BAB 3

Operasi Piksel dan Histogram

Setelah bab ini berakhir, diharapkan pembaca memahami berbagai bahasan berikut.

- ✓ Operasi piksel
- ✓ Menggunakan histogram citra
- ✓ Meningkatkan kecerahan
- ✓ Meregangkan kontras
- ✓ Kombinasi kecerahan dan kontras
- ✓ Membalik citra
- ✓ Pemetaan nonlinear
- ✓ Pemotongan aras keabuan
- ✓ Ekualisasi histogram

3.1 Operasi Piksel

Pada pengolahan citra terdapat istilah operasi piksel atau kadang disebut operasi piksel-ke-piksel. Operasi piksel adalah operasi pengolahan citra yang memetakan hubungan setiap piksel yang bergantung pada piksel itu sendiri. Jika f(y, x) menyatakan nilai sebuah piksel pada citra f dan g(y, x) menyatakan piksel hasil pengolahan dari f(y, x), hubungannya dapat dinyatakan dengan

$$g(y,x) = T(f(y,x)) \tag{3.1}$$

Dalam hal ini, T menyatakan fungsi atau macam operasi yang dikenakan terhadap piksel f(y, x). Model operasi inilah yang akan dibahas di bab ini, termasuk pembahasan pengolahan citra berbasis histogram.

3.2 Menggunakan Histogram Citra

Histogram citra merupakan diagram yang menggambarkan frekuensi setiap nilai intensitas yang muncul di seluruh piksel citra. Nilai yang besar menyatakan bahwa piksel-piksel yang mempunyai intensitas tersebut sangat banyak.

Pada citra berskala keabuan, jumlah aras keabuan (biasa disimbolkan dengan L) sebanyak 256. Nilai aras dimulai dari 0 hingga 255. Adapun histogram untuk suatu aras dinyatakan dengan `hist(k+1)` dengan k menyatakan nilai aras (0 sampai dengan L-1). Jadi, `hist(k+1)` menyatakan jumlah piksel yang bernilai k. Penggunaan k+1 pada `hist` diperlukan mengingat dalam *Octave* dan *MATLAB* tidak ada indeks nol atau `hist(0)`. Cara menghitung `hist(k+1)` ditunjukkan pada algoritma berikut.

ALGORITMA 3.1 – Menghitung histogram citra aras keabuan

Masukan:

- f(M, N) : citra berukuran M baris dan N kolom
- L : jumlah aras keabuan

1. Buatlah larik hist sebanyak 2^L elemen dan isi dengan nol.
2. FOR i ← 1 TO M
   FOR j ← 1 TO N

```
            hist(f(M, N)+1) ← hist(f(M, N)+1) + 1
        END-FOR
    END-FOR
```

Contoh berikut menunjukkan cara membuat histogram citra innsbruck.png.

Program : histo.m

```
function histo(Img)
% HISTO Digunakan sebagai contoh pembuatan histogram

[jum_baris, jum_kolom] = size(Img);
Img = double(Img);

Histog = zeros(256, 1);
for baris=1 : jum_baris
    for kolom=1 : jum_kolom
        Histog(Img(baris, kolom)+1) = ...
            Histog(Img(baris, kolom)+1) + 1;
    end
end

% Tampilkan dalam bentuk diagram batang
Horis = (0:255)';
bar(Horis, Histog);
```

Akhir Skrip

Perlu diketahui, (0:255) untuk membentuk nilai dari 0,1,2, dan seterusnya sampai dengan 255. Dengan kata lain, (0:255) membentuk larik 1 x 256. Tanda ‘ di belakang (0:255) menyatakan operasi transpos, sehingga hasilnya berupa larik berukuran 256x1. Perintah **bar** digunakan untuk membuat diagram batang.

Catatan

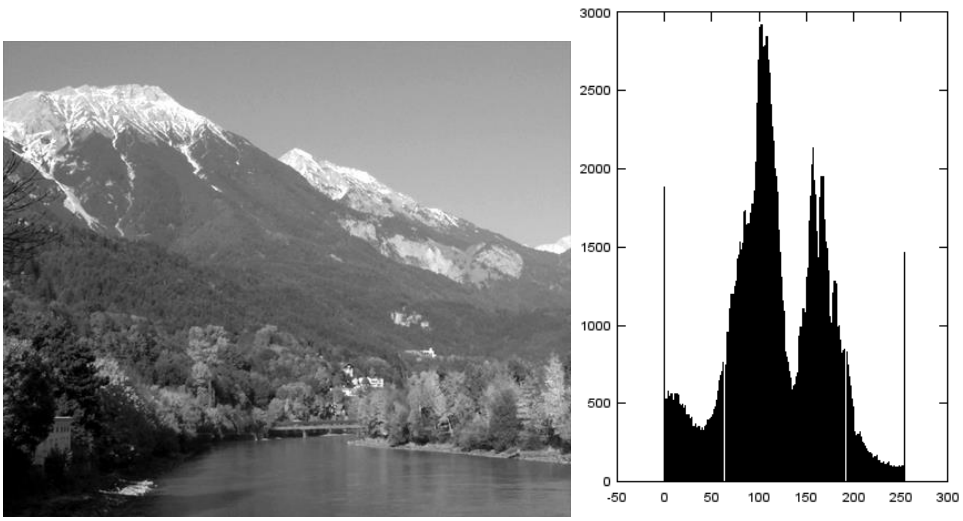
Misalnya, B berisi 0, 10, 8, …, 5, 20 (berukuran 1 x 256). Bila dilakukan operasi transpos (B'), diperoleh matriks berukuran 256 x 1. Hasilnya seperti berikut:

0
10
8
…
5
20

Dengan memanggil

```
>> Img = imread('C:\Image\innsbruck.png'); ↵
>> histo(Img); ↵
```

diperoleh hasil seperti terlihat pada Gambar 3.1. Perhatikan keberadaan satu garis yang cukup panjang di posisi intensitas nol, yang berasal dari bagian citra yang berwarna hitam. Adapun puncak histogram di posisi intensitas sekitar 90 menyatakan warna dominan abu-abu. Garis panjang di sisi kanan menyatakan warna putih.

