



PENERAPAN METODE RETINEX UNTUK MENINGKATKAN KECERAHAN CITRA PADA HASIL SCREENSHOT

Soeb Aripin¹, Guidio Leonarde Ginting², Natalia Silalahi³

^{1,2} Dosen Tetap STMIK Budi Darma

³ Dosen Tetap AMIK STIEKOM Sumatera Utara

^{1,2} Jl. Sisingamangaraja No. 338 Simpang Limun Medan

³ Jl. Abdul Haris Nasution No. 19 Medan

ABSTRAK

Perbaikan kualitas diperlukan karena seringkali citra yang diuji mempunyai kualitas yang buruk, salah satunya adalah peningkatan citra menggunakan metode Retinex dimana Retinex ini adalah teknik yang digunakan untuk mendapatkan citra baru dengan kontras yang lebih baik daripada kontras dari citra asalnya. Terdapat empat tahapan utama yang akan dilakukan untuk mendapatkan sebuah citra baru sampai dengan proses penampilan statistik citra, yaitu proses menampilkan citra awal pada window, perbaikan kontras citra, menampilkan histogram citra dan menampilkan statistik citra. Perbaikan kualitas diperlukan karena citra pada hasil screenshot mempunyai kualitas yang buruk, untuk itu dilakukan perbaikan citra dengan meningkatkan kualitas citra menggunakan metode Retinex. Metode Retinex ini adalah metode yang digunakan untuk mendapatkan citra baru dengan kontras yang lebih baik daripada kontras dari citra asalnya.

Kata Kunci: Citra, Retinex, Screenshot

ABSTRACT

Improved quality is required because often the image tested has poor quality, one of which is image enhancement using Retinex method where Retinex is a technique used to get a new image with better contrast than contrast from its original image. There are four main stages to be done To obtain a new image up to the process of displaying image statistics, ie the process of displaying the initial image in the window, improving the image contrast, displaying the image histogram and display the image statistics. Quality improvement is required because the image on the screenshot has poor quality, By improving image quality using Retinex method. This Retinex method is a method used to get a new image with better contrast than the contrast of its original image.

Keywords: Image, Retinex, Screenshot

I. PENDAHULUAN

Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra yang berkualitas lebih baik daripada citra masukannya. Dalam meningkatkan mutu citra ada beberapa teknik yang digunakan antara lain *image enhancement*, berupa proses perbaikan citra dengan meningkatkan kualitas citra baik kontras maupun kecerahan. *image restoration*, proses memperbaiki model citra, *color image processing*, suatu proses yang melibatkan citra berwarna, baik berupa *image enhancement*, *image restoration* atau yang lainnya dan masih banyak lagi teknik-teknik yang lain.

Perbaikan kualitas citra (*image enhancement*) merupakan salah satu bidang yang cukup populer hingga saat ini. Kecerahan suatu citra sangat diperlukan agar seseorang dapat melihat dan mengamati citra tersebut dengan jelas tanpa adanya halangan seperti gangguan dengan menggunakan *image enhancement*. Suatu citra dapat mengalami gangguan yang berupa kerusakan pada beberapa

bagian/blok *pixel* pada proses pengiriman/transmisi atau proses penyimpanan. Kerusakan ini adalah bentuk kesalahan yang utama pada suatu citra, misalnya citra hasil *screenshot*.

Screenshot adalah suatu gambar tampilan yang diambil dari layar monitor seperti perangkat komputer, tablet PC, dan *smartphone*. Hasil citra dari *screenshot* memiliki tingkat ketajaman dan kehalusan citra yang kecil, sehingga untuk mendapatkan citra yang lebih baik biasanya dilakukan pembesaran terhadap citra tersebut secara langsung namun memiliki tingkan kecerahan yang kurang baik berdasarkan penelitian sebelumnya[1].

Tingkat ketajaman dan kecerahan citra merupakan salah satu parameter yang utama dalam pemrosesan citra digital. Meningkatkan kecerahan citra pada perbaikan kualitas citra *hasil screenshot* memiliki informasi gambar yang kaya, tetapi sering kali gambar yang dimiliki rusak parah (*degradasi*), misalnya citra mengalami derau (*noise*) pada saat pengiriman melalui saluran transmisi, citra terlalu terang/gelap, citra kurang tajam, kabur, mengandung cacat atau kebisingan, warnanya terlalu kontras, kurang tajam, *blur* (kabur), dan sebagainya. Salah satu untul meniingkatkan kualitas citra menggunakan



