

Histogram Citra

Informasi penting mengenai isi citra digital dapat diketahui dengan membuat histogram citra. Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas *pixel* dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (*relative*) dari intensitas pada citra tersebut. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (*brightness*) dan kontras (*contrast*) dari sebuah gambar. Karena itu, histogram adalah alat bantu yang berharga dalam pekerjaan pengolahan citra baik secara kualitatif maupun kuantitatif.

6.1 Membuat Histogram

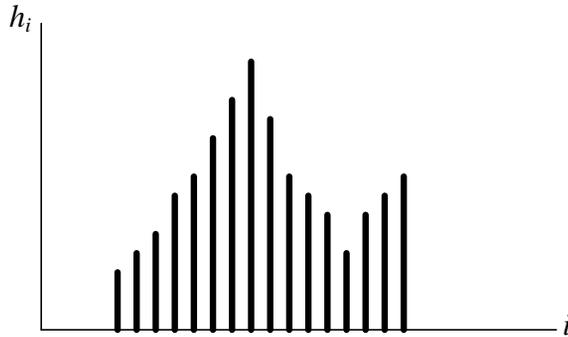
Misalkan citra digital memiliki L derajat keabuan, yaitu dari nilai 0 sampai $L - 1$ (misalnya pada citra dengan kuantisasi derajat keabuan 8-bit, nilai derajat keabuan dari 0 sampai 255). Secara matematis histogram citra dihitung dengan rumus

$$h_i = \frac{n_i}{n}, \quad i = 0, 1, \dots, L - 1 \quad (6.1)$$

yang dalam hal ini,

n_i = jumlah *pixel* yang memiliki derajat keabuan i
 n = jumlah seluruh *pixel* di dalam citra

Plot h_i versus f_i dinamakan **histogram**. Gambar 6.1 adalah contoh sebuah histogram citra. Secara grafis histogram ditampilkan dengan diagram batang. Perhatikan dari persamaan 6.1 bahwa nilai n_i telah dinormalkan dengan membaginya dengan n . Nilai h_i berada di dalam selang 0 sampai 1.



Gambar 6.1. Histogram citra

Sebagai contoh, misalkan matriks di bawah ini menyatakan citra digital yang berukuran 8×8 *pixel* dengan derajat keabuan dari 0 sampai 15 (ada 16 buah derajat keabuan):

$$\begin{bmatrix} 3 & 7 & 7 & 8 & 10 & 12 & 14 & 10 \\ 2 & 0 & 0 & 0 & 1 & 8 & 15 & 15 \\ 14 & 6 & 5 & 9 & 8 & 10 & 9 & 12 \\ 12 & 12 & 11 & 8 & 8 & 10 & 11 & 1 \\ 0 & 2 & 3 & 4 & 5 & 13 & 10 & 14 \\ 4 & 5 & 0 & 0 & 1 & 0 & 2 & 2 \\ 15 & 13 & 11 & 10 & 9 & 9 & 8 & 7 \\ 2 & 1 & 0 & 10 & 11 & 14 & 13 & 12 \end{bmatrix}$$

Tabulasi perhitungan histogramnya ditunjukkan pada Tabel 6.1. Mudah dilihat bahwa semakin besar nilai n_i maka semakin besar pula nilai h_i .

Tabel 6.1 Perhitungan histogram

i	n_i	$h_i = n_i/n$ ($n = 64$)
0	8	0.125
1	4	0.0625
2	5	0.078125

3	2	0.03125
4	2	0.03125
5	3	0.046875
6	1	0.015625
7	3	0.046875
8	6	0.09375
9	3	0.046875
10	7	0.109375
11	4	0.0625
12	5	0.078125
13	3	0.046875
14	4	0.0625
15	3	0.046875

6.2 Algoritma Perhitungan Histogram

Algoritma perhitungan histogram ditunjukkan pada Algoritma 6.1 [HEN95]. Citra masukan mempunyai 256 derajat keabuan yang nilai-nilainya dari 0 sampai 255. Intensitas *pixel* disimpan di dalam `Image[0..N-1][0..M-1]`, sedangkan histogram disimpan di dalam tabel `Hist[0..255]`.

