

## *Pengembangan Aplikasi Konversi Video Digital Greyscale Menjadi Video Digital Berwarna Dengan Menggunakan Metode Global Image Matching*

Ketut Widianara<sup>1</sup>, Made Windu Antara Kesiman<sup>2</sup>, I Made Gede Sunarya<sup>3</sup>  
Jurusan Pendidikan Teknik Informatika  
Universitas Pendidikan Ganesha  
Singaraja, Bali

E-mail: widian45@yahoo.com<sup>1</sup>, dekndu@yahoo.com<sup>2</sup>, imadegedesunarya@gmail.com<sup>3</sup>

**Abstrak**—Penelitian ini bertujuan untuk merancang dan mengimplementasikan sebuah aplikasi yang dapat mengkonversi video digital greyscale menjadi video digital berwarna secara cepat, mudah dan hasil pewarnaan yang memuaskan. *Input* dari aplikasi ini berupa video greyscale dan citra warna acuan, dan *output* dari aplikasi ini adalah sebuah video berwarna dengan nuansa warna yang mengacu pada citra warna acuan yang di-*input*-kan.

Aplikasi ini dikembangkan dengan menggunakan metode *Global Image Matching* yang diimplementasikan pada lingkungan bahasa pemrograman *Java* dengan menggunakan editor *NetBeans 7.2.1*. Pada aplikasi yang dibuat terdapat tiga proses utama yaitu proses ekstrak video, proses pewarnaan, dan proses *render*. Proses ekstrak merupakan proses memecah video menjadi *frame-frame* dan audio. Proses pewarnaan merupakan proses yang mengimplementasikan metode *Global Image Matching*, dimana *frame-frame* hasil ekstraksi akan dilakukan pewarnaan pada proses ini. Sedangkan proses *render* merupakan proses penggabungan *frame-frame* hasil pewarnaan dengan *file* audio hasil ekstraksi menjadi sebuah video berwarna. Berdasarkan hasil pengujian, aplikasi konversi video digital greyscale menjadi video digital berwarna ini dapat melakukan proses konversi dengan sangat mudah dan cepat.

**Kata kunci**—*Citra Warna Acuan, Ekstraksi, Frame, Global Image Matching, Java, NetBeans, Rendering, Video Digital Greyscale.*

**Abstract**—*This research aims to design and implement an application that could convert a greyscale video into a colored video digital, easily and staining results were satisfactory. Input from the applications is greyscale video and color reference image, and the output of this application was a video-colored with shades of color that refers to the reference image input.*

*This application was developed by using the method of Global Image Matching was implemented in the Java programming language environment with editor used NetBeans 7.2.1. On an application made there were three main processes that extract the video, the process of coloring, and render process. The extract process was the process of breaking into the video and audio frames. Coloring process was a process that implements the Global Image Matching method, in which frame-frame extraction results would be coloring in this process. While the render process was the process of combining frames staining results with the extracted audio files into a video-colored. Based on test results, greyscale digital video conversion application to digital video color conversion process can be done very easily and quickly.*

**Keywords**— *Color Reference Image, Extraction, Frame, Global Image Matching, Java, NetBeans, Rendering, Greyscale Digital Video.*

### I. PENDAHULUAN

Video *greyscale* merupakan video yang tersusun dari *frame-frame* yang hanya memiliki satu nilai kanal pada setiap *pixel*, dengan kata lain  $RED = GREEN = BLUE$ . Video *greyscale* sering disebut sebagai video hitam putih (*black and white*), ini disebabkan karena video *greyscale* pada tiap *pixel* *frame*-nya berwarna dari hitam sampai putih. Umumnya video *greyscale* kurang memiliki daya tarik jika dibandingkan dengan video berwarna (*true color*), ini disebabkan karena warna pada gambar video berwarna merupakan *powerful descriptor* yang sering digunakan untuk pengenalan objek maupun ekstraksi objek dari suatu citra. Selain itu kebanyakan video *greyscale* yang ada berkaitan dengan sejarah dari suatu peristiwa. Video seperti ini tentunya sangat

penting untuk kita simak, akan tetapi dengan warna video yang *greyscale* menjadikan video *greyscale* kurang menarik untuk disimak. Untuk itu suatu upaya restorasi video diperlukan untuk mengubah video *greyscale* menjadi video berwarna. Restorasi yang mungkin dilakukan yaitu mewarnai video *greyscale* dengan menggunakan teknik pentransferan warna.

Pentransferan warna dapat dilakukan dengan cara manual, yaitu dengan menggunakan *software* desain grafis untuk mewarnainya. Seperti yang kita ketahui fps (*frame rate/seconds*) dari suatu video berkisar 25-30 fps [1]. Bisa dibayangkan waktu yang dibutuhkan untuk memproses video yang berdurasi 60 detik/ lebih akan membutuhkan waktu yang lama. Selain itu, agar hasil pewarnaan *frame* lebih bagus maka dibutuhkan tenaga ahli yang paham mengenai proses pewarnaan *frame* pada perangkat lunak desain grafis, dan untuk mendatangkan tenaga ahli tentunya membutuhkan biaya yang tidak sedikit.

Berdasarkan permasalahan tersebut perlu dikembangkannya suatu aplikasi yang dapat mengkonversi video digital *greyscale* menjadi video digital berwarna. Aplikasi yang dikembangkan diharapkan dapat melakukan pewarnaan pada video *greyscale* dengan baik, mudah, cepat, dan hasil dari konversi video ini mampu memberikan persepsi yang jelas dan lebih menarik untuk disimak.

## II. KAJIAN TEORI

### A. Citra Digital

Citra menurut kamus *Webster*, adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda. Citra merupakan istilah lain untuk gambar sebagai salah satu komponen multimedia yang memegang peranan penting dalam informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks yaitu citra kaya dalam informasi.