metode *Retinex*. Metode *Retinex* ini adalah metode yang digunakan untuk mendapatkan citra baru dengan kontras yang lebih baik daripada kontras dari citra asalnya. Metode *Retinex* suatu metode yang digunakan untuk perbaikan citra yang bersifat *point processing*, yang artinya proses ini hanya tergantung dari nilai intensitas (*gray level*) satu *pixel*, tidak tergantung dari *pixel* lain yang ada disekitarnya. Di sini diasumsikan bahwa citra memiliki *range gray level* dari 0 sampai 255. Terdapat empat tahapan utama yang akan dilakukan untuk mendapatkan sebuah citra baru sampai dengan proses penampilan statistik citra, yaitu proses menampilkan citra awal pada *window*, perbaikan kontras citra, menampilkan *histogram* citra dan menampilkan statistik citra.

II. TEORITIS

A. Screenshot

Screenshot atau *screen capture* dibutuhkan saat anda ingin mendokumentasikan segala sesuatu yang tampak pada layer monitor jendela komputer kedalam *image* yang pasip (Wahana Komputer, 2004). *Printscreen* atau *screenshot* berfungsi untuk menangkap gambar layar *hanphone*, sehingga dapat dipakai untuk mendokumentasi atau kebutuhan lain yang sering di sebut *mobile printscreen* (Yuniar Supradi, 2009). Berdasarkan pengertian di atas penulis dapat mengambil kesimpulan bahwa *screenshot* adalah suatu dokumentasi gambar tampilan yang diambil dari layar monitor seperti perangkat komputer, tablet PC dan *smartphone*.

B. Metode Retinex

Metode *Retinex* adalah salah satu teknik *error and erasure correction* yang paling baik dan dijadikan standar dalam banyak bidang diantaranya komunikasi satelit dan *mobile, magnetic recording*, dan *high-definition television*. Dalam komunikasi nirkabel, data yang dikirimkan oleh transmitter tentu akan mendapatkan noise dari kanal (*channel*) yang bisa mengakibatkan kerusakan pada data. Reed-Solomon Decoder mampu mengembalikan data yang rusak tersebut. Salah satu kelebihan Reed-Solomon adalah *non-binary code* (data diolah dalam *word*) sehingga kemampuan koreksi data lebih banyak (Xiao-Chun, 2006:23). Adapun rumus-rumus yang digunakan dalam metode Reed Solomon adalah sebagai berikut :

Untuk mendapatkan masing-masing nilai R, G dan B dihitung dengan rumus sebagai berikut:

$$\begin{aligned} \text{Nilai R} &= c \text{ and } 255 \\ \text{Nilai G} &= (c \text{ and } 65280) / 256 \\ \text{Nilai B} &= ((c \text{ and } 16711680) / 256) \end{aligned}$$

Dimana

$$f_0(x,y) = \left(\frac{f^R(x,y) + f^G(x,y) + f^B(x,y)}{3} \right) \dots \dots \dots (1)$$

Dimana f^R adalah nilai komponen red
 f^G adalah nilai komponen green
 f^B adalah nilai komponen blue

$$f(x,y) = [\pi_{(s,t) \in S_{xy}}(s,t)]^{1/mn} \dots \dots (2)$$

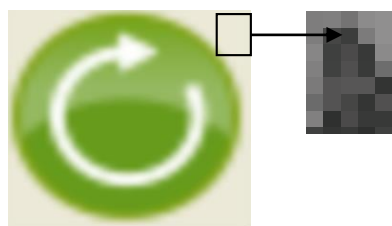
Dimana

xy adalah koordinat pixel pada citra
 Mn adalah dimensi citra (pixel)
 S,t adalah intensitas pixel
 Π adalah perkalian nilai pixel yang terkena filter

III. ANALISA dan PEMBAHASAN

A. Analisa Metode Retinex

Perbaikan citra hasil gambar *screenshot* dapat diperbaiki dengan salah satu metode perbaikan citra yaitu, melakukan perbaikan pada citra digital yang terkena *Noise* menjadi citra yang lebih bagus dengan melakukan perbaikan pada citra tersebut. Dalam sistem ini citra yang terkena *noise-noise* tersebut akan direduksi dengan menggunakan metode *Retinex*



Gambar 1. *Noise* pada Citra

Nilai intensitas citra disimpan dalam data *Jpeg* dalam bentuk nilai biner dengan nilai antara 0 dan 1. Jadi jumlah *pixel*nya adalah jumlah kelipatan 24 bit.

