

# PENGEMBANGAN APLIKASI PENGUBAH LATAR BELAKANG VIDEO

Rudy Adipranata<sup>1</sup>, Djoni Haryadi Setiabudi<sup>2</sup>, Henry Tedjowarsito  
Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra,  
Jl. Siwalankerto 121-131, Surabaya 60236  
E-mail : rudya@petra.ac.id<sup>1</sup>, djonih@petra.ac.id<sup>2</sup>

**ABSTRAK:** Pada penelitian ini dikembangkan aplikasi untuk mengubah latar belakang video. Aplikasi pengubahan latar belakang video ini banyak digunakan pada *broadcasting* untuk menghasilkan tayangan yang lebih menarik. Dengan menggunakan aplikasi ini, obyek seperti penyiar dapat berada pada studio yang mempunyai latar belakang statis dan kemudian dilakukan penggantian sehingga latar belakang menjadi lebih baik atau dinamis. Input dapat berasal dari webcam atau dapat pula berasal dari dua buah file video, dimana satu file adalah video sumber dengan latar belakang yang akan diganti dan file yang lain adalah latar belakang pengganti. Aplikasi dikembangkan dengan menggunakan metode untuk melakukan pendeteksian latar belakang dan melakukan proses penggantian latar belakang yang telah terdeteksi tersebut. Output aplikasi dapat disimpan menjadi file video.

**Kata kunci:** pengubahan latar belakang video, *broadcasting*

**ABSTRACT:** At this research we developed an application to substitute video background. The application can be used in broadcasting to broadcast more interesting program. By using this application, the object for example presenter, can stay in studio having static background and then the application will do background substitution to show better or dynamic background. Input for this application could come from webcam or two video files, where one file is the video source with background which will be substituted and the other file is the substitution background. The application is developed by using method to do detection of background and the substitution will be done for the detected background. Output of the application could be saved in video file.

**Keywords:** video background substitution, *broadcasting*

## PENDAHULUAN

Sebagai salah satu bagian yang cukup penting dalam dunia hiburan, *broadcasting* merupakan bagian yang cukup rumit. Biasanya acara *broadcasting* memerlukan latar belakang yang beragam sehingga acara menjadi lebih menarik. Pembuatan latar belakang secara fisik yang bagus dan menarik membutuhkan biaya yang cukup tinggi karena membutuhkan materi-materi yang banyak dan beragam serta tingkat kesulitan pengerjaan yang tinggi.

Sehubungan dengan kemajuan teknologi informasi, sekarang ini dimungkinkan untuk membuat latar belakang dengan bantuan komputer. Permasalahan yang timbul adalah bagaimana menggantikan latar belakang asli yang telah direkam dengan latar belakang buatan komputer tersebut.

Pada penelitian ini dilakukan pengembangan aplikasi untuk mengubah latar belakang video. Dengan menggunakan aplikasi ini, latar belakang obyek utama dapat diganti dengan latar belakang lain sehingga menghasilkan tampilan yang menarik. Sebagai contoh, penyiar dapat melakukan pembacaan

berita dari studio yang mempunyai latar belakang hanya berupa dinding berwarna polos. Sebelum disiarkan, terlebih dahulu dilakukan penggantian latar belakang dengan menggunakan aplikasi ini, sehingga tampilan menjadi lebih menarik.

## FORMAT FILE AVI

AVI (*Audio Video Interleave*) adalah sebuah *file* video standar yang didefinisikan oleh Microsoft [1]. Pada format *file* AVI dimungkinkan untuk terdapat banyak *data streams* yang berupa *video stream* atau *audio stream*. Dalam sebuah *file* AVI, jumlah maksimum *video stream* yang dapat disimpan adalah satu buah. Sedangkan untuk *audio stream*, sebuah *file* AVI dapat menyimpan satu atau lebih *audio stream*. Tetapi juga dimungkinkan dalam sebuah *file* AVI tidak terdapat *audio stream* sama sekali.