Suatu citra dapat didefinisikan sebagai fungsi  $f(x,y)$  berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo  $f$  di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut, jika nilai x, y, dan nilai amplitudo  $f$  secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital [2].

Citra terbagi menjadi 2 yaitu :

a. Citra diam, adalah citra tunggal yang tidak bergerak.

b. Citra bergerak (video) adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberi kesan pada mata kita sebagai gambar yang bergerak.

### B. Jenis Citra

Berdasarkan nilai *pixel* dalam , jenis citra dapat dibedakan sebagai berikut [2].

a) Citra biner, merupakan citra dengan nilai *pixel* antara intensitas maksimum dan intensitas minimum. Citra biner juga disebut sebagai citra B&W (black and white) atau citra monokrom.

b) Citra *greyscale*, merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap *pixel*, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuandan putih.

c) Citra warna, adalah citra dengan lebih dari 1 nilai pada setiap *pixel*-nya. Warna merah, hijau, dan biru (Red, Green, Blue atau RGB) adalah warna primer untuk mencampur cahaya yang disebut dengan additive primary color yang berbeda dengan subtractive primary color untuk mencampur cat (Cyan, Magenta, Yellow).

### C. Video Digital

Video merupakan gabungan gambar-gambar mati yang dibaca berurutan dalam suatu waktu dengan kecepatan tertentu. Gambar-gambar yang digabung tersebut dinamakan *frame* dan kecepatan pembacaan gambar disebut dengan *frame rate*, dengan satuan fps(*frame per second*). Karena dimainkan dalam kecepatan yang tinggi maka tercipta ilusi gerak yang halus, semakin besar nilai *frame rate* maka akan semakin halus pergerakan yang ditampilkan [3].

Karakteristik dari video digital mempengaruhi kualitas terhadap suatu video. Adapun karakteristik yang dimiliki oleh suatu video ada tiga yaitu resolusi, kedalaman bit, dan laju *frame* [4].

### D. FFmpeg

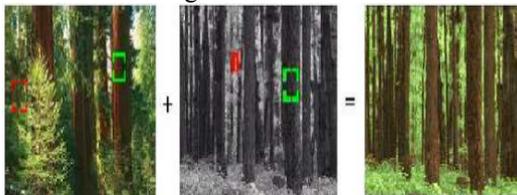
FFmpeg adalah program komputer yang dapat merekam, mengkonversikan dan *streaming* audio dan video digital dalam berbagai format. FFmpeg merupakan aplikasi *command line* yang terdiri dari kumpulan pustaka perangkat lunak bebas/*open source*. Termasuk *libavcodec*, *library* untuk audio

codec/ video codec yang digunakan oleh beberapa proyek lain, dan *libavformat*, *library* untuk audio / video mux kontainer dan demux kontainer. Nama proyek yang berasal dari grup video standar MPEG, di tambahkan "FF" untuk "fast forward" [5].

FFmpeg dikembangkan pada Linux, tetapi dapat dikompilasi di banyak sistem operasi, termasuk Apple Inc Mac OS X, Microsoft Windows dan AmigaOS. Sebagian besar platform komputasi dan set instruksi arsitektur *microprocessor* juga di dukung, seperti x86 (IA-32 dan x86-64), PPC (PowerPC), ARM, DEC Alpha, SPARC, dan arsitektur MIPS.

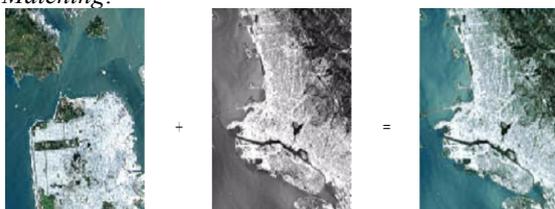
#### E. Global Image Matching

Pentransferan warna secara keseluruhan dari gambar warna ke gambar *greyscale* dapat dilakukan dengan mencocokkan tingkat kecerahan dan informasi tekstur diantara kedua gambar [6]. Lebih lanjut, teknik ini dikembangkan dengan memperbolehkan pengguna untuk mencocokkan area dari kedua gambar dengan menggunakan kotak-kotak contoh (*swatches*). Teknik ini dapat diterapkan pada gambar yang bermacam-macam, dengan tekstur dan tingkat kecerahan yang benar-benar berbeda dengan sukses.



Gambar 1. Transfer Warna dengan Swatches [7]

Teknik pentransferan warna tanpa *swatches* merupakan teknik pentransferan warna dimana warna-warna yang akan ditransfer dari citra warna ke dalam citra *greyscale* ditentukan secara otomatis oleh komputer. Komputer akan mencocokkan setiap sampel warna yang diperoleh dari citra warna ke dalam citra *greyscale* dengan acuan tingkat kecerahan warna. Salah satu metode yang menerapkan konsep pentransferan warna tanpa *swatches* adalah metode *Global Image Matching*.



Gambar 2.14 Transfer Warna dengan Tanpa Swatches [7]

Ada beberapa tahapan yang harus dilakukan dalam proses pentransferan warna dengan menggunakan teknik *Global Image Matching* [8]. Berikut akan dijelaskan tahapan-tahapan dari metode *Global Image Matching*.

#### a) Merubah Ruang Warna RGB ke dalam Ruang Warna $l\alpha\beta$

Ruang warna  $l\alpha\beta$  merupakan ruang warna yang dikembangkan oleh Ruderman. Ruang warna ini memiliki 3 buah *channel* dimana  $l$  merupakan *luminance*, komponen  $\alpha$  merupakan komponen penyusun warna yaitu dari kuning sampai biru dan komponen  $\beta$  merupakan komponen penyusun warna yaitu dari merah sampai dengan hijau. Cara untuk mengubah citra dengan ruang warna RGB menjadi citra dengan ruang warna  $l\alpha\beta$ , dapat menggunakan matriks konversi sebagai berikut.