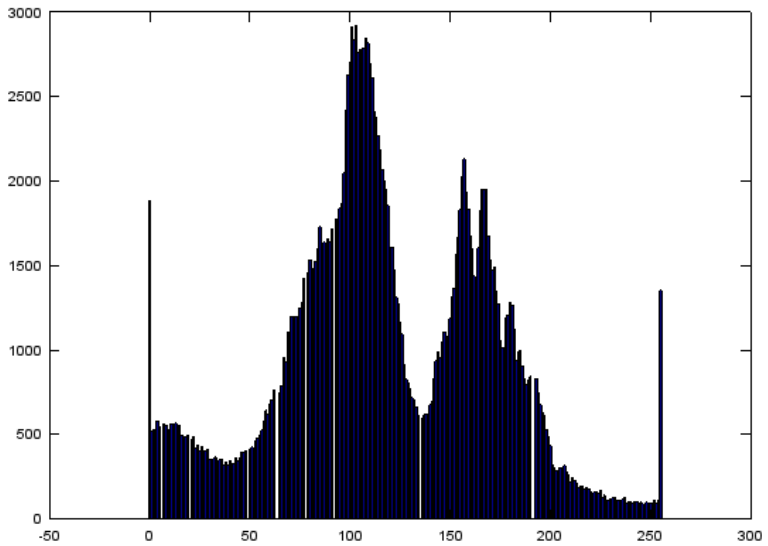


Gambar 3.1 Citra innsbruck.png dan histogramnya

Untuk kemudahan dalam mengamati histogram, fungsi bawaan bernama **imhist** dapat dimanfaatkan. Contoh penggunaannya:

```
>> Img=imread('C:\Image\innsbruck.png'); ↵
>> imhist(Img); ↵
```

Hasilnya ditunjukkan pada Gambar 3.2.



Gambar 3.2 Hasil histogram dengan imhist

Untuk mengetahui nilai histogram, diperlukan perintah seperti berikut:

```
>> [Histog, aras] = imhist(Img); ↵
```

Dengan cara seperti itu, `Histog` berupa larik yang berisi jumlah piksel setiap nilai aras dalam argumen `aras`. Namun, diagram tidak dibuat.

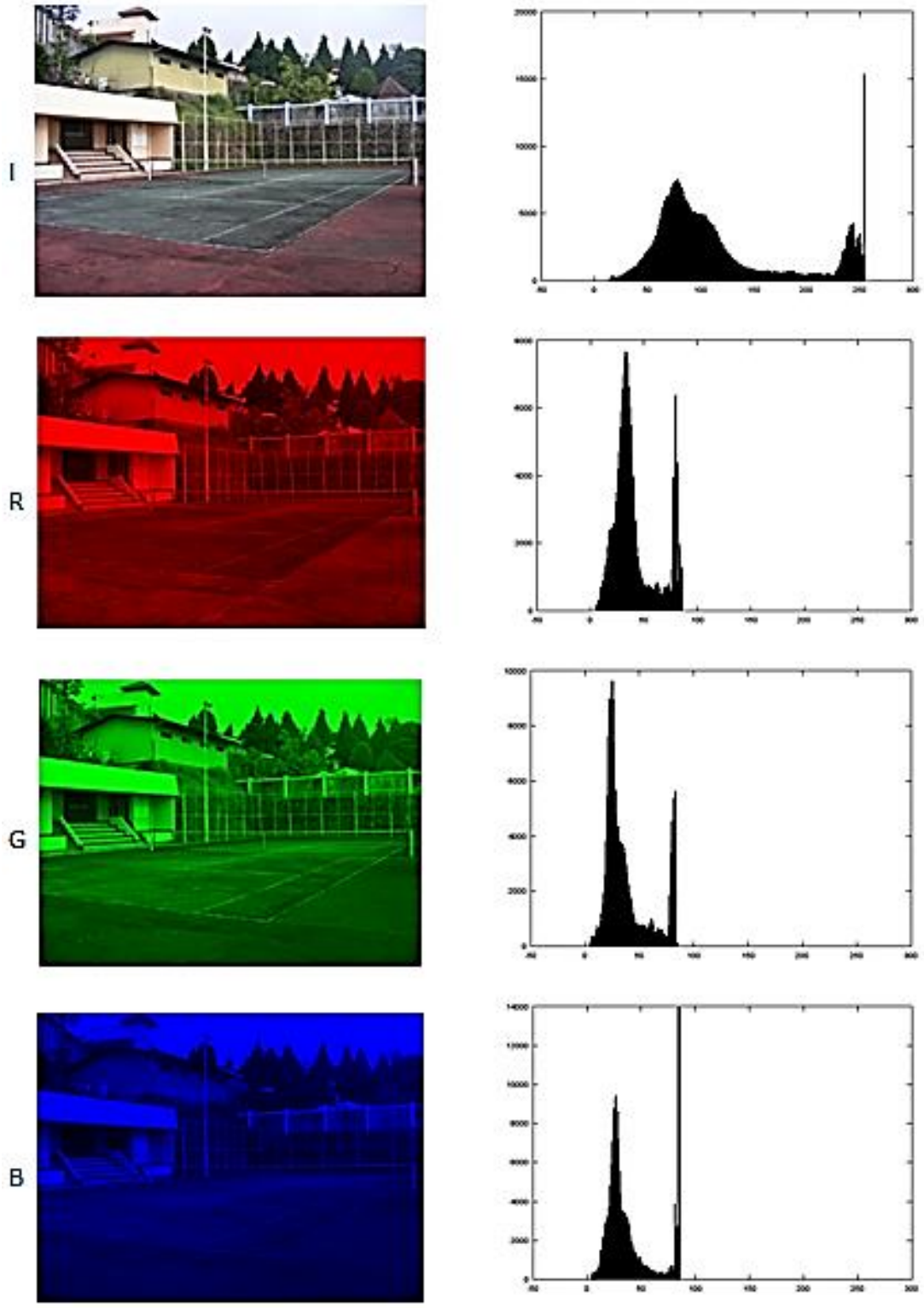
Catatan

- Sayangnya fungsi **imhist** pada *Octave* saat buku ini ditulis masih menyisakan *bug*.
- Pada pengolahan citra terkadang dijumpai istilah histogram ternormalisasi. Artinya, nilai histogram dibagi dengan jumlah piksel dalam citra, sehingga menjadi angka-angka pecahan bernilai kurang dari satu dan jumlah totalnya satu.

Pada pengolahan citra, histogram mempunyai peran yang cukup penting. Manfaat yang dapat didapatkan seperti berikut.

1. Berguna untuk mengamati penyebaran intensitas warna dan dapat dipakai untuk pengambilan keputusan misalnya dalam peningkatan kecerahan atau peregangan kontras serta sebaran warna.
2. Berguna untuk penentuan batas-batas dalam pemisahan objek dari latarbelakangnya.
3. Memberikan persentase komposisi warna dan tekstur intensitas untuk kepentingan identifikasi citra.

