

# Implementasi Image Enhancement Menggunakan Homomorphic Filtering

Rudy Adipranata<sup>1</sup>, Cherry Galatia Ballangan<sup>2</sup>, William Susanto  
Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra  
Jl. Siwalankerto 121-131, Surabaya  
Telp. 8439040  
Email : rudy@petra.ac.id<sup>1</sup>, cherry@petra.ac.id<sup>2</sup>

## Abstract

Nowadays, the need of digital images is increasing because of the advantages of the digital image technologies. Some of the advantages are the easiness in obtaining, duplicating and processing the image. Because of the high demand of digital images, the need of the high quality digital images is also increasing. If the quality of the image is good, we expect to obtain the information needed from the image. The software, which is developed in this paper, is one of the efforts to do image enhancement.

The software is created based on homomorphic filtering. This algorithm attenuates the low frequencies data and amplifying the high frequencies of the image. The high frequencies data represent the detail of the images, therefore amplifying this frequencies will cause image enhancement and sharpening.

This software uses C++ programming language, with Microsoft Visual C++ 6.0<sup>®</sup> compiler. The experiment results show that proper Butterworth Highpass Filter's parameters choices will determines the quality of image result.

Keywords : image enhancement, homomorphic filtering

## 1. Pendahuluan

Penggunaan gambar digital semakin hari semakin meningkat karena kelebihan-kelebihan yang dimiliki oleh gambar digital tersebut, antara lain kemudahan dalam mendapatkan gambar, memperbanyak gambar, pengolahan gambar dan lain-lain. Semakin pentingnya peran gambar digital dalam kehidupan sehari-hari ini makin meningkatkan kebutuhan akan gambar digital yang bebas dari adanya *noise* sehingga gambar digital yang ada dapat terlihat dengan jelas. *Noise* yang dimaksud disini adalah detail gambar yang kurang jelas karena pencahayaan yang kurang bagus. Dari gambar digital yang jelas itu nantinya diharapkan dapat memberikan informasi yang tepat bagi orang yang hendak memanfaatkan gambar digital tersebut.

*Color constancy* atau ketetapan warna adalah salah satu keistimewaan dari sistem penglihatan manusia, yang mengusahakan agar warna yang diterima dari suatu benda terlihat sama meskipun berada pada kondisi pencahayaan yang berbeda-beda. Misalnya apel akan terlihat berwarna hijau pada saat siang hari dengan pencahayaan yang utama adalah putih matahari. Apel tersebut juga akan terlihat berwarna hijau pada saat matahari terbenam atau dengan pencahayaan berwarna merah. Hal ini yang membantu kita untuk mengidentifikasi suatu benda.

Karena kelebihan dari *color constancy* itu, maka dikembangkan algoritma yang dapat mengakomodasikan *color constancy* sehingga dapat dimanfaatkan untuk sistem penglihatan pada robot ataupun *computer vision*. Algoritma ini dikenal dengan nama algoritma *retinex*. Salah satu pengembangan dari algoritma ini adalah algoritma *homomorphic filtering*. Kunci dari algoritma ini adalah suatu desain filter yang tepat pada domain frekuensi untuk menghasilkan *image enhancement*. Pada penelitian ini, diimplementasikan image enhancement dengan menggunakan *homomorphic filtering* tersebut.

## 2. Retinex

Pada tahun 1971, Edwin Land mengungkapkan teori mengenai *Retinex* (*Color Constancy*, n.d.). *Retinex* sendiri berasal dari kata “*retina*” dan “*cortex*”, yang berarti retina dan otak pada manusia sama-sama digunakan dalam proses melihat dan mengidentifikasi obyek. Teori *retinex* digunakan untuk mengusahakan ketetapan warna / *color constancy*. Yang dimaksud *color constancy* ini adalah warna dari obyek yang dilihat diusahakan agar mempunyai warna yang relatif tetap meskipun disinari dengan kondisi pencahayaan yang berbeda-beda. Contoh dari *color constancy* ini dimiliki oleh manusia, seperti pada buah apel yang terlihat hijau pada siang hari, dengan pencahayaan utama berwarna putih sinar matahari. Buah apel itu juga akan terlihat hijau pada waktu matahari terbenam, dengan pencahayaan berwarna merah. Beberapa spesies yang bukan manusia, seperti monyet dan ikan mas juga memiliki *color constancy*.