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \dots\dots(1)$$

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \dots\dots(2)$$

#### b) Penyesuaian Tingkat Kecerahan Warna

Citra warna dan citra *greyscale* memiliki perbedaan tingkat kecerahan yang cukup tinggi, hal ini dapat berpengaruh di dalam proses pewarnaan sehingga diperlukan adanya suatu proses *luminance remapping*, yaitu suatu proses menggeser dan menskalakan tingkat kecerahan citra warna agar sesuai dengan tingkat kecerahan citra *greyscale*. Terdapat beberapa tahapan dalam melaksanakan proses penyesuaian tingkat kecerahan citra, adapun tahapan tersebut adalah sebagai berikut.

1. Membuat histogram tingkat kecerahan gambar warna.
2. Membuat histogram tingkat kecerahan gambar *greyscale*.
3. Melakukan proses *luminance remapping*.

#### c) Menghitung Statistik Pixel

Proses ini bertujuan untuk mencari kecocokkan antara citra warna dengan citra *greyscale*. Terdapat beberapa tahapan yang harus dilakukan untuk mencari kecocokkan antara citra warna dengan citra *greyscale* yaitu:

1. Pemilihan sampel pixel warna pada citra warna

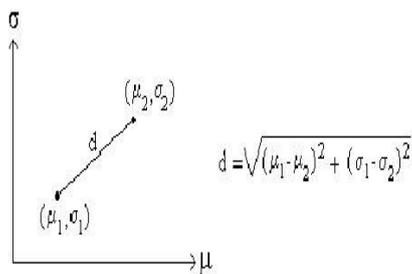
Sampel warna merupakan contoh *pixel* warna yang nantinya akan diambil nilai kromatik warnanya yang akan digunakan sebagai pigmen warna *pixel* pada citra *greyscale*.

2. Perhitungan statistik

Perhitungan statistik yang dimaksud disini adalah perhitungan *mean* (rata-rata) dan perhitungan standar deviasi (simpangan baku). Perhitungan statistik dilakukan pada kedua citra inputan. Pada citra warna, perhitungan dilakukan pada masing-masing sampel sedangkan pada citra *greyscale* dilakukan pada keseluruhan masing-masing *pixel*. Dalam perhitungan nilai rata-rata dan simpangan baku setiap *pixel* dilakukan dengan menggunakan matriks 5 x 5.

d) Mencocokkan Pixel

Setiap *pixel* pada citra *greyscale* dicocokkan dengan *pixel* sampel yang didapat dari citra warna. Proses pencocokan dilakukan dengan menghitung bobot rata-rata dan simpangan baku di sekelilingnya. Dengan menganggap rata-rata ( $\mu$ ) dan simpangan baku ( $\sigma$ ) tersebut sebagai kordinat, maka *pixel* gambar warna yang paling cocok adalah *pixel* yang memiliki jarak terpendek ( $d$ ) dengan *pixel* gambar *greyscale*. Ilustrasi pencocokan *pixel* dapat dilihat seperti Gambar 3 berikut.



Gambar 3. Ilustrasi Pencocokan Pixel [6]

e) Pewarnaan Citra Greyscale

Ketika ditemukan *pixel* yang paling cocok, nilai kromatik  $l$  dan  $\beta$  ditransfer ke *pixel* target pada citra *greyscale*, sementara tingkat kecerahan *pixel* tetap disimpnan.

Setelah proses pewarnaan di atas, maka citra *greyscale* yang telah memiliki warna dalam *lab* color space harus diubah kembali ke RGB color space untuk menampilkannya. Perubahan ruang warna *lab* ke dalam ruang warna RGB dibutuhkan matriks konversi berikut

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{6}}{6} & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} \dots\dots\dots(3)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 4.4679 & -3.5873 & 0.1193 \\ -1.2186 & 2.3809 & -0.1624 \\ 0.0497 & -0.2439 & 1.2045 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \dots\dots\dots(4)$$

III METODOLOGI

A. Analisis Perangkat Lunak

Pada sub bab analisis perangkat lunak ini, akan dipaparkan tahapan dari *Software/ System Development Life Cycle* (SDLC) yang meliputi hal-hal berikut.

1. Kebutuhan Perangkat Lunak

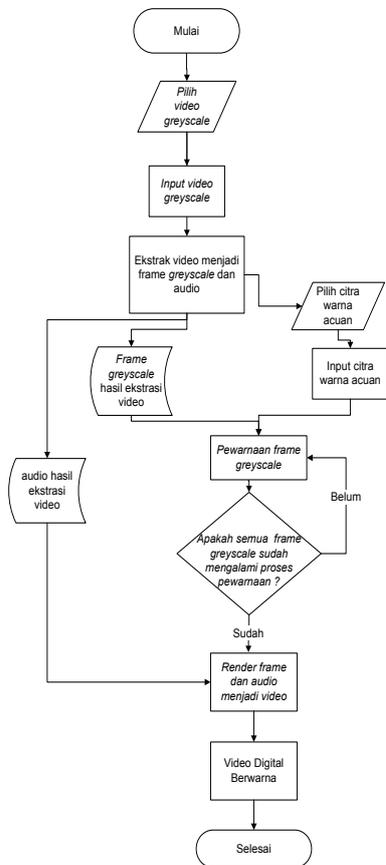
Berdasarkan analisis terhadap pengembangan aplikasi konversi video digital *greyscale* menjadi video digital berwarna, maka terdapat beberapa proses yang dapat implementasikan, yaitu:

- Melakukan ekstrasi dari video *greyscale* menjadi *frame-frame* dengan format *bitmap* dan sebuah *file* audio dengan format *mp3*.
- Melakukan operasi perubahan ruang warna dari ruang warna RGB ke ruang warna *lab* dan sebaliknya dari ruang warna *lab* ke ruang warna RGB.
- Melakukan penyesuaian tingkat kecerahan *pixel* dengan cara melakukan perataan histogram pada *framegreyscale* dan citra warna acuan dan melakukan proses spesifikasi histogram pada *frame* dan citra acuan hasil perataan histogram.
- Melakukan pengambilan *pixel* sampel dari citra warna acuan dengan jumlah yang sesuai dengan yang di-input-kan *user*.
- Melakukan penghitungan statistika *pixel* yang meliputi penghitungan rata-rata dan penghitungan standar deviasi *pixel*.
- Melakukan pencocokan nilai *pixel*, pencocokan nilai *pixel* didasarkan atas nilai perhitungan pada operasi statistik.
- Melakukan proses pewarnaan *framegreyscale* menjadi *frame* berwarna.
- Melakukan penggabungan antara *frame-frame* yang sudah diwarnai dengan *file* audio menjadi sebuah video berwarna.