Gambar 6.2 memperlihatkan histogram citra kapal (512×512). Beberapa program komersil seperti *Adobe Photoshop*, *Paintshop*, dan *PolyView*, dapat digunakan untuk membangkitkan histogram citra.

```

void histogram(citra Image, int N, int M, float Hist[256])
/* Menghitung histogram untuk citra Image yang berukuran N ´ M.
   Histogram disimpan di dalam tabel Hist yang bertipe riil(float).
*/
{
    int i, j, n;

    /* inisialisasi Hist[0..255] dengan 0 */
    for(i=0;i<=255;i++)
        Hist[i]=0;

    for(i=0;i<=N-1;i++)
        for(j=0;j<=M-1;j++)
            Hist[Image[i][j]]=Hist[Image[i][j]]+1;

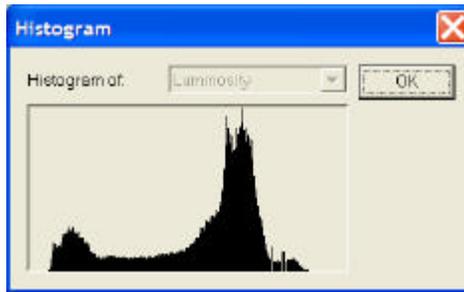
    /*normalisasi Hist[i] dengan jumlah seluruh pixel*/
    n=N*M;
    for(i=0;i<=255;i++)
        Hist[i]=Hist[i]/(float)n;
}

```

Algoritma 6.2. Perhitungan histogram citra



(a) kapal 512×512 , 8-bit



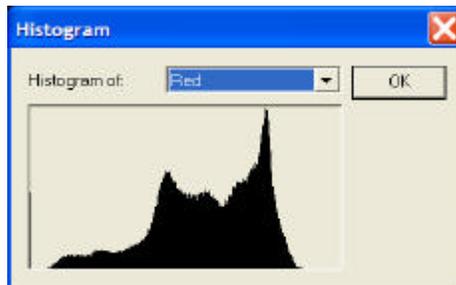
(b) Histogram citra kapal (by PolyView)

Gambar 6.2. Citra Kapal (512×512) dan histogramnya

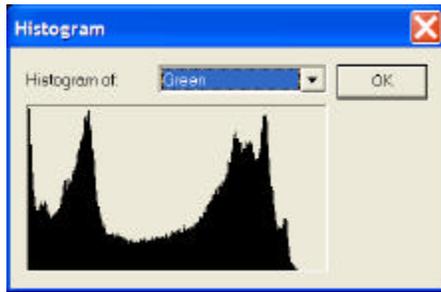
Khusus untuk citra berwarna, histogramnya dibuat untuk setiap kanal *RGB* (merah, hijau, dan biru). Misalnya citra berwarna *pepper* 512×512 *pixel* 8-bit, pada Gambar 6.3, histogramnya ada tiga buah, masing-masing untuk komponen merah, hijau, dan biru. Histogram tersebut dihasilkan dari program [WIC01].



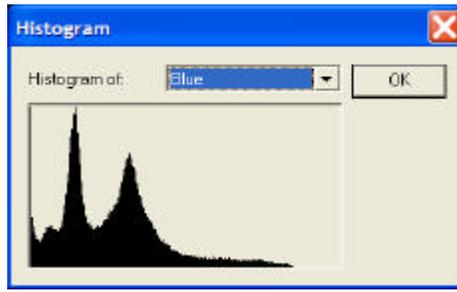
(a) *pepper* (color), 512×512 , 24-bit



(b) Histogram untuk kanal merah



(c) Histogram untuk kanal hijau



(d) Histogram untuk kanal biru

Gambar 6.3. Citra berwarna pepper dan histogram masing-masing kanal warnanya

Histogram citra banyak memberikan informasi penting sebagai berikut:

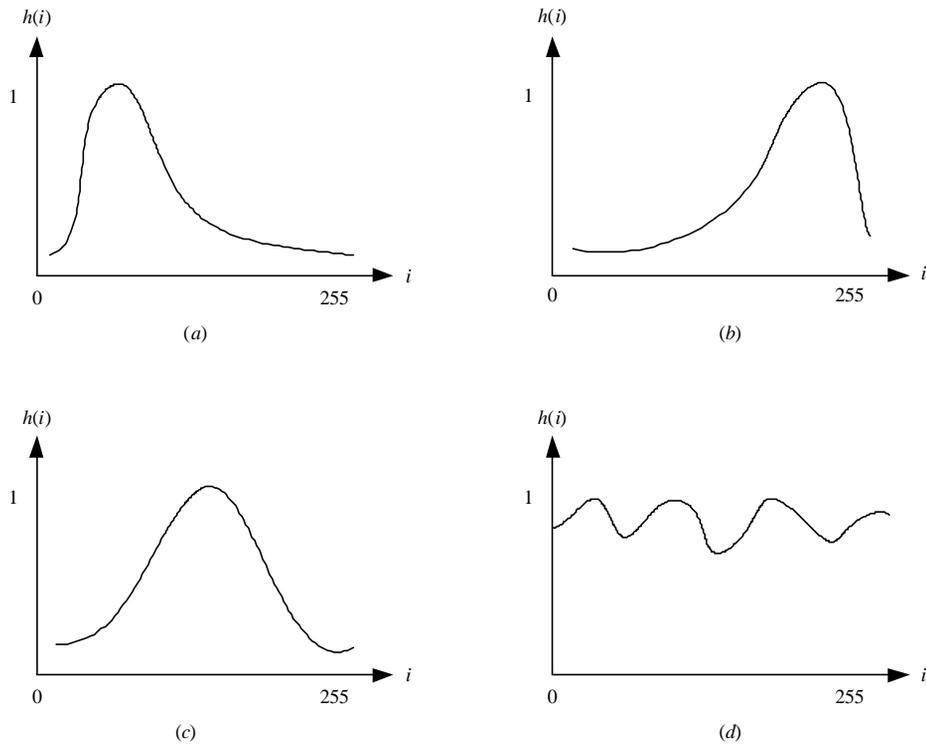
1. Nilai h_i menyatakan peluang (*probability*) *pixel*, $P(i)$, dengan derajat keabuan i . Jumlah seluruh nilai h_i sama dengan 1, atau

$$\sum_{i=0}^{L-1} h_i = 1$$

Peluang suatu *pixel* memiliki derajat keabuan lebih kecil atau sama dengan derajat keabuan tertentu adalah jumlah h_i untuk $0 \leq i \leq j$, atau

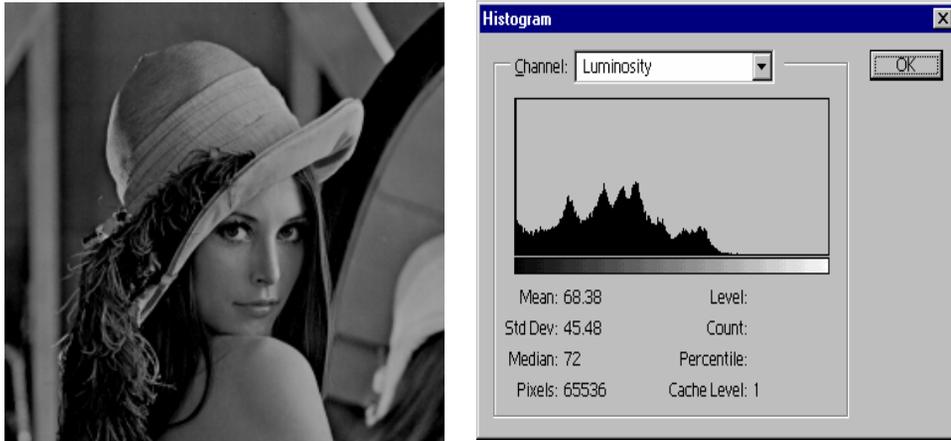
$$P(i \leq j) = \sum_{i=0}^j h_i, \quad 0 \leq j \leq L-1$$

2. Puncak histogram menunjukkan intensitas *pixel* yang menonjol. Lebar dari puncak menunjukkan rentang kontras dari gambar. Citra yang mempunyai kontras terlalu terang (*overexposed*) atau terlalu gelap (*underexposed*) memiliki histogram yang sempit. Histogramnya terlihat hanya menggunakan setengah dari daerah derajat keabuan. Citra yang baik memiliki histogram yang mengisi daerah derajat keabuan secara penuh dengan distribusi yang merata pada setiap nilai intensitas *pixel* (Gambar 6.4).