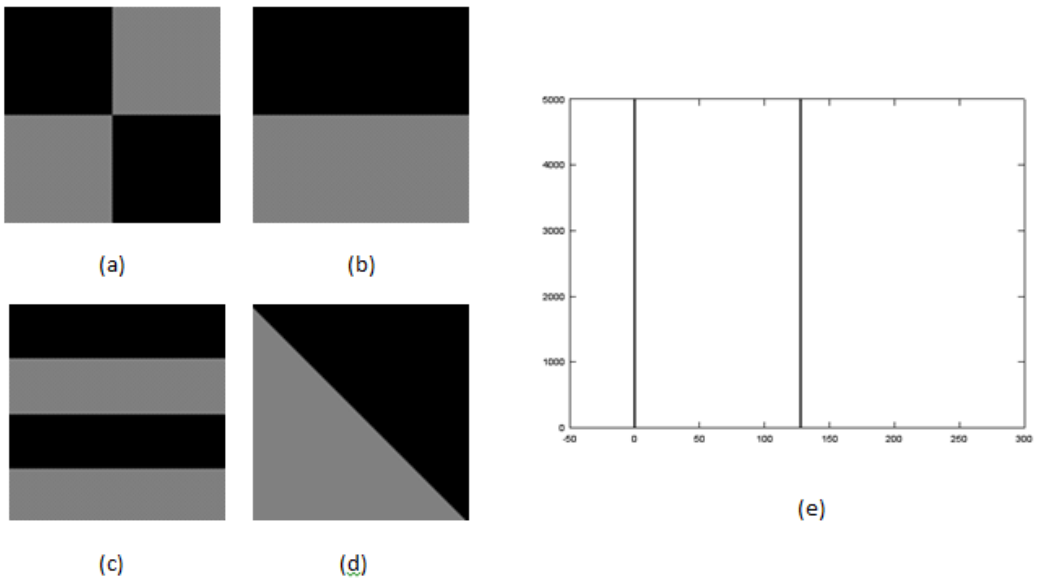
Khusus pada citra berwarna, histogram dapat diterapkan pada gabungan komponen-komponen RGB penyusunnya ataupun per komponen. Gambar 3.3 menunjukkan contoh mengenai hal itu. Pada gambar tersebut, I menyatakan histogram gabungan intensitas warna, R untuk komponen warna merah, G untuk komponen warna hijau, dan B untuk komponen warna biru.



Gambar 3.3 Histogram pada citra berwarna secara menyeluruh (I), merah (R), hijau (G), dan biru (B)

Histogram tidak mencerminkan susunan posisi warna piksel di dalam citra. Oleh karena itu, histogram tidak dapat dipakai untuk menebak bentuk objek yang

terkandung di dalam citra. Sebagai contoh, Gambar 3.4 memperlihatkan empat buah citra yang memiliki histogram yang sama, tetapi bentuk masing-masing jauh berbeda. Dengan demikian, histogram tidak memberikan petunjuk apapun tentang bentuk yang terkandung dalam keempat citra tersebut.



Gambar 3.4 Empat buah citra (a),(b),(c), dan (d) yang memiliki histogram yang sama (e), tetapi mempunyai informasi yang jauh berbeda

3.3 Meningkatkan Kecerahan

Operasi dasar yang sering dilakukan pada citra adalah peningkatan kecerahan (*brightness*). Operasi ini diperlukan dengan tujuan untuk membuat gambar menjadi lebih terang.

Secara matematis, peningkatan kecerahan dilakukan dengan cara menambahkan suatu konstanta terhadap nilai seluruh piksel. Misalkan, f(y, x) menyatakan nilai piksel pada citra berskala keabuan pada koordinat (y, x). Maka, citra baru

$$g(y,x) = f(y,x) + \beta \qquad (3.2)$$

telah meningkat nilai kecerahan semua pikselnya sebesar β terhadap citra asli f(y, x). Apabila β berupa bilangan negatif, kecerahan akan menurun atau menjadi lebih gelap.

Sebagai contoh, terdapat citra seperti pada Gambar 3.5(a). Citra tersebut dapat dicerahkan dengan memberikan perintah seperti berikut.

```
>> Img = imread('C:\Image\absam.png'); ↵
>> C = Img + 60; ↵
>> imshow(C); ↵
```

Hasilnya ditunjukkan pada Gambar 3.5(b).

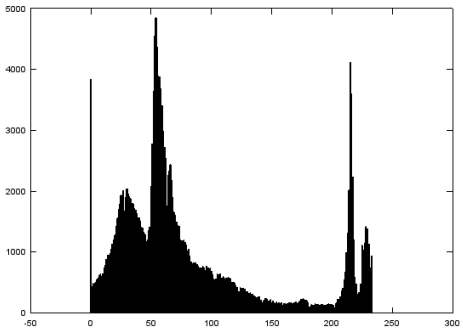
(a) Sebelum dicerahkan (b) Sesudah dicerahkan

Gambar 3.5 Efek pencerahan gambar

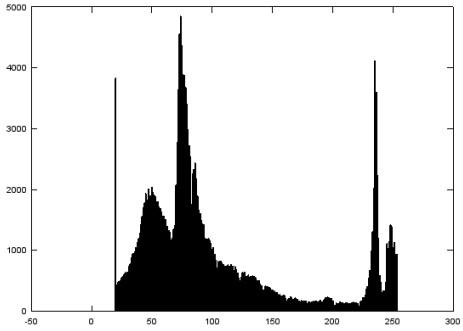
Jika dilihat melalui histogram, peningkatan kecerahan sebenarnya berefek pada penggeseran komposisi intensitas piksel ke kanan bila β berupa bilangan positif atau ke kiri jika β berupa bilangan negatif di Persamaan 3.2. Gambar 3.6 memperlihatkan keadaan ketika pencerahan dilakukan.

(a) Citra dengan kecerahan rendah

(b) Citra dengan kecerahan ditambah 20



(c) Histogram dari gambar (a)



(d) Histogram dari gambar (b)

Gambar 3.6 Histogram pada peningkatan citra. Komposisi jumlah intensitas per aras keabuan tidak berubah

Perhatikan, warna hitam (ditandai dengan garis tunggal yang menonjol di ujung kiri histogram) ikut tergeser. Jadi, warna hitam tidak lagi menjadi hitam kalau peningkatan kecerahan dilakukan dengan cara seperti di depan.

