"PENGGUNAAN HISTOGRAM STRETCHING UNTUK IMPROVISASI IMAGE CONTRAS DALAM GRAY-SCALE IMAGE"

Oleh :
Dwi Agus Diartono

LATAR BELAKANG

Kebutuhan akan pengolahan terhadap suatu citra sekarang ini dirasakan dapat membantu k epada p engguna u ntuk k eperluan-keperluan t ertentu, s eperti k eperluan u ntuk dapat mendeteksi suatu objek yang tidak jelas suapaya menjadi lebih jelas atau hal yang lainnya, dari kebutuhan tersebut maka dapat digunakan suatu metode yang digunakan untuk menampilkan suatu edges ataupun bentuk kurva dari sebuah citra yang diolah, untuk dapat diolah citra tersebut harus terlebih dahulu dilakukan pendeteksian terhadap edge tersebut.

Tiap-tiap pixel dalam gray-scale image mempunyai a brightness dari 0 sampai 255, dimana 0 adalah untuk warna hitam dan 255 adalah untuk warna putih. Sebuah histogram menampilkan angka dari pixel-pixel dengan variasi level brightness. Nilai "0" pada kiri histogram menunjukkan jumlah pixel hitam Nilai "255" pada kanan dari histogram menunjukkan angka terhadap pixel yang berwarna putih. Histogram adalah kemungkinan distribusi tingkat kecerahan.

Sebuah gambar memiliki kontras yang rendah pada saat melengkapi range yang mungkin tidak digunakan. Sebagai contoh dalam citra yang ditunjukkan diatas hanya menggunakan nilai 11 sampai 97 dari 0 sampai range 255. Dari citra yang ada tersebut maka dapat kita lakukan perubahan-perubahan dengan mendasarkan terhadap nilai-nilai yang kita dapatkan untuk kemudian kita rubah nilai tersebut dengan nilai baru dan dari nilai baru tersebut akan kita gambarkan kembali.

OPERASI HISTOGRAM

Dalam operasi Histogram ini menggambarkan histogram dari graylevel suatu citra yang memberikan distribusi graylevel dari pixel dalam citra. Histogram suatu graylevel dikenali sebagai sekumpulan M (jumlah graylevel yang mungkin) yang menjelaskan persentase image di satu nilai graylevel. Histogram satu image dikenali sebagai

$$h_i = \frac{n_i}{n_t}$$

untuk I – 0 ... (M-1)

dimana n_i adalah jumlah pixel dalam citra di gray level ke-I dan n_t adalah jumlah total pixel dalam suatu image.

PEMROSESAN WARNA CITRA

Analisa warna cahaya sudah dilakukan beberapa abad yang lampau. Di abad 17 Sir Isaac Newton menunjukkan sorotan cahaya matahari melalui gelas prisma yang memunculkan pelangi/bianglala warna. Newton menarik kesimpulan bahwa cahaya putih terjadi dari banyak warna yang berbeda.tabel dibawah ini daftar daerah 6 warna utama dan sekelompok panjang gelombangnya dalam nanometer.

Tabel panjang gelombang untuk 6 daerah warna utama.

Color	Panjang Gelombang
Violet	400 - 450 nm
Blue	450 – 480 nm
Green	480 – 550 nm
Yellow	550 – 580 nm
Orange	580 – 610 nm
Red	610 – 700 nm

Dalam perkembangannya ,Clerk E.Maxwell menunjukkan bahwa satu warna citra bisa dibuat menggunakan tiga warna citra. Warna tersebuat adalah red(R), blue(B), dan green(G), semua warna yang ada merupakan pencampuran dari ketiga warna berikut.

