

Perbaikan Kualitas Citra

Perbaikan kualitas citra (*image enhancement*) merupakan salah satu proses awal dalam pengolahan citra (*image preprocessing*). Perbaikan kualitas diperlukan karena seringkali citra yang dijadikan objek pembahasan mempunyai kualitas yang buruk, misalnya citra mengalami derau (*noise*) pada saat pengiriman melalui saluran transmisi, citra terlalu terang/gelap, citra kurang tajam, kabur, dan sebagainya. Melalui operasi pemrosesan awal inilah kualitas citra diperbaiki sehingga citra dapat digunakan untuk aplikasi lebih lanjut, misalnya untuk aplikasi pengenalan (recognition) objek di dalam citra

7.1 Lingkup Proses Perbaikan Kualitas Citra

Yang dimaksud dengan perbaikan kualitas citra adalah proses mendapatkan citra yang lebih mudah diinterpretasikan oleh mata manusia. Pada proses ini, ciri-ciri tertentu yang terdapat di dalam citra lebih diperjelas kemunculannya [DUL97]. Secara matematis, *image enhancement* dapat diartikan sebagai proses mengubah citra $f(x, y)$ menjadi $f'(x, y)$ sehingga ciri-ciri yang dilihat pada $f(x, y)$ lebih ditonjolkan.

Proses-proses yang termasuk ke dalam perbaikan kualitas citra [DUL97]:

1. Perubahan kecerahan gambar (*image brightness*)
2. Peregangan kontras (*contrast stretching*)
3. Perubahan histogram citra.
4. Pelembutan citra (*image smoothing*)
5. Penajaman (*sharpening*) tepi (*edge*).
6. Pewarnaan semu (*pseudocolouring*)
7. Perubahan geometrik

Beberapa operasi *image enhancement* (4 dan 5) dapat dipandang sebagai operasi penapisan untuk memperoleh citra yang lebih baik. Operasi penapisan adalah operasi konvolusi citra $f(x, y)$ dengan penapis $h(x, y)$:

$$f'(x, y) = h(x, y) * f(x, y) \quad (7.1)$$

atau dalam ranah frekuensi:

$$F'(u, v) = H(u, v)F(u, v) \quad (7.2)$$

Pada umumnya, $f(x, y)$ sudah diketahui sehingga persoalannya adalah memilih $h(x, y)$ sedemikian rupa sehingga $f'(x, y)$ merupakan citra yang menonjolkan ciri tertentu dari $f(x, y)$.

7.2 Pengubahan Kecerahan Gambar (*Image Brightness*)

Untuk membuat citra lebih terang atau lebih gelap, kita melakukan pengubahan kecerahan gambar. Kecerahan/kecemerlangan gambar dapat diperbaiki dengan menambahkan (atau mengurangi) sebuah konstanta kepada (atau dari) setiap *pixel* di dalam citra. Akibat dari operasi ini, histogram citra mengalami pergeseran.

Secara matematis operasi ini ditulis sebagai

$$f(x, y)' = f(x, y) + b \quad (7.3)$$

Jika b positif, kecerahan gambar bertambah, sebaliknya jika b negatif kecerahan gambar berkurang.

