

PERANGKAT LUNAK PERBAIKAN KUALITAS CITRA DIGITAL MODEL RGB DAN IHS DENGAN OPERASI PENINGKATAN KONTRAS

Tole Sutikno, Kartika Firdausy, Eko Prasetyo

Center for Electrical Engineering Research and Solutions (CEERS)

Program Studi Teknik Elektro Fakultas Teknologi Industri Universitas Ahmad Dahlan

Kampus III UAD, Jln. Prof Soepomo Yogyakarta, Telp. 0274 379418 psw 220; Fax. (0274)-381523,

e-mail: ceers@uad.ac.id, tholes2000@yahoo.com, kartikaf@indosat.net.id

ABSTRAKSI

Citra dengan kontras rendah mempunyai kualitas yang relatif kurang baik, informasinya sulit untuk diinterpretasikan secara langsung oleh mata manusia. Citra seperti ini dapat diperbaiki kualitasnya dengan operasi peningkatan kontras (*contrast stretching*). Tujuan penelitian ini adalah merancang perangkat lunak perbaikan kualitas citra melalui operasi peningkatan kontras pada citra model RGB dan IHS. Pada penelitian ini digunakan operasi titik (*point operation*), yang akan memproses tiap piksel citra masukan secara individu, yaitu dengan membuat titik yang cenderung gelap menjadi lebih gelap dan yang cenderung terang menjadi lebih terang. Pengembangan perangkat lunak peningkatan kontras citra dilakukan menggunakan Borland Delphi 6.0, yaitu kelas *TBitmap*. Pengujian dilakukan pada citra standar yang berupa citra skala keabuan (*grayscale*) atau citra warna (*color images*) dengan format file *BMP* dan *JPEG*. Berdasarkan pengujian pada beberapa sampel citra uji menunjukkan bahwa perangkat lunak yang dirancang dapat digunakan untuk meningkatkan kontras suatu citra. Perbaikan kualitas citra ini dibuktikan dengan melebarnya histogram citra.

Kata kunci: *image enhancement, contrast stretching, point operation, RGB, IHS*

1. PENDAHULUAN

Pada pengolahan citra digital, terdapat beberapa model warna yang digunakan untuk merepresentasikan suatu citra. Model warna tersebut, diantaranya model RGB, model IHS, model YIQ, model CIE, dan model CMY. Model RGB lebih sering digunakan karena bagus untuk menampilkan informasi warna, tetapi model ini tidak cocok untuk beberapa operasi pengolahan citra. Ada beberapa operasi pengolahan citra yang mengharuskan penggunaan model selain RGB, yaitu model IHS. Di antara operasi pengolahan citra yang menggunakan model ini adalah operasi pengenalan pola (*pattern recognition*) dan operasi pemampatan citra (*image compression*).

Setiap citra yang akan diolah atau diproses dapat dianalisis dengan melihat *image histogram*-nya. *Image histogram* adalah grafik yang menunjukkan frekuensi suatu nomor warna. Sumbu horizontal menunjukkan nomor warna yang dipakai (0-255). Sumbu vertikal menunjukkan jumlah titik yang menggunakan warna yang bersangkutan [4]. Tetapi tidak semua citra dapat dianalisis dengan baik, contohnya pada citra dengan tingkat kekontrasan rendah. Citra ini dapat terjadi karena kondisi pencahayaan yang tidak seragam atau kurang mendapat efek cahaya. Kondisi ini membuat citra tersebut akan kelihatan kurang kontras. Salah satu cara untuk memperbaikinya dengan menggunakan efek peningkatan kontras (*contrast stretching*). Dalam hal ini operasi yang digunakan adalah operasi titik (*point operation*), di mana tingkat keabuan yang satu dipetakan ke tingkat keabuan yang lain sesuai dengan fungsi yang diberikan. Operasi titik adalah operasi terhadap citra

di mana setiap titik diolah secara tidak gayut terhadap titik-titik yang lain [1]. Dengan efek peningkatan kontras diharapkan detail dari citra terlihat lebih jelas sehingga citra akan dengan mudah diamati dan dianalisis.

