

PERBAIKAN CITRA DIGITAL DENGAN MENGGUNAKAN METODE RETINEX

Rudy Adipranata¹, Cherry Galatia Ballangan², Leono Epatha
Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya
Email : rudya@petra.ac.id¹, cherry@petra.ac.id²

ABSTRAK

Dewasa ini penggunaan citra digital semakin meningkat karena kelebihan-kelebihan yang dimiliki oleh citra digital tersebut, antara lain kemudahan dalam mendapatkan gambar, memperbanyak gambar, pengolahan gambar dan lain-lain. Tetapi tidak semua citra digital memiliki tampilan visual yang memuaskan mata manusia. Ketidakpuasan itu dapat timbul karena adanya *noise*, kualitas pencahayaan pada citra digital yang terlalu gelap atau terlalu terang. Sehingga diperlukan metode untuk dapat memperbaiki kualitas citra digital tersebut.

Salah satu metode yang dapat digunakan untuk memperbaiki kualitas citra digital adalah metode Retinex. Metode Retinex ini dapat digunakan untuk memperbaiki kualitas citra digital yang berhubungan dengan pencahayaan yaitu dengan mempertahankan *color constancy*. *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 sebuah 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 inilah yang membantu manusia untuk mengidentifikasi suatu benda. Dengan metode Retinex, obyek pada *citra digital* yang terkena efek pencahayaan sehingga menjadi berwarna lain, dapat diperbaiki menjadi sehingga menjadi berwarna menyerupai sebenarnya.

Metode Retinex ini bekerja dengan cara memisahkan citra digital ke dalam dua bagian, yaitu *reflectance* dan *illumination*. Dengan adanya pemisahan tersebut maka dapat dilakukan kemungkinan pembuangan efek cahaya dari pencahayaan depan atau belakang serta memperbaiki warna pada citra digital dengan membuang iluminasi yang biasanya membuat warna berubah.

Kata kunci: perbaikan citra digital, Retinex, *color constancy*

1. PENDAHULUAN

Penggunaan kamera digital sebagai alat untuk mengambil citra saat ini sudah menggantikan kamera analog karena kepraktisannya ataupun kemudahannya. Tetapi terkadang hasil citra yang didapatkan memiliki tampilan visual yang kurang baik. Tampilan visual yang kurang baik tersebut dapat disebabkan oleh faktor dalam atau faktor luar. Faktor dalam adalah faktor karena kualitas kamera digital yang kurang baik, sehingga hasil citra yang dihasilkan juga kurang baik, sedangkan faktor luar adalah

faktor yang disebabkan karena adanya *noise*, kualitas pencahayaan pada citra digital yang terlalu gelap atau terlalu terang.

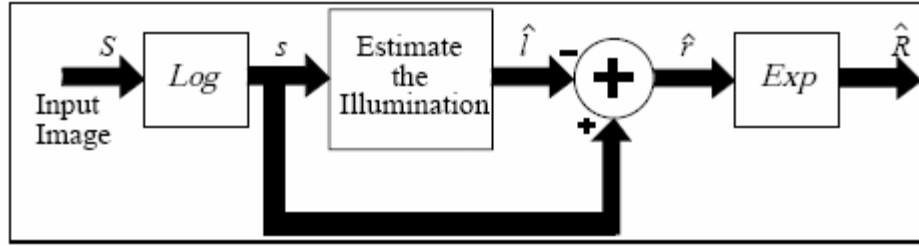
Salah satu faktor luar yang diakibatkan oleh pencahayaan akan menyebabkan sebuah benda mempunyai warna yang berbeda dari warna aslinya. Hal ini sering juga disebut dengan *color constancy*. *Color constancy* adalah salah satu keistimewaan dari sistem penglihatan manusia dimana warna yang diterima mata 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. Tetapi jika diambil dengan kamera digital, maka *color constancy* tersebut menjadi tidak ada. Karena itu dikembangkan algoritma yang dapat mengakomodasikan *color constancy* sehingga dapat dimanfaatkan untuk *computer vision*. Algoritma ini dikenal dengan nama algoritma Retinex.

