

## Pendeteksian Tepi (*Edge Detection*)

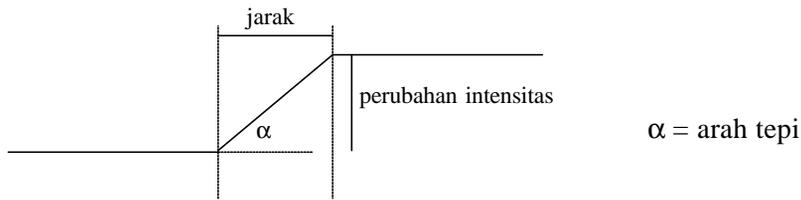
Peningkatan kualitas citra (*image enhancement*) bertujuan menghasilkan citra dengan kualitas yang lebih baik dibandingkan dengan citra semula. Langkah selanjutnya dalam pengolahan citra adalah **analisis citra** (*image analysis*). Analisis citra bertujuan mengidentifikasi parameter-parameter yang diasosiasikan dengan ciri (*feature*) dari objek di dalam citra, untuk selanjutnya parameter tersebut digunakan dalam menginterpretasi citra. Analisis citra pada dasarnya terdiri dari tiga tahapan: ekstraksi ciri (*feature extraction*), segmentasi, dan klasifikasi.

Faktor kunci dalam mengekstraksi ciri adalah kemampuan mendeteksi keberadaan tepi (*edge*) dari objek di dalam citra. Setelah tepi objek diketahui, langkah selanjutnya dalam analisis citra adalah segmentasi, yaitu mereduksi citra menjadi objek atau region, misalnya memisahkan objek-objek yang berbeda dengan mengekstraksi batas-batas objek (*boundary*). Langkah terakhir dari analisis citra adalah klasifikasi, yaitu memetakan segmen-segmen yang berbeda ke dalam kelas objek yang berbeda pula.

### 8.1 Definisi Tepi

---

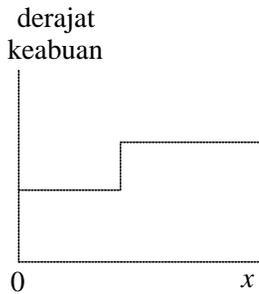
Yang dimaksud dengan tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat (Gambar 8.1). Perbedaan intensitas inilah yang menampakkan rincian pada gambar. Tepi biasanya terdapat pada batas antara dua daerah berbeda pada suatu citra. Tepi dapat diorientasikan dengan suatu arah, dan arah ini berbeda-beda pada bergantung pada perubahan intensitas.



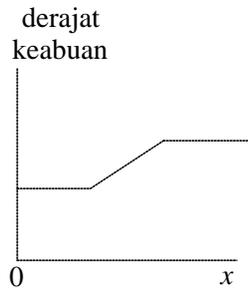
**Gambar 8.1** Model tepi satu-matra

Perhatikan Gambar 8.2. Ada tiga macam tepi yang terdapat di dalam citra digital. Ketiganya adalah:

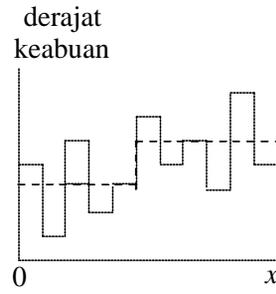
1. Tepi curam  
Tepi dengan perubahan intensitas yang tajam. Arah tepi berkisar  $90^\circ$ .
2. Tepi landai  
Disebut juga tepi lebar, yaitu tepi dengan sudut arah yang kecil. Tepi landai dapat dianggap terdiri dari sejumlah tepi-tepi lokal yang lokasinya berdekatan.
3. Tepi yang mengandung derau (*noise*)  
Umumnya tepi yang terdapat pada aplikasi *computer vision* mengandung derau. Operasi peningkatan kualitas citra (*image enhancement*) dapat dilakukan terlebih dahulu sebelum pendeteksian tepi.