Bagaimana kalau ingin mencerahkan pada citra berwarna? Secara prinsip, hal itu sama saja dengan pada citra berskala keabuan. Tentu saja, dalam hal ini, penambahan konstanta dilakukan pada ketiga komponen penyusun warna. Contoh:

```
>> RGB = imread('c:\Image\bunga.png'); ↵
>> RGB2 = RGB + 80; ↵
```

Gambar 3.7 memperlihatkan perbedaan antara gambar pada keadaan awal dan setelah dicerahkan. Gambar 3.7(a) menyatakan citra pada `RGB` dan Gambar 3.7(b) menyatakan citra pada `RGB2`.

(a) Keadaan awal (b) Citra yang telah dicerahkan

Gambar 3.7 Peningkatan kecerahan pada citra berwarna

3.4 Meregangkan Kontras

Kontras dalam suatu citra menyatakan distribusi warna terang dan warna gelap. Suatu citra berskala keabuan dikatakan memiliki kontras rendah apabila distribusi warna cenderung pada jangkauan aras keabuan yang sempit. Sebaliknya, citra mempunyai kontras tinggi apabila jangkauan aras keabuan lebih terdistribusi secara melebar. Kontras dapat diukur berdasarkan perbedaan antara nilai intensitas tertinggi dan nilai intensitas terendah yang menyusun piksel-piksel dalam citra.

Perlu diketahui, citra dengan kontras rendah acapkali terjadi karena kondisi pencahayaan yang jelek ataupun tidak seragam. Hal itu dapat diakibatkan oleh sensor-sensor penangkap citra yang tidak linear (Jain, 1989).

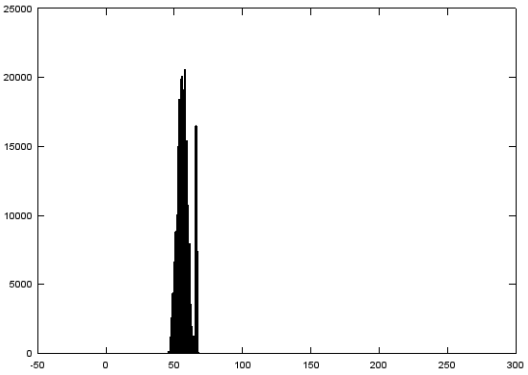
Agar distribusi intensitas piksel berubah perlu dilakukan peregangan kontras. Hal ini dilaksanakan dengan menggunakan rumus

$$g(y,x) = \alpha\, f(y,x) \qquad (3.3)$$

Berdasarkan rumus di atas, kontras akan naik kalau $\alpha > 1$ dan kontras akan turun kalau $\alpha < 1$.

Sebelum mempraktikkan peregangan kontras, perhatikan Gambar 3.8. Gambar tersebut sengaja dibuat ekstrem sempit agar memiliki kontras yang rendah. Hal ini dapat dilihat pada histogramnya.

(a) Citra gembala.png



(b) Histogram gambar (a)

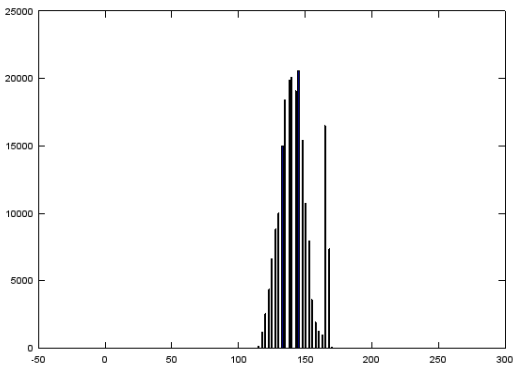
Gambar 3.8 Contoh citra dengan kontras rendah

Sekarang akan dicoba untuk meregangkan kontras dengan cara seperti berikut:

```
>> Img = imread('C:\Image\gembala.png'); ↵
>> K = 2.5 * Img; ↵
```

Gambar 3.9 memperlihatkan hasil peregangan kontras dan bentangan histogramnya.

(a) Citra hasil peregangan kontras



(b) Histogram gambar (a)

Gambar 3.9 Hasil peregangan kontras dengan α = 2,5

Kalau dilihat dari histogram pada Gambar 3.9(b), tampak bahwa distribusi intensitas warna menjadi melebar dan bergeser ke kanan terhadap keadaan terdahulu. Namun, karena distribusi cenderung ke aras keabuan yang tinggi, maka warna yang dihasilkan cenderung keputih-putihan.

3.5 Kombinasi Kecerahan dan Kontras

Operasi peningkatan kecerahan dan peregangan kontras dapat dilakukan sekaligus untuk kepentingan memperbaiki citra. Secara umum, gabungan kedua operasi tersebut dapat ditulis menjadi

$$g(y,x) = \alpha f(y,x) + \beta \quad (3.4)$$

Namun, kalau yang dikehendaki adalah melakukan pengaturan agar aras keabuan pada citra f yang berkisar antara f_1 dan f_2 menjadi citra g dengan aras antara g_1 dan g_2, rumus yang diperlukan adalah

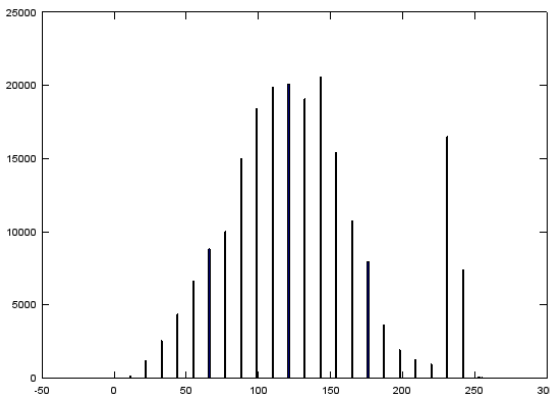
$$g(y,x) = g_1 + \left(\frac{g_2 - g_1}{f_2 - f_1}\right)[f(y,x) - f_2] \quad (3.5)$$

Mengacu histrogram pada Gambar 3.9(b), rumus di atas dapat diterapkan. Pertama, distribusi histogram perlu digeser ke kiri. Selanjutnya, baru dikenakan peregangan kontras. Implementasinya seperti berikut.

```
>> Img = imread('C:\Image\gembala.png'); ↵
>> C = Img - 45; ↵
>> K = C * 11; ↵
```

Dengan cara seperti itu, akan dihasilkan citra yang lebih tegas, sebagaimana diperlihatkan pada Gambar 3.10.