2. RETINEX

Metode Retinex ini dikemukakan oleh Edwin Land pada tahun 1971. Melalui eksperimen yang dilakukan olehnya, terlihat bahwa bahwa sistem penglihatan manusia mampu secara praktis mengenal dan mencocokkan warna-warna di bawah sebuah *range illumination* berbeda yang luas, hal ini dikenal dengan *Color Constancy Phenomenon*. Teori Retinex berhubungan dengan kompensasi untuk efek *illumination* (pencahayaan) pada citra. Tujuan utama dalam metode Retinex adalah untuk memisahkan *image S* ke dalam dua buah *image* yang berbeda, yaitu *reflectance image R* dan *illumination image L*, di mana pada setiap titik (x,y) dalam *image* domain,

$$S(x,y) = R(x,y) \cdot L(x,y) \quad (1)$$

Pada gambar 1 terdapat gambaran dekomposisi *image S* menjadi *R* dan *L*. Keuntungan dari dekomposisi *image* itu adalah kemungkinan pembuangan efek cahaya dari pencahayaan depan atau belakang, dan memperbaiki warna-warna dalam *image* dengan membuang iluminasi yang membuat warna berubah.



Gambar 1. Gambaran Umum Retinex

Algoritma yang dipakai dalam metode Retinex ini adalah sebuah multi-resolusi yang memulai algoritmanya pada *image I* (*illumination image*) yang terkasar atau *image I* dengan resolusi terendah, dan diperbesar dengan teknik *pixel replication* dan menggunakan hasil dari *image scaling* tersebut sebagai inisialisasi untuk layer resolusi selanjutnya. Dengan mengubah *image* ke dalam domain logaritma, maka didapat $s=\log S$, $l=\log L$, $r=\log R$, maka $s=l+r$. Algoritma mengambil *spatial smoothness* dari *illumination field*. Sebagai tambahan, pengetahuan dari *dynamic range* terbatas dari *reflectance* digunakan sebagai sebuah *constraint* dalam proses *recovery*.

Karena natur fisik dari *reflecting objects* adalah bahwa *reflecting objects* hanya memantulkan bagian dari cahaya insiden, sehingga *reflectance* tidak diperbolehkan berada pada *range* $Re[0, 1]$, dan $L \geq S$ dapat diimplementasikan untuk $l \geq s$.

Mengumpulkan semua asumsi-asumsi di atas ke dalam satu formula, maka didapatkan fungsi penalti [1]:

$$\text{Minimalkan:} \quad F[l] = \int_{\Omega} \left(|\nabla l|^2 + \alpha(l-s)^2 + \beta |\nabla(l-s)|^2 \right) dx dy \quad (2)$$

dengan batasan: $l \geq s$ dan $\nabla l \cdot \vec{n} = 0$ pada $\partial\Omega$,

dimana Ω adalah sokongan dari *image*, $\partial\Omega$ adalah batasnya, dan \vec{n} adalah normal dari batas tersebut. α dan β adalah parameter bebas yang merupakan bilangan real non negatif. Dalam fungsi $F[l]$, penalti pertama adalah $(|\nabla l|^2)$ yang memberlakukan *spatial smoothness* pada *illumination image*. Penalti kedua adalah $((l-s)^2)$ yang memberlakukan sebuah kedekatan antara l dan s . Perbedaan antara kedua *image* ini adalah pasti r , dimana term ini diberatkan oleh parameter bebas α . Sebagai tambahan dapat diberlakukan solusi l menjadi $l \geq s$. Penalti ketiga adalah $((\nabla r)^2)$ yang memberlakukan *reflectance image* r menjadi sebuah *image* dengan tampilan yang lebih

baik. Hal ini membuat r menjadi *smooth* secara *spatial*, dan diberatkan oleh parameter bebas β .