Sebuah eksperimen untuk membuktikan *color constancy* dimiliki oleh manusia dilakukan dengan percobaan sebagai berikut (*Color Constancy*, n.d.). Sebuah lukisan berjudul “Mondrian” (diambil dari nama Piet Mondrian yang pernah melukis lukisan yang mirip), terdiri dari banyak petak-petak berwarna ditunjukkan kepada seseorang. Lukisan itu disinari oleh tiga buah sinar berwarna putih, satu sinar diproyeksikan melalui *filter* berwarna merah, satu lagi diproyeksikan melalui *filter* berwarna hijau, dan yang terakhir diproyeksikan melalui *filter* berwarna biru. Orang itu kemudian diminta untuk mengatur intensitas dari tiap-tiap sinar agar sebuah petak dari gambar itu kelihatan berwarna putih. Si peneliti kemudian mengukur intensitas dari masing-masing sinar yang dipantulkan dari obyek tadi. Orang itu kemudian diminta untuk mengidentifikasi warna dari petak yang ada disebelahnya, misalnya ternyata berwarna hijau. Si peneliti mengatur agar intensitas ketiga sinar yang dipantulkan dari petak berwarna hijau itu memiliki nilai yang sama dengan sebelumnya. Dengan *color constancy* yang dimiliki manusia, orang tadi tetap melihat warna hijau pada petak berwarna hijau, putih pada petak berwarna putih, dan juga pada petak yang lain sesuai dengan warna aslinya.

Masalah yang sering terjadi adalah apabila gambar diambil menggunakan kamera digital, warna benda dapat berubah tergantung dari kondisi pencahayaan pada saat itu. Untuk itulah diperlukan algoritma teori *retinex* yang dapat menyediakan *color constancy* pada gambar. Teori *retinex* ini berdasarkan pada model dimana setiap gambar yang ada merupakan hasil perkalian (*product*) dari dua komponen, yaitu komponen pencahayaan / *illumination part* dan komponen obyek / *reflectance part*. *Retinex* sendiri terus dikembangkan menjadi berbagai algoritma yang berbeda-beda, antara lain (Du, 2002):

- *Random Walk Algorithms* oleh Edwin Land pada tahun 1971.
- *Homomorphic Filtering* oleh Edwin Land pada tahun 1986 dan D. J. Jobson pada tahun 1997.
- *Solving Poisson Equation* oleh B. K. P. Horn pada tahun 1974.

Algoritma *retinex* yang akan dipakai adalah *homomorphic filtering*. Berdasar pada teori bahwa *image* terdiri dari dua komponen, yaitu *illumination* dan *reflectance* dan dengan mengetahui bahwa *illumination part* mendominasi frekuensi rendah dan *reflectance part* mendominasi frekuensi tinggi pada *frequency domain*, maka apabila kedua komponen itu dapat dipisahkan dengan melemahkan frekuensi rendah dan menguatkan frekuensi tinggi akan menimbulkan perbaikan (*enhancement*) dan penajaman (*sharpening*) pada *image*. Sebuah *image* mempunyai persamaan sebagai berikut:

$$f(x,y) = i(x,y) * r(x,y) \quad (1)$$

Dimana  $f(x,y)$  adalah sebuah *image* merupakan hasil perkalian (*product*) dari  $i(x,y)$  yang merupakan komponen *illumination* dengan  $r(x,y)$  yang merupakan komponen *reflectance*. Untuk mengimplementasikan *homomorphic filtering*, pertama-tama kedua komponen yang membentuk *image* ini harus bisa dipisahkan. Dengan menggunakan sifat dari logaritma, maka *product* dari kedua komponen itu dapat dipisahkan ke dalam bentuk

$$\begin{aligned} z(x,y) &= \ln f(x,y) \\ &= \ln i(x,y) + \ln r(x,y) \end{aligned} \quad (2)$$