Untuk dapat melakukan pembacaan *file* AVI, dapat digunakan *file* DLL (*Dynamic Link Library*) yang telah disediakan oleh sistem operasi Windows, diantaranya adalah [1]:

- MSVFW32.DLL  
(*Microsoft Video for Windows DLL - NT 4.0*)

- AVIFIL32.DLL  
(AVIFILE API for Reading and Writing AVI Files and Streams)
- AVICAP32.DLL  
(AVI Capture Window Class)
- MCIAVI32.DLL  
(Video for Windows MCI Driver)
- MSACM32.DRV  
(Microsoft Audio Compression Manager)
- MSACM32.DLL  
(More Microsoft Audio Compression Manager)
- MSRLE32.DLL  
(Microsoft RLE Video Codec)
- IR32\_32.DLL  
(Intel Indeo 3.2 Video Codec)
- MSVIDC32.DLL  
(Microsoft Video 1 Codec)
- ICCVID.DLL  
(Cinepak for Windows 32 - Radius)

**MODEL WARNA HSV**

Model warna HSV mempunyai tiga buah komponen warna yaitu *hue* (*H*), *saturation* (*S*), dan *value* (*V*) [2]. Model warna HSV juga dikenal dengan model warna HSB (*Hue, Saturation, Brightness*). Karakteristik komponen warna HSV adalah sebagai berikut:

- Komponen *Hue* (*H*) merepresentasikan ukuran komposisi secara spektral dari sebuah warna, di mana memiliki nilai dinyatakan dalam besaran sudut, antara 0° hingga 360°.
- Komponen *Saturation* (*S*) merepresentasikan kejernihan dari sebuah warna, dengan nilai antara 0 hingga 1.
- Komponen *Value* (*V*) merepresentasikan tingkat *brightness* dari suatu warna. Nilai *value* (*V*) mempunyai jarak antara 0 hingga 1.

Model warna HSV dapat diperoleh dengan melakukan transformasi dari model warna RGB dengan menggunakan formula berikut [2]:

$$H = \begin{cases} H1, & \text{if } B \leq G \\ 360^\circ - H1, & \text{if } B > G \end{cases} \quad (1)$$

$$H1 = \cos^{-1} \left\{ \frac{0.5[(R-G) + (R-B)]}{\sqrt{(R-G)^2 + (R-B)(R-B)}} \right\} \quad (2)$$

$$V = \frac{1}{3}(R + G + B) \quad (3)$$

$$S = 1 - \frac{3}{(R + G + B)}[\min(R, G, B)] \quad (4)$$

Model warna HSV dapat pula diubah menjadi model RGB dengan menggunakan formula berikut [2]:

Jika  $0^\circ \leq H < 120^\circ$ ,

$$B = V(1 - S) \quad (5)$$

$$R = V \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad (6)$$

$$G = 1 - (R + B) \quad (7)$$

Jika  $120^\circ \leq H < 240^\circ$ ,

$$R = V(1 - S) \quad (8)$$

$$G = V \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad (9)$$

$$B = 1 - (R + G) \quad (10)$$

Jika  $240^\circ \leq H \leq 360^\circ$ ,

$$G = V(1 - S) \quad (11)$$

$$B = V \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad (12)$$

$$R = 1 - (G + B) \quad (13)$$

**MODEL WARNA YCrCb**

YCrCb memiliki tiga buah komponen warna yaitu *luma* (*Y*), *red chroma* (*Cr*) dan *blue chroma* (*Cb*). Untuk dapat mengubah model RGB ke YCrCb dapat digunakan formula berikut [3]:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \frac{1}{256} \begin{bmatrix} 65 & 129.057 & 25.064 \\ -37.945 & -74.494 & 112.439 \\ 112.439 & -94.154 & -18.285 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (14)$$

Sedangkan untuk mengubah model YCrCb ke model RGB dilakukan *inverse* dari bentuk sebelumnya sebagai berikut [3]:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{256} \begin{bmatrix} 298.082 & 0 & 408.583 \\ 298.082 & -100.291 & -208.120 \\ 298.082 & 516.411 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} \quad (15)$$

**KEMIRIPAN PADA JARAK VEKTOR**

Kemiripan pada jarak vektor dapat dihitung pada vektor RGB, HSV, ataupun YCrCb vektor. Dengan

merepresentasikan warna dari suatu piksel pada sumbu XYZ, maka dapat dihasilkan suatu titik seperti yang ditunjukkan pada Gambar 1. Panjang vektor yang dihasilkan dari 2 titik pada 2 piksel yang berbeda dengan asumsi X sebagai nilai dari *red*, Y sebagai nilai dari *green*, dan Z sebagai nilai dari *blue* dapat dihitung melalui formula sebagai berikut [4]:

$$D(a,b) = [(X_{(a)}-X_{(b)})^2 + (Y_{(a)}-Y_{(b)})^2 + (Z_{(a)}-Z_{(b)})^2]^{1/2} \quad (16)$$