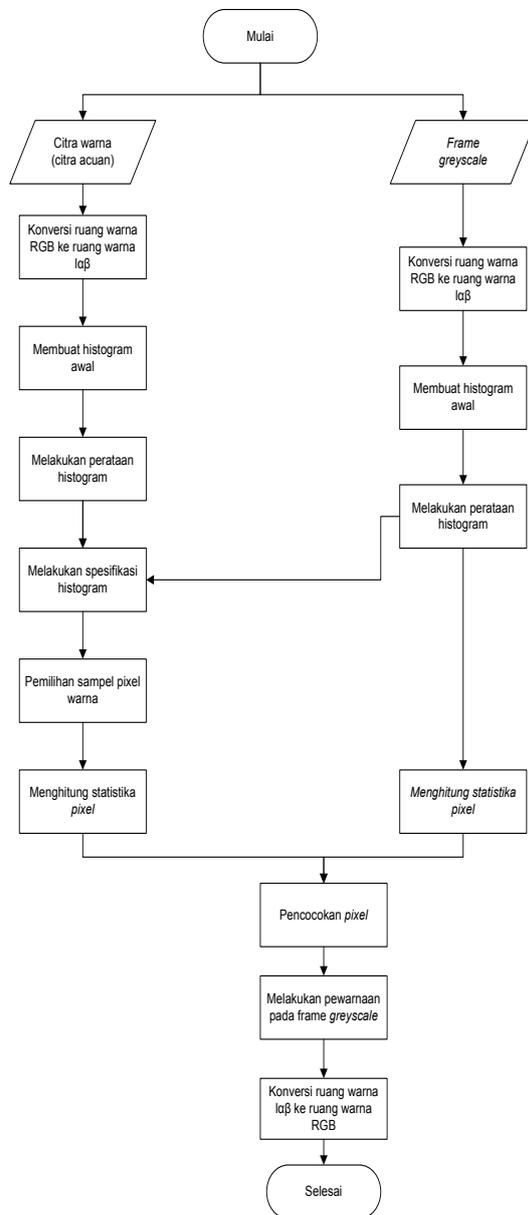
2. Tujuan Pengembangan Perangkat Lunak

Adapun tujuan dari pengembangan perangkat lunak aplikasi konversi video digital *greyscale* menjadi video digital berwarna adalah sebagai berikut.

- a. Dapat melakukan ekstraksi dari video *greyscale* menjadi *frame-frame* dengan format *bitmap* dan sebuah *file* audio dengan format mp3.
  - b. Dapat melakukan operasi perubahan ruang warna dari ruang warna RGB ke ruang warna *lab* dan sebaliknya dari ruang warna *lab* ke ruang warna RGB.
  - c. Dapat melakukan penyesuaian tingkat kecerahan *pixel* dengan cara melakukan perataan histogram pada *framegreyscale* dan citra acuan dan melakukan proses spesifikasi histogram pada *frame* dan citra acuan hasil perataan histogram.
  - d. Dapat melakukan pengambilan *pixel* sampel dari citra warna acuan dengan jumlah yang sesuai dengan yang di-*input*-kan *user*.
  - e. Dapat melakukan penghitungan statistika *pixel* yang meliputi penghitungan rata-rata dan penghitungan standar deviasi *pixel*.
  - f. Dapat melakukan pencocokan nilai *pixel*, pencocokan nilai *pixel* didasarkan atas nilai perhitungan pada operasi statistik.
  - g. Dapat melakukan proses pewarnaan *frame-framegreyscale* menjadi *frame-frame* berwarna.
  - h. Dapat melakukan penggabungan antara *frame-frame* yang sudah diwarnai dengan *file* audio menjadi sebuah video berwarna.
3. *Masukan dan Keluaran Perangkat Lunak*  
 Adapun masukan dan keluaran dari perangkat lunak aplikasi konversi video digital *greyscale* menjadi video digital berwarna, adalah sebagai berikut.
    - a. *Masukan Perangkat Lunak*
      - Video *greyscale* dengan format AVI , FLV, WEBM, MKV, MPEG, MP4, dan WMV.
      - Parameter jumlah *frame* per-detik (fps) untuk proses ekstraksi video.
      - Citra acuan pewarnaan dengan format *bitmap* (\*.*bmp*) 24 bit.
      - Parameter jumlah *pixel* sampel yang akan digunakan sebagai acuan pewarnaan.
      - Format *file* video *output* hasil pewarnaan video *greyscale*.
    - b. *Keluaran Perangkat Lunak*
      - *Frame* hasil ekstraksi video *greyscale* dengan format *bitmap* (\*.*bmp*).
      - Audio hasil ekstraksi video dengan format mp3.
2. *Perancangan Perangkat Lunak*  
 Terdapat beberapa tahapan dalam perancangan perangkat lunak aplikasi konversi video digital *greyscale* menjadi video digital berwarna. Adapun tahapan-tahapan tersebut yaitu.
    - a. *Batasan Perancangan Perangkat Lunak*  
 Masalah yang akan dibahas pada skripsi ini dibatasi pada hal-hal sebagai berikut.
      1. Video digital *greyscale* yang dimaksud adalah video yang apabila diekstrak akan terdiri dari *frame-frame* yang memiliki warna skala keabuan (*greyscale*) berekstensi *bitmap* (\*.*bmp*) 24 bit.
      2. Format video masukan berupa AVI , FLV, WEBM, MKV, MPEG, MP4, dan WMV.
      3. Citra warna/ acuan menggunakan format RGB - 24 bit dan berekstensi BMP , PNG dan JPEG/ JPG.
    - b. *Alur Kerja Perangkat Lunak*  
 Alur kerja perangkat lunak konversi video digital *greyscale* menjadi video digital berwarna dimulai dari user meng-*input*-kan video *greyscale*, dimana video *greyscale* akan diekstrak menjadi *frame-frame greyscale* dan *file* audio. Setelah proses ekstrak selesai dilanjutkan ke proses pewarnaan *frame*, pewarnaan *frame* dilakukan secara *frame by frame* dengan warna acuan diperoleh dari citra warna acuan yang di-*input*-kan *user*. Proses pewarnaan akan dilakukan pada semua *frame* hasil ekstraksi. Setelah semua *frame* diwarnai, kemudian dilanjutkan ke tahap *rendering frame-frame* hasil pewarnaan dan audio hasil ekstraksi menjadi video digital yang sudah berwarna (*true color*). Alur kerja dari aplikasi konversi video digital *greyscale* menjadi video digital berwarna digambarkan dalam *flowchart* pada Gambar 4 dan Gambar 5 merupakan *flowchart* dari proses pewarnaan yang dijabarkan lebih detail.