Adapun nilai citra diatas adalah sebagai berikut:
 11110000111100001111110001100001111000011
 111111111100001111000

01111100011110000111100001111100111100001
 111000011111010101100

001111000011111111111100001111000011111111
 111100001111000011010

111111110000111100001100000111110000111100
 001111011000111100001

111000011111111111100001111000011111000011
 100001111000011110101

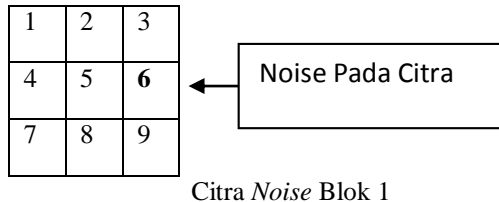
01111000011110000111100011011111000011110
 000111111111111000011

110000111010111111100001111000011111111

Nilai 111100001111000011111111 = 24 bit mewakili 1 *pixel* dari citra. Jika jumlah bit citra di atas adalah 1.800.000 bit, maka jumlah *pixel*nya adalah 1.800.000/24 bit = 75.000 *pixel*. Pada



reduksi *noise* ini dilakukan per blok citra dengan ukuran per bloknya adalah 3 x 3 *pixel* atau 9 *pixel*, maka jumlah blok adalah 1.800.000 : 9 = 200.000 blok. Proses reduksi *noise* dilakukan per blok citra yang terdiri dari 9 *pixel* (3x3), maka citra tersebut dilakukan pembagian *pixel* dalam ukuran 3 x 3, di mana *pixel* yang berada di posisi tengah merupakan *pixel* yang akan diuji. penghitungan nilai komponen warna *RGB*-nya dilakukan dalam 9 *pixel*. Adapun nilai matriks dari gambar diatas adalah sebagai berikut



Nilai *pixel* 1 dalam *biner* 24 bit adalah:

Pixel 1 $f(0,0) = 111100001111000011111111$

Pixel 2 $f(0,1) = 100000011110000111110000$

Pixel 3 $f(0,2) = 110000001111000011111111$

Pixel 4 $f(1,0) = 101110011110000111110000$

Pixel 5 $f(1,1) = 11110000111110011110011$

Pixel 6 $f(1,2) = 110110011110000101110010$

Pixel 7 $f(2,0) = 010100001111000011100111$

Pixel 8 $f(2,1) = 100000011110000111110111$

Pixel 9 $f(2,2) = 101110111110000100111100$

Untuk mendapatkan masing-masing nilai R, G dan B dihitung dengan rumus sebagai berikut:

Nilai R = c and 255

Nilai G = $(c$ and 65280) / 256

Nilai B = $((c$ and 16711680) / 256) / 256

Dimana c adalah nilai *pixel* citra

Perhitungan Nilai RGB:

1. Nilai *pixel* (0,0) adalah

111100001111000011111111

Nilai R = 111100001111000011111111 and 11111111

= 11111111 dalam biner = 255 (desimal)

Nilai G = (111100001111000011111111 and 1111111100000000) / 100000000

= 11110000 (biner) = 240 (desimal)

Nilai B = ((111100001111000011111111 and 111111110000000000000000) /

100000000) / 100000000 = 11110000 = 240 (desimal)

Sehingga diperoleh nilai R G B *pixel* (0,0):

R = 11110001 = 255 (desimal)

G = 10110100 = 240 (desimal)

B = 10010000 = 240 (desimal)

2. Nilai *pixel* (0,1) adalah

100000011110000111110000 (24 bit).

Nilai R = 100000011110000111110000 and 11111111

= 11110000 dalam biner = 240 (desimal)

Nilai G = (100000011110000111110000 and 1111111100000000) / 100000000

= 11100001 (biner) = 225 (desimal)

Nilai B = ((100000011110000111110000 and 111111110000000000000000) /

100000000) / 100000000

= 10000001 = 129 (desimal)

Sehingga diperoleh nilai R G B *pixel* (0,1):

R = 11110001 = 240 (desimal)

G = 10110100 = 225 (desimal)

B = 10010000 = 129 (desimal)

Sampai nilai *pixel* (2,2) adapun RGB semua *pixel* pada citra noise blok 1 dimasukkan ke dalam matriks seperti berikut.

255,240,240	240,225,129	255,240,192
240,225,185	243,252,249	114,225,217
231,240,80	252,48,187	156,240,187

Selanjutnya nilai RGB per *pixel* di atas di konversi ke nilai *grayscale* dengan cara mencari nilai rata-rata per *pixel*.

Selanjutnya nilai *grayscale* semua *pixel* pada citra noise blok 1 dimasukkan ke dalam matriks nilai *grayscale* seperti berikut.