D = panjang vektor

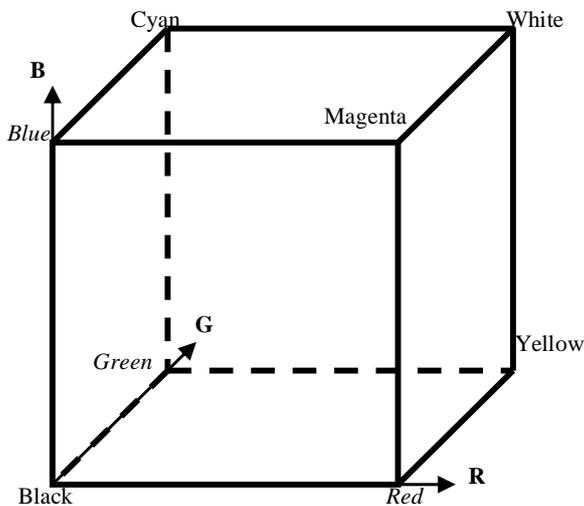
a = piksel ke a

b = piksel ke b

$X_{(n)}$  = nilai X dari piksel n

$Y_{(n)}$  = nilai Y dari piksel n

$Z_{(n)}$  = nilai Z dari piksel n

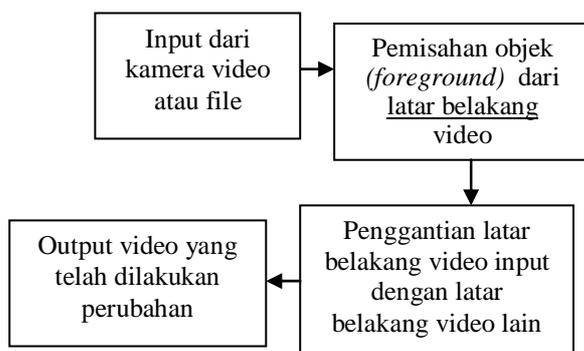


Gambar 1. RGB pada sumbu XYZ

Panjang vektor yang dihasilkan mewakili kemiripan warna antar 2 piksel tersebut.

### PERENCANAAN SISTEM

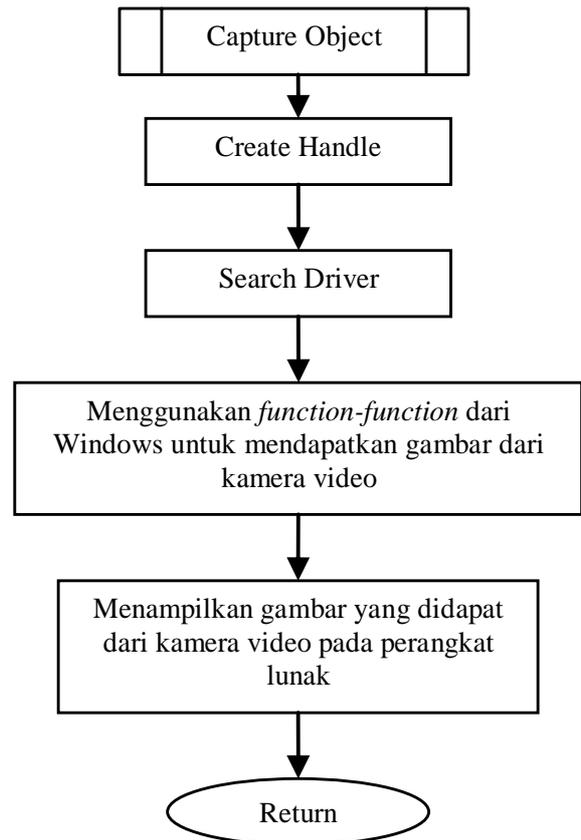
Secara garis besar, cara kerja sistem aplikasi yang dikembangkan dapat dilihat pada blok diagram di Gambar 2.



Gambar 2. Blok Diagram Sistem

Proses input dari kamera video dapat dilihat pada diagram alir pada Gambar 3.

Hal yang dilakukan pada proses input dari kamera video pertama kali adalah membuat *handle* dan mencari *driver* yang digunakan oleh kamera. Kemudian digunakan fungsi dari Windows menggunakan *library* seperti terdapat pada bagian 2 untuk mendapatkan data *stream* dari kamera dan kemudian menampilkan data *stream* tersebut ke layar monitor.



Gambar 3. Diagram Alir Input Kamera

Setelah mendapat data *stream* dari kamera video, maka dapat dilakukan perubahan latar belakang input dengan menggunakan latar belakang pengganti. Terdapat beberapa metode yang dapat digunakan untuk melakukan penggantian ini, yaitu *RGB color range*, *HSV color range*, *RGB similarity*, *HSV similarity*, dan *YcrCb similarity*.