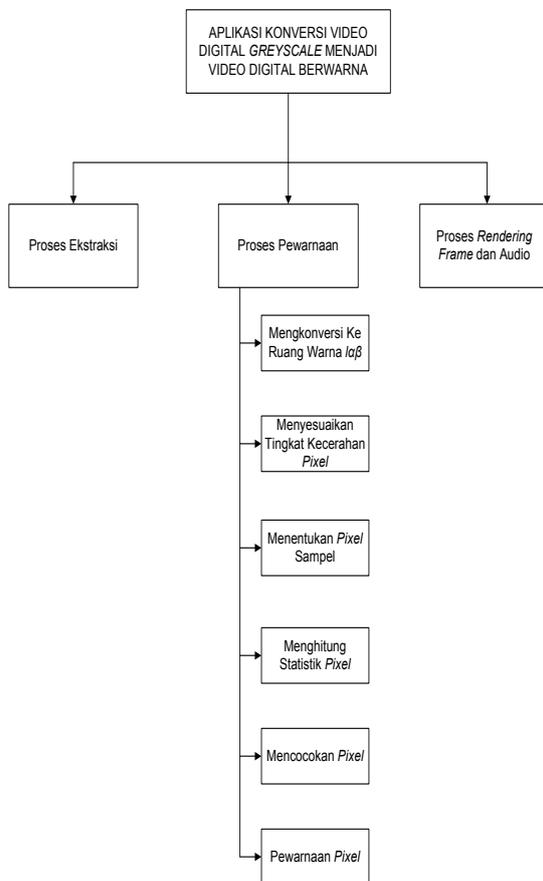
Gambar 4. Flowchart Aplikasi Konversi Video Digital Greyscale Menjadi Video Digital Berwarna



Gambar 5. Flowchart Proses Pewarnaan Aplikasi Konversi Video Digital Greyscale Menjadi Video Digital Berwarna

c. Perancangan Arsitektur Perangkat Lunak

Pada tahapan ini menggambarkan bagian-bagian modul, struktur ketergantungan antar modul, dan hubungan antar modul dari perangkat lunak yang dibangun. Pada bagian ini terdapat *structure chart* dan *flowchart* sebagai kendali fungsional yang digambarkan seperti Gambar 6 untuk perangkat lunak aplikasi konversi video digital greyscale menjadi video digital berwarna.



Gambar 6. Structure Chart Aplikasi Konversi Video Digital Greyscale Menjadi Video Digital Berwarna

Dari Gambar 6 kita dapat mengetahui bahwa dalam aplikasi yang akan dikembangkan terdapat tiga proses utama dan 6 sub proses dari proses pewarnaan. Setiap sub proses dari proses pewarnaan merupakan implementasi dari metode *Global Image Matching*. Penggunaan metode *Global Image Matching* pada pengembangan aplikasi ini disebabkan karena metode *Global Image Matching* merupakan teknik pentransferan warna yang paling efisien jika dibandingkan dengan teknik pentransferan warna yang lain, misalnya teknik pentransferan dengan menggunakan *swatches*.

#### d. Perancangan Struktur Data Perangkat Lunak

Struktur data utama yang digunakan dalam perangkat lunak aplikasi konversi video digital greyscale menjadi video digital berwarna ini adalah struktur data yang terdapat pada bahasa pemrograman *java* yaitu matriks *arrayofarrayofshort*, matriks

*arrayofarrayofinteger*, matriks *array of array of double*, larik *arrayofinteger*, larik *array of double*, *string*, *bufferedImage* dan *integer*. Sedangkan struktur data bentukan meliputi tipe dan objek. Tipe data objek pada perangkat lunak aplikasi konversi video digital greyscale menjadi video digital berwarna dapat berupa tampilan GUI (*Graphic Unit Interface*). Objek untuk tampilan GUI yang digunakan adalah objek-objek yang terdapat pada bahasa pemrograman *java* dengan menggunakan editor *Netbeans*, diantaranya yaitu *Jframe*, *Jinternalframe*, *Jtabbedpane*, *Jpopupmenu*, *Jpanel*, *Jlabel*, *Jbutton*, *Jtextfield*, *Jcheckbox*, *Jprogressbar*, *Jradiobutton*, *Jfilechooser* dan *Jspinner*, *Jscrollpane*.