Persentase warna Red, Blue dan Green dalam satu warna diketahui sebagai

$$r = \frac{R}{R + B + G}$$

$$b = \frac{B}{R + B + G}$$

$$g = \frac{G}{R + B + G}$$

trichromatic coefficients warna yaitu :

Dimana R,B,dan G jumlah cahaya red,blue,dan green, secara berurutan. Trichromatic coefficients sumbu X untuk red,r, sumbu Y untuk warna green,g. Trichromatic coefficients blue,b bisa dihitung menggunakan b=1-r-g. contoh warna merah di chart C.I.E denan panjang gelombang 625 nm memiliki trichromatic coefficients berikut : r=71%,b=0%, dan g=29%, menghasilkan warna kombinasi 0% white dan memenuhi corak warna rd 71% dan green 29%

Warna Pixel

Pada kartu dan penampil warna, jumlah warna yang dapat ditampilkan berhubungan dengan jumlah bit memori yang tersedia untuk menyimpan warna. Jumlah warna yang dapat dihasilkan oleh sebuah kartu penampil dapat dihitung dengan menggunakan rumus :

$Jumlah\ warna = 2^n$

dengan :

n = bits per pixel

Sebagai contoh, kartu EGA menggunakan 4 bit untuk menyimpan satu pixel sehingga kartu EGA dapat menampilkan maksimum sampai 16 (2^4) warna. Kartu VGA menggunakan 8 bits per pixel atau sama dengan 256 (2^8) warna.

Resolusi

Selain warna, mutu dari kartu tampilan juga ditentukan dari resolusi yang dapat dihasilkan. Resolusi adalah jumlah pixel yang dapat ditampilkan untuk setiap garis garis horisontal dan vertikal. Sebagai contoh, kartu monochrome mempunyai resolusi maksimum sebesar 720 * 348 pixel, ini berarti bahwa kartu tersebut dapat menampilkan maksimum 720 pixel pada garis horisontal dan 348 pixel pada garis vertikal.

Pseudocolor

Pseudocolor graylevel citra merupakan graylevel sendiri dalam citra yang harus dipetakan ke sekumpulan warnaimage red,blue dan green.

Dengan mempertimbangkan satu citra dengan nilai graylevel N diskrit digambarkan oleh fungsi F. satu image bisa dipertimbnangkan sebagai sekumpulan nilai graylevel yang diberikan oleh fungsi F untuk semua pxel dalam citra. Berikutnya, fungsi F ini bisa dipetakan ke tiga ufngsi R,B dan G yang mengghasilkan keluaran warna merah (red), biru (blue), dan hijau (green). Gambar dibawah ini menunjukkan blok diagram pseudocolor suatu citra. Graylevel fungsi citra F(x,y) dipetakan ke tiga citra R(x,y),B(x,y) dan G(x,y). setiap citra ini kemudian digunakan untuk memodulasikan tampilan gambar warna red,blue dan green .

Untuk menampilkan citra graylevel, fungsi pemetaan,R,G,dan B berisi fungsi pemetaan yang sama dengan nilai masukan graylevel F. tiga fungsi pemetaan bisa digunakan untuk mengendalikan keterangan (brgihtness) dan kontras (contrast) citra dengan memodifikasi semua persamaan fungsi pemetaan.

Untuk menghasilkan citra kontras maka R=B=G=0.5F. keluaran nilai intensitas ketiga citra adalah setengah dari nilai graylevel masukan citra.

Untuk menghasilkan graylevel yang highlight (menyorot) nilai T di warna merah, persamaan berikut ini bisa digunakan yaitu :

$$G(x,y) = \begin{cases} R=F;B=G=0 & \text{untuk } F(x,y)=T \\ R=G=B=F & F(x,y) <> T \end{cases}$$

Di persamaan ini, untuk F(x,y) sama untuk nilai graylevel T, citra biru dan hijau diset = 0 atau untuk intensitas rendah dan citra merah diset sama dengan masukan nilai graylevelnya. Untuk semua nilai graylevel yang lain citra merah,biru,dan hijau diset sama dengan masukan nilai graylevelnya, menghasilkan citra putih dan hitam untuk nilai masukan graylevel ini.