(a) Citra hasil pengaturan kecerahan dan peregangan kontras



(b) Histogram gambar (a)

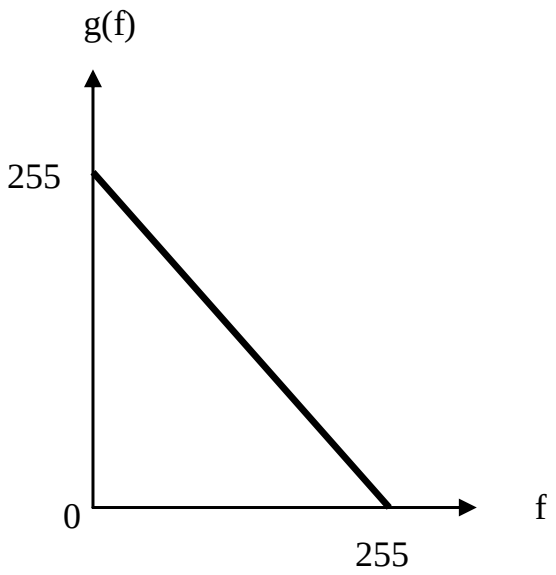
Gambar 3.10 Hasil pengaturan kecerahan dan peregangan kontras menggunakan Persamaan 3.5

3.6 Membalik Citra

Bila pernah melihat film hasil kamera analog, gambar yang terekam dalam film tersebut berkebalikan dengan foto saat dicetak, yang dikenal sebagai film negatif. Citra seperti ini biasa digunakan pada rekam medis; misalnya hasil fotografi rontgen. Hubungan antara citra dan negatifnya untuk yang beraras keabuan dapat dinyatakan dengan rumus:

$$g(y,x) = 255 - f(y,x) \qquad (3.6)$$

Hubungan di atas dapat digambarkan seperti secara grafis pada Gambar 3.11.



Gambar 3.11 *Pembalikan citra*

Gambar 3.11 menunjukkan bahwa kalau f(y, x) bernilai 255, g(y, x) bernilai 0. Sebaliknya, kalau f(y, x) bernilai 0, g(y, x) bernilai 255. Jika bit yang digunakan bukan 8 tetapi 4, persamaan untuk membalik citra berubah menjadi

$$g(y,x) = 15 - f(y,x) \qquad (3.7)$$

Untuk mempraktikkan Persamaan 3.7, perintah berikut dapat dicoba:

```
>> Img = imread('C:\Image\lena256.png'); ↵
>> R = 255 - Img; ↵
```

Dengan memberikan

```
R = 255 - Img;
```

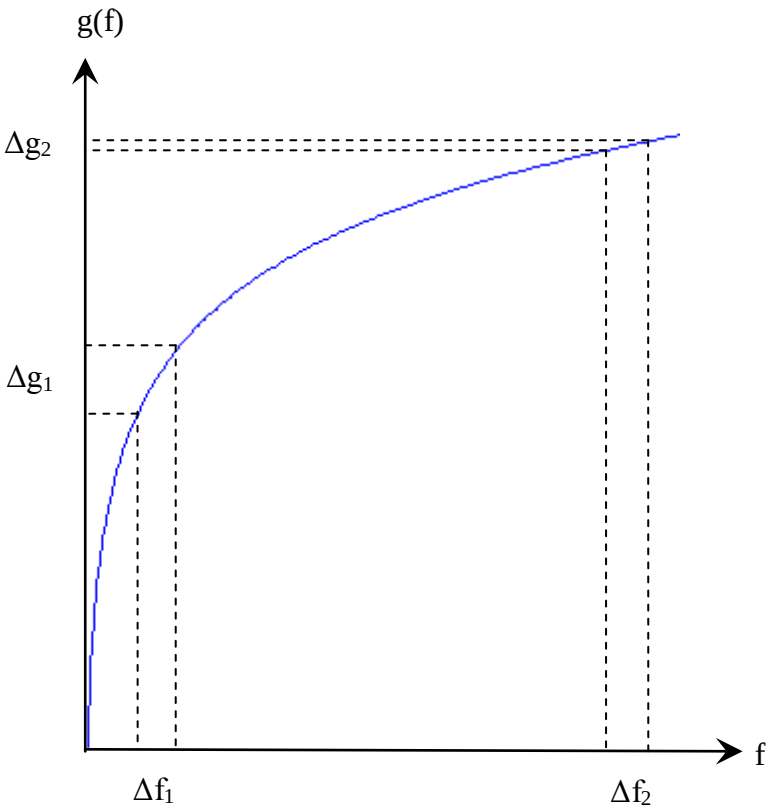
maka `R` berisi kebalikan dari citra di `Img`. Citra asli dan citra negatif yang dihasilkan diperlihatkan pada Gambar 3.12.

(a) Citra asli (b) Citra hasil pembalikan

Gambar 3.12 Pembalikan citra

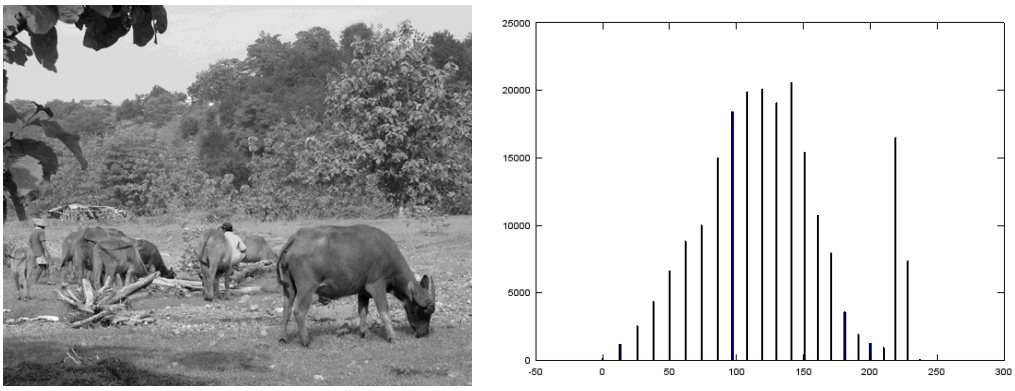
3.7 Pemetaan Nonlinear

Dalam pengolahan citra, terkadang diperlukan pemetaan intensitas piksel yang tidak menggunakan cara linear seperti yang telah dibahas, melainkan menggunakan pendekatan nonlinear. Kalau suatu citra berisi bagian yang cerah dan bagian yang gelap yang cukup ekstrem, akan lebih baik kalau digunakan cara nonlinear. Sebagai contoh, dapat digunakan fungsi logaritma, yang membuat bagian yang gelap (intensitas rendah) lebih dicerahkan daripada yang berintensitas tinggi, karena memuat banyak detail yang penting. Gambar 3.13 memperlihatkan keadaan tersebut.