#### e. Perancangan Antar Muka Perangkat Lunak

Perancangan antar muka perangkat lunak bertujuan untuk menggambarkan bentuk rancangan atarmuka perangkat lunak yang meliputi rancangan layar utama, rancangan layar ekstrak video, rancangan layar pewarnaan *frame*, rancangan layar rendering dan rancangan layar hasil pewarnaan.

## IV. PEMBAHASAN

### A. Lingkungan Implementasi Perangkat Lunak

Perangkat lunak konversi video digital greyscale menjadi video digital berwarna dikembangkan pada lingkungan perangkat keras komputer (*notebook*) yang memiliki spesifikasi sebagai berikut:

- Monitor 14 inci.
- Processor Intel Core 2 Duo T6670 2.20 GHz.
- Random access memory (RAM) 2.00 GB.

Untuk lingkungan perangkat lunak, aplikasi konversi video digital greyscale menjadi video digital berwarna dibangun pada lingkungan sistem operasi Ubuntu 12.04.1 LTS. Sedangkan untuk bahasa pemrograman yang digunakan adalah bahasa pemrograman *Java* dengan *java version 1.7.0\_10* dan editor yang digunakan adalah *Netbeans 7.2.1*. Selain itu peneliti juga menggunakan aplikasi *FFmpeg* untuk proses pengolahan video.

### B. Implementasi Arsitektur Perangkat Lunak Konversi Video Digital Greyscale Menjadi Video Digital Berwarna

Structure chart perangkat lunak aplikasi pewarnaan citra greyscale diimplementasikan menggunakan bahasa pemrograman *Java* dengan

editor Netbeans 7.2.1. Tabel 1 berikut ini merupakan tabel pemetaan *class* implementasi.

Tabel 1. Tabel Pemetaan Arsitektur Terhadap *Class* Implementasi

Bagian Perancangan Arsitektur	Unit Implementasi	Penjelasan Unit Implementasi
Proses Ekstrak	<i>ClassUtama.java</i>	Pada <i>class</i> ini terdapat <i>method</i> untuk mengekstrak video <i>greyscale</i> yang di-inputkan oleh <i>user</i> menjadi <i>frame-frame</i> dengan format <i>bitmap</i> dan menjadi sebuah <i>file</i> audio dengan format <i>mp3</i> .
Proses Pewarnaan	<i>ClassUtama.java</i>	Pada <i>class</i> ini terdapat beberapa <i>method</i> untuk meminta <i>input-an</i> dari <i>user</i> yang terkait dengan proses pewarnaan
	<i>ImageProcessing.java</i>	Pada <i>class</i> ini diimplementasikan <i>method-method</i> untuk melakukan proses pewarnaan yang sesuai dengan metode <i>Global Image Matching</i> .
	<i>ImageMatrix.java</i>	Pada <i>class</i> ini diimplementasikan <i>method-method</i> untuk melakukan pembacaan <i>file</i> citra, untuk mendapatkan nilai RGB dari suatu citra, untuk mengubah nilai-nilai RGB menjadi suatu <i>file</i> citra.
	<i>ImagePreview.java</i>	<i>Class</i> ini digunakan untuk menampilkan hasil dari <i>testing</i> pewarnaan <i>frame greyscale</i> .
Proses Rendering	<i>ClassUtama.java</i>	Pada <i>class</i> <i>ClassUtama.java</i> terdapat <i>method</i> untuk menggabungkan <i>frame-frame</i> hasil pewarnaan dan <i>file</i> audio menjadi sebuah video.

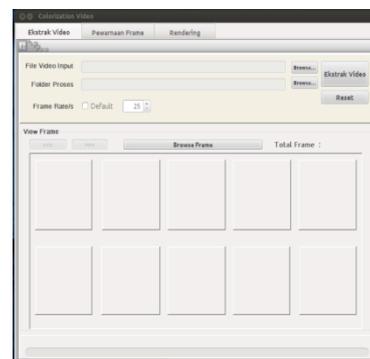
### C. Implementasi Layar Antarmuka Perangkat Lunak

Rancangan layar antarmuka perangkat lunak aplikasi konversi video digital *greyscale* menjadi video digital berwarna diimplementasikan menggunakan *class-class* yang terdapat pada *frame* di editor Netbeans 7.2.1 IDE. Gambar 7

sampai Gambar 11 merupakan implementasi dari rancangan antarmuka yang direncanakan.

#### 1. Form Utama Perangkat Lunak

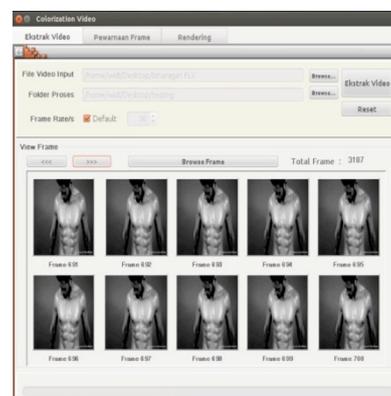
Form utama merupakan *form* yang pertama kali tampil saat aplikasi dijalankan. Dalam aplikasi konversi video digital *greyscale* menjadi video digital berwarna, *form* utama dibagi menjadi tiga *tab* yaitu *tab* ekstrak video, *tab* pewarnaan *frame* dan *tab* rendering. Implementasi *form* utama seperti pada Gambar 7 berikut.