Kondisi yang perlu dan cukup untuk fungsi minimisasi $F[l]$ didapat melalui persamaan Euler-Lagrange:

$$\forall (x, y) \in \Omega, \begin{cases} \partial F[l] = 0 = -\Delta + \alpha(l-s) - \beta \Delta(l-s), & l > s \\ l = s & l = s \end{cases} \quad (3)$$

Secara langkah demi langkah, algoritma Retinex adalah sebagai berikut:

1. Input : Input dalam algoritma ini adalah sebuah *image* s dengan ukuran $[N, M]$, dan terdapat dua parameter yaitu α dan β .
2. Inisialisasi : Hitung sebuah *Gaussian* Piramid dari *image* s . Piramid ini dibangun dengan menghaluskan (*smoothing*) *image* dengan kernel K_{PYR}

$$K_{pyr} = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \quad (4)$$

dan kemudian mengecilkan *image* dengan menggunakan rasio 2:1. Proses ini diulang p kali dan menghasilkan sebuah deret dari *image-image* $\{s_k\}_{k=1}^p$. *Image* s_1 adalah *image* asli, dan s_p adalah *image* dengan resolusi paling kasar dalam piramid.

Definisi dari numerical inner product adalah sebagai berikut:

$$\langle G, F \rangle = \sum_{n,m} G[n, m] F[n, m] \quad (5)$$

dan numerical Laplacian pada resolusi ke k^{th} sebagai berikut

$$\Delta_k G = G * k_{LAP} 2^{-2(k-1)} \quad (6)$$

Tentukan $k=p$, berarti dimulai pada lapisan resolusi yang paling kasar, dan tentukan kondisi initial $l_0 = s_p$.

Tentukan T_i - jumlah dari langkah-langkah yang sedang berlangsung pada setiap lapisan resolusi i .

3. Perulangan utama : Untuk layer resolusi ke- k ,

❖ Hitung

$$G_B \leftarrow \Delta s_k \quad (7)$$

❖ For $j=1, \dots, T_k$ do:

(a) Hitung gradien dari

$$G_A \leftarrow \Delta l_{j-1}, \quad (8)$$

$$G \leftarrow G_A + \alpha(l_{j-1} - s_k) - \beta(G_A - G_B) \quad (9)$$

(b) Hitung μ_{NSD}

$$\mu_A \leftarrow \langle G, G \rangle, \quad (10)$$

$$\mu_B \leftarrow \langle G, \Delta G \rangle, \quad (11)$$

$$\mu_{NSD} \leftarrow \mu_A / (\alpha \mu_A + (1+\beta) \mu_B). \quad (12)$$

Lengkapi iterasi NSD :

$$I_j \leftarrow I_{j-1} - \mu_{NSD} \cdot G, \quad (13)$$

$$(c) \quad I_j \leftarrow \max\{I_j, s_k\} \quad (14)$$

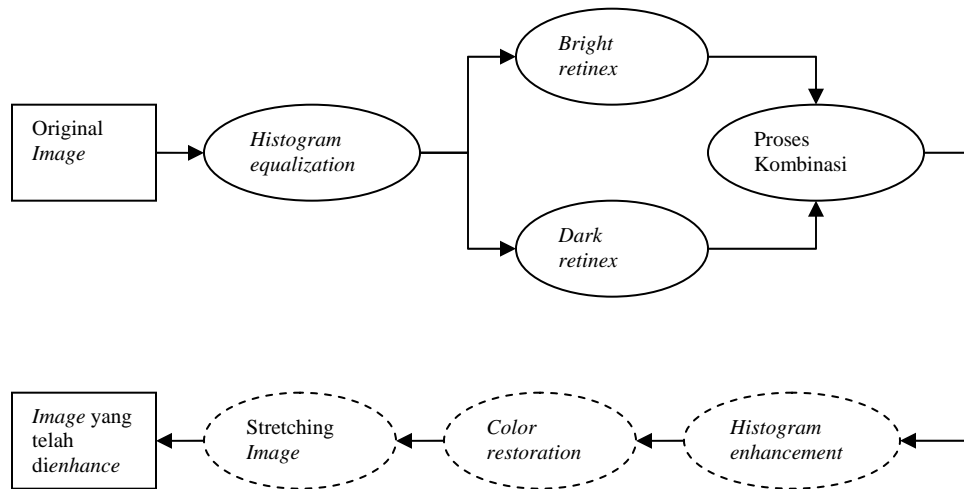
❖ End j Loop;