Gambar 3.13 Pemetaan dengan fungsi logaritma

Gambar 3.14 menunjukkan bahwa dengan menggunakan selang Δf yang sama pada f, ternyata memberikan selang yang berbeda pada g. Dengan kata lain, terjadi pengaturan atau variasi intensitas berbeda pada intensitas rendah dan intensitas tinggi. Peningkatan yang tajam dilakukan pada area yang gelap (yang nilai intensitasnya rendah). Sifat pemetaan yang tidak seragam itulah yang dikatakan sebagai pemetaan nonlinear. Gambar 3.14 memperlihatkan efek pemetaan nonlinear berdasarkan citra gembala.png (Gambar 3.8 (a)).



(a) Citra pemetaan dengan logaritma (b) Histogram gambar (a)

Gambar 3.14 Contoh hasil penggunaan pemetaan nonlinear

Kode yang digunakan untuk melakukan pemetaan di atas seperti berikut:

```
>> Img = imread('C:\Image\gembala.png'); ↵
>> C = log(1+double(Img)); ↵
>> C2 = im2uint8(mat2gray(C)); ↵
```

Penambahan angka 1 pada fungsi **log** dimaksudkan untuk menghindari kegagalan dalam menghitung logaritma alami untuk bilangan nol. Karena fungsi **log** bekerja pada area bilangan real maka penggunaan **double(Img)** diperlukan. Selanjutnya, mengingat hasil pada `C` berupa bilangan real, diperlukan konversi balik ke tipe **uint8** (8 bit). Hal ini dikerjakan melalui

```
C2 = im2uint8(mat2gray(C));
```

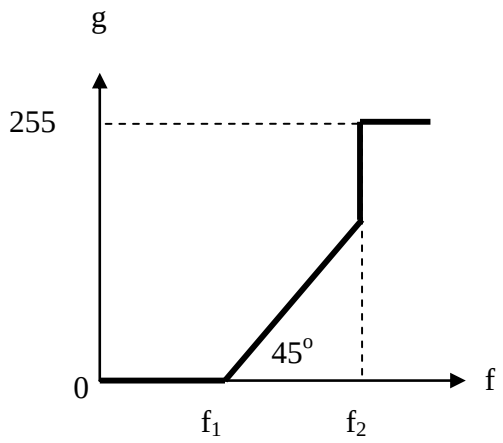
Pertama-tama, **mat2gray** dipanggil agar semua nilai pada larik `C` berada di dalam jangkauan [0, 1]. Lalu, agar nilai berada pada jangkauan [0, 255], **im2uint8** dipanggil.

3.8 Pemotongan Aras Keabuan

Efek pemotongan (*clipping*) diperoleh bila dilakukan operasi seperti berikut:

$$g(y,x) = \begin{cases} 0, & x \le f_1 \\ f(y,x), & f_1 < f(y,x) < f_2 \\ 255, & x \ge f_2 \end{cases} \quad (3.8)$$

Nilai g dinolkan atau dipotong habis untuk intensitas asli dari 0 hingga f_1 karena dipandang tidak mengandung informasi atau objek menarik. Demikian pula untuk nilai intensitas dari f_2 ke atas, yang mungkin hanya mengadung derau. Gambar 3.15 menyajikan diagram penggunaan rumus tersebut.



Gambar 3.15 Contoh pemotongan aras keabuan dengan pola sangat tidak linear atau patah-patah

Untuk mempraktikkan rumus dalam Persamaan 3.8, kode berikut dapat digunakan.

Program : potong.m

```
function [Hasil] = potong(berkas, f1, f2)
% POTONG Menghasilkan citra dengan level 0 s/d f1
%      serta f2-255 dinolkan

Img = imread(berkas);
[tinggi, lebar] = size(Img);

Hasil = Img;
```

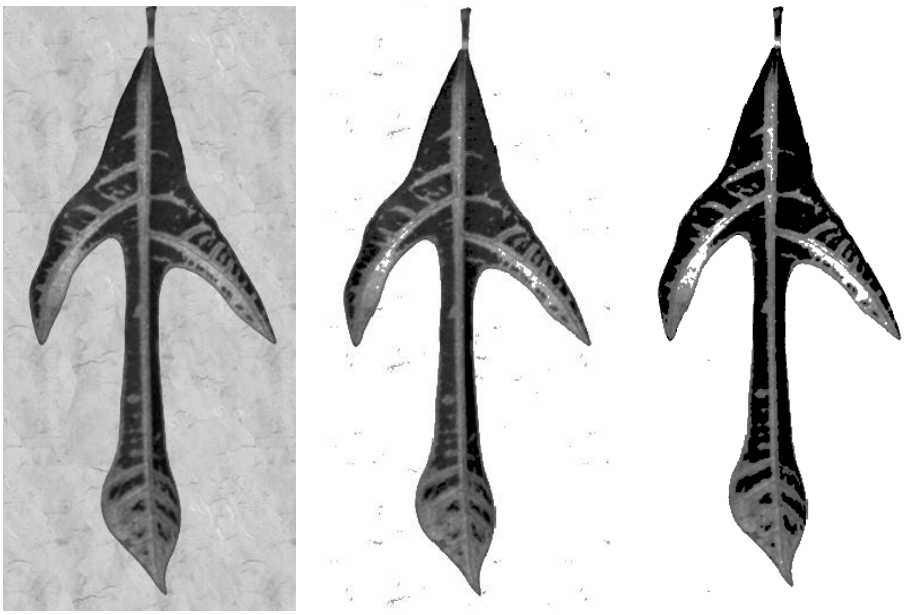
```
for baris=1 : tinggi
    for kolom=1 : lebar
        if Hasil(baris, kolom) <= f1
            Hasil(baris, kolom) = 0;
        end

        if Hasil(baris, kolom) >= f2
            Hasil(baris, kolom) = 255;
        end
    end
end
```

Akhir Program

Skrip di atas dapat dipanggil dengan menyertakan nama *file* berisi citra berskala keabuan, batas rendah dan batas tinggi untuk kepentingan pemotongan pada citra. Sebagai contoh, pemanggilan seperti berikut dapat diberikan:

```
>> H = potong('C:\Image\daun.png', 30, 170); ↵
>> imshow(H); ↵
```