Gambar 7. Form Utama Perangkat Lunak

#### 2. Tab Ekstrak Video

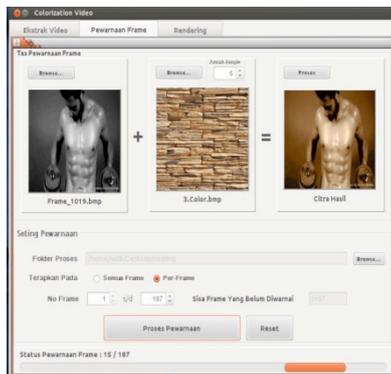
*Tab* ekstrak video merupakan tempat *user* untuk melakukan proses ekstrak video *greyscale* menjadi *frame-frame* dan *file* audio. *File-file frame* hasil dari ekstraksi video akan diberi nama *Frame\_1*, *Frame\_2*...*Frame\_n*, dimana *n* merupakan total *frame* hasil ekstraksi. Total *frame* diperoleh dari lama durasi video dalam detik dikali dengan nilai *frame rate/second* yang di-inputkan *user*. Setelah proses ekstrak selesai, *frame-frame* akan ditampilkan pada panel *view frame*. pada Gambar 8 merupakan implementasi dari *tab* ekstrak video.



Gambar 8. Tab Ekstrak Video

### 3. Layar Tab Pewarnaan Frame

Tab pewarnaan *frame* merupakan implementasi dari metode *Global Image Matching*, dimana *user* dapat melakukan proses pewarnaan *frame-frame* hasil dari ekstraksi video *greyscale*. Dalam tab pewarnaan *frame* terdapat bagian untuk melakukan *testing* pewarnaan pada salah satu *frame* yang akan diwarnai, sehingga *user* dapat menentukan citra warna acuan yang tepat untuk proses pewarnaan. Proses pewarnaan pada tab pewarnaan *frame* dapat dilakukan pada semua *frame* atau beberapa *frame* saja dengan memilih *option* "Terapkan Pada". Adapun implementasi dari tab pewarnaan *frame* adalah seperti Gambar 9.



Gambar 9. Tab Pewarnaan *Frame*

### 4. Form Hasil Pewarnaan

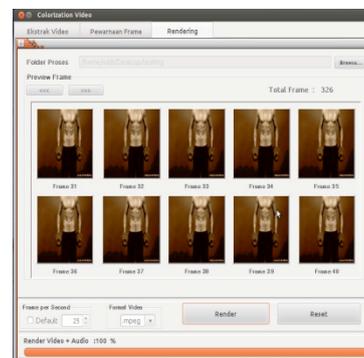
*Form* hasil pewarnaan adalah *form* yang digunakan untuk menampilkan lebih detail *frame* hasil *testing* pewarnaan. Pada *form* ini pengguna dapat melakukan *zoom in*, *zoom out* dan pengembalian ke ukuran semula dari *frame* yang dilihat. Gambar 10 adalah implementasi dari *form* hasil pewarnaan.



Gambar 10. *Form* Hasil Pewarnaan

### 5. Tab Rendering

Tab ini merupakan tab yang digunakan dalam penggabungan *frame-frame* hasil proses pewarnaan dengan audio dari hasil ekstraksi video *greyscale* menjadi video digital berwarna. Proses *render* baru bisa dilakukan jika semua *frame-frame* hasil ekstraksi diwarnai. Dalam *tabrendering*, *user* akan memilih format *filevideo* yang tersedia untuk menjadi keluaran dari aplikasi konversi video digital *greyscale* menjadi video digital berwarna. *Format file* video yang dapat dipilih adalah AVI, FLV, WEBM, MKV, MPEG, MP4, dan WMV. Gambar 11. merupakan implementasi dari *tabrendering*.



Gambar 11. *Tab Rendering*

### D. Pengujian Perangkat Lunak

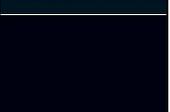
Secara umum hasil pengujian fungsional menunjukkan bahwa sistem sudah dapat menangani data masukan yang tidak valid dan valid dan dapat melakukan proses pewarnaan video *greyscale* menjadi video berwarna dengan baik. Berikut akan ditampilkan hasil perbandingan *frame-frame* hasil pengujian aplikasi konversi video digital *greyscale* menjadi video digital berwarna.

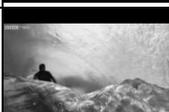
#### 1. Video "Surfer (Old Times)"

Video "Surfer (Old Times)" dapat anda lihat di situs Youtube dengan alamat <http://www.youtube.com/watch?v=YqhUDHLAsU>. Video ini berdurasi 46 detik. Dengan nilai fps yaitu 26, *frame* yang didapat setelah dilakukan ekstraksi yaitu 1216 *frame* dengan resolusi *frame* 320 x 240 *pixel*. Pada proses pewarnaan, pewarnaan dilakukan pada semua *frame* yang mengacu pada 1 citra warna acuan. Waktu yang dibutuhkan untuk melakukan pewarnaan pada 1216 *frame* yaitu 335711 milidetik, sehingga waktu proses per *frame*-nya yaitu 276,08 milidetik. Tabel 2 berikut menampilkan

perbandingan beberapa *frame* sebelum dan sesudah diwarnai.

Tabel 2. Hasil Perbandingan *Frame* Sebelum Diwarnai dan Setelah Diwarnai pada Video "Surfer (Old Times)"

Hasil Pengujian			
Citra Warna Acuan			
Pixel Sampel = 15			
No Frame	Frame Greyscale	Frame Output	T
1			276,08
100			
200			
300			
500			
650			
800			
900			

1000		
1150		
1216		

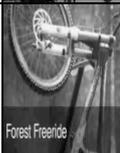
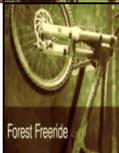
Ket : T= waktu proses per *frame* (milyardetik)

## 2. Video "Forest Freeride"

File video "Forest Freeride" merupakan video yang berdurasi 2 menit 7 detik. Video "Forest Freeride" dapat anda saksikan di situs Youtube dengan alamat [http://www.youtube.com/watch?v=3u\\_Vi4Eb\\_nQ](http://www.youtube.com/watch?v=3u_Vi4Eb_nQ). Setelah dilakukan ekstraksi dengan fps sama dengan 30, didapatkan *frame* sebanyak 3812 *frame* dengan resolusi 426 x 240 *pixel*. Pada proses pewarnaan video "Forest Freeride" ini menggunakan beberapa citra warna acuan. Pemakaian citra warna acuan yang lebih dari satu disebabkan karena pada video ini terdiri dari latar tempat yang berbeda-beda. Tabel hasil perbandingan *frame* sebelum dan sesudah diwarnai ditunjukkan pada Tabel 3 berikut.