4. Update *resolution layer* selanjutnya :

Jika $k > 1$, hasil I_{Tk} di *up scale* (rasio 2:1) dengan *pixel replication* ke dalam I_0 yang baru, inisialisasi untuk *resolution layer* $k-1$ selanjutnya. *Resolution layer* di-update $k=k-1$, dan algoritma tetap berlangsung dengan berjalan lagi ke langkah ke-3. Jika $k=1$, maka hasil I_{T1} adalah output akhir dari algoritma.

3. DESAIN SISTEM

Secara garis besar, sistem terdiri dari dua bagian utama, yaitu: bagian proses utama *image* dan bagian proses pilihan. Secara garis besar, sistem yang akan dibuat adalah seperti terlihat pada gambar di bawah ini.



keterangan:



-Proses Utama



-Proses Pilihan

Gambar 2. Diagram Sistem

Proses pertama adalah *histogram equalization* yang merupakan perbaikan pada suatu *image* dengan berfokus pada perbaikan kekontrasan (*contrast enhancement*). Tujuan proses ini adalah untuk mendapatkan *image* baru yang mempunyai histogram seragam (*uniform histogram*). *Image* ini didapatkan dengan menggunakan histogram kumulatif yang dinormalisasikan dari nilai-nilai per piksel dari *image* asli dan histogram ini digunakan sebagai fungsi *gray scaling mapping*. Cara ini bekerja dengan sangat baik untuk citra-citra yang memiliki detail-detail pada bagian gelap. Langkah-langkah dalam proses *histogram equalization* ini adalah sebagai berikut.

- ✓ Dapatkan nilai piksel dari sebuah *image* dengan mengakses tiap-tiap piksel.
- ✓ Tentukan nilai piksel yang paling tinggi dari citra tersebut.
- ✓ Bangun histogram kumulatif dengan menghitung jumlah dari nilai-nilai piksel yang unik dan simpan dalam sebuah array. Setiap elemen array selain menyimpan nilai piksel untuk piksel yang berhubungan dengan elemen ini juga menyimpan jumlah dari elemen yang sebelumnya, sehingga elemen ke-1 akan menyimpan nilai piksel ke-1 dan jumlah elemen ke-0, sedangkan untuk elemen yang ke-255 akan menyimpan jumlah nilai elemen ke-255, 254, 253, dan seterusnya.
- ✓ Normalisasi histogram kumulatif dengan mengalikan masing-masing elemen dengan nilai piksel maksimum dibagi dengan jumlah piksel yang ada
- ✓ Kemudian lakukan pemetaan *image* asli ke *image* yang ter-histogram equalization dengan menggunakan array yang telah ternormalisasi sebagai fungsi mapping.

Proses berikutnya adalah proses Retinex yang diimplementasikan ke dalam dua macam *image*. *Image* pertama yang digunakan adalah *image* asli yang disebut dengan *bright retinex*. Sedangkan *image* kedua yang digunakan adalah *image* yang telah dilakukan proses *inverse* yang disebut dengan *dark retinex*. Hasil dari proses *dark retinex* ini akan dilakukan proses *inverse* kembali. Kemudian dilakukan penggabungan (proses kombinasi) pada hasil dari kedua proses yaitu dengan cara menghitung rata-rata dari kedua proses tersebut.