Borland Delphi 6.0 menyediakan banyak fasilitas untuk membuat aplikasi-aplikasi pengolah grafis dan multimedia, beberapa fungsi dan metode khusus disediakan untuk menangani masalah-masalah tersebut. Metode kelas (*class method*) dalam Borland Delphi 6.0 dapat didefinisikan sebagai metode yang diasosiasikan pada suatu tipe data kelas. Oleh sebab itu, kelas ini dapat dipanggil tanpa harus dibuat *instance*-nya. *Instance* dapat diartikan sebagai contoh nyata atau contoh langsung dari suatu kelas. *TBitmap* merupakan salah satu kelas yang khusus menangani masalah grafis. Dalam *TBitmap* terkandung informasi yang cukup lengkap pada suatu citra. Penggunaan kelas *TBitmap* akan secara otomatis mengatur sistem palet dari suatu citra. Sebelum digunakan, kelas *TBitmap* harus dideklarasikan terlebih dahulu.

Saat ini terdapat ratusan macam format berkas (*file*) yang dapat digunakan sesuai kebutuhan. Format gambar tersebut, di antaranya format *BMP*, *JPEG*, *GIF*, *ICO*, *WMF* dan lain sebagainya. Cara penyimpanan suatu citra juga sangat tergantung pada kebutuhan penggunaan citra tersebut. Untuk mengirim gambar melalui internet, kompresi data sangat diperlukan dan harus kompatibel dengan komputer yang menerimanya. Bila ingin menggunakan format gambar untuk animasi atau presentasi, maka kecepatan untuk menampilkan data lebih penting dibanding dengan kompresi data. Pada

penelitian ini digunakan 2 macam format gambar, yaitu BMP dan JPEG.

2. METODE PENELITIAN

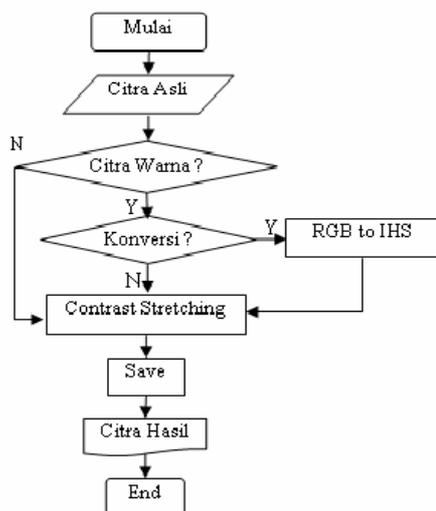
2.1 Analisis Kebutuhan Sistem

Pada penelitian ini digunakan citra diam (*still images*) 2 dimensi yang direpresentasikan ke dalam bentuk matriks berukuran $N \times M$. Matriks tersebut dapat juga dianggap sebagai larik 2 dimensi atau *array* 2 dimensi, karena memiliki 2 koordinat citra, yaitu koordinat x dan koordinat y . Pendefinisian larik 2 dimensi secara konvensional dapat menyebabkan terjadinya pemborosan memori.

Representasi citra dengan sistem vektor lebih relevan dibandingkan dengan struktur matriks. Dalam hal ini, vektor digunakan untuk memanipulasi suatu citra dalam bentuk matriks (larik 2 dimensi) agar menjadi larik 1 dimensi. Sehingga, dengan menggunakan teknik ini objek atau citra dapat dianggap sebagai larik 1 dimensi dari suatu vektor. Oleh karena itu diperlukan tipe data yang sesuai. Tipe data yang digunakan untuk menangani sistem vektor adalah tipe data pointer *PbyteArray*. Dengan menggunakan tipe data ini, jumlah maksimum memori akan bersifat dinamis, sesuai dengan ukuran citra yang sedang digunakan.

2.2 Perancangan Sistem

Perancangan sistem dibentuk dibentuk dalam 2 tahap, yaitu menyusun algoritma dalam bentuk diagram alir (*flowchart*) dan implementasi algoritma. Pada Gambar 1 ditunjukkan *flowchart* program yang akan dirancang.