Tabel 3. Hasil Perbandingan *Frame* Sebelum Diwarnai dan Setelah Diwarnai pada Video "Forest Freeride"

No	Input		Frame Output	S	T
	Frame Greyscale	Citra Warna Acuan			
104				20	632
140					
200					

No	Input		Frame Output	S	T
	Frame Greyscale	Citra Warna Acuan			
220				50	937
300					
480				30	676
590					
1110				50	
1900				30	
3415				30	

Ket : S = Jumlah sampel *pixel*  
 T = Waktu proses per *frame* (milidetik)

#### E. Evaluasi Hasil Pengujian

Berdasarkan hasil pengujian di atas, diketahui bahwa aplikasi konversi video digital *greyscale* menjadi video digital berwarna sudah mampu melakukan proses pewarnaan pada video *greyscale*. Hal ini dibuktikan dengan terjadinya perubahan warna pada *frame greyscale* yang awal mulanya hanya memiliki nuansa warna keabuan saja tapi setelah dilakukan proses pewarnaan maka *frame* yang dihasilkan memiliki warna yang beragam (*true color*). Keberhasilan sistem sangat

ditentukan oleh citra warna acuan yang di-*input*-kan. Citra warna acuan yang memiliki tingkat kecerahan yang mendekati tingkat kecerahan *frame greyscale* akan menyebabkan hasil pewarnaan lebih baik jika dibandingkan dengan citra warna acuan yang memiliki tingkat kecerahan yang sangat berbeda dengan *frame greyscale*.

## V. PENUTUP

### A. Simpulan

Berdasarkan penelitian yang telah dilakukan diperoleh simpulan sebagai berikut.

1. Implementasi metode *Global Image Matching* pada aplikasi konversi video digital *greyscale* menjadi video digital berwarna menghasilkan suatu aplikasi yang mampu melakukan pewarnaan video *greyscale* dengan cepat dan mudah sehingga dapat dilakukan oleh semua orang.
2. Kecepatan dalam pewarnaan video *greyscale* sangat ditentukan oleh jumlah *pixel* sampel yang di-*input*-kan.
3. Hasil pewarnaan video yang diperoleh sangat ditentukan oleh citra warna acuan yang di-*input*-kan.

Untuk pengembangan lebih lanjut dari aplikasi konversi video digital *greyscale* menjadi video digital berwarna, adapun saran yang dapat diberikan adalah sebagai berikut.

- a. Adanya pengembangan algoritma penyesuaian tingkat kecerahan yang lebih spesifik sehingga menghasilkan tingkat kecerahan *frame greyscale* dan citra warna acuan akan memiliki lebih baik dan nantinya akan berimbang pada hasil pewarnaan yang lebih bagus.
- b. Adanya pengembangan algoritma yang dapat melakukan restorasi pada *frame* hasil pewarnaan, sehingga dapat meminimalisir bercak-bercak yang dihasilkan dari proses pewarnaan *frame*.
- c. Adanya pengembangan algoritma yang dapat mendeteksi bentuk, sehingga kita dapat menentukan dengan lebih tepat objek yang akan diwarnai dari suatu *frame* hasil ekstraksi.

## REFERENSI

- [1] Pusat Pengembangan Bahan Ajar Universitas Mercu Buana . 2011. *Video & Televisi*. Tersedia pada <http://kk.mercubuana.ac.id/files/92052-3-574637347358.doc>.
- [2] Putra, Darma.2010. *Pengolahan Citra Digital*. Yogyakarta: Andi.

- [3] Budiman, Asep. 2009. Aplikasi *Steganography* pada Video dengan Metode *Least Significant Bit* (LSB). Skripsi (tidak diterbitkan). Teknik Informatika S1, Universitas Indonesia. Tersedia pada <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-asepbudima-18351>.
- [4] Andikaputra, Gusti Bagus Agung Pradnya. 2011. Karakteristik Video Digital. Tersedia pada [http://digilib.itelkom.ac.id/index.php?option=com\\_content&view=article&id=866:karakteristik-video-digital&catid=21:itp-informatika-teori-dan-pemograman&Itemid=14](http://digilib.itelkom.ac.id/index.php?option=com_content&view=article&id=866:karakteristik-video-digital&catid=21:itp-informatika-teori-dan-pemograman&Itemid=14) (diakses pada tanggal 20 Desember 2012).
- [5] <http://id.wikipedia.org/wiki/FFmpegpdf> (diakses tgl 18 Desember 2012).
- [6] Dewi, Lilyana. 2003. Perencanaan Dan Pembuatan Aplikasi Untuk Transfer Warna Ke Gambar *Greyscale* Dengan Metode *Global Image Matching*. Skripsi (tidak diterbitkan). Teknik Informatika S1, Universitas Kristen Petra. Tersedia pada [http://dewey.petra.ac.id/jiunkpe\\_dg\\_1519.html](http://dewey.petra.ac.id/jiunkpe_dg_1519.html).
- [7] Welsh, et.al. 2002. "Transferring Color to *Greyscale* Image". Tersedia pada [www.cse.iitd.ernet.in/~pkalra/csl783/assignment1/colorize.pdf](http://www.cse.iitd.ernet.in/~pkalra/csl783/assignment1/colorize.pdf) (diakses tgl 17 Desember 2012).
- [8] Putra, I Made Agus Wirahadi. 2012. Pengembangan Aplikasi Citra Digital Untuk Mengubah Citra *Greyscale* Menjadi Citra Berwarna. Skripsi (tidak diterbitkan). Jurusan Pendidikan Teknik Informatika, Undiksha Singaraja.