of Tpixel;
APixel=^arraypixel;
```

Record TPixel ini digunakan untuk menyimpan nilai warna dari suatu pixel. Pada *Record TPixel* ini data dari suatu pixel yang disimpan adalah data dari warna *red, green,* dan *blue.* Data-data inilah yang nantinya yang akan menyimpan warna dari suatu pixel yang ada pada *image*. Variabel R, G, dan B dalam *Tpixel* nantinya akan terisi nilai dari 0 sampai 255.

Record TPixel dalam aplikasi morphing tidak akan langsung digunakan melainkan

dibutuhkan nilai *array* dari *record Tpixel*, dimana *array* tersebut tergantung dari besar *image* yang akan diproses

3. *Record Tframe*

```
Tframe = record
   RGB :byte;
   Kode :Boolean;
end;
```

Record Tframe digunakan untuk menyimpan nilai warna dari hasil deformasi sebuah image. Hasil dan pengambilan pixel image dibedakan menjadi dua buah record dikarenakan pada sebuah image hasil, selain dibutuhkan variabel R, G, dan B untuk menyimpan warna-warna yang ada, diperlukan juga variabel bantuan yaitu kode. Variabel kode akan berisi pernyataan *true* atau *false*. Variabel kode akan bernilai *true* apabila pada koordinat atau *array* yang dimaksud, pixelnya sudah terisi, dan akan bernilai *false* apabila pada koordinat teersebut, nilai pixelnya tidak terisi. Variable kode ini akan membantu program untuk mengetahui titik mana saja yang belum terisi, yang perlu diisikan dengan fungsi interpolasi.

3.1.4.4 *Procedure* dan *Function*

Selain *record,* aplikasi ini menggunakan juga beberapa *procedure* dan *function* untuk mempermudahkan pembuatan aplikasi. *Procedure* yang ada digunakan untuk memisahkan proses yang satu dengan yang lain sehingga tidak terjadi perulangan perhitungan yang nantinya akan menyusahkan pembuatan aplikasi. Beberapa *procedure* dan *function* yang ada dalam aplikasi *morphing* adalah :

Tabel 3.1. *Procedure* dan *function*

Prosedure / function pada aplikasi image morphing	*Fungsi*
procedure reset_morph();	Digunakan untuk me-reset semua variable yang dipakai dalam pembuatan aplikasi image morphing
procedure JPEGtoBMP(file_name:string);	Digunakan untuk mengubah file image ber-format JPEG menjadi BMP agar dapat diproses. Dalam procedure ini perubahan dari format JPEG akan disimpan dahulu menjadi image yang ber-format BMP. File BMP yang akan digunakan untuk proses image morphing.

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Pada bab ini akan dilakukan implementasi dan pengujian terhadap sistem aplikasi *morphing*. Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman yang akan digunakan. Setelah implementasi maka dilakukan pengujian terhadap sistem aplikasi *morphing* dan akan dilihat kekurangan-kekurangan pada aplikasi yang baru untuk pengembangan sistem selanjutnya.

Aplikasi ini dibangun dengan menggunakan bahasa *visual Delphi 7.0,* dengan menggunakan sistem operasi *windows xp sp2*

4.2 Pengujian

Pengujian yang dilakukan terhadap dua video sebagai video hasil field morphing dan defomation technique, hasil dari analisis terhadap pengujian yang telah dilakukan antara lain :

1. Jumlah *frame* dan garis *feature* sangat mempengaruhi waktu dan hasil proses metode *field morphing* dan *deformation technique*.
2. Metode *field morphing* lebih baik dari segi hasil, sedangkan metode *deformation technique* lebih baik dari segi waktu.

3. Kecepatan dan keberhasilan proses penyisipan tergantung pada kemampuan hardware.
4. Format video yang bisa diproses hanya format microsof t AVI

5.1 Kesimpulan

Beberapa kesimpulan yang dapat diambil adalah sebagai berikut :

1. Jumlah frame dan garis feature sangat mempengaruhi waktu dan hasil dari proses field morphing dan deformation techniques.
2. Metode field morphing lebih baik dari segi hasil, sedangkan metode deformation techniques lebih baik dari segi waktu.

5.2 Saran

Beberapa saran untuk pengembangan sistem selanjutnya adalah sebagai berikut :

1. Mengembangkan algoritma *morphing* yang ada untuk melakukan *morphing* 3 dimensi
2. Menambah jeni *Format file* untuk menyimpan animasi hasil *morphing* misalnya *.mpeg, *.mpg dan sebagainya.
3. Menggabungkan aplikasi *morphing* dengan aplikasi pengenalan suara sehingga hasilnya dapat digabungkan antara fiile *video* dan *file audio*.
4. Menambahkan atau menggabungkan algoritma *morphing* yang lain agar hasil animasi yang didapatkan lebih baik dan dapat mengetahui kelebihan dan kelemahannya masing-masing.