Gambar 1. Diagram alir sistem

User dapat menggunakan 2 jenis masukan, citra skala keabuan dan citra warna. Bila masukan berupa citra skala keabuan, maka program akan meneruskan ke tahap berikutnya, yaitu peningkatan kontras. Bila citra warna, program akan melakukan pilihan proses konversi. Bila ya (Y) citra warna model RGB akan dilakukan proses konversi ke

model IHS. Proses konversi citra warna dilakukan untuk menghasilkan citra skala keabuan. Bila tidak (N) citra warna model RGB langsung dikenakan efek peningkatan. Tahap berikutnya adalah penyimpanan hasil peningkatan kontras. Citra hasil dapat disimpan ke dalam format BMP maupun JPEG.

Berikut algoritma yang diterapkan pada proses perbaikan kualitas citra menggunakan efek peningkatan kontras.

1. Algoritma untuk menghitung frekuensi intensitas

Algoritma ini digunakan untuk menghitung jumlah piksel (cacah piksel) yang menggunakan suatu intensitas warna atau menghitung frekuensi intensitas (berapa kali intensitas suatu warna digunakan). Berikut adalah algoritma yang digunakan pada citra skala keabuan.

- Ambil bit-bit nilai piksel pada bitmap
`FProses.Imagel.Picture.Bitmap.PixelFormat = pf8bit;`
- Lakukan inisialisasi terhadap variabel cacah $c[i]=0$;
- Lakukan perulangan untuk setiap piksel
`for i:= 0 to 255;`
- Analisis setiap piksel pada citra
`inc(c[PC[i]]);`
- Lakukan perulangan untuk setiap piksel
`For i:= 0 to FProses.Imagel.Picture.Height-1;`
`For j:= 0 to FProses.Imagel.Picture.Width-1;`
 Setelah semua piksel dianalisis, nilai cacah maksimum dari semua nilai keabuan dihitung. Algoritmanya adalah sebagai berikut:
- Set nilai cacah maksimum
`Max <> 0 then cMax=max;`
`Max = 0 then cMax=0;`
- Lakukan perulangan untuk setiap piksel citra
`For i:= 0 then 255;`

Pada format citra warna 24 bit (*true color*), algoritma yang diterapkan hampir sama. Perbedaannya, pada format ini setiap piksel memiliki 3 elemen warna sehingga memerlukan 3 *byte* dalam memori.

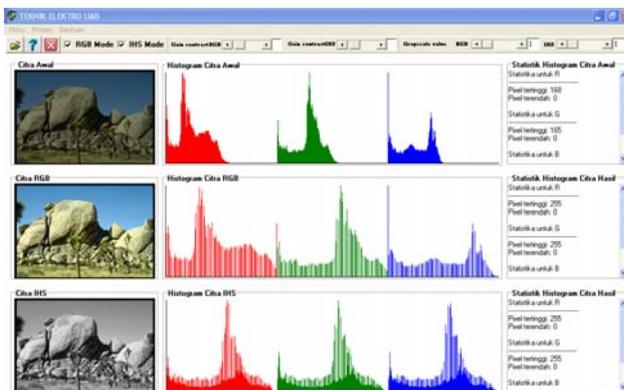
- Algoritma proses peningkatan kontras
 Algoritma ini digunakan untuk mengatur kontras suatu citra. Untuk mengatur kontras suatu citra, diperlukan transformasi dengan menggunakan konstanta-konstanta tertentu. Operasi peningkatan kontras dilakukan secara terpisah pada masing-masing format citra. Sebelum disimpan dalam bentuk *array*, dilakukan pembatasan nilai maksimum dan minimum pada hasil pengontrasan citra. Berikut algoritma efek peningkatan kontras untuk citra skala keabuan:
 - Ambil bit-bit nilai piksel pada bitmap

- ```
FProses.Image1.Picture.Bitmap.PixelF
ormat = pf8bit;
```
- b. Lakukan perulangan untuk setiap piksel  
For i:= 0 to  
FProses.Image1.Picture.Height-1;  
For j:= 0 to  
FProses.Image1.Picture.Width-1;
  - c. Ambil nilai piksel yang akan diproses  
PC:=  
FProses.Image1.Picture.Bitmap.Scanli  
ne[i];  
PH:=  
FProses.Image2.Picture.Bitmap.Scanli  
ne[i];
  - d. Tentukan nilai piksel hasil  
temp:= Round(K\*(PC[j]-P)+P);
  - e. Lakukan pembatasan, yaitu jika nilai piksel  
hasil melebihi 255, maka nilai piksel akan  
diset 255, sebaliknya jika nilai piksel hasil  
kurang dari 0, maka nilai piksel akan di set  
0.  
If (temp > 255) then temp:= 255;  
If (temp < 0 then) temp:= 0;
  - f. Letakkan nilai piksel hasil pada penyangga  
data citra  
PH[i]:= temp;