Setelah dilakukan proses kombinasi Retinex, proses selanjutnya adalah *histogram enhancement*. Proses ini dilakukan untuk membuat histogram dari *image* lebih seragam (uniform), tetapi tanpa kehilangan warna-warna aslinya. Pada proses ini digunakan invers transformasi dari Gaussian Error Function dimana persamaan transformasinya adalah sebagai berikut:

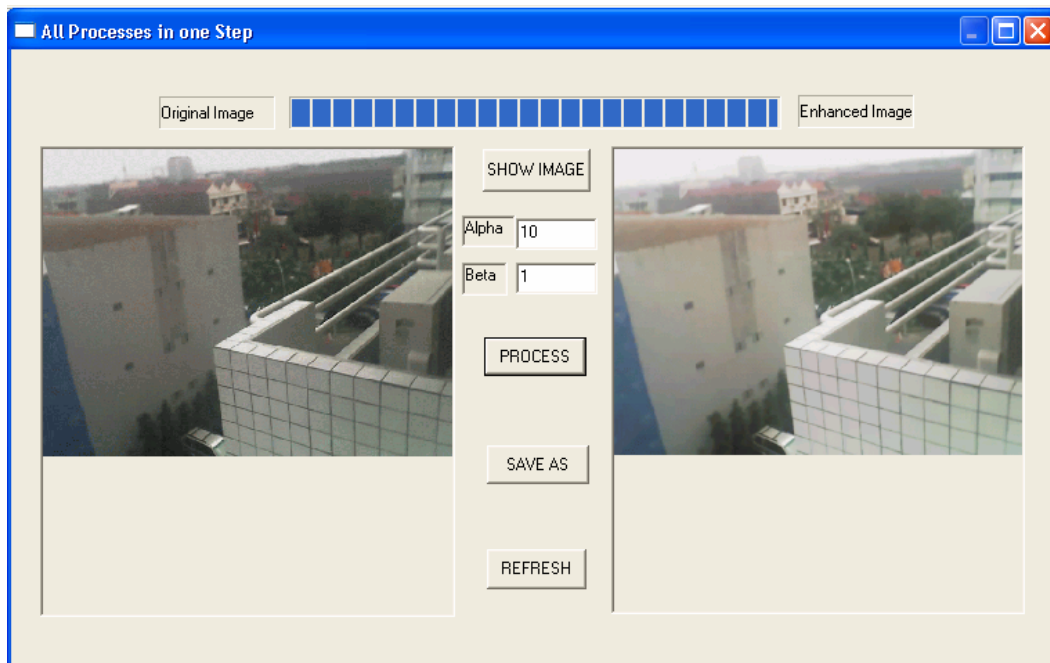
$$HD^{-1}(g) = \frac{\text{erf}\left(\left(g - \frac{1}{2}\right)2\sigma\right)}{2.\text{erf}\sigma} + \frac{1}{2}, \quad (15)$$

Proses berikutnya adalah *color restoration* yang bertujuan untuk mengembalikan detail warna yang telah hilang. Proses *color restoration* ini dilakukan pada setiap *channel* warna. Proses ini akan mengalikan nilai piksel pada *channel* warna tertentu dengan nilai piksel *channel* warna itu sendiri yang telah dibagi dengan jumlah nilai-nilai piksel dari ketiga *channel* warna.

Proses yang terakhir adalah *image stretching*. Proses ini digunakan untuk menambah *visual range* dari *image*. Idenya dengan secara berulang-ulang mengubah warna-warna yang jarang dipakai ke dalam warna yang sering dipakai.