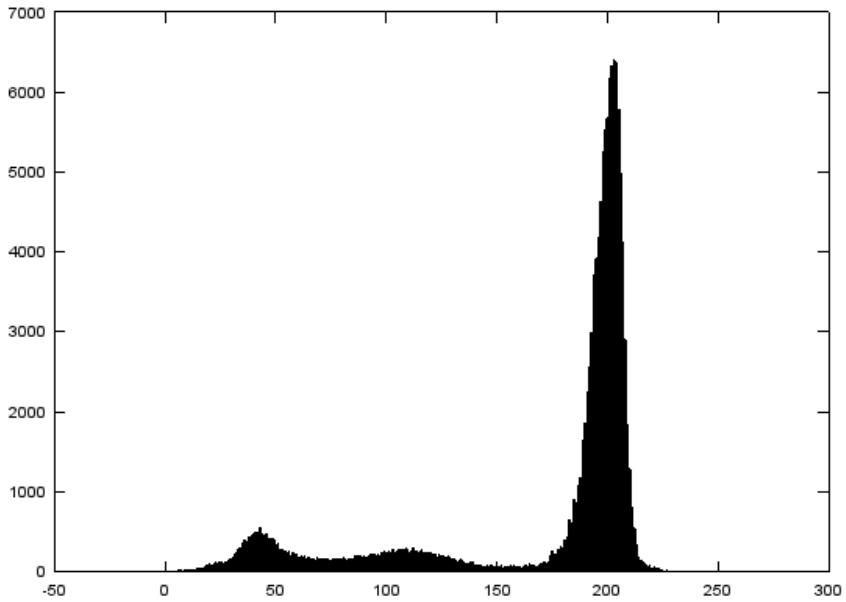
Dengan cara seperti itu, hasil pemrosesan ditampilkan. Gambar 3.16 menunjukkan contoh daun.png dalam keadaan asli dan hasil pemotongan pada dua tingkat ambang. Pada Gambar 3.16(b), sedikit noktah warna latar belakang masih muncul.



(a) Citra asli daun.tif (b) f_1=30, f_2=170 (c) f_1=50, f_2=150

Gambar 3.16 Efek pemotongan aras keabuan

Untuk melakukan percobaan dalam menentukan f_1 dan f_2, kekhasan histogram citra perlu dipertimbangkan. Gambar 3.17 memperlihatkan histogram daun.png.



Gambar 3.17 Histogram daun.png

Nilai intensitas yang berposisi sebagai lembah dalam histogram pada Gambar 3.18 (sekitar 40 untuk f_1 dan 160 untuk f_2) berpotensi menjadi nilai ambang,

3.9 Ekualisasi Histogram

Ekualisasi histogram merupakan suatu cara yang bertujuan untuk memperoleh histogram yang intensitasnya terdistribusi secara seragam pada citra. Namun, dalam praktik, hasilnya tidak benar-benar seragam (Jain, 1989). Pendekatan yang dilakukan adalah untuk mendapatkan aras keabuan yang lebih luas pada daerah yang memiliki banyak piksel dan mempersempit aras keabuan pada daerah yang berpiksel sedikit. Efeknya dapat digunakan untuk meningkatkan kontras secara menyeluruh. Perlu diketahui, ekualisasi histogram termasuk sebagai pemetaan nonlinear.

Misalnya, histogram untuk setiap aras keabuan dinyatakan dengan

```
hist[i+1]
```

Dalam hal ini, i bernilai 0, 1, 2, .., L-1, dengan L menyatakan jumlah aras keabuan. Akumulasi histogram untuk piksel yang memiliki aras k dinyatakan dengan

$$c[k+1] = \sum_{i=1}^{k} hist[k+1], k = 0,1,2, \dots, L-1 \quad (3.8)$$

Selanjutnya, aras k akan diganti dengan a dengan ketentuan sebagai berikut:

$$a_k = round((L-1)\ \frac{c[k+1]}{N}), k = 0,1,2, \dots, L-1 \quad (3.9)$$

Dalam hal ini, N menyatakan jumlah piksel pada citra.

Untuk memahami proses dalam ekualisasi histogram, lihatlah contoh pada Tabel 3.1.

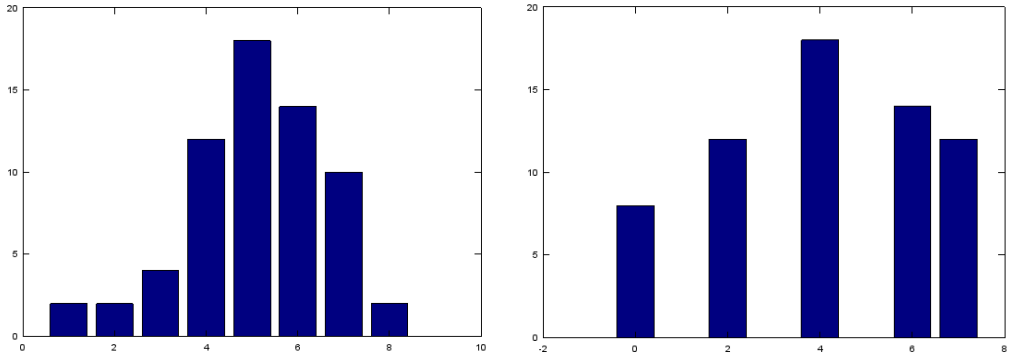
Tabel 3.1 Proses ekualisasi histogram

I	Aras	hist[i]	c[i]	a(i)
1	0	2	2	0
2	1	2	4	0
3	2	4	8	0
4	3	12	20	2
5	4	18	38	4
6	5	14	52	6
7	6	10	62	7
8	7	2	64	7
	L=8	N=64		

Pada contoh di atas, yang diarsir dengan warna hijau muda menyatakan keadaan awal citra. Dalam hal ini, citra mengandung N=64 piksel (8x8) dengan jumlah aras keabuan berupa 8. Selanjutnya, berdasarkan nilai `hist[i]` maka `c[i]` dihitung. Selanjutnya, `a[i]` dapat dihitung berdasar Persamaan 3.9. Dalam hal ini, setiap nilai

- 0 atau 1 pada citra akan diganti dengan 0;
- 3 akan diganti dengan 2;
- 4 tidak diganti (tetap);
- 5 diganti dengan 6;
- 6 dan 7 diganti dengan 7.

Gambar 3.18 memperlihatkan keadaan sebelum dan sesudah ekualisasi histogram. Tampak bahwa di sekitar batang histogram yang paling tinggi terjadi perenggangan dan perbedaan dengan yang lebih rendah mengecil.



(a) Histogram awal (b) Histogram setelah ekualisasi

Gambar 3.18 Efek ekualisasi histogram

Algoritma untuk melakukan penggantian nilai intensitas pada citra ditunjukkan berikut ini.