Pada citra warna, algoritma efek peningkatan kontras yang diterapkan sama, hanya saja pada citra warna dilakukan pada masing-masing elemen warna (R, G dan B).

### 3. HASIL DAN PEMBAHASAN

Program diimplementasikan menjadi suatu perangkat lunak *contrast stretching*. Dengan menggunakan paket pemrograman Delphi 6.0 dibuat sebuah *project* untuk peningkatan kontras. *Project* pada Delphi adalah suatu *file* yang memuat informasi tentang sekumpulan form, unit dan beberapa hal yang lain dalam program aplikasi. Sebenarnya *project* merupakan program aplikasi itu sendiri. *File* utama *project* disimpan dengan ekstensi .dpr. Pada saat dijalankan, *file project* inilah yang selalu dikompilasi menjadi *file* yang dapat dijalankan, yaitu *file* berekstensi .exe dan .dll. Tampilan perangkat lunak perbaikan kualitas citra yang dirancang ditunjukkan pada Gambar 2.

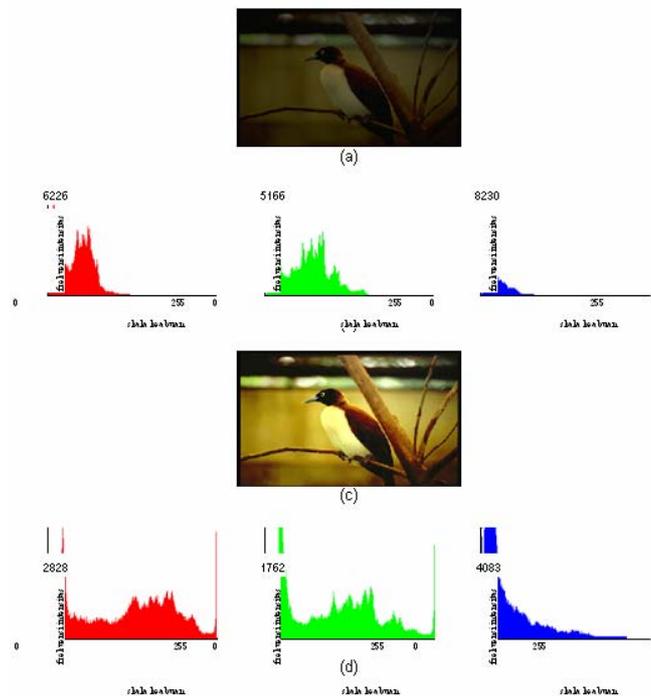


Gambar 2. Tampilan perangkat lunak perbaikan kualitas citra yang dirancang

Pengujian program dilakukan menggunakan beberapa contoh citra uji (*test images*) yang mengalami gangguan kontras. Berikut hasil penelitian menggunakan efek *contrast stretching* pada beberapa citra uji.

#### 1. Citra gelap

Citra uji yang memiliki kontras rendah akan cenderung terlihat gelap. Citra dengan karakteristik seperti ini diperbaiki dengan mengatur koefisien penguatan kontrasnya (G). Gambar 2 adalah contoh peningkatan kontras pada citra Bird.bmp yang memiliki kontras rendah. Hasil citra direpresentasikan dengan model RGB kemudian disimpan dengan menggunakan format *file* JPEG.