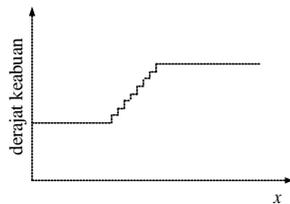
(a) Tepi curam



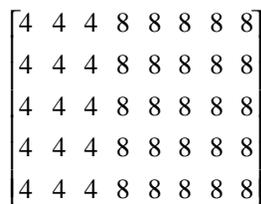
(b) tepi landai



(c) tepi curam dengan derau



(d) *break down* tepi landai



(e) citra dengan tepi curam



(f) citra dengan tepi landai

**Gambar 8.2** Jenis-jenis tepi

## 8.2 Tujuan Pendeteksian Tepi

---

Pendeteksian tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra.

Tujuan operasi pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Karena tepi termasuk ke dalam komponen berfrekuensi tinggi, maka pendeteksian tepi dapat dilakukan dengan penapis lolos-tinggi.

Terdapat beberapa teknik yang digunakan untuk mendeteksi tepi, antara lain:

1. Operator gradien pertama (*differential gradient*)
2. Operator turunan kedua (*Laplacian*)
3. Operator kompas (*compass operator*)

## 8.3 Pendeteksian Tepi dengan Operator Gradien Pertama

---

Perubahan intensitas yang besar dalam jarak yang singkat dipandang sebagai fungsi yang memiliki kemiringan yang besar. Kemiringan fungsi biasanya dilakukan dengan menghitung turunan pertama (*gradient*). Karena citra  $f(x,y)$  adalah fungsi dwimatra dalam bentuk diskrit, maka turunan pertamanya adalah secara parsial, masing-masing dalam arah- $x$  dan dalam arah- $y$ , sebagai berikut:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \quad (8.1)$$

yang dalam hal ini,

$$G_x = \frac{\partial f(x, y)}{\partial x} = \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \quad (8.2)$$

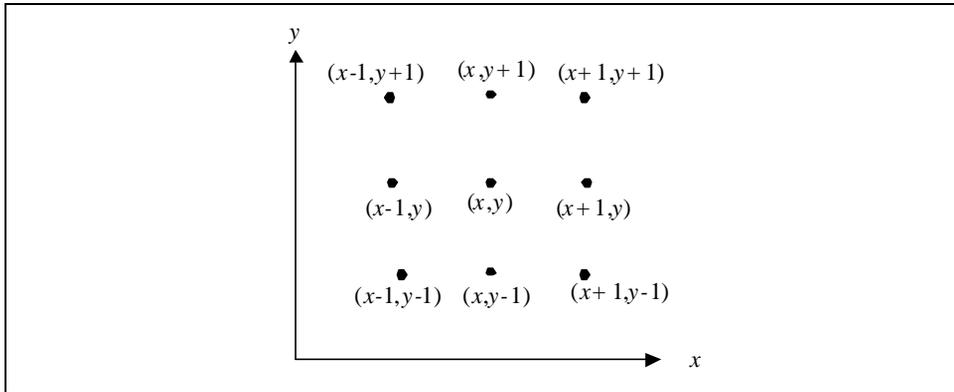
$$G_y = \frac{\partial f(x, y)}{\partial y} = \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \quad (8.3)$$

Biasanya  $\Delta x = \Delta y = 1$ , sehingga persamaan turunan pertama menjadi:

$$G_x = \frac{\partial f(x, y)}{\partial x} = f(x + 1, y) - f(x, y) \quad (8.4)$$

$$G_y = \frac{\partial f(x, y)}{\partial y} = f(x, y + 1) - f(x, y) \quad (8.5)$$

Titik-titik yang terlibat dalam perhitungan turunan pertama diperlihatkan pada Gambar 8.3.



**Gambar 8.3** Titik-titik yang dilibatkan dalam perhitungan gradien

Kedua turunan tersebut dapat dipandang sebagai dua buah *mask* konvolusi sebagai berikut:

$$G_1(x) = [-1 \ 1] \quad \text{dan} \quad G_1(y) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

**Contoh 8.1.** [LOW91] Misalkan terdapat sebuah  $5 \times 5$  citra dengan dua derajat keabuan sebagai berikut:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Hasil perhitungan gradien setiap *pixel* di dalam citra adalah sebagai berikut:

Citra	Gradien-x	Gradien-y	Arah gradien
$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 & * \\ 0 & 0 & -1 & 0 & * \\ 0 & -1 & 0 & 0 & * \\ 0 & -1 & 0 & 0 & * \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & * \end{bmatrix}$	$