ALGORITMA 3.2 – Melaksanakan ekualisasi histogram citra aras keabuan

Masukan:
- f(M, N) : citra berukuran M baris dan N kolom
- n : jumlah piksel dalam citra

Keluaran
- g(M, N) : citra yang telah mengalami ekualisasi histogram

1. Hitung faktor penyekalaan: $\alpha \leftarrow 255 / n$
2. Hitung histogram citra menggunakan Algoritma 3.1 dengan hasil berupa hist
3. c[1] ← α * hist[1]
4. FOR i←1 TO L-1
5. c[i+1] ← c[i] + round(α * hist[i+1])
6. END-FOR
7. FOR i ← 1 TO M
8. FOR j ← 1 TO N
9. g(y,x) ← c[f(y, x)]
10. END-FOR
11. END-FOR

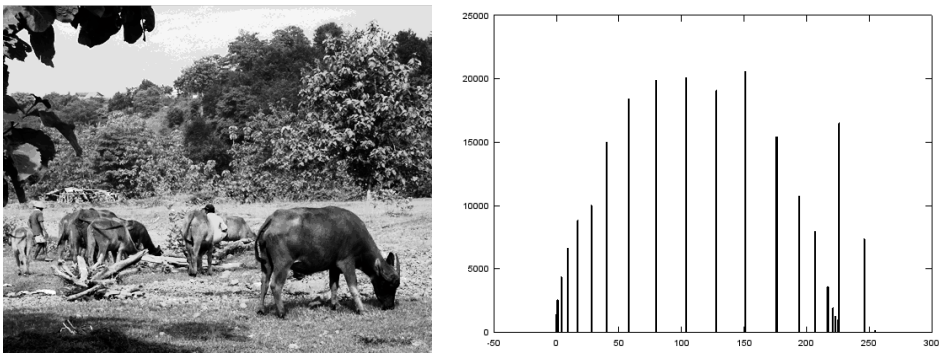
Berikut adalah contoh skrip yang digunakan untuk melakukan ekualisasi terhadap gambar gembala.png.

Program : ekualisasi.m

```
% EKUALISASI Contoh untuk melakukan ekualisasi histogram

Img = imread('c:\Image\gembala.png');
[jum_baris, jum_kolom] = size(Img);

L = 256;
Histog = zeros(L, 1);
for baris=1 : jum_baris
    for kolom=1 : jum_kolom
        Histog(Img(baris, kolom)+1) = ...
            Histog(Img(baris, kolom)+1) + 1;
    end
end

alpha = (L-1) / (jum_baris * jum_kolom);

C(1) = alpha * Histog(1);
for i=1 : L-2
    C(i+1) = C(i) + round(alpha * Histog(i+1));
end

for baris=1 : jum_baris
    for kolom=1 : jum_kolom
        Hasil(baris, kolom) = C(Img(baris, kolom));
    end
end

Hasil = uint8(Hasil);
imshow(Hasil);
```

Akhir Program

Gambar 3.19 menunjukkan contoh hasil citra berdasarkan pemrosesan di atas, yang memperlihatkan dengan jelas posisi peregangan dan pemadatan garis-garis histogram.



(a) Hasil ekualisasi histogram (b) Histogram gambar a

Gambar 3.19 Hasil ekualisasi histogram dan histogramnya

Perlu diketahui, pernyataan

```
Hasil = uint8(Hasil);
```

Pada ekualisasi.m diperlukan untuk membuat hasil bertipe **uint8** mengingat

```
Hasil(baris, kolom) = C(Img(baris, kolom));
```

memberikan Hasil bertipe **double**. Hal itu disebabkan C memang bertipe **double**.

Latihan

1. Jelaskan yang dimaksud operasi piksel.
2. Jelaskan pengertian histogram citra.
3. Terdapat citra 4x4 dengan rincian nilai kecerahan piksel seperti berikut.

7	7	7	7
6	5	5	6
5	5	5	5
2	4	0	1

Aras keabuan sebanyak 8 buah. Bagaimana histogram citra tersebut dalam bentuk angka?

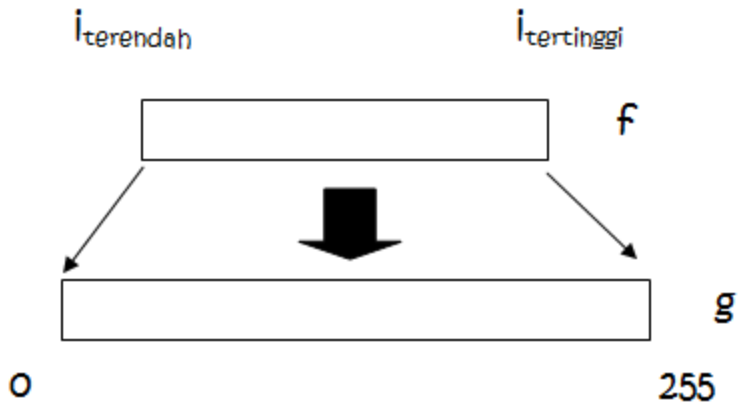
4. Apa bedanya histogram citra asli dengan histogram citra yang ternormalisasi? Terapkan histogram ternormalisasi untuk citra pada soal 3.
5. Apakah histogram dapat dipakai untuk mengenali objek yang terkandung dalam citra secara langsung? Jelaskan!
6. Jelaskan yang dimaksud dengan citra dengan kontras yang rendah. Apakah efeknya?
7. Jelaskan bahwa peningkatan kecerahan melalui rumus

   $$g(y,x) = f(y,x) + \beta$$

   tidak memberikan efek peregangan kontras.
8. Jelaskan bahwa dengan menggunakan peningkatan kecerahan saja, warna hitam justru menjadi tidak tegas lagi.
9. Suatu citra berskala keabuan memiliki nilai aras keabuan terendah berupa a dan nilai aras keabuan tertinggi sebesar b. Bagaimana cara meregangkan kontras pada citra agar aras keabuan terendah berupa 0 dan nilai aras keabuan tertinggi berupa 255?
10. Terangkan proses untuk membentuk film negatif berdasarkan suatu citra.
11. Apa sebenarnya tujuan penggunaan fungsi logaritmik alami dalam operasi piksel?
12. Berdasarkan Tabel 3.1, berapa jumlah piksel untuk setiap aras keabuan (0 sampai dengan 7) setelah ekualisasi histogram dilakukan.
13. Buatlah suatu program yang dapat digunakan untuk meregangkan kontras dari suatu citra sehingga nilai aras keabuan terendah pada aras 0 dan nilai aras keabuan tertinggi dalam histogramnya akan diletakkan pada aras 255. Rumus yang dapat dipakai:

    $$g(y,x) = (f(y,x) - i_{terendah}) x \frac{255}{i_{tertinggi} - i_{terendah}}$$

    dengan i menyatakan intensitas. Gambar berikut menunjukkan skema konversinya.


$i_{terendah}$
$i_{tertinggi}$
f
g
0
255