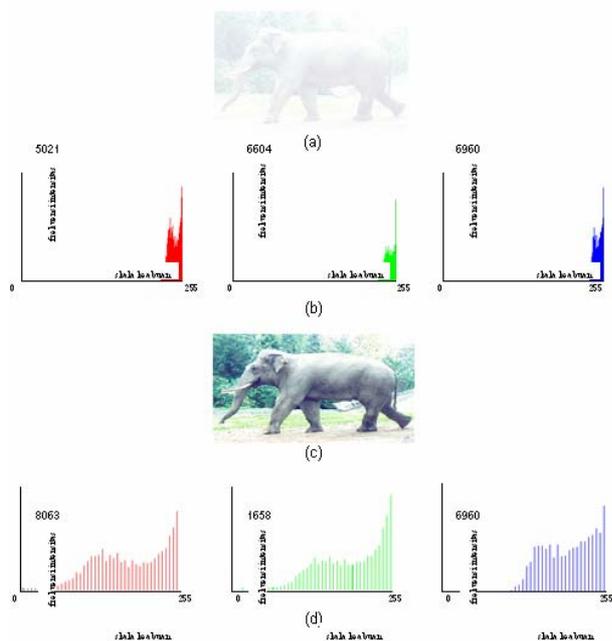
Gambar 2. Contrast Stretching pada citra Bird.bmp (a) Citra asli; (b) Histogram citra asli (R, G, B), (c) Citra hasil format JPEG, (d) Histogram citra hasil format JPEG

Pada Gambar 2.a terlihat bahwa citra asli tampak gelap/lebih redup, dimana koefisien penguatan kontras (G) = 1 dan nilai skala keabuan (P) diatur = 0. Histogram citra asli masing-masing komponen (R,G,B) menyempit dan berada di kiri, hal ini disebabkan karena citra asli tersebut mengalami penurunan kontras, ditunjukkan pada Gambar 2.b. Gambar 2.c adalah citra hasil RGB setelah diberi efek peningkatan kontras (koefisien penguatan kontras (G) = 3; nilai skala keabuan (P) = 0). Setelah dilakukan peningkatan kontras terlihat bahwa citra hasil RGB lebih tajam dari citra asli. Citra hasil RGB tersebut disimpan dengan format *file* JPEG (Bird\_hasil.jpg). Histogram masing-masing komponen (R,G,B) pada citra hasil melebar/renggang karena memiliki jangkauan nilai

keabuan yang lebar. Gambar 2.d adalah histogram citra hasil RGB format *file* JPEG setelah mengalami peningkatan kontras.

## 2. Citra terang

Citra uji yang memiliki kontras tinggi akan sulit untuk dilakukan analisis, karena gambar terlalu tajam (skala keabuan citra hampir mendekati putih). Penanganan citra dengan karakteristik seperti ini dilakukan dengan mengatur koefisien penguatan kontras ( $G$ ) dan nilai skala keabuannya ( $P$ ). Gambar 3 adalah contoh peningkatan kontras pada citra *Elephant.bmp* yang memiliki kontras tinggi. Hasil citra direpresentasikan dalam model RGB, kemudian disimpan dalam format *file* BMP.

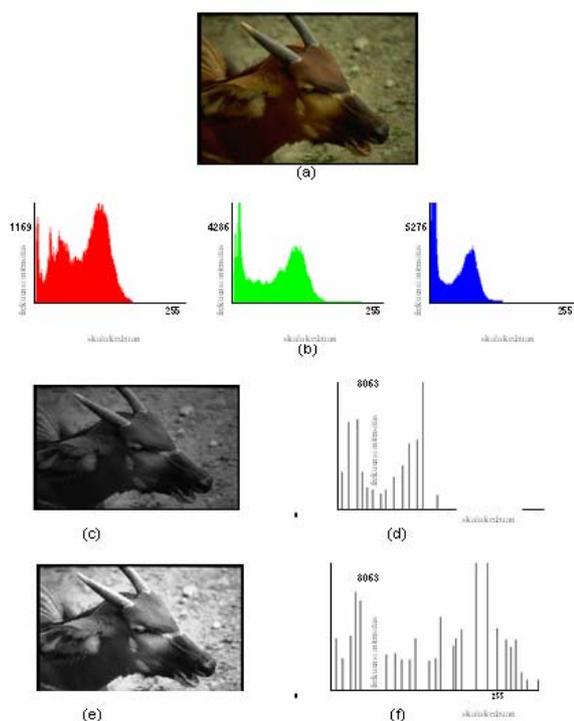


**Gambar 3.** *Contrast Stretching* pada citra *Elephant.bmp*, (a) Citra asli; (b) Histogram citra asli (R, G, B); (c) Citra hasil RGB format BMP; (d) Histogram citra hasil format BMP