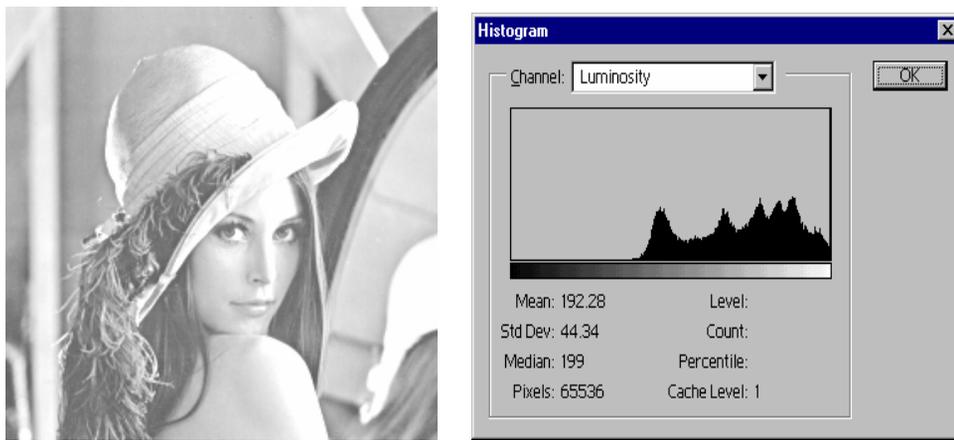


Gambar 6.4. (a) citra gelap, (b) citra terang, (c) citra normal (normal brightness), (d) normal brightness dan high contrast

Gambar 6.5 memperlihatkan tiga buah citra Lena. Citra Lena yang pertama terlalu gelap. Histogramnya banyak menumpuk pada bagian kiri karena citra tersebut mengandung banyak nilai intensitas yang dekat dengan 0 (hitam). Citra Lena yang kedua terlalu terang. Histogramnya banyak menumpuk pada bagian kanan karena citra tersebut mengandung banyak nilai intensitas yang dekat dengan 255 (putih). Citra Lena yang ketiga adalah citra yang normal (bagus). Histogramnya tersebar merata di seluruh daerah derajat keabuan. Tiga buah histogram tersebut dihasilkan dengan program *Adobe Photoshop*.

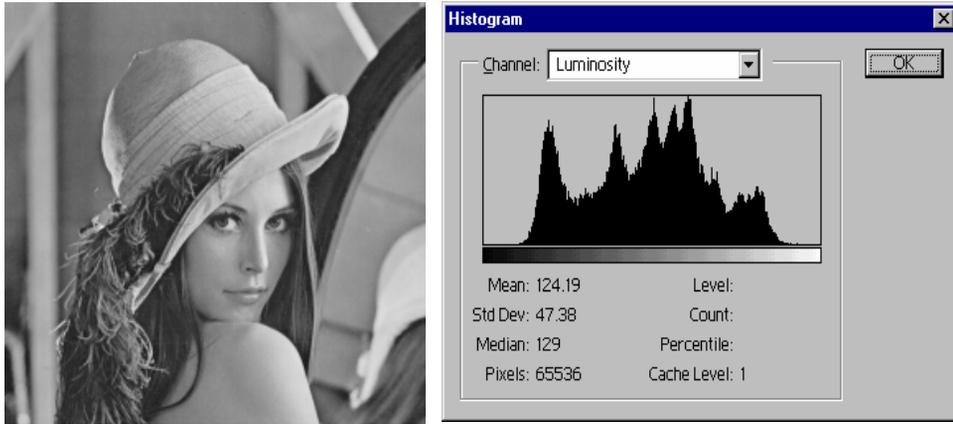


(a) Kiri: citra Lena yang terlalu gelap; kanan: histogramnya (by *Photoshop*)



(b) Kiri: citra Lena yang terlalu terang; kanan: histogramnya

Gambar 6.5. *Beragam-histogram dari beberapa kasus citra Lena*



(c) Kiri: citra Lena yang bagus (normal); kanan: histogramnya

Gambar 6.5 (lanjutan).