*RGB Color range* merupakan metode untuk membandingkan nilai dari suatu warna (RGB) secara langsung. Pada metode ini dibutuhkan nilai *min*, *max* serta warna pembanding yang dapat diatur oleh pengguna. Nilai warna pembanding adalah nilai warna latar belakang yang digunakan pada input video sedangkan nilai *min* serta *max* adalah nilai toleransi dari warna pembanding tersebut. Proses perubahan

dilakukan pada semua *frame* video. Untuk setiap *frame* akan dilihat warna setiap piksel, dimana jika warna tersebut terletak antara warna pembanding – *min* dan warna pembanding + *max*, maka piksel tersebut akan diganti dengan piksel pada latar belakang pengganti yang mempunyai lokasi sama.

Serupa dengan *RGB color range*, *HSV color range* juga melakukan perbandingan antara nilai HSV video input terhadap range nilai HSV pembanding yang ditentukan oleh pengguna. Jika nilai HSV video input terletak di antara range HSV pembanding, maka dilakukan penggantian piksel.

Metode *RGB similarity*, *HSV similarity* dan *YCrCb similarity* melakukan perhitungan jarak antara nilai input dengan nilai pembanding menggunakan formula (16). Jika hasil perhitungan jarak lebih kecil dari nilai *threshold*, maka dilakukan penggantian piksel pada *frame* input dengan piksel pada latar belakang pengganti yang mempunyai lokasi sama. Nilai pembanding ditentukan oleh pengguna dimana nilai ini adalah nilai warna latar belakang yang digunakan pada input video. Begitu pula nilai *threshold* juga ditentukan oleh pengguna. Diagram alir metode *RGB similarity* dapat dilihat pada Gambar 4.

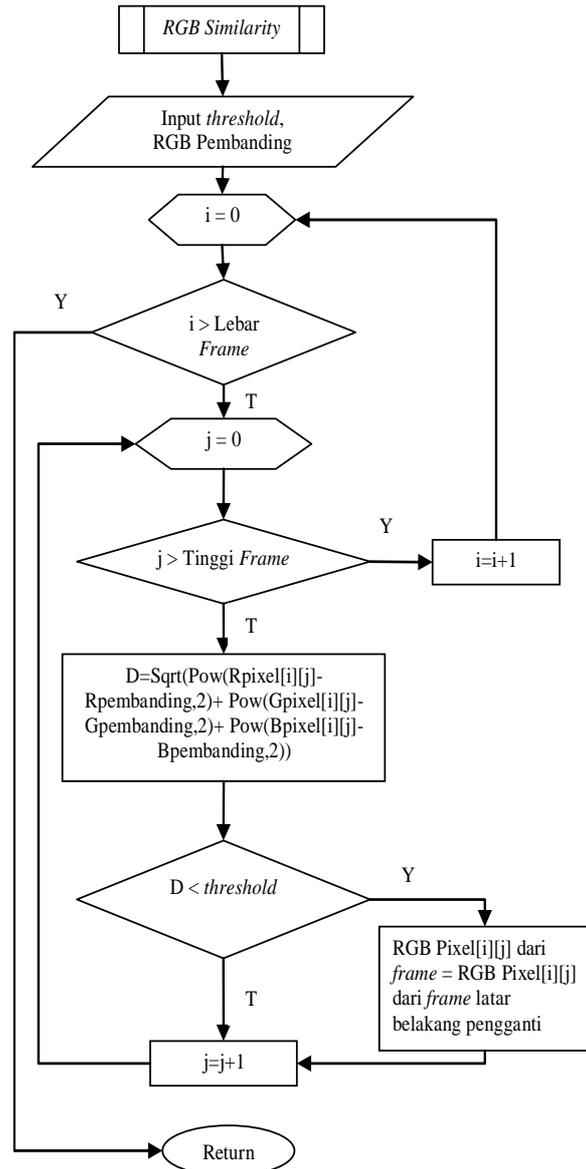
Metode *HSV similarity* serta *YCrCb similarity* mempunyai langkah-langkah yang hampir sama seperti *RGB similarity*, hanya saja yang dihitung jaraknya bukan komponen RGB melainkan komponen HSV atau YCrCb. Sama seperti metode *color range*, metode *RGB similarity*, *HSV similarity* serta *YCrCb similarity* juga diaplikasikan pada setiap *frame*.

**IMPLEMENTASI DAN PENGUJIAN**

Aplikasi diimplementasikan dengan menggunakan bahasa pemrograman Delphi 7.0, sedangkan untuk pengujian, digunakan komputer dengan spesifikasi sebagai berikut:

- *Processor Intel Pentium IV – 2,8 GHz*
- *Memory 1GB DDR RAM*
- *VGA Card ATI Radeon 9200 128 MB*
- *Sistem operasi Microsoft Windows XP Service Pack 2*

Pada Gambar 5 ditampilkan salah satu *frame* input video yang akan diganti latar belakangnya, dimana latar belakang asli menggunakan layar berwarna hijau.