Pada Gambar 3.a, citra asli adalah citra warna 8 bit. Citra asli tersebut tergolong citra yang terang karena pada histogramnya, nilai skala keabuan masing-masing komponen (R, G, B) mendominasi di sebelah kanan, seperti ditunjukkan pada Gambar 3.b. Koefisien penguatan kontrasnya ( $G$ ) = 1 dan nilai skala keabuannya ( $P$ ) = 0. Citra tersebut masih sulit untuk dianalisis sehingga parameter kontrasnya perlu diatur. Gambar 3.c adalah citra hasil model RGB. Citra tersebut koefisien penguatan kontrasnya ( $G$ ) = 6, nilai skala keabuannya ( $P$ ) diatur maksimum = 255, terlihat bahwa citra hasil RGB lebih menonjol sehingga citra dapat dilakukan analisis. Histogram citra hasil masing-masing komponen (R, G, B) mengalami pelebaran, sesuai dengan fungsi yang dihasilkan seperti pada Gambar 3.d.

## 3. Citra skala keabuan

Pada citra skala keabuan masing-masing komponen warna (R,G,B) mempunyai nilai piksel yang sama. Citra asli model RGB dikonversi ke citra model IHS, kemudian diberi efek penguatan kontras. Gambar 4 adalah citra skala keabuan yang dihasilkan melalui proses konversi, dari citra model RGB dikonversi ke citra model IHS. Gambar 4.a adalah citra asli RGB warna 8 bit sedangkan Gambar 6.b adalah histogram citra asli masing-masing komponen (R,G,B). Sebelum dilakukan proses peningkatan kontras, untuk menghasilkan citra skala keabuan masing-masing komponen warna (R, G, B) dikonversi terhadap komponen IHS yaitu pada komponen *intensity* (I). Citra warna RGB yang telah dikonversi akan menjadi citra skala keabuan seperti Gambar 4.c.



**Gambar 4.** *Contrast Stretching* pada citra *Anoa.bmp*; (a) Citra asli; (b) Histogram citra asli (R,G,B); (c) Citra konversi IHS; (d)Histogram citra IHS; (e) Citra hasil *contrast stretching*; (f) Histogram citra hasil IHS

Pada citra skala keabuan histogram citra hanya dilukiskan pada 1 warna karena nilai piksel masing-masing komponen (R,G,B) sama. Gambar 4.d adalah histogram citra IHS sebelum dikenakan efek peningkatan kontras. Kemudian citra IHS dinaikkan kontrasnya (koefisien penguatan ( $G$ ) = 2), terlihat bahwa citra semakin terang seperti Gambar 4.e dan histogram citranya juga melebar seperti Gambar 4.f.

#### 4. KESIMPULAN

Berdasarkan pengujian pada beberapa sampel citra uji menunjukkan bahwa perangkat lunak yang dirancang dapat digunakan untuk meningkatkan kontras suatu citra. Perbaikan kualitas citra ini dibuktikan dengan melebarnya histogram citra. Pada citra yang memiliki kontras rendah, dapat ditingkatkan dengan mengatur koefisien nilai kontras dan nilai skala keabuannya.

#### PUSTAKA

- [1] Achmad, B. dan Firdausy, K., 2005, *Teknik Pengolahan Citra Digital menggunakan Delphi*, Ardi Publishing, Yogyakarta
- [2] Andreswari, D., 2002, *Peningkatan Mutu Citra Menggunakan Borland Delphi 5.0*, Skripsi S-1, Universitas Ahmad Dahlan, Yogyakarta.
- [3] Munir, R., 2004, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Penerbit Informatika, Bandung.
- [4] Nalwan, A., 1997, *Pengolahan Gambar secara Digital*, PT. Elex Media Komputindo, Jakarta.