198	245	229
217	248	185
184	162	194

Matriks Nilai *Grayscale* Citra Noise Blok 1

Pada citra Gambar 3.4 dilakukan penambahan *noise salt and papper* pada citra yang ber-*noise*. *Noise Salt and Papper* merupakan *pixel* memiliki warna gelap dan terang yang nilai 0 dan 255 dan penambahan ke dalam citra dilakukan secara acak. Proses penambahan *noise* pada citra pada Gambar 3.4 dengan mengganti nilai *pixel* secara acak dengan nilai 0 dan 255 seperti pada Gambar 3.5.

198	245	229	198	245	229
217	248	185	217	248	0
0	162	194	0	162	194

Matriks Citra Hasil Penambahan Noise



Pada proses filter yang digunakan pada reduksi *noise* dengan *Retinex* berukuran 3 x 3 yang semuanya bernilai 1 dengan menggunakan rumus:

Rumus diatas berarti semua *pixel* yang mewakili nilai intensitas dari citra yang terkena jendela filter dikalikan lalu dipangkatkan dengan 1/mn. Pada Gambar diatas dilakukan filter dengan menggunakan rumus di atas:

198	245	229
217	248	0
0	162	194

Matriks Citra Noise Blok 1

Hasil filter yang terbentuk =

$$(198*245*229*217*248*0*0*162*194)^{1/9} = 0$$

198	245	229
217	0	0
0	162	194

Matriks Citra Noise Blok 1 Hasil Filter *Retinex*

Pada masing-masing filter dilakukan proses sampai semua blok citra *noise* habis di- filter dengan cara menggeser jendela filter sebanyak satu pilsel ke kanan per proses

198	245	229			
217	0	0			
0	162	194			

Matriks Citra *Noise* Blok 1 Sebelum Digeser Dengan menggeser satu *pixel* ke kanan maka hasilnya dapat dilihat pada matriks dibawah ini

198	245	229	Xx		
217	0	0	Xx		
0	162	194	Xx		

Matriks Citra Noise Blok 1 Sesudah Digeser



Gambar 2. Hasil akhir proses Retinex

IV. KESIMPULAN

Dari hasil penelitian dan analisa maka dapat diambil kesimpulan, dimana kesimpulan-kesimpulan tersebut kiranya dapat berguna bagi para pembaca, sehingga penelitian ini dapat lebih bermanfaat.

1. Proses perbaikan peningkatan kecerahan citra dilakukan dengan cara melakukan proses operasi titik *pixel* penyeleksian terhadap proses pembesaran citra hasil gambar *screenshot* dengan cara melakukan peroses pencerahan citra
2. Metode *retinex* dapat diterapkan dalam pencerahan citra dengan cara proses dengan menggunakan perkalian kernel dan menghitung nilai tengah *pixel* yang telah dikalikan dengan kernel dan menjadi sebuah citra baru dengan kecerahan citra *pixel* yang lebih jelas dari aslinya

DAFTAR PUSTAKA

- [1] Aripin Soeb and Sunandar Hery, "Perancangan Aplikasi Perbaikan Citra Pada Hasil Screenshot Menggunakan Metode Interpolasi Linier," Pelita Informatika Budi Darma, vol. 15, no. 1, pp. 51-58, 2016.
- [2] Darma Putra. 2010. Pengolahan Citra Digital. Yogyakarta. Andi
- [3] Harry Ramza dan Yohanes Dewanto. 2007. Teknik Pemograman Menggunakan Matlab. Jakarta, Grasindo.
- [4] Hendrayudi. 2009. VB 2008 Untuk Berbagai Keperluan Pemograman. Jakarta, PT. Gramedia.
- [5] Jogiyanto HM. 2005. Analisis & Desain. Yogyakarta. Andi
- [6] Yuniar Supardi. 2009. Aplikasi Populer Handphone. Jakarta, PT. Gramedia.
- [7] Julius Hermawan. 2005. Analisa Desain Dan Pemrograman Berorientasi Objek Dengan UML Dan Visual Basic.Net. Yogyakarta. Andi
- [8] Kortoko Dwi Hartomo. 2006. Implementasi Metode Interpolasi Linier Untuk Pembesaran Reselusi Citra. TEKNOIN, 11, 0853-8697
- [9] Setia Budi Sasongko. 2010. Metode Numrik Dengan Scilab. Yogyakarta. Andi
- [10] Wahana Komputer. 2008. Cepat Menguasai Visual Studio .Net 2008 Express. Yogyakarta, Penerbit Andi
- [11] Wahana Komputer. 2004. Teknik Pengolahan Image Bitmap dengan GIMP. Yogyakarta, Penerbit Andi