Algoritma pengubahan kecerahan gambar ditunjukkan pada Algoritma 7.1. 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 hasil pengubahan tetap disimpan di dalam citra `Image`.

```
void ImageBrightness(citra Image, int N, int M, int b)
/* Mengubah kecerahan citar Image yang berukuran N ´ M dengan penambahan
intensitas setiap pixel sebesar b.
*/
{
    int i, j, n;

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

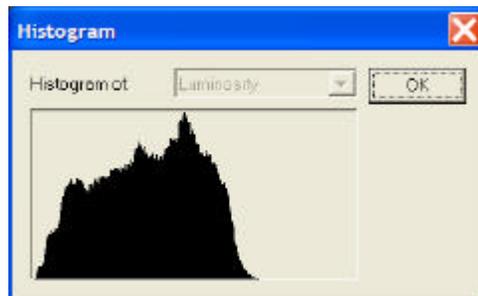
Algoritma 7.1. Perhitungan histogram citra

Nilai *pixel* hasil pengubahan mungkin \leq derajat keabuan minimum (0) atau \geq derajat keabuan maksimum (255). Karena itu, *pixel* tersebut perlu dilakukan *clipping* ke nilai keabuan minimum atau ke nilai keabuan maksimum.

Sebagai contoh, Gambar 7.1(a) adalah citra Zelda (beserta histogramnya) yang tampak gelap, sedangkan Gambar 7.1(b) adalah citra Zelda (beserta histogramnya) yang lebih terang (nilai $b = 80$). Perhatikan histogramnya. Sebelum operasi penambahan kecerahan, histogramnya menumpuk di bagian kiri. Setelah penambahan kecerahan, histogramnya bergeser ke bagian kanan.



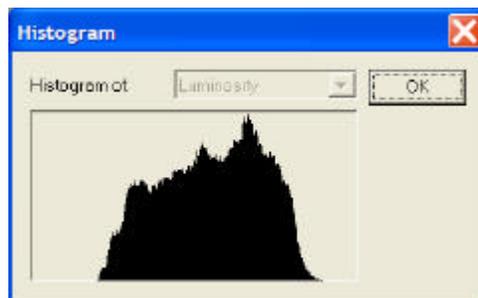
(a) Citra Zelda (orisinil)



(b) Histogram citra Zelda (orisinil)



(c) Citra Zelda setelah penambahan kecerahan dengan $b = 80$



(d) Histogram citra Zelda setelah penambahan kecerahan

Gambar 7.1. Citra Zelda; Atas: sebelum operasi penambahan kecerahan terlihat agak gelap; Bawah: Zelda setelah operasi penambahan kecerahan dengan $b = 80$.

7.3 Peregangan Kontras

Kontras menyatakan sebaran terang (*lightness*) dan gelap (*darkness*) di dalam sebuah gambar. Citra dapat dikelompokkan ke dalam tiga kategori kontras: citra kontras-rendah (*low contrast*), citra kontras-bagus (*good contrast* atau *normal contrast*), dan citra kontras-tinggi (*high contrast*). Ketiga kategori ini umumnya dibedakan secara intuitif.

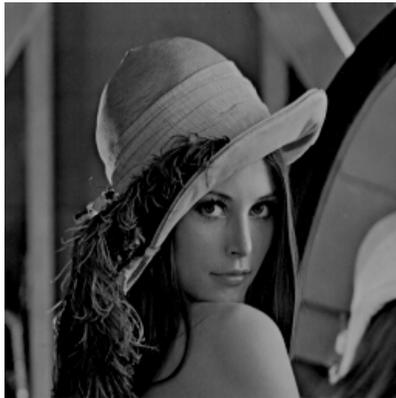
Citra kontras-rendah dicirikan dengan sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Dari histogramnya terlihat sebagian besar derajat keabuannya terkelompok (*clustered*) bersama atau hanya menempati sebagian kecil dari rentang nilai-nilai keabuan yang mungkin. Jika pengelompokan nilai-nilai *pixel* berada di bagian kiri (yang berisi nilai keabuan yang rendah), citranya cenderung gelap. Jika pengelompokan nilai-nilai *pixel* berada di bagian kanan (yang berisi nilai keabuan yang tinggi), citranya cenderung terang. Tetapi, mungkin saja suatu citra tergolong kontras-rendah meskipun tidak terlalu terang atau tidak terlalu gelap bila semua pengelompokan nilai keabuan berada di tengah histogram.

Citra kontras-bagus memperlihatkan jangkauan nilai keabuan yang lebar tanpa ada suatu nilai keabuan yang mendominasi. Histogram citranya memperlihatkan sebaran nilai keabuan yang relatif seragam.

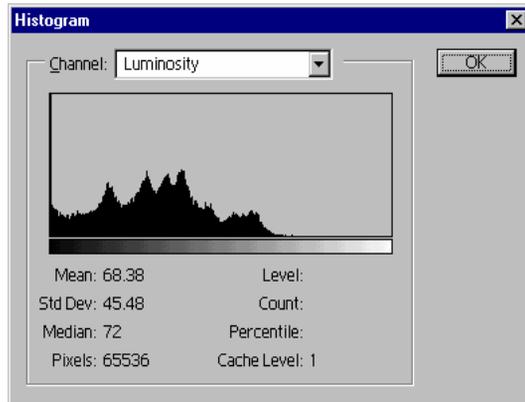
Citra kontras-tinggi, seperti halnya citra kontras bagus, memiliki jangkauan nilai keabuan yang lebar, tetapi terdapat area yang lebar yang didominasi oleh warna gelap dan area yang lebar yang didominasi oleh warna terang. Gambar dengan langit terang dengan latar depan yang gelap adalah contoh citra kontras-tinggi. Pada histogramnya terlihat dua puncak, satu pada area nilai keabuan yang rendah dan satu lagi pada area nilai keabuan yang tinggi.

Citra dengan kontras-rendah dapat diperbaiki kualitasnya dengan operasi peregangan kontras. Melalui operasi ini, nilai-nilai keabuan *pixel* akan merentang dari 0 sampai 255 (pada citra 8-bit), dengan kata lain seluruh nilai keabuan *pixel* terpakai secara merata.

Gambar 7.2 memperlihatkan tiga buah citra Lena yang masing-masing memiliki kontras-rendah, kontras-tinggi, dan kontras-bagus. Ketiga histogram ini dihasilkan dengan program *Adobe Photoshop*.



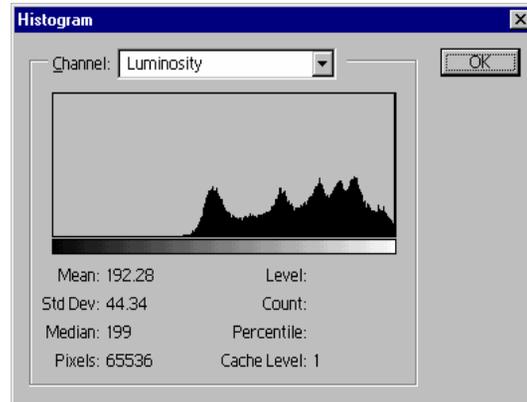
(a) Citra Lena yang terlalu gelap
(kontras rendah)



Histogram



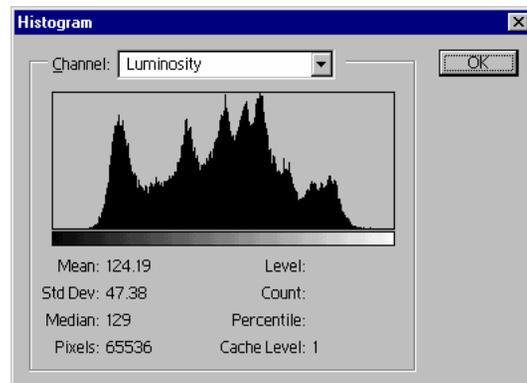
(b) Citra Lena yang terlalu terang
(kontras tinggi)



Histogram



(c) Citra Lena yang bagus (normal)
(kontras bagus)



Histogram

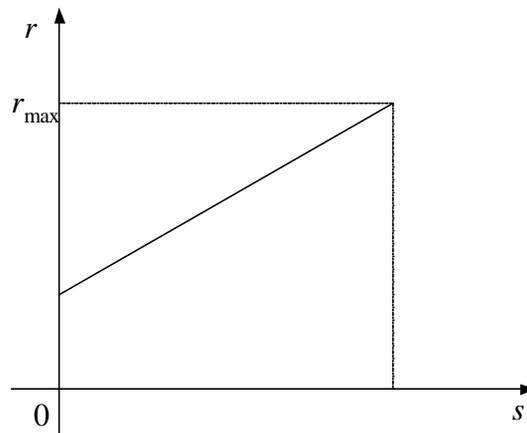
Gambar 7.2. Tiga buah citra Lena dengan tiga macam kontras.

Algoritma peregangan kontras adalah sebagai berikut:

1. Cari batas bawah pengelompokan *pixel* dengan cara memindai (*scan*) histogram dari nilai keabuan terkecil ke nilai keabuan terbesar (0 sampai 255) untuk menemukan *pixel* pertama yang melebihi nilai ambang pertama yang telah dispesifikasikan.
2. Cari batas atas pengelompokan *pixel* dengan cara memindai histogram dari nilai keabuan tertinggi ke nilai keabuan terendah (255 sampai 0) untuk menemukan *pixel* pertama yang lebih kecil dari nilai ambang kedua yang dispesifikasikan.
3. *Pixel-pixel* yang berada di bawah nilai ambang pertama di-*set* sama dengan 0, sedangkan *pixel-pixel* yang berada di atas nilai ambang kedua di-*set* sama dengan 255.
4. *Pixel-pixel* yang berada di antara nilai ambang pertama dan nilai ambang kedua dipetakan (diskalakan) untuk memenuhi rentang nilai-nilai keabuan yang lengkap (0 sampai 255) dengan persamaan:

$$s = \frac{r - r_{\max}}{r_{\min} - r_{\max}} \times 255 \quad (7.4)$$

yang dalam hal ini, r adalah nilai keabuan dalam citra semula, s adalah nilai keabuan yang baru, r_{\min} adalah nilai keabuan terendah dari kelompok *pixel*, dan r_{\max} adalah nilai keabuan tertinggi dari kelompok *pixel* (Gambar 7.3).



Gambar 7.3 Peregangan kontras

7.4 Pengubahan Histogram Citra

Untuk memperoleh histogram citra sesuai dengan keinginan kita, maka penyebaran nilai-nilai intensitas pada citra harus diubah. Terdapat dua metode pengubahan citra berdasarkan histogram:

1. Perataan historam (*histogram equalization*)
Nilai-nilai intensitas di dalam citra diubah sehingga penyebarannya seragam (*uniform*).
2. Spesifikasi histogram (*histogram spesification*)
Nilai-nilai intensitas di dalam citra diubah agar diperoleh histogram dengan bentuk yang dispesifikasikan oleh pengguna.

Kedua macam pengubahan histogram citra ini dibahas lebih rinci di dalam upabab 7.5 dan 7.6 di bawah ini.

7.5 Perataan Histogram

Sebagaimana telah dijelaskan pada pembahasan terdahulu, histogram citra memberikan informasi tentang penyebaran intensitas *pixel-pixel* di dalam citra. Misalnya, citra yang terlalu terang atau terlalu gelap memiliki histogram yang sempit.