**Gambar 4. Diagram Alir RGB Similarity**



**Gambar 5. Frame Input Video**

Pengujian pertama dilakukan dengan menggunakan metode *RGB color range* dimana warna pembanding adalah hijau sesuai dengan warna latar belakang video input. Hasil penggantian latar belakang video input dengan latar belakang pengganti dapat dilihat pada Gambar 6. Dari hasil penggantian tersebut terlihat bahwa latar belakang hasil masih mempunyai cukup banyak bagian yang merupakan latar belakang asli yaitu pada bagian sebelah kanan atas kepala, sudut kanan atas gambar serta sudut kanan bawah gambar.

Pengujian kedua dilakukan dengan menggunakan metode *HSV color range*. Sama seperti *RGB color range*, nilai pembanding adalah warna hijau yang telah diubah ke komponen warna HSV. Hasil proses dapat dilihat pada Gambar 7.

Dari hasil pengujian tersebut terlihat bahwa semua bagian latar belakang asli telah tergantikan dengan latar belakang pengganti. Tidak lagi terlihat latar belakang asli seperti pada pengujian sebelumnya.

Pengujian berikutnya dilakukan dengan menggunakan metode *RGB similarity*. Warna pembanding yang digunakan adalah warna hijau sesuai dengan warna latar belakang video input. Hasil proses dapat dilihat pada Gambar 8. Dari hasil pengujian tersebut dapat dilihat bahwa terdapat bagian yang masih belum dapat diganti yaitu di sebelah kanan atas kepala serta di sudut kanan atas gambar.

Dilakukan pula pengujian dengan metode *HSV Similarity* menggunakan parameter seperti pada pengujian yang lainnya, hanya saja warna pembanding telah diubah menjadi model warna HSV. Hasil pengujian ini dapat dilihat pada Gambar 9.

Pengujian dengan metode *HSV Similarity* ini menghasilkan hasil yang cukup baik, mirip dengan hasil yang didapat dengan menggunakan metode *HSV color range*, dimana hampir semua latar belakang asli telah dapat diganti dengan latar belakang pengganti. Hanya saja masih terdapat sedikit bagian yang masih belum dapat diganti yaitu di sebelah kanan atas kepala.



Gambar 6. Hasil Proses Menggunakan Metode *RGB Color Range*



Gambar 8. Hasil Proses Menggunakan Metode *RGB Similarity*



Gambar 7. Hasil Proses Menggunakan *HSV Color Range*



Gambar 9. Hasil Proses Menggunakan *HSV Similarity*

Untuk hasil pengujian dengan menggunakan metode *YCbCr Similarity*, dapat dilihat pada Gambar 10. Hasil pengujian dengan metode tersebut menunjukkan bahwa masih terdapat cukup banyak bagian latar belakang asli yang belum dapat diganti dengan latar belakang pengganti. Hal ini mirip dengan hasil yang didapat dari pengujian dengan menggunakan metode *RGB color range*.

Pengujian juga dilakukan terhadap waktu proses keempat metode yang digunakan. Dengan menggunakan *frame* berukuran 640x480 piksel, didapat rata-rata waktu proses tercepat untuk sebuah *frame* adalah dengan menggunakan metode *RGB color range*.



**Gambar 10.** Hasil Proses Menggunakan Metode *YCbCr Similarity*

## KESIMPULAN

Dari hasil pengujian yang dilakukan, dapat diambil kesimpulan bahwa hasil penggantian latar belakang yang dilakukan terbaik dengan menggunakan metode *HSV color range*, sedangkan kecepatan proses tertinggi diperoleh dengan menggunakan metode *RGB color range*.

Metode yang digunakan pada aplikasi ini semuanya membutuhkan parameter input dari pengguna sehingga memerlukan beberapa kali percobaan (*trial and error*) agar hasil yang didapat sesuai dengan harapan.

## DAFTAR PUSTAKA

1. *AVI file format*. <http://www.sonicspot.com/guide/avifile.html>. Akses terakhir 30 Desember 2008.
2. Gonzalez, R.C. dan Woods, R.E. 2002. *Digital image processing second edition*. New Jersey: Prentice Hall.
3. Hsu, R. L. 2002. *Face Detection and Modeling for Recognition*. Doctoral Thesis. Department of Computer Science and Engineering, Michigan State University, Michigan.
4. Castleman, K.R. 1996. *Digital image processing international edition*. New Jersey: Prentice Hall.