Untuk menjaga apabila ada piksel yang bernilai 0 yang akan menimbulkan *error* pada proses logaritma, seluruh piksel akan ditambahkan dengan *offset* sebesar 1 agar terhindar dari *error* tersebut. Karena karakteristik yang dimiliki oleh komponen *illumination* dan *reflectance* bekerja pada *domain* frekuensi, maka harus dilakukan transformasi *fourier* pada *image*,

$$\begin{aligned} F\{z(x,y)\} &= F\{\ln f(x,y)\} \\ &= F\{\ln i(x,y)\} + F\{\ln r(x,y)\} \\ Z(u,v) &= Fi(u,v) + Fr(u,v) \end{aligned} \quad (3)$$

Dimana  $Fi(u,v)$  dan  $Fr(u,v)$  adalah transformasi *fourier* dari  $\ln i(x,y)$  dan  $\ln r(x,y)$ .

Setelah dipindah dalam *domain* frekuensi, barulah *image* tersebut diproses dengan menggunakan *filter* yang sesuai agar tujuan awal dapat tercapai yaitu untuk melemahkan frekuensi rendah dan memperkuat frekuensi tinggi sehingga terjadi *image enhancement* dan *image sharpening* dengan formula:

$$\begin{aligned} S(u,v) &= H(u,v) Z(u,v) \\ &= H(u,v) Fi(u,v) + H(u,v) Fr(u,v) \end{aligned} \quad (4)$$

Dimana  $S(u,v)$  adalah transformasi *fourier* dari *image* yang telah diproses. Sehingga untuk mendapatkan hasil yang sebenarnya perlu dikembalikan ke dalam *spatial domain* dengan formula (5).

$$\begin{aligned} s(x,y) &= F^{-1}\{S(u,v)\} \\ &= F^{-1}\{H(u,v) Fi(u,v)\} + F^{-1}\{H(u,v) Fr(u,v)\} \\ &= i'(x,y) + r'(x,y) \end{aligned} \quad (5)$$

Langkah terakhir adalah menghilangkan operasi logaritma yang dilakukan diawal proses dengan cara melakukan operasi eksponensial sehingga dapat diperoleh *enhanced image* yang diinginkan dilambangkan dengan  $g(x,y)$ , yaitu

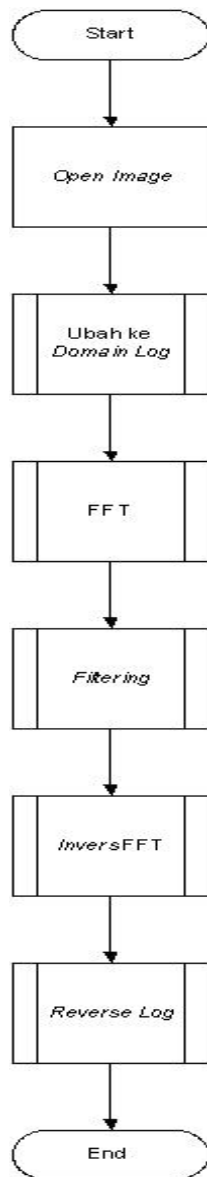
$$\begin{aligned} g(x,y) &= \exp(s(x,y)) \\ &= e^{i'(x,y)} * e^{r'(x,y)} \end{aligned} \quad (6)$$

### 3. Desain Sistem

Secara garis besar, sistem dari algoritma *homomorphic filtering* ini dibagi menjadi 5 bagian, yaitu:

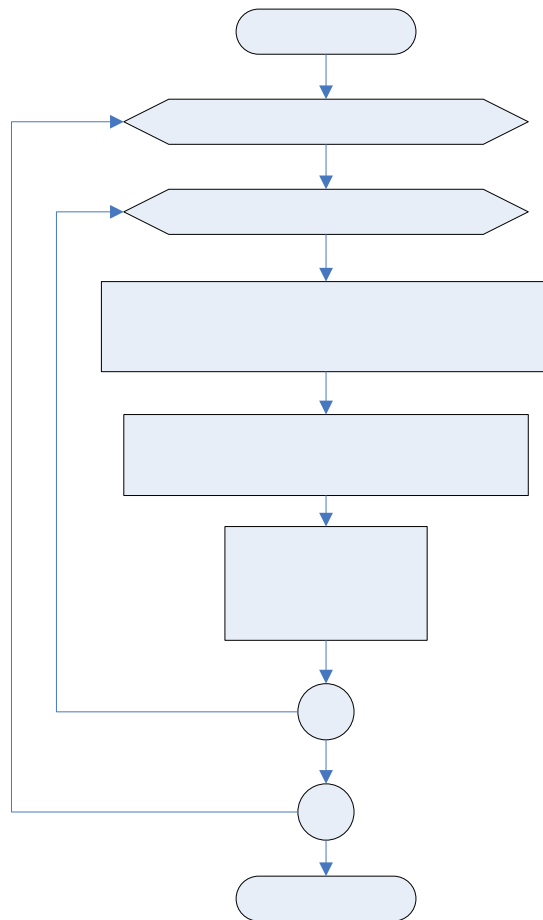
1. Mengubah gambar ke dalam *log domain*.
2. Melakukan transformasi *fourier* pada gambar.
3. Melakukan proses *filter* terhadap gambar.
4. Melakukan *inverse fourier* pada gambar.
5. Menghilangkan sifat *log* dari gambar.

Untuk menjalankan algoritma ini diperlukan sebuah *image* yang hendak diproses. *Image* kemudian diubah dalam *log domain* sebelum dikenai transformasi *fourier*. Setelah diubah dalam *frequency domain*, barulah *image* dapat dikenakan *filter* yang bekerja pada domain frekuensi. Apabila proses *filter* telah selesai, *image* dikembalikan lagi ke *spatial domain* dengan menggunakan *inverse fourier*. Setelah itu untuk menghilangkan *log* yang dikenakan di awal algoritma, *image* diproses dengan menggunakan *exponential* agar nilainya kembali semula. *Flowchart* keseluruhan sistem dapat dilihat pada gambar 1.



Gambar 1. *Flowchart Homomorphic Filtering*

Sebagai filter yang digunakan untuk melemahkan frekuensi rendah dan memperkuat frekuensi tinggi, digunakan *Butterworth Highpass Filter* yang dimodifikasi dengan menggunakan *High Frequency Emphasis Filtering*. Flowchart dari *Butterworth Highpass Filter* yang telah dimodifikasi tersebut terdapat pada gambar 2 di bawah ini.



Gambar 2. Flowchart Butterworth Highpass Filter

#### 4. Implementasi dan Hasil Pengujian

Pada penelitian ini, implementasi dilakukan dengan bahasa pemrograman Visual C++ 6.0 dan dilakukan pengujian terhadap gambar dengan melakukan pengubahan pada parameter yang terdapat pada *Butterworth Highpass Filter*. Berikut ini ditampilkan hasil pengujian pada gambar-gambar di bawah.



Gambar 3. Gambar Input yang Digunakan

Dengan menggunakan beberapa kombinasi parameter yang terdapat pada tabel 1, maka dapat dilihat gambar hasil setelah proses *image enhancement* pada gambar 4 hingga 7.

Tabel 1. Kombinasi Parameter *Butterworth Highpass Filter*

Parameter	Output - 1 (Gambar 4)	Output - 2 (Gambar 5)	Output - 3 (Gambar 6)	Output - 4 (Gambar 7)
<i>Cutoff Freq.</i>	800	800	800	500
<i>Order</i>	2	2	2	2
<i>Alpha</i>	0.9	0.9	0.9	0.9
<i>Beta</i>	0.2	0.4	0.42	0.4



Gambar 4. Gambar Output-1



Gambar 5. Gambar Output-2



Gambar 6. Gambar Output-3



Gambar 7. Gambar Output-4

Dari Output-1, diperoleh gambar yang lebih terang dari gambar aslinya. Demikian juga pada Output-2, hanya saja pada gambar Output-2 gambar terlihat lebih berwarna. Hal ini disebabkan oleh parameter *beta* dimana parameter ini merupakan modifikasi dari *highpass filter*. Pada gambar Output-3 dapat dilihat bahwa penambahan pada parameter *beta* mengakibatkan gambar menjadi lebih terang. Dengan membandingkan nilai parameter dari Output-2 dan Output-4 dapat dilihat juga bahwa parameter *cutoff frequency* juga berperan pada terangnya sebuah gambar, dimana nilai parameter *cutoff frequency* yang lebih kecil membuat gambar menjadi lebih terang.

Disamping melakukan pengujian terhadap perubahan parameter Butterworth Highpass Filter, dilakukan juga pengujian dengan membandingkan hasil *homomorphic filtering* dengan metode image enhancement lain yaitu *histogram equalization*. Hasil dari kedua *filter* terdapat pada gambar di bawah.



Gambar 8. Hasil *Homomorphic Filtering*



Gambar 9. Hasil *Histogram Equalization*

Dari hasil *histogram equalization*, terlihat bahwa detail warna terlihat dengan jelas tetapi gambar yang dihasilkan menjadi tidak serupa dengan aslinya. Sedangkan hasil *homomorphic filtering*, mempertahankan gambar sehingga tetap serupa dengan aslinya.

## 5. Kesimpulan

Berdasarkan pengujian yang telah dilakukan pada maka dapat disimpulkan bahwa:

- Parameter *beta* berfungsi sebagai *offset* yang akan ditambahkan pada nilai *highpass filter* sehingga frekuensi yang mendekati nol tidak dieliminasi begitu saja. Semakin besar parameter *beta*, gambar yang dihasilkan akan semakin terang.
- Semakin kecil parameter *cutoff frequency*, gambar yang akan dihasilkan semakin terang.
- Parameter *beta* dan *cutoff frequency* dapat saling menyeimbangkan sehingga menghasilkan dua gambar yang sama dari parameter yang berbeda.
- Hasil gambar yang diproses dengan *homomorphic filtering* akan tetap mempunyai warna yang sesuai dengan warna asli gambar tersebut.



## 6. Daftar Pustaka

- Color constancy – wikipedia. (n.d.). *Color Constancy*. <http://en.wikipedia.org/wiki/Retinex>, diakses terakhir tanggal 28 Oktober 2005.
- Du, Yu. (2002). *Approaches for retinex and their relations*. University of California. [www.cse.ucsc.edu/classes/ee264/Winter02/duyu.ppt](http://www.cse.ucsc.edu/classes/ee264/Winter02/duyu.ppt), diakses terakhir tanggal 18 Oktober 2005.
- Staab, Brian. (2004). *Homomorphic filtering of color images*. Rochester Institute of Technology. [www.cis.rit.edu/~bms4555/files/papers/homomorphic.doc](http://www.cis.rit.edu/~bms4555/files/papers/homomorphic.doc), diakses terakhir tanggal 28 Oktober 2005.
- Fourier transform – wikipedia. (n.d.). *Fourier Transform*. [http://en.wikipedia.org/wiki/Fourier\\_transform](http://en.wikipedia.org/wiki/Fourier_transform), diakses terakhir tanggal 18 Desember 2005.
- Gonzalez, Rafael C dan Woods, Richard E. (2002). *Digital image processing (2<sup>nd</sup> ed.)*. New Jersey: Prentice Hall.