Agar kita memperoleh citra yang baik, maka penyebaran nilai intensitas harus diubah. Teknik yang lazim dipakai adalah **perataan histogram** (*histogram equalization*). Tujuan dari perataan histogram adalah untuk memperoleh penyebaran histogram yang merata, sedemikian sehingga setiap derajat keabuan memiliki jumlah *pixel* yang relatif sama.

Karena histogram menyatakan peluang *pixel* dengan derajat keabuan tertentu, maka rumus menghitung histogram ditulis kembali sebagai fungsi peluang

$$P_r(r_k) = \frac{n_k}{n} \quad (7.5)$$

yang dalam hal ini,

$$r_k = \frac{k}{L-1}, \quad 0 \leq k \leq L-1 \quad (7.6)$$

yang artinya, derajat keabuan (k) dinormalkan terhadap derajat keabuan terbesar ($L-1$). Nilai $r_k = 0$ menyatakan hitam, dan $r_k = 1$ menyatakan putih dalam skala keabuan yang didefinisikan.

Contohnya, jika $L = 8$, maka nilai-nilai r_k dinyatakan di dalam tabel 7.1.

Tabel 7.1 Nilai-nilai r_k jika $L = 8$

k	r_k
0	$0/7 = 0$
1	$1/7$
2	$2/7$
3	$3/7$
4	$4/7$
5	$5/7$
6	$6/7$
7	$7/7 = 1$

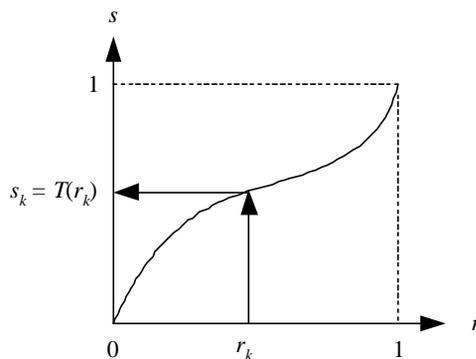
Yang dimaksud dengan perataan histogram adalah mengubah derajat keabuan suatu *pixel* (r) dengan derajat keabuan yang baru (s) dengan suatu fungsi transformasi T , yang dalam hal ini $s = T(r)$. Gambar 7.4 memperlihatkan transformasi r menjadi s .

Dua sifat yang dipertahankan pada transformasi ini:

1. Nilai s merupakan pemetaan 1 ke 1 dari r . Ini untuk menjamin representasi intensitas yang tetap. Ini berarti r dapat diperoleh kembali dari s dengan transformasi *invers*:

$$r = T^{-1}(s) \quad , \quad 0 \leq s \leq 1 \quad (7.7)$$

2. Untuk $0 \leq r_i \leq 1$, maka $0 \leq T(r) \leq 1$. Ini untuk menjamin pemetaan T konsisten pada rentang nilai yang diperbolehkan.



Gambar 7.4 Fungsi transformasi

Untuk fungsi histogram yang menerus,

$$s = T(r) = \int_0^r P_r(w)dw \quad , \quad 0 \leq r \leq 1 \quad (7.8)$$

yang dalam hal ini w adalah peubah bantu.

Dalam bentuk diskrit, nilai-nilai s diperoleh dengan persamaan berikut:

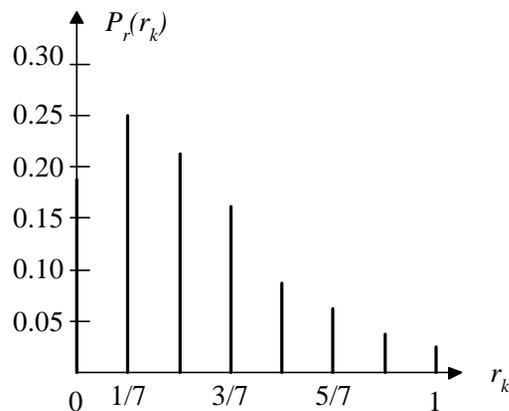
$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k P_r(r_j) \quad (7.9)$$

yang dalam hal ini, $0 \leq r_k \leq 1$, $k = 0, 1, 2, \dots, L - 1$

Contoh 7.1. [GON77] Misalkan terdapat citra yang berukuran 64×64 dengan jumlah derajat keabuan (L) = 8 dan jumlah seluruh *pixel* (n) = $64 \times 64 = 4096$:

k	r_k	n_k	$P_r(r_k) = n_k/n$
0	$0/7 = 0.00$	790	0.19
1	$1/7 = 0.14$	1023	0.25
2	$2/7 = 0.29$	850	0.21
3	$3/7 = 0.43$	656	0.16
4	$4/7 = 0.57$	329	0.08
5	$5/7 = 0.71$	245	0.06
6	$6/7 = 0.86$	122	0.03
7	$7/7 = 1.00$	81	0.02

Gambar 7.5 adalah histogram citra semula sebelum perataan.



Gambar 7.5. Histogram citra sebelum perataan

Perhitungan perataan histogram adalah sbb:

$$s_0 = T(r_0) = \sum_{j=0}^0 P_r(r_j) = P_r(r_0) = 0.19$$

$$s_1 = T(r_1) = \sum_{j=0}^1 P_r(r_j) = P_r(r_0) + P_r(r_1) = 0.19 + 0.25 = 0.44$$

$$s_2 = T(r_2) = \sum_{j=0}^2 P_r(r_j) = P_r(r_0) + P_r(r_1) + P_r(r_2) = 0.19 + 0.25 + 0.21 = 0.65$$

dan seterusnya, diperoleh:

$$\begin{array}{ll} s_3 = 0.81 & s_6 = 0.98 \\ s_4 = 0.89 & s_7 = 1.00 \\ s_5 = 0.95 & \end{array}$$

Karena pada citra ini hanya ada 8 nilai intensitas, maka nilai-nilai s_k harus dibulatkan ke nilai-nilai r yang terdekat:

$$\begin{array}{l} s_0 = 0.19 \text{ lebih dekat ke nilai } 1/7 (= 0.14), \text{ maka } s_0 = 1/7 \\ s_1 = 0.44 \text{ lebih dekat ke nilai } 3/7 (= 0.43), \text{ maka } s_1 = 3/7 \\ s_2 = 0.65 \text{ lebih dekat ke nilai } 5/7 (= 0.71), \text{ maka } s_2 = 5/7 \\ s_3 = 0.81 \text{ lebih dekat ke nilai } 6/7 (= 0.86), \text{ maka } s_3 = 6/7 \\ s_4 = 0.89 \text{ lebih dekat ke nilai } 6/7 (= 0.86), \text{ maka } s_4 = 6/7 \\ s_5 = 0.95 \text{ lebih dekat ke nilai } 7/7 (= 1.00), \text{ maka } s_5 = 7/7 \\ s_6 = 0.98 \text{ lebih dekat ke nilai } 7/7 (= 1.00), \text{ maka } s_6 = 7/7 \\ s_7 = 1.00 \text{ lebih dekat ke nilai } 7/7 (= 1.00), \text{ maka } s_7 = 7/7 \end{array}$$

Hasil transformasinya :

k	r_k	s_k
0	0	1/7
1	1/7	3/7
2	2/7	5/7
3	3/7	6/7
4	4/7	6/7
5	5/7	1
6	6/7	1
7	1	1

Terlihat dari contoh di atas hanya lima nilai intensitas yang terisi (1/7, 3/7, 5/7, 6/7, dan 1).

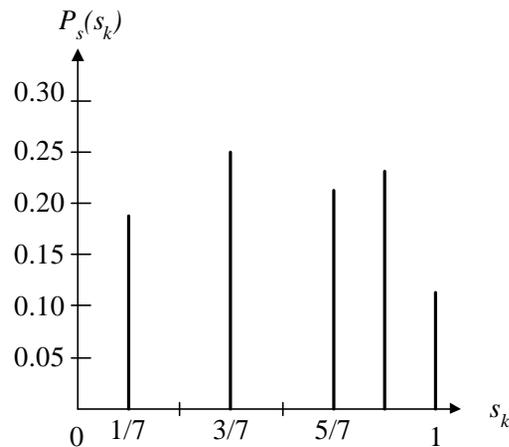
Notasi untuk tiap hasil transformasi didefinisi ulang menjadi:

$$s_0 = 1/7, s_1 = 3/7, s_2 = 5/7, s_3 = 6/7, s_4 = 1$$

Karena $r_0 = 0$ dipetakan ke $s_0 = 1/7$, terdapat 790 *pixel* hasil transformasi yang memiliki nilai intensitas $1/7$. Selanjutnya, $s_1 = 3/7$ memiliki 1023 *pixel*, $s_2 = 5/7$ memiliki 850 *pixel*. Juga, karena r_3 dan r_4 dipetakan ke nilai yang sama, $s_3 = 6/7$, maka jumlah *pixel* yang bernilai $6/7$ adalah $656 + 329 = 985$. Jumlah *pixel* hasil transformasi diringkas pada tabel di bawah ini:

s_k	n_k	$P_s(s_k) = n_k/n$
1/7	790	0.19
3/7	1023	0.25
5/7	850	0.21
6/7	$656 + 329 = 958$	0.23
7/7	$245 + 122 + 81 = 448$	0.11

Gambar 7.5 adalah histogram citra hasil perataan. ■



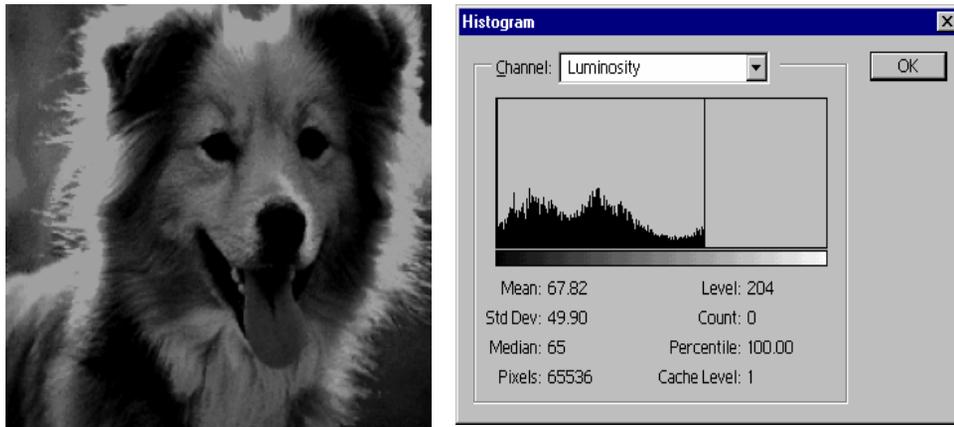
Gambar 7.5. Histogram citra hasil perataan

Gambar 7.6 memperlihatkan perataan histogram pada citra anjing *collie*. Pada mulanya citra *collie* terlihat terlalu gelap. Histogramnya menumpuk pada daerah derajat keabuan bagian kiri. Dengan teknik perataan histogram, citra anjing *collie* terlihat lebih bagus. Hal ini dapat dilihat juga pada histogramnya yang tersebar merata di seluruh daerah derajat keabuan.

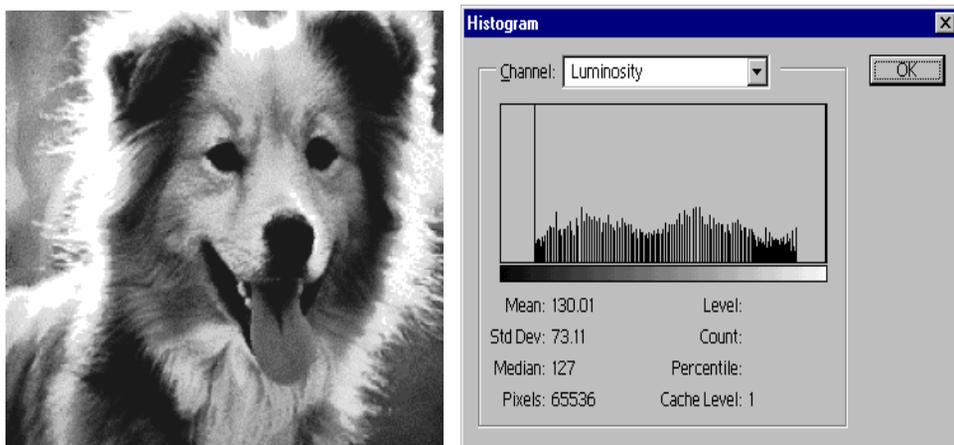
Meskipun perataan histogram bertujuan menyebarkan secara merata nilai-nilai derajat keabuan, tetapi seringkali histogram hasil perataan tidak benar-benar tersebar secara merata (misalnya pada contoh di atas). Alasannya adalah :

1. Derajat keabuan terbatas jumlahnya. Nilai intensitas baru hasil perataan merupakan pembulatan ke derajat keabuan terdekat.
2. Jumlah *pixel* yang digunakan sangat terbatas.

Agar hasil perataan benar-benar seragam sebarannya, maka citra yang diolah haruslah dalam bentuk malar (*continue*), yang dalam praktek ini jelas tidak mungkin.



(a) Kiri: citra anjing *collie* yang terlalu gelap; Kanan: histogramnya



(b) Kiri: citra anjing *collie* setelah perataan histogram; kanan: histogramnya

Gambar 7.6. Contoh perataan histogram pada citra anjing *collie*

Algoritma perataan histogram ditunjukkan pada Algoritma 7.2 [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]`. Histogram citra semula disimpan di dalam tabel `Hist[0..255]` yang bertipe riil. Histogram hasil perataan disimpan di dalam `HistEq[0..255]` yang bertipe *integer*.

```

void PerataanHistogram(citra Image, int N, int M)
/* Mengubah citra Image yang berukuran N ´ M dengan melakukan perataan
  histogram (histogram equalization).
*/
{
  int i, j;
  float sum, float Hist[256];
  int HistEq[256];          /* histogram hasil perataan */

  histogram(Image,N,M,Hist); /* hitung histogram citra */
  for(i=0;i<256;i++)
  {
    sum=0.0;
    for (j=0;j<=i;j++)
      sum=sum+Hist[j];
    HistEq[i]=floor(255*sum);
  }
  /* update citra sesuai histogram hasil perataan */
  for(i=0;i<=N-1;i++)
    for(j=0;j<=M-1;j++)
      Image[i][j]=HistEq[Image[i][j]];
}

```

Algoritma 7.2 Perataan histogram citra

7.6 Spesifikasi Histogram

Perataan histogram memetakan histogram citra semula menjadi histogram yang seragam. Bila histogram yang diinginkan tidak seragam, maka cara ini tidak dapat digunakan.

Metode spesifikasi histogram (*histogram specification*) memberikan cara menghasilkan histogram yang ditentukan oleh pengguna. Cara pembentukan histogramnya memanfaatkan sifat pada perataan histogram. Bila fungsi transformasi pada perataan histogram menghasilkan histogram semula menjadi histogram yang seragam, maka fungsi balikkannya (*inverse*) memetakan histogram yang seragam menjadi histogram semula. Sifat ini dapat dimanfaatkan untuk mengubah histogram citra menjadi histogram lain yang tidak seragam.

Dasar teorinya adalah sebagai berikut: misalkan $P_r(r)$ dan $P_z(z)$ masing-masing adalah histogram citra semula dan histogram yang diinginkan. Fungsi transformasi T mula-mula memetakan intensitas citra semula menjadi histogram yang seragam dengan cara perataan histogram,

$$s = T(r) = \int_0^r P_r(w)dw$$

Jika histogram yang diinginkan sudah dispesifikasikan, kita dapat melakukan perataan histogram pula dengan fungsi transformasi G :

$$v = G(z) = \int_0^z P_z(w)dw \quad (7.10)$$

Balikan (*invers*) dari fungsi G ,

$$z = G^{-1}(v) \quad (7.11)$$

akan menghasilkan histogram yang diinginkan kembali.

Dengan mengganti v dengan s pada persamaan yang terakhir,

$$z \approx G^{-1}(s) \quad (7.12)$$

maka kita dapat memperoleh nilai intensitas yang diinginkan. Hasil yang diperoleh merupakan hampiran karena kita mencoba menemukan nilai s yang transformasinya mendekati nilai z .

Algoritma spesifikasi histogram adalah sebagai berikut:

1. Misalkan $P_r(r)$ adalah histogram citra semula. Lakukan perataan histogram terhadap citra semula dengan fungsi transformasi T ,

$$s = T(r) = \int_0^r P_r(w)dw$$

Dalam bentuk diskrit, nilai-nilai s diperoleh dengan persamaan berikut:

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k P_r(r_j)$$

2. Tentukan histogram yang diinginkan, misalkan $P_z(z)$ adalah histogram yang diinginkan. Lakukan perataan histogram dengan fungsi transformasi G ,

$$v = G(z) = \int_0^z P_z(w)dw$$

Dalam bentuk diskrit, nilai-nilai v diperoleh dengan persamaan berikut:

$$v_k = G(z_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k P_z(z_j)$$

3. Terapkan fungsi transformasi balikan, $z = G^{-1}(s)$ terhadap histogram hasil langkah 1. Caranya adalah dengan mencari nilai-nilai s yang memberi nilai z terdekat.

Dengan kata lain, histogram nilai-nilai intensitas pada citra semula dipetakan menjadi intensitas z pada citra yang diinginkan dengan fungsi

$$z = G^{-1}[T(r)]$$

Ketiga langkah di dalam algoritma spesifikasi histogram di atas digambarkan dalam bagan pada Gambar 7.7.



Gambar 7.7 Langkah-langkah metode spesifikasi histogram

Contoh 7.2. [GON77] Tinjau kembali citra yang berukuran 64×64 dengan jumlah derajat keabuan (L) = 8 dan jumlah seluruh *pixel* (n) = $64 \times 64 = 4096$. Tabel histogram citra semula dan tabel histogram yang diinginkan adalah sebagai berikut:

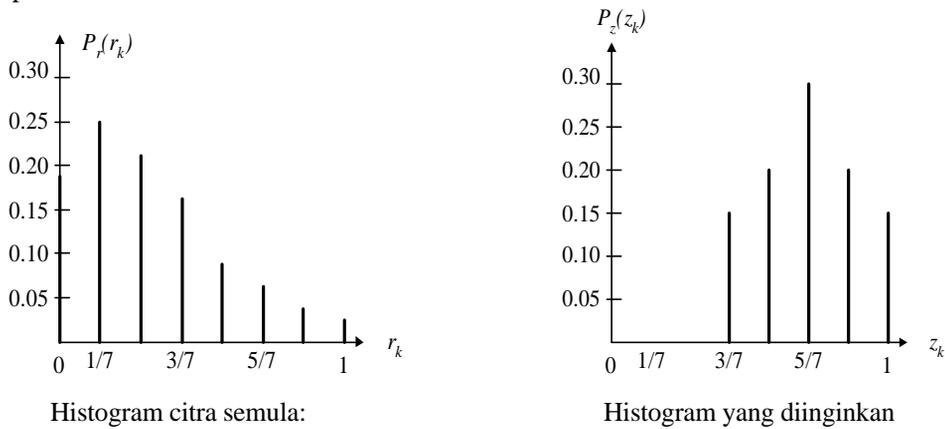
Tabel histogram citra semula

r_k	n_k	$P_r(r_k) = n_k/n$
0/7 = 0.00	790	0.19
1/7 = 0.14	1023	0.25
2/7 = 0.29	850	0.21
3/7 = 0.43	656	0.16
4/7 = 0.57	329	0.08
5/7 = 0.71	245	0.06
6/7 = 0.86	122	0.03
7/7 = 1.00	81	0.02

Tabel histogram yang diinginkan

z_k	$P_z(z_k)$
0/7 = 0.00	0.00
1/7 = 0.14	0.00
2/7 = 0.29	0.00
3/7 = 0.43	0.15
4/7 = 0.57	0.20
5/7 = 0.71	0.30
6/7 = 0.86	0.20
7/7 = 1.00	0.15

Histogram citra semula dan histogram yang diinginkan diperlihatkan secara grafis pada Gambar 7.8.



Gambar 7.8 Histogram citra semula dan histogram yang diinginkan

Langkah-langkah pembentukan histogram adalah sebagai berikut:

Langkah 1: Hasil perataan histogram terhadap citra semula,

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k P_r(r_j)$$

telah dilakukan (lihat Contoh 7.1), dan ini hasilnya:

$r_j \text{ ® } s_k$	n_k	$P_s(s_k) = n_k/n$
$r_0 \rightarrow s_0 = 1/7$	790	0.19
$r_1 \rightarrow s_1 = 3/7$	1023	0.25
$r_2 \rightarrow s_2 = 5/7$	850	0.21
$r_3, r_4 \rightarrow s_3 = 6/7$	$656 + 329 = 958$	0.23
$r_5, r_6, r_7 \rightarrow s_4 = 7/7$	$245 + 122 + 81 = 448$	0.11

Langkah 2: Lakukan perataan terhadap histogram yang diinginkan, $P_z(z)$, dengan persamaan

$$v_k = G(z_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k P_z(z_j)$$

Hasilnya adalah sbb:

$$\begin{array}{ll} v_0 = G(z_0) = 0.00 & v_4 = G(z_4) = 0.35 \\ v_1 = G(z_1) = 0.00 & v_5 = G(z_5) = 0.65 \\ v_2 = G(z_2) = 0.00 & v_6 = G(z_6) = 0.85 \\ v_3 = G(z_3) = 0.15 & v_7 = G(z_7) = 1.00 \end{array}$$

Langkah 3: Gunakan transformasi $z = G^{-1}(s)$ untuk memperoleh nilai z dari nilai s hasil perataan histogram.

$$\begin{array}{l} s_0 = 1/7 \approx 0.14 \text{ paling dekat dengan } 0.15 = G(z_3), \text{ jadi } G^{-1}(0.14) = z_3 = 1/7 \\ s_1 = 3/7 \approx 0.43 \text{ paling dekat dengan } 0.35 = G(z_4), \text{ jadi } G^{-1}(0.43) = z_4 = 4/7 \\ s_2 = 5/7 \approx 0.71 \text{ paling dekat dengan } 0.65 = G(z_5), \text{ jadi } G^{-1}(0.71) = z_5 = 5/7 \\ s_3 = 6/7 \approx 0.86 \text{ paling dekat dengan } 0.85 = G(z_6), \text{ jadi } G^{-1}(0.86) = z_6 = 6/7 \\ s_4 = 1 \approx 1.00 \text{ paling dekat dengan } 1.00 = G(z_7), \text{ jadi } G^{-1}(1.00) = z_7 = 1 \end{array}$$

Diperoleh pemetaan langsung sebagai berikut:

$$\begin{array}{ll} r_0 = 0 \rightarrow z_3 = 3/7 & r_4 = 4/7 \rightarrow z_6 = 6/7 \\ r_1 = 1/7 \rightarrow z_4 = 4/7 & r_5 = 5/7 \rightarrow z_7 = 1 \\ r_2 = 2/7 \rightarrow z_5 = 5/7 & r_6 = 6/7 \rightarrow z_7 = 1 \\ r_3 = 3/7 \rightarrow z_6 = 6/7 & r_7 = 1 \rightarrow z_7 = 1 \end{array}$$

Penyebaran pixel:

Karena $r_0 = 0$ dipetakan ke $z_3 = 3/7$, maka terdapat 790 *pixel* hasil transformasi yang memiliki nilai intensitas 3/7.

Karena $r_1 = 1/7$ dipetakan ke $z_4 = 4/7$, maka terdapat 1023 *pixel* hasil transformasi yang memiliki nilai intensitas 4/7.

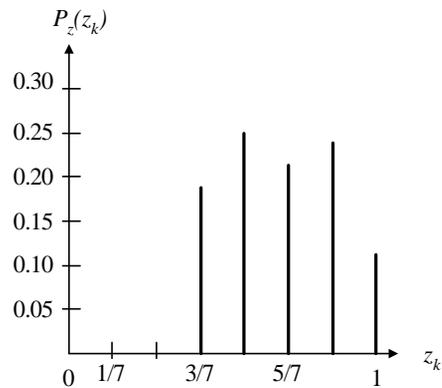
Karena $r_2 = 2/7$ dipetakan ke $z_5 = 5/7$, maka terdapat 850 *pixel* hasil transformasi yang memiliki nilai intensitas 5/7.

Karena $r_3 = 3/7$ dan $r_4 = 4/7$ dipetakan ke $z_6 = 6/7$, terdapat $245 + 122 + 81 = 448$ *pixel* hasil transformasi yang memiliki nilai intensitas 1.

Selanjutnya, tidak ada *pixel* yang mempunyai intensitas $z_0 = 0$, $z_1 = 1/7$, dan $z_2 = 2/7$, karena tidak ada r_k yang dipetakan ke nilai-nilai z tersebut.

z_k	n_k	$P_z(z_k) = n_k/n$
0	0	0.00
1/7	0	0.00
2/7	0	0.00
3/7	790	0.19
4/7	1023	0.25
5/7	850	0.21
6/7	985	0.24
1	448	0.11

Histogram yang terbentuk:



Seperti yang sudah disebutkan sebelum ini, histogram yang diperoleh merupakan hampiran dari histogram yang dispesifikasikan karena kita mencoba menemukan nilai s yang transformasinya mendekati nilai z .

Dalam praktek, mungkin terdapat ambiguitas pada nilai transformasi balikan, $G^{-1}(s)$. Dengan kata lain, nilai transformasi balikan dari s ke z tidak tunggal. Hal ini terjadi karena: (i) proses pembulatan $G^{-1}(s)$ ke nilai intensitas terdekat, atau (ii) terdapat nilai intensitas yang tidak terisi di dalam histogram spesifikasi. Solusi termudah untuk masalah ini adalah memilih nilai z yang terdekat dengan histogram yang dispesifikasikan.

Algoritma Spesifikasi Histogram ditunjukkan pada Algoritma 7.3. 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]`. Hasil perataan histogram dari citra semula disimpan kembali di dalam matriks `Image[0..N-1][0..M-1]`. Histogram yang dispesifikasikan disimpan di dalam `Spec[0..255]`. Histogram hasil perataan dari `Spec` disimpan di dalam tabel `SpecEq[0..255]`. Histogram hasil transformasi balikan disimpan di dalam tabel `InvHist[0..255]`.

```
void SpesifikasiHistogram(citra Image, int N, int M, float Spec[256])
/* Mengubah citra Image yang berukuran N x M berdasarkan histogram yang
dispesifikasikan oleh pengguna (Spec).
*/
{
    float sum, Hist[256];
    int i, j, minj, minval, HistEq[256], SpecEq[256], InvHist[256];

    /* lakukan perataan histogram terhadap citra semula */
    histogram(Image,N,M,Hist); /* hitung histogram citra */
    for(i=0;i<256;i++)
    {
        sum=0.0;

```

```

for (j=0;j<=i;j++)
    sum=sum+Hist[j];
HistEq[i]=floor(255*sum);
}

/* lakukan perataan histogram terhadap citra Spec */
for(i=0;i<=255;i++)
{
    sum=0.0;
    for (j=0;j<=i;j++)
        sum=sum+Spec[j];
    SpecEq[i]=floor(255*sum);
}

/* lakukan transformasi balikan */
for(i=0;i<=N-1;i++)
{
    minval=abs(HistEq[i] - SpecEq[0]);
    minj=0;
    for(j=0;j<=255;j++)
        if (abs(HistEq[i] - SpecEq[j]) < minval)
        {
            minval = abs(HistEq[i] - SpecEq[j]);
            minj=j;
        }
    InvHist[i]=minj;
}

/* update citra setelah pembentukan histogram */
for(i=0;i<=N-1;i++)
    for(j=0;j<=M-1;j++)
        Image[i][j]=InvHist[Image[i][j]];

```

Algoritma 7.3 Pengubahan citra berdasarkan histogram yang dispesifikasikan

7.7 Pelembutan Citra (*Image Smoothing*)

Pelembutan citra (*image smoothing*) bertujuan untuk menekan gangguan (*noise*) pada citra. Gangguan tersebut biasanya muncul sebagai akibat dari hasil penerokan yang tidak bagus (*sensor noise*, *photographic grain noise*) atau akibat saluran transmisi (pada pengiriman data).

Gangguan pada citra umumnya berupa variasi intensitas suatu *pixel* yang tidak berkorelasi dengan *pixel-pixel* tetangganya. Secara visual, gangguan mudah dilihat oleh mata karena tampak berbeda dengan *pixel* tetangganya. Gambar 7.9 adalah citra Lena yang mengalami gangguan berupa *spike* atau *speckle* yang tampil pada gambar dalam bentuk bercak putih atau hitam seperti beras.

Pixel yang mengalami gangguan umumnya memiliki frekuensi tinggi (berdasarkan analisis frekuensi dengan transformasi Fourier). Komponen citra yang berfrekuensi rendah umumnya mempunyai nilai *pixel* konstan atau berubah sangat lambat. Operasi pelembutan citra dilakukan untuk menekan komponen yang berfrekuensi tinggi dan meloloskan komponen yang berfrekuensi rendah.



Gambar 7.9. Citra Lena yang mengalami gangguan berupa spike

Operasi pelembutan dapat dilakukan pada ranah spsial maupun pada ranah frekuensi. Pada ranah spasial, operasi pelembutan dilakukan dengan mengganti intensitas suatu *pixel* dengan rata-rata dari nilai *pixel* tersebut dengan nilai *pixel-pixel* tetangganya. Jadi, diberikan citra $f(x,y)$ yang berukuran $N \times M$. Citra hasil pelembutan, $g(x,y)$, didefinisikan sebagai berikut:

$$g(x, y) = \frac{1}{d} \sum_{r=m_1}^{m_2} \sum_{s=n_1}^{n_2} f(x+r, y+s) \quad (7.13)$$

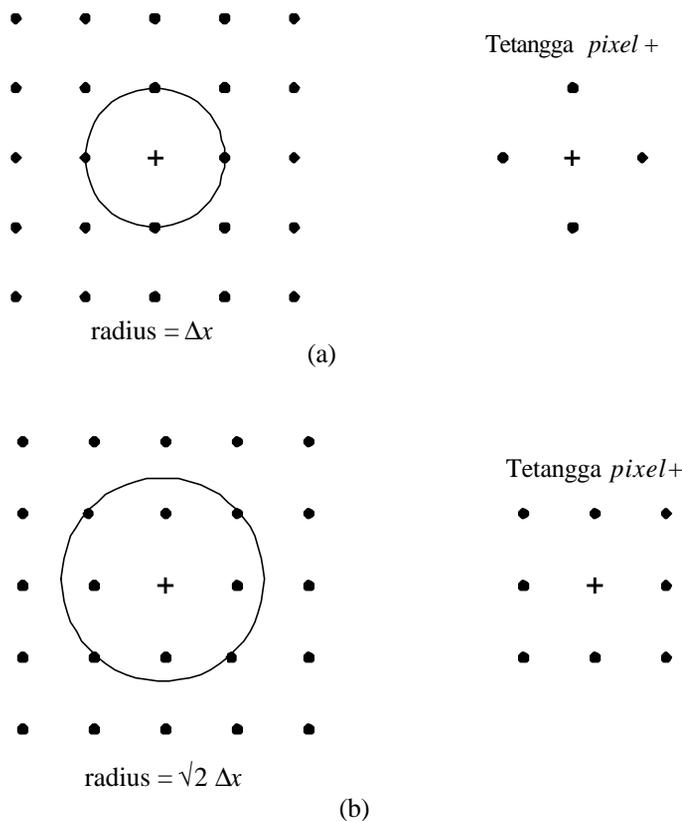
yang dalam hal ini d adalah jumlah *pixel* yang terlibat dalam perhitungan rata-rata. Gambar 7.10 memperlihatkan dua buah skema perata-rataan [GON77]. Pada skema pertama, tetangga sebuah *pixel* adalah *pixel-pixel* yang berjarak Δx , sedangkan pada skema kedua tetangga sebuah *pixel* adalah *pixel-pixel* yang berjarak paling jauh $\sqrt{2} \Delta x$.

Operasi perata-rataan di atas dapat dipandang sebagai konvolusi antara citra $f(x,y)$ dengan penapis $h(x,y)$:

$$g(x,y) = f(x,y) * h(x,y) \quad (7.14)$$

Penapis h disebut **penapis merata** (*mean filter*). Dalam ranah frekuensi, operasi konvolusi tersebut adalah

$$G(u,v) = F(u,v)H(u,v) \quad (7.15)$$



Gambar 7.10. Skema perata-rataan

Contoh penapis rerata yang berukuran 3×3 dan 2×2 adalah seperti di bawah ini (elemen yang bertanda • menyatakan posisi $(0, 0)$ dari *pixel* yan dikonvolusi):

$$(i) \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & \bullet 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad (ii) \begin{bmatrix} \bullet 1/4 & 1/4 \\ 1/4 & 1/4 \end{bmatrix}$$

Algoritma pelembutan citra dengan penapis 3×3 ditunjukkan pada Algoritma 7.4.

```
void PerataanCitra(citra Image, citra ImageResult, int N, int M)
/* Melembutkan citra Image yang berukuran N x M dengan melakukan
konvolusi citra Image dengan penapis rerata yang berukuran 3 x 3. Hasil
pelembutna disimpan di dalam ImageResult.
*/
{ int i, j;
  for (i=1; i<=N-1; i++)
```

```

for(j=1; j<=M-1; j++)
{
  ImageResult[i][j]=
    Image[i-1][j-1] + Image[i-1][j] + Image[i-1,j+1]+
    Image[i][j-1] + Image[i][j] + Image[i,j+1] +
    Image[i+1][j-1] + Image[i+1][j] + Image[i+1,j+1];
  ImageResult[i][j]=ImageResult[i][j]/9;
}
}

```

Algoritma 7.4. Operasi pelembutan citra dengan penapis rerata 3 × 3.

Operasi penapisan ini mempunyai efek pemerataan derajat keabuan, sehingga gambar yang diperoleh tampak lebih kabur kontrasnya. Efek pengaburan ini disebut efek *blurring*. Gambar 7.11 adalah hasil pelembutan citra Lena dari Gambar 7.9 dengan penapis rata-rata 3 × 3.

Efek pengaburan yang dihasilkan dari penapis rata-rata dapat dikurangi dengan prosedur pengambangan berikut:

$$g(x, y) = \begin{cases} \frac{1}{d} \sum_{r=m_1}^{m_2} \sum_{s=n_1}^{n_2} f(x+r, y+s) & \text{jika } \left| f(x, y) - \frac{1}{d} \sum_{r=m_1}^{m_2} \sum_{s=n_1}^{n_2} f(x+r, y+s) \right| > T \\ f(x, y), & \text{lainnya} \end{cases} \tag{7.16}$$

dengan T adalah nilai ambang yang dispesifikasikan.



Gambar 7.11. Citra Lena yang sudah dilembutkan dengan penapis rerata 3 × 3

Penapis $h(x,y)$ pada operasi pelembutan citra disebut juga **penapis lolos-rendah** (*low-pass filter*), karena penapis tersebut menekan komponen yang berfrekuensi tinggi (misalnya *pixel* gangguan, *pixel* tepi) dan meloloskan komponen yang berfrekuensi rendah.

Penapis Lolos-Rendah

Penapis rata-rata adalah salah satu penapis lolos-rendah yang paling sederhana. Aturan untuk penapis lolos-rendah adalah [GAL95]:

1. Semua koefisien penapis harus positif
2. Jumlah semua koefisien harus sama dengan 1

Jika jumlah semua koefisien lebih besar dari 1, maka konvolusi menghasilkan penguatan (tidak diinginkan). Jika jumlah semua koefisien kurang dari 1, maka yang dihasilkan adalah penurunan, dan nilai mutlak setiap *pixel* di seluruh bagian citra berkurang. Akibatnya, citra hasil pelembutan tampak lebih gelap.

Ilustrasi konvolusi dengan penapis rata-rata 3×3 terhadap citra yang mengandung *pixel* derau diperlihatkan di bawah ini. *Pixel* yang mengalami gangguan dimisalkan bernilai 17, sedangkan nilai *pixel* tetangganya (yang tidak mengalami gangguan) bernilai rendah, misalkan 8. Efek dari penapis lolos-rendah adalah sbb: *pixel-pixel* tetangga tidak mengalami perubahan (kecuali bila terdapat perbedaan nilai atau gradien antara *pixel-pixel* yang bertetangga), sedangkan *pixel* derau nilainya turun menjadi 9:

$$\begin{bmatrix} 8 & 8 & 8 & 8 \\ 8 & 17 & 8 & 8 \\ 8 & 8 & 8 & 8 \end{bmatrix}$$

(i) sebelum konvolusi

$$\begin{bmatrix} 8 & 8 & 8 & 8 \\ 8 & 9 & 9 & 8 \\ 8 & 8 & 8 & 8 \end{bmatrix}$$

(ii) setelah konvolusi

Nilai 9 ini diperoleh dari hasil perhitungan konvolusi:

$$f'(1,1) = (8 + 8 + 8 + 8 + 17 + 8 + 8 + 8 + 8)/9 = 81/9 = 9$$

Selain dengan penapis rata-rata, penapis lolos-rendah lain yang dapat digunakan pada operasi pelembutan adalah:

$$(i) \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$

$$(ii) \begin{bmatrix} 1/10 & 1/10 & 1/10 \\ 1/10 & 1/5 & 1/10 \\ 1/10 & 1/10 & 1/10 \end{bmatrix}$$

$$(iii) \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$

Jika citra hasil penapisan lolos-rendah dikurangi dari citra semula (yang mengandung derau), maka yang dihasilkan adalah peningkatan relatif komponen citra yang berfrekuensi tinggi tanpa peningkatan komponen derau. Akibatnya, citra hasil pengurangan muncul lebih tajam dari citra semula. Ini dapat digunakan untuk menonjolkan bagian citra yang tidak jelas, misalnya tertutup oleh kabut atau awan. Aplikasi ini dapat diterapkan untuk mendapatkan citra kota Jakarta yang lebih bagus daripada citra kota Jakarta yang tertutup oleh kabut.

Penapis lolos-rendah yang disebutkan di atas merupakan penapis linjar (*linear*). Operasi pelembutan dapat juga dilakukan dengan menggunakan penapis nirlinjar, yaitu:

- a. Penapis minimum (*min filter*)
- b. Penapis maksimum (*max filter*)
- c. Penapis median (*median filter*)

Penapis nirlinjar sebenarnya tidak termasuk kategori operasi konvolusi yang lazim. Cara kerja penapis tersebut berbeda dari penapis linjar. Operasi dengan penapis nirlinjar dihitung dengan mengurutkan nilai intensitas sekelompok *pixel*, lalu mengganti nilai *pixel* yang sedang diproses dengan nilai tertentu dari kelompok tersebut (misalnya nilai median dari kelompok *pixel*, nilai maksimum atau nilai minimum dari kelompok *pixel*)

Penapis Median

Penapis nirlinjar yang akan dijelaskan adalah penapis median. Penapis ini dikembangkan oleh Tukey. Pada penapis median, suatu “jendela” (*window*) memuat sejumlah *pixel* (ganjil). Jendela digeser titik demi titik pada seluruh daerah citra. Pada setiap pergeseran dibuat jendela baru. Titik tengah dari jendela ini diubah dengan nilai median dari jendela tersebut.

Sebagai contoh, tinjau jendela berupa kelompok *pixel* (berbentuk kotak diarsir) pada sebuah citra pada Gambar 7.12(a). *Pixel* yang sedang diproses adalah yang mempunyai intensitas 35. Urutkan *pixel-pixel* tersebut:

9 10 10 10 **10** 10 11 12 35

Median dari kelompok tersebut adalah 10 (dicetak tebal). Titik tengah dari jendela (35) sekarang diganti dengan nilai median (10). Hasil dari penapis median diperlihatkan pada Gambar 7.12(b). Jadi, penapis median menghilangkan nilai *pixel* yang sangat berbeda dengan *pixel* tetangganya.

13	10	15	14	18
12	10	10	10	15
11	11	35	10	10
13	9	12	10	12
13	12	9	8	10

(a) *Pixel* bernilai 35 terkena derau

13	10	15	14	18
12	10	10	10	15
11	11	10	10	10
13	9	12	10	12
13	12	9	8	10

(b) 35 diganti dengan median dari kelompok 3×3 *pixel*

Gambar 7.12. Penghilangan derau dengan penapis median 3×3 .

Selain berbentuk kotak, jendela pada penapis median dapat bermacam-macam bentuknya, seperti palang (*cross*), lajur vertikal (*vertical strip*), atau lajur horizontal (*horizontal strip*). Gambar 7.13 adalah hasil pelembutan citra dari Gambar 7.9 dengan penapis median 3×3 .

Dari kedua contoh penapis (penapis rerata dan penapis median), dapat dilihat bahwa penapis median memberikan hasil yang lebih baik dibandingkan penapis rerata untuk citra yang mengalami gangguan dalam bentuk *spike* berupa bercak-bercak putih.



Gambar 7.13. Citra Lena yang dilembutkan dengan penapis median.

Cara lain yang dapat dilakukan pada pelembutan citra adalah merata-ratakan derajat keabuan setiap *pixel* dari citra yang sama yang diambil berkali-kali. Misalnya untuk gambar yang sama direkam dua kali, lalu dihitung intensitas rata-rata untuk setiap *pixel*:

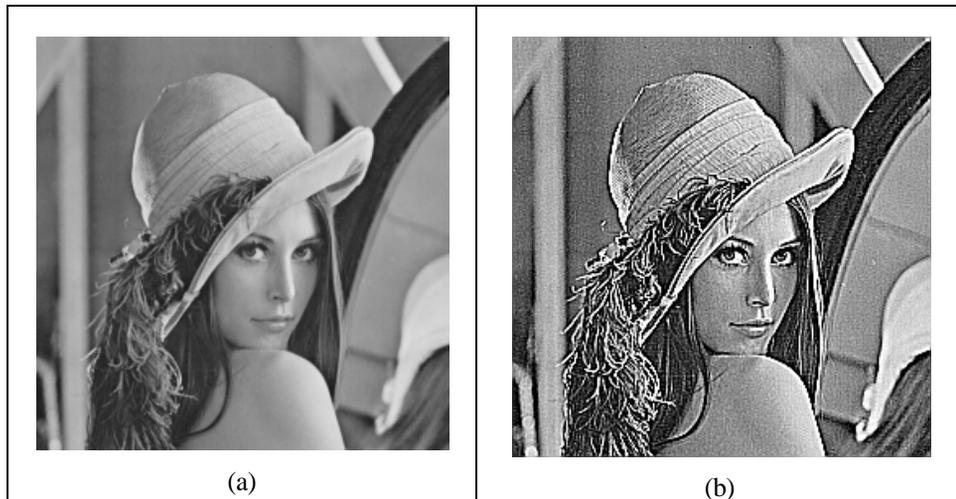
$$f'(x,y) = \frac{1}{2} \{ f_1(x, y) + f_2(x, y) \} \quad (7.16)$$

7.8 Penajaman Citra (*Image Sharpening*)

Operasi penajaman citra bertujuan memperjelas tepi pada objek di dalam citra. Penajaman citra merupakan kebalikan dari operasi pelembutan citra karena operasi ini menghilangkan bagian citra yang lembut.

Operasi penajaman dilakukan dengan melewati citra pada **penapis lolos-tinggi** (*high-pass filter*). Penapis lolos-tinggi akan meloloskan (atau memperkuat) komponen yang berfrekuensi tinggi (misalnya tepi atau pinggiran objek) dan akan menurunkan komponen berfrekuensi rendah. Akibatnya, pinggiran objek terlihat lebih tajam dibandingkan sekitarnya.

Karena penajaman citra lebih berpengaruh pada tepi (*edge*) objek, maka penajaman citra sering disebut juga **penajaman tepi** (*edge sharpening*) atau peningkatan kualitas tepi (*edge enhancement*). Gambar 7.14 adalah citra Lena setelah ditajamkan gambarnya.



Gambar 7.14 (a) Citra Lena semula, (b) Citra Lena setelah penajaman

Selain untuk mempertajam gambar, penapis lolos-tinggi juga digunakan untuk mendeteksi keberadaan tepi (*edge detection*). Dalam hal ini, *pixel-pixel* tepi ditampilkan lebih terang (*highlight*) sedangkan *pixel-pixel* bukan tepi dibuat gelap (hitam). Masalah pendeteksian tepi akan dibahas dalam pokok bahasan tersendiri.

Penapis Lolos-Tinggi

Aturan penapis lolos-tinggi [GAL95]:

1. koefisien penapis boleh positif, negatif, atau nol
2. jumlah semua koefisien adalah 0 atau 1

Jika jumlah koefisien = 0, maka komponen berfrekuensi rendah akan turun nilainya, sedangkan jika jumlah koefisien sama dengan 1, maka komponen berfrekuensi rendah akan tetap sama dengan nilai semula.

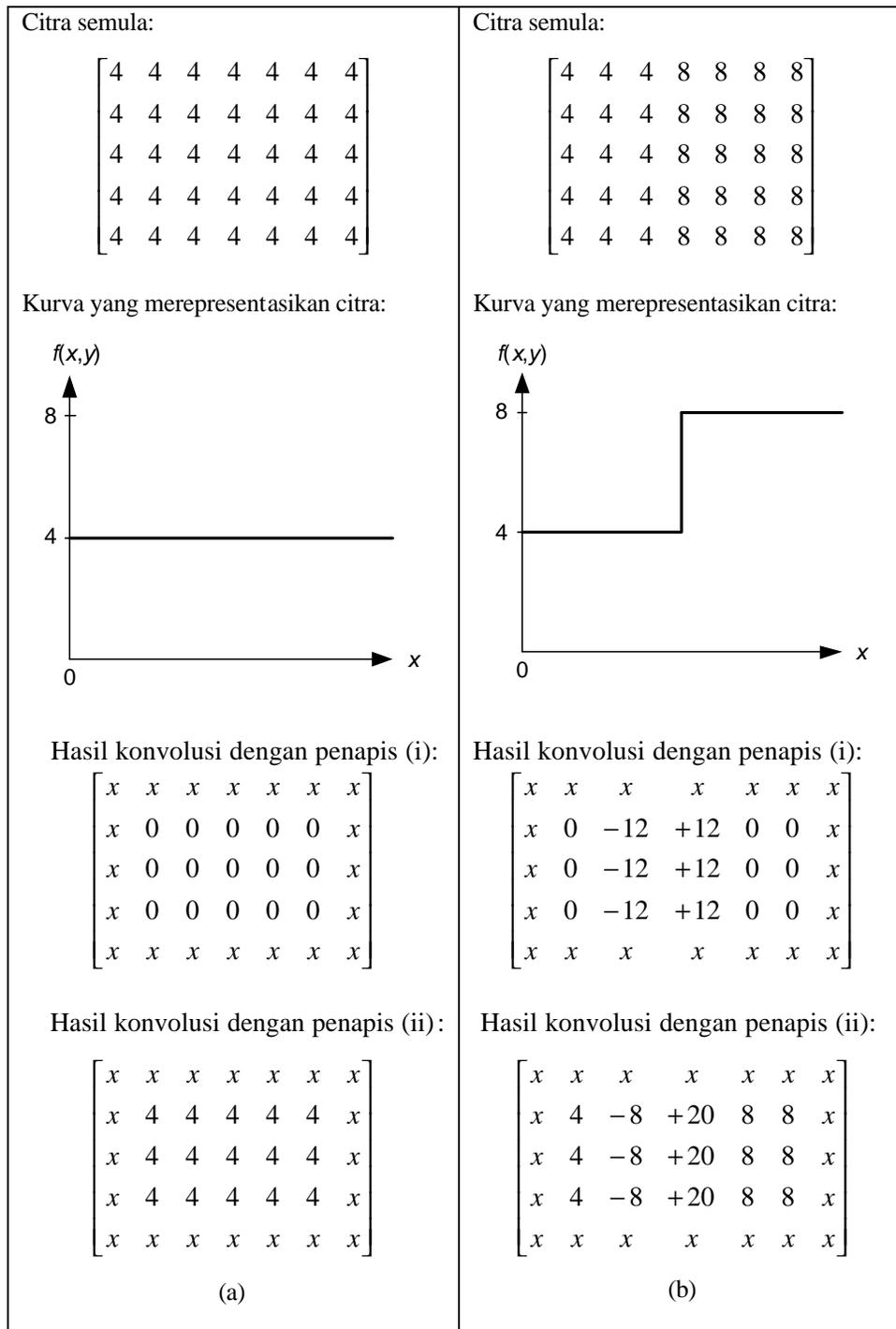
Contoh-contoh penapis lolos-tinggi:

$$\begin{array}{lll}
 \text{(i)} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} & \text{(ii)} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} & \text{(iii)} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \\
 \Sigma = 0 & \Sigma = 1 & \Sigma = 1 \\
 \\
 \text{(iv)} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix} & \text{(v)} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} & \text{(vi)} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \\
 \Sigma = 1 & \Sigma = 0 & \Sigma = 0
 \end{array}$$

Nilai koefisien yang besar di titik pusat penapis memainkan peranan kunci dalam proses konvolusi. Pada komponen citra dengan frekuensi tinggi (yang berarti perubahan yang besar pada nilai intensitasnya), nilai tengah ini dikalikan dengan nilai *pixel* yang dihitung. Koefisien negatif yang lebih kecil di sekitar titik tengah penapis bekerja untuk mengurangi faktor pembobotan yang besar. Efek netto adalah, *pixel-pixel* yang bernilai besar diperkuat, sedangkan area citra dengan intensitas *pixel* konstan tidak berubah nilainya.

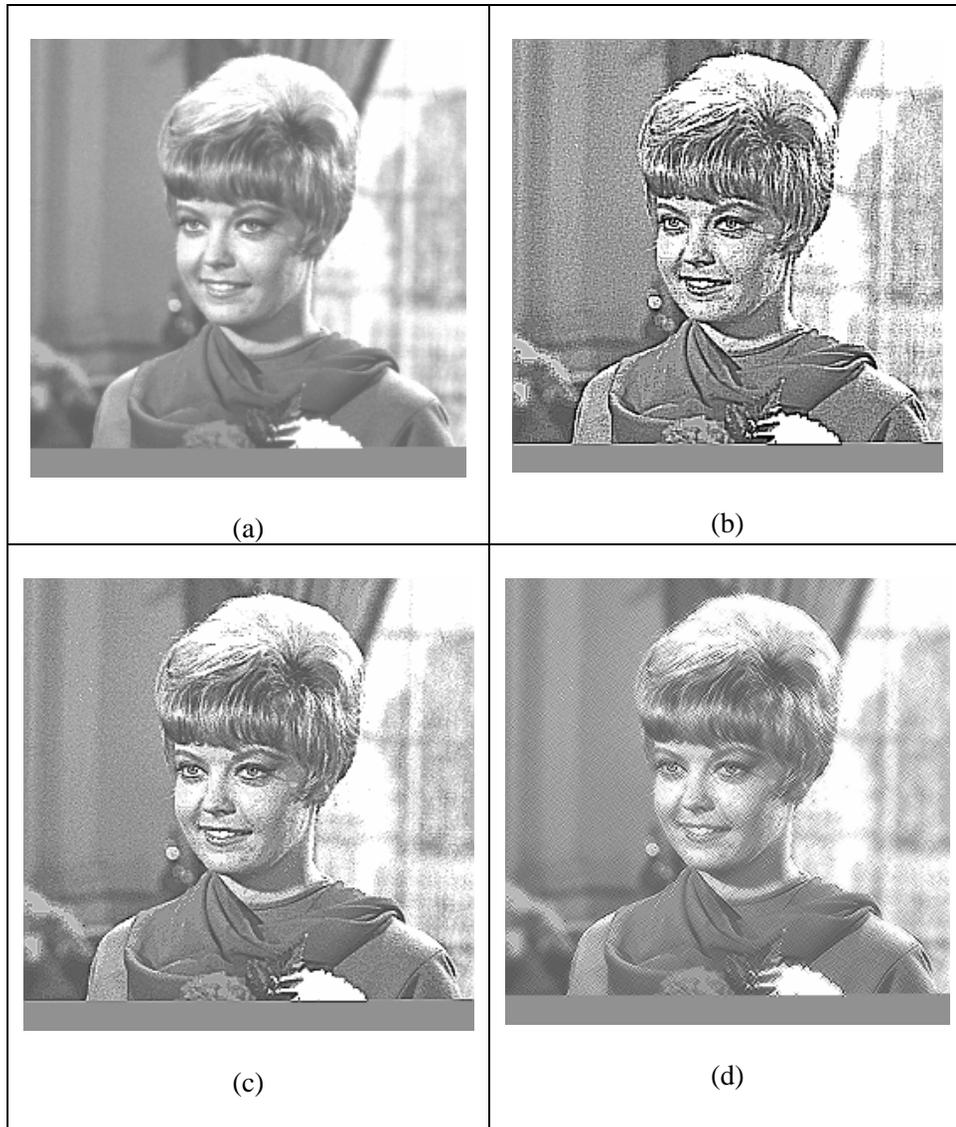
Gambar 7.15 memperlihatkan konvolusi dengan penapis lolos-tinggi, gambar (a) adalah citra yang tidak mempunyai *pixel* tepi, dan gambar (b) adalah citra yang mempunyai *pixel* tepi. Penapis lolos-tinggi yang digunakan adalah penapis (i) dan (ii).

Karena koefisien penapis mengandung nilai negatif, maka konvolusi mungkin saja menghasilkan *pixel* bernilai negatif. Meskipun intensitas bernilai negatif menarik, tetapi kita tidak dapat menampilkannya. Untuk alasan terakhir ini, implementasi konvolusi men-set nilai negatif menjadi nilai 0. Cara lainnya adalah dengan mengambil nilai mutlaknya atau menskalakan semua nilai-nilai *pixel* secara menaik sehingga nilai yang paling negatif menjadi 0.



Gambar 7.15 Hasil konvolusi dengan penapis lolos-tinggi: (a) citra yang tidak memiliki pixel tepi, (b) citra yang mengandung pixel-pixel tepi

Gambar 7.16 adalah contoh lain penajaman gambar terhadap citra *girl*, masing-masing dengan penapis (ii), (iii), dan (iv).



Gambar 7.16 (a) citra *girl* sebelum penajaman; (b), (c), dan (d) masing-masing adalah hasil penajaman dengan penapis lolos-tinggi (ii), (iii), dan (iv)

7.9 Pewarnaan Semu

Pewarnaan semu adalah proses memberi warna tertentu pada nilai-nilai *pixel* suatu citra skala-abu pada suatu citra berdasarkan kriteria tertentu, misalnya suatu warna tertentu untuk suatu interval derajat keabuan tertentu. Hal ini dilakukan karena mata manusia mudah membedakan banyak jenis warna.

7.10 Koreksi Geometrik

Koreksi geometrik dilakukan pada citra yang memiliki gangguan yang terjadi pada waktu proses perekaman citra, misalnya pergeseran koordinat citra (translasi), perubahan ukuran citra, dan perubahan orientasi koordinat citra (*skew*). Proses koreksi geometri untuk meningkatkan kualitas citra tersebut disebut juga koreksi geometri. Koreksi geometri yang sederhana adalah dengan operasi geometri sederhana seperti rotasi, translasi, dan penskalaan citra.

Gambar 7.17 kiri adalah citra kota San Fransisco yang condong (*skew*) ke kanan. Rotasi sejauh 6° berlawanan arah jarum jam menghasilkan perbaikan yang ditunjukkan pada Gambar 8.11 kanan.



(a)



(b)

Gambar 7.17 (a) Citra San Fransisco yang condong ke kanan; (b) Hasil rotasi sejauh 6° berlawanan arah jarum jam.