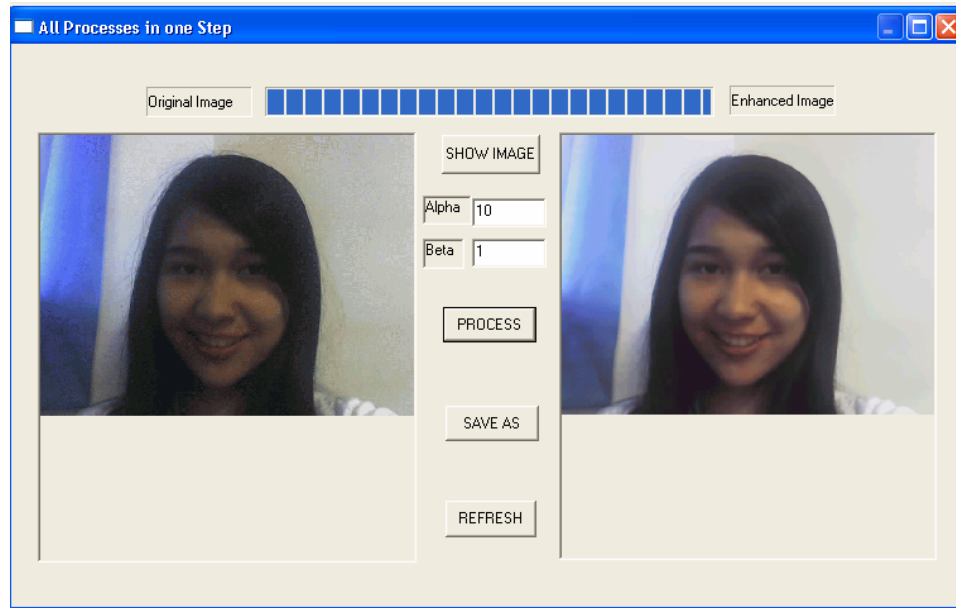
4. IMPLEMENTASI DAN HASIL

Pada implementasi sistem perbaikan citra ini digunakan bahasa pemrograman Microsoft Visual C++ 6.0. Setelah selesai dilakukan implementasi, dilakukan pengujian terhadap citra untuk melihat hasil perbaikan citra tersebut. Berikut ini ditampilkan hasil pengujian aplikasi yang dikembangkan.



Gambar 3. Pengujian Hasil Perbaikan Citra

Di pengujian yang terlihat pada gambar 3, citra diambil dengan menggunakan kamera VGA yang terdapat pada telepon seluler. Dari hasil pengujian tersebut tampak bagian yang agak gelap menjadi lebih terang. Pada gambar di bawah ini juga ditampilkan pengujian terhadap citra lain yang berupa wajah.



Gambar 4. Pengujian Hasil Perbaikan Citra

Dari pengujian di gambar 4 terlihat bahwa warna kulit yang buram atau gelap karena pencahayaan menjadi berwarna terang menyerupai warna kulit yang sesungguhnya. Begitu pula warna dinding yang agak keabu-abuan menjadi putih.

5. KESIMPULAN

Dari hasil pengujian dapat disimpulkan beberapa hal sebagai berikut:

- Pengujian proses *image enhancement* dengan menggunakan algoritma Retinex, menggunakan nilai dalam range satu sampai seratus (1-100), dengan nilai *alpha* dan *beta* yang berubah-ubah, didapatkan kesimpulan bahwa jika nilai *alpha* lebih besar dari *beta* ($\alpha > \beta$) maka akan menghasilkan *image* yang lebih tajam, lebih terang dan lebih halus, sedangkan jika menggunakan nilai *alpha* yang lebih kecil dari *beta* ($\alpha < \beta$) akan menghasilkan *image* yang tidak baik secara visual, berupa *image* yang tidak jelas detailnya.
- Hasil pengujian dari proses *color restoration* tidak terlalu memuaskan. *Image* hasil proses *color restoration* terlihat lebih gelap dibandingkan dengan *image* aslinya.

DAFTAR PUSTAKA

- [1] Devir, Zvi, and Tridensky, Liliya, *Dynamic Range Compression while Improving the Visual Quality of Color Images*. Project Report Geometric Image Processing Laboratory, www.cs.technion.ac.il/~gip/projects/s2002/Zvi_Devir, Desember 2005

- [2] G.D. Finlayson, S.D. Hordley, and M.S. Drew, *Removing Shadows From Images Using Retinex*. ECCV02, pages 823-836, 2002.
- [3] Kimmel Ron, *A Variational Framework for Retinex*. International Journal of Computer Vision 52(1), pages 7-23,2003.
- [4] Gonzalez, R.C. & Woods, R.E, *Digital Image Processing*, New Jersey: Prentice Hall, 2002.