Representasi Warna

Ada banyak skema untuk merepresentasikan warna. Sistem Red-Green-Blue (**RGB**) secara luas digunakan dalam perangkat keras computer. Dalam merepresentasikan warna ada tiga komponen yaitu : Red (**R**), Green (**G**) and Blue (**B**). biasanya komponen-komponen tersebut dinormalisasikan dari nilai 0 sampai 1. warna **RGB** bisa satu titik dalam bentuk kerucut dengan tiga komponen sebagai sudutnya. Warna hitam merupakan gabungan dari nilai **R=0, G=0, B=0** dan untuk warna putih dengan nilai **R=1, G=1, B=1**. Intensitas maksimum adalah warna biru, contoh : **R=0, G=0, B=1**. Bayangan keabuan terletak sepanjang garis diagonal dari sudut warna hitam ke sudut warna putih.

Sistem Cyan-Magenta-Yellow (**CMG**) merupakan varian dari sistem **RGB** yang secara luas digunakan dalam industri pencetakan. Hubungan antara system RGB dan CMY (untuk komponen-komponen ternormalisasi) adalah trivial yang diberikan sebagai berikut :

$$\begin{aligned} C &= 1 - R \\ M &= 1 - B \\ Y &= 1 - G \end{aligned} \qquad (1)$$

Sistem Hue-Saturation-Value (**HSV**) yang awalnya diperkenalkan oleh Smith[1] dan secara luas dipergunakan di perangkat lunak visualisasi. Sistem warna **HSV** mempergunakan kerucut terbalik, dalam kerucut satu warna merupakan satu titik yang dikenali dengan tiga komponen yaitu Hue, Saturation dan value.

Value merupakan jarak titik dari puncak kerucut. Diukur sepanjang sumbu utama. **V=0** adalah puncak kerucut dan sesuai untuk hitam (nilai komponen yang lain disini tidak relevan). Satu titik pada dasar kerucut dengan **V=1** dan titik dimana sumbu utama memotong dasar kerucut sesuai untuk warna putih. Titik lanjutan sepanjang sumbu utama disesuaikan untuk bayangan keabuan (shades of grey)

Hue sudutnya diukur mengitari sumbu utama dari kerucut. Untuk titik di luar lingkaran k erucut yang d isesuaikan d engan t eori w arna ; **H=0** d isesuaikan u ntuk w arna merah, $H = 0.333...$(atau $120°$) disesuaikan dengan warna hijau dan $H = 0.666...$ (atau $240°$) disesuaikan dnegan warna biru. **H=1** lengkap satu rotasi dan disesuaikan dengan kembali ke warna merah . Warna komplementer berada pada sisi yang sama sekali berlawanan dari kerucut (dibedakan oleh $180°$dalam **H**). titikpada lingkaran dasar kerucut (pada **V=1**) secara teori memiliki warna dengan intensitas maksimum. Titik-titik diluar kerucut (yaitu $V < 1$)memiliki warna yang sama yang disesuaikan dengan warna dilingkaran dasar, tetapi berada pada intensitas yang berkurang.

Saturation adalah jarak dari sumbu utama , yang diproyeksikan pada dasar kerucut. Diukur dengan kemurnian warna , yang dikembangkan dari penipisan dengan warna putih. Untuk warna yang terletak pada dasar kerucut maka **S=0** pada sumbu utama dan disesuaikan ke warna putih , **S=1** terletak pada edge kerucut dan disesuaikan dengan warna murni, c ontoh **H=0, S=1, V=1** a dalah w arna m erah m urni d engan i ntensitas m aksimum.

Titik yang lebih lanjut dari kerucut (yaitu V<1) disesuaikan dengan intensitas yang berkurang.

Keuntungan utama dari system *HSV* adalah penyesuaian kondisi secara tertutup bagi persepsi manusia pada warna, efek penentuan dan manipulasi warna dalam sistem HSV yang sangat intuitive dan bisa diprediksikan. Memungkinkan untuk mentranformasikan sistem HSV dan RGB. (Smith,1996)

Histogram Equalization

Histogram equalization menyeragamkan pendistribusian kembali nilai-nilai graylevel pixel dalam satu citra dimana sejumlah pixel berada di salah satu graylevel yang kurang lebih sama. Gray level mentransfoormasikan equalisasi histogram suatu image dengan :

Rumus :

$$g_i = \frac{M-1}{n_t} \sum_{j=0}^{i} n_i$$

Dimana :

nt = total jumlah pixel dalam citra,
ni = jumlah pixel di graylevel I, dan
M = total jumlah yang mungkin dari graylevel

Erotion dan Dilation

Operasi *Erotion* dan *Dilation* membuat obyek untuk menjadi lebih kecil dan lebih besar. Operasi ini berharga bagi dirinya sendiri dan merupakan dasar bagi operasi *Opening* dan *Closing*.

Erotion (Binary) Filter

Erotion Biner (Pengikisan biner) suatu obyek untuk mengurangi daerah geometrinya dengan mensetting kontur pixel-pixel obyek ke nilai latarbelakang. Erotion dikenali sebagai komplemen hasil dilation dari obyek A dengan fungsi struktur B,

Erotion (Gray Level) Filter

Pengikisan gray-level suatu citra yang digunakan untuk penghalusan (smooth) daerah gray-level positif. Daerah yang tererosi dikenali sebagai perbedaan minimum antara daerah lokal citra dan memberikan mask gray-level sebagai berikut :

$$A \ominus B = \min[A(x+I,y+j) - B(I,j)],$$

Dimana koordiinat x+I,y+j dikenali melalui citra A dan koordinat I,j dikenali melalui mask B. untuk kasus khusus ketika semua nilai mask = 0, erosi graylevel berkurang untuk filter minimum.

Dilation (Graylevel) Filter

Dilasi Graylevel satu citra digunakan untuk penghalusan daerah graylevel negatif . Dilasi Graylevel dikenali sebagai jumlah maksimum satu daerah lokal suatu citra dan memberikan satu mask graylevel berikut :

$$A \oplus B = \max[A(x+i,y+j) + B(i,j)],$$

Dimana koordiinat x+i,y+j dikenali melalui citra A dan koordinat i,j dikenali melalui mask B. Untuk kasus khusus ketika semua nilai mask = 0, dilasi graylevel berkurang untuk filter maksimum.

Dilation (Binary) Filter

Dilasi biner suatu obyek menambah daerah geometrinya dengan mensetting pixel-pixel latarbelakang yang berbatasan/berdekatan ke suatu kontur obyek unutk nilai

graylevel obyek. Dilasi dikenali sebagai union dari semua vektor tambahan semua pixel *a* dalam sutu obyek A dengan semua pixel *b* dalam fungsi struktur B.

$$A \oplus B = \{t \in Z^2 : t = a+b, a \in A, b \in B\},$$

Dimana vektor t adalah elemen ruang citra (image space) Z^2

Pembahasan transformasi graylevel ke warna meliputi perubahan yang terjadi pada data gambar dari graylevel sampai ke pencampuran yang dihasilkan oleh tiga warna dasar yaitu merah,hijau dan biru menggunakan fungsi sinuidal dan frekuensi untuk melihat kekontrasan dan terang atau tidaknya gambar.

Fungsi transformasi yang digunakan seperti yang ditunjukkan di gambar bawah . Fungsi Sinusoidal ini berisi daerah yang secara relatif berisi nilai konstan di sekitar puncaknya seperti daerah dimana perubahan dengan cepat terjadi disekitar lembahnya.Dengan perubahan fase dan frekuensinya di setiap sinusoid memungkinkan untuk memberi tekanan (dalam warna) pada range di skala gray.

Jika tiga transformasi memiliki phase dan frekuensi yang sama , keluaran citra akan monokrom. Satu perubahan kecil pada phase diantara ketiga transformasi akan menghasilkan perubahan yang kecil di pixel dimana gray level mengumpul disekitar puncak dalam sinusoida, terutama jika sinisoid memiliki profile yang luas (yaitu frekuensi rendah).

Pixel dengan nilai graylevel di bagian yang curam dari sinusoida ditandai dengan warna yang lebih kuat jika menghasilkan perbedaan yang signifikan antara amplitudo ketiga sinusoida yang disebabkan oleh perpindahan phase ketiganya.

File gambar yang digunakan format grafik raw.

Perubahan phase dihitung berdasarkan fungsi sinusnya ditambah dengan pergeserannya untuk menyatakan pencampuran warna dengan nilai yang berbeda. Untuk menentukan graylevel pergeseran ketiga warna dasar (red,green, dan blue)dibuat sama, kemudian dalam perubahan intensitas pencampuran ketiga warna yang dihasilkan diperoleh dengan

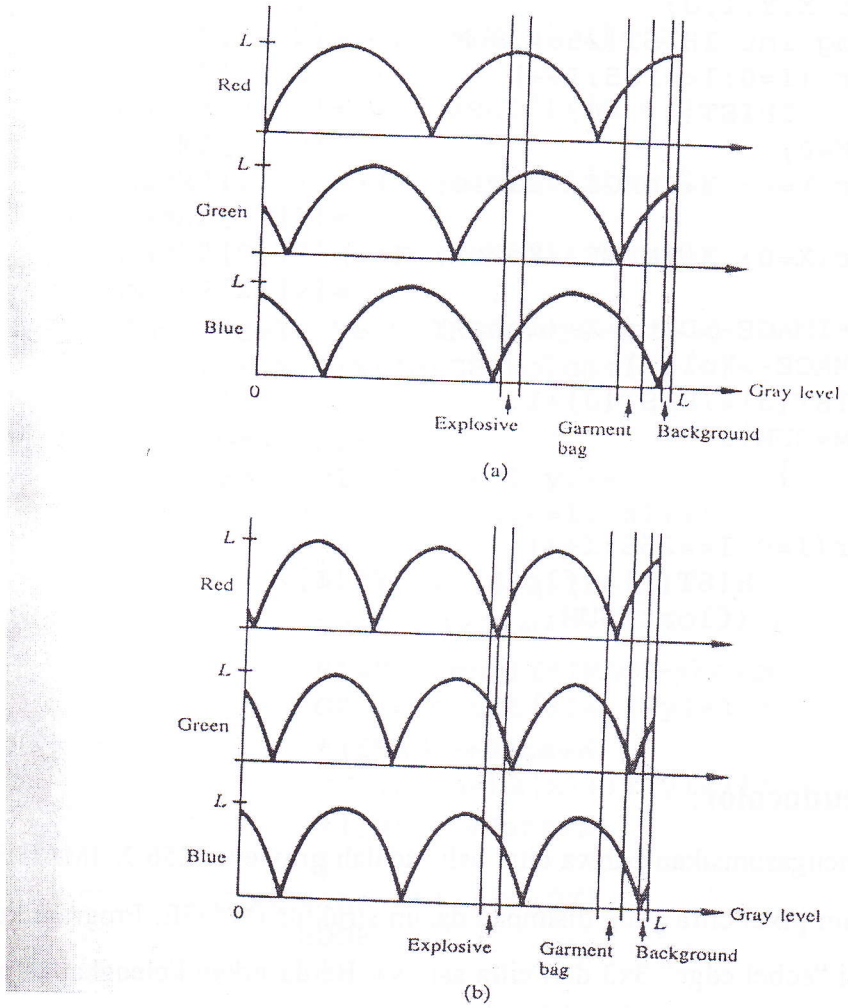
memberikan nilai pada masing-masing warna yang berbeda untuk menghasilkan warna dengan intensitas sesuai dengan persentasi ketiga warna yang diberikan.

Perhitungan intensitas dilakukan dengan pergesaran phase dari ketiga fungsi sinosoida warna tersebut.

Karena fungsi yang digunakan merupakan fungsi sinusoida yang memiliki dua puncak optimal yaitu bagian puncak dengan nilai positif dan bagian puncak yang lain bernilai negatif. Untuk menentukan transformasi dari gray level ke intensitas warna menggunakan fungsi yang positif maka perhitungan untuk fungsi sinus yang dihasilkan dibuat absolut agar bila diperoleh nilai negatif menjadi nilai positif semua.Dan semua puncak sinusoida bernilai positif.

Untuk menyatakan frekuensinya, perhitungan yang dilakukan dengan mengalikan dua fungsi sinusoidanya. Dengan perhitungan tersebut untuk bisa diketahui dalam setengah gelombang frekuennya bisa ditentukan apapkah gambar dalam keadaan kontrans, gelap atau terang (menentukan contranst dan brightness).

Gabungan kedua fungsi ini (dalam menentukan phase dan frekuensinya) berguna untuk menentukan kecerahan, kekontrasan, warna suatu gambar.



Gambar 1.
Fungsi transformasi digunakan untuk gambar

Algoritma Operasi Histogram.

Program mengasumsikan bahwa citra asli adalah graylevel 256 x IMAGE ->baris X IMAGE -> Kolom pixel citra.

Program kemudian menghitung histogram citra dan menyimpan hasil histogramnya dalam array floating point HIST[].

```
Histogram(struct Image *IMAGE, float HIST[])
    {
```

```
int X,Y,I,J;
long int IHIST[256],SUM;
for (I=0;I<=255;I++)
     IHIST[I] =0;
SUM=0;
For Y=0; Y<IMAGE->Baris; Y++)
{
for(X=0; X<IMAGE->Kolom; X++)
{
J=*IMAGE->Data+X+(long)Y
*IMAGE->kolom);
IHIST[J]=IHIST[J]+1;
SUM=SUM+1;
     }
}
for(I=0;I<=255;I++)
       HIST[I]=(float) IHIST[I]/
       (float)SUM;

}
```

Algoritma Pseudocolor:

Program mengasumsikan bahwa citra asli adalah graylevel 256 X IMAGE→Baris X IMAGE→ kolom pixel citra yang disimpan dalam struktur IMAGE. Program kemudian melakukan deteksi "sobel edge" 3x3 dari citra aslinya. Berdasarkan kelengkapan program, edge dalam citra disorot dengan warna merah dan disimpan dalam struktur RED,GREEN dan BLUE. Struktur tiga warna diasumsikan berada dalam ukuran warna dan tipe yang sama sebagai citra asli.

```
Pseudo_edge(struct image *IMAGE,struct
Image *RED,struct Image * Blue, struct
Image *GREEN, int T)
     {
     int X,Y,x1,y1, mask1[3][3];
     int GX,GY,EDGE;
     long int,R,R1;
     mask1[0][0]=-1; mask[1][0]=-2;
     mask1[2][0]=-1;
     mask1[0][1]=-1; mask[1][1]= 0;
```

```
mask1[2][0]= 0;
mask1[0][2]= 1; mask[1][2]= 2;
mask1[2][2]= 1;
mask2[0][0]=-1; mask2[1][0]= 0;
mask2[2][0]= 1;
mask2[0][1]=-2; mask2[1][1]= 0;
mask2[2][1]= 2;
mask2[0][2]=-1; mask2[1][2]= 0;
mask2[2][2]= 1;
for  (Y=1; Y<IMAGE→baris-1;Y++)
    for(X=1;X<IMAGE>kolom-1;X++)
     {
     GX=0;GY=0;
     For (y1=-1;y1<=1;y1++)
        For (x1=-1;x1<=1; x1++)
          {
          R=X+x1+(long)(Y+y1)*
          IMAGE→kolom;
          R1=X+(long)Y*IMAGE→kolom;
          GX += mask1[x1+1][y1+1]*
          *(IMAGE→Data+R);
          GY += mask2[x1+1][y1+1]*
          *(IMAGE→Data+R);
          }
     EDGE=abs(GX)+abs(GY);
     If (EDGE > T)
          {
          *(RED→Data+R1)=255;
          *(BLUE→Data+R1)=0;
          *(GREEN→Data+R1)=0;
     else
          *(RED→Data+R1)=
          *(IMAGE→Data+R1);
          *(BLUE→Data+R1)=;
          *(IMAGE→Data+R1);
          *(GREEN→Data+R1)=;
          *(IMAGE→Data+R1);
          }
        }
     }
```

Algoritma Histogram Equalization:

Program mengasumsikan bahwa citra asli = 256 graylevel x IMAGE→baris x IMAGE→kolom pixel citra yang disimpan dalam struktur IMAGE. Program akan menghitung histogram dari citra menggunakan program histigram yang diberikan dalam penyimpanan teks. Histogram dihasilkan dalam array floating point HIST[I]. Terakhir program menyimpan citra yang terqualisasi dalam struktur IMAGE1.

```
Histogram_Equalization(struct Image * IMAGE,
Struct Image *IMAGE1)
      {
      int X,Y,I,J;
      int HISTEQ[256];
      float HIST[256],SUM;
      Histogram(IMAGE,HIST);
      For (J=0; J<=I; J++)
           SUM=SUM+HIST[J];
      HISTEQ[I]=(int)(255*SUM+.5);
      }
for (Y=0; Y<IMAGE→baris; Y++)
   {
      for(X=0; X<IMAGE→kolom; X++)
      {
      (IMAGE→Data+X+(long)Y*
      IMAGE→kolom)=
      HISTEQ[*(IMAGE→Data+X+(long)Y*
      IMAGE→kolom)];
      }
   }
}
```

Erotion dan Dilation.

Operasi *Erotion* dan *Dilation* membuat obyek untuk menjadi lebih kecil dan lebih besar. Operasi ini berharga bagi dirinya sendiri dan merupakan dasar bagi operasi *Opening* dan *Closing*.

Operasi Erotion membuat suatu obyek menjadi lebih kecil dengan membuang atau mengikis jauh pixel-pixel pada *edge*nya.

Gambar 3 menunjukkan hasil pengikisan bentuk persegi panjang di gambar 2

Dilation membuat obyek menjadi lebih besar dengan menambahkan pixel-pixel di sekitar edgenya. Gambar 4 menunjukkan hasil dilation bentuk persegi panjang di gambar 2. Dengan mengumpulkan beberapa pixel nol (0) dimana berikutnya diubah ke pixel 200 (ditunjukkan sebagai bintang). (Phillips, 1992)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	200	200	200	200	200	200	0	0
0	0	200	200	200	200	200	200	0	0
0	0	200	200	200	200	200	200	0	0
0	0	200	200	200	200	200	200	0	0
0	0	200	200	200	200	200	200	0	0
0	0	200	200	200	200	200	200	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Gambar 2
Satu Citra biner dengan background = 0 dan obyek = 200

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	200	200	200	200	0	0	0
0	0	200	200	200	200	200	200	0	0
0	0	200	200	200	200	200	200	0	0
0	0	200	200	200	200	200	200	0	0
0	0	200	200	200	200	200	200	0	0
0	0	0	200	200	200	200	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Gambar 3
Hasil Pengikisan (erotion) obyek persegi panjang di gambar 2

0	0	0	0	0	0	0	0	0	0
0	***	***	***	***	***	***	***	***	0
0	***	200	200	200	200	200	200	***	0
0	***	200	200	200	200	200	200	***	0
0	***	200	200	200	200	200	200	***	0
0	***	200	200	200	200	200	200	***	0
0	***	200	200	200	200	200	200	***	0
0	***	200	200	200	200	200	200	***	0
0	***	***	***	***	***	***	***	***	0
0	0	0	0	0	0	0	0	0	0

Gambar 4

Hasil pelebaran (dilation) obyek persegi panjang di gambar 2

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	200	200	200	200	0	0	0
0	0	0	200	200	200	200	0	0	0
0	0	0	200	200	200	200	0	0	0
0	0	0	200	200	200	200	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Gambar 5

Hasil pengikisan (eroting) obyek persegi panjang di gambar 2 menggunakan treshold 2

0	0	0	0	0	0	0	0	0	0
0	0	0	***	***	***	***	0	0	0
0	0	200	200	200	200	200	200	0	0
0	***	200	200	200	200	200	200	***	0
0	***	200	200	200	200	200	200	***	0
0	***	200	200	200	200	200	200	***	0
0	***	200	200	200	200	200	200	***	0
0	0	200	200	200	200	200	200	0	0
0	0	0	***	***	***	***	0	0	0
0	0	0	0	0	0	0	0	0	0

Gambar 6

Hasil pelebaran (dilation) obyek persegi panjang di gambar 3 menggunakan treshold 2

Pembacaan terhadap file gambar yang dipilih maka dari hasil pembacaan tersebut akan menghasilkan data yang siap diolah kedalam bentuk statistik untuk dibuat kedalam histogram, histogram distribusi frequency dihitung, seperti angka pada statistik, ketika sebuah histogram mempunyai puncak yang tunggal, statistik ini juga membantu didalam membaca file image, distribusi frekuensi histogram dihitung, juga sejumlah statistik yang lain. Pada saat histogram puncaknya dominan, statistik ini sering berguna untuk menggambarkan puncaknya.

"Range interes" dihitung dari histogram beradasarkan persentase parameter. Dalam contoh yang ditunjukkan diatas , nilai 1% dan 99% ditentukan menggunakann histogram. Normalnya "range interest" (Range yang asli) distretche secara penuh dengan range 0 to 255 untuk memperbaiki kecerahan dari suatu citra. Nilai persentase, berpengaruh, dikenali berapa banyakpixel yang lengkap dari 0 (hitam) dan 255 (putih).

StretchFactor yang normalnya 1.00, digunakan untuk menghitung ScaleFactor histogram:

```
StretchedRange := OriginalRange + ROUND( StretchFactor * (255 - OriginalRange) );
ScaleFactor := StretchedRange / OriginalRange;
```

Intensity setiap pixel diinspected dan ditandai kembali berdasarkan pada ScaleFactor:

```
NewIntensity := ROUND( ScaleFactor * (OldIntensity - OriginalRangeLeft) );
```

Penggunaan warna selain dari warna dasar dilakukan pencampuran dari ketiga warna dasar yaitu merah, hijau dan biru dan hasil yang diperoleh berdasarkan persentase dari ketiga warna tersebut apakah warna yang dominan merah, hijau atau biru. Bila nilai yang diberikan sama untuk ketiga warna dasar tersebut akan menghasilkan warna gray-level.

Selain menentukan warna lain dari warna dasar juga dilihat kekontrasan, terang dan gelapnya permukaan data gambar. Dalam menentukan kekontrasan (contrast) dan kecerahan (brightness) dengan menentukan frekuensi dari ketiga warna dasar.

Histogram merupakan cara paling efektif untuk mengimprovisasikan contrast dari gray-scale image.Histogram stretching warna image untuk mengaplikasikan teknik memperlihatkan warna ke dalam R, G dan B dengan menghasilkan image dengan contras yang tinggi, tetapi juga memperkenalkan warna yang tidak diinginkan digeser. Sebagai contoh image yang kurang kontras dari kuning tulip dengan daun hijau ketika histostretched menampilkan kuning tulip tetapi hampir menyerupai daun biru.

DAFTAR PUSTAKA

[Alwi98] **Alwi, H., Soenjono Dardjowidjojo, Hans Lapoliwa, Anton M. Moeliono,** *Tata Bahasa Baku Bahasa Indonesia;* Departemen Pendidikan dan Kebudayaan Republik Indonesia, Jakarta 1998

[Frakes92] **Frakes, B. William.,** dan **Baeza Tates Richardo,** *Informaton Retrieval : Data Structure and Algorithms*; Prentice Hall, New Jersey.

[John] **Johnson, S. C.,** *YACC: Yet Another Compiler-Compiler*; AT&T Bell Laboratories, New Jersey.
http://www.csc. calpoly.edu/~gfischer/450/doc/yacc/paper.txt

[Lesk] **Lesk, M. E.** dan **Schmidt, E.,** *Lex – A Lexical Analyzer Generator.*
http://www.cs. ucsb.edu/~cs160/machines/lex-docs.txt

[Sage81] **Sager, N.** *Natural Language Information Processing: A Computer Grammar of English and Its Aplications*; Addison-Wesley Publishing Company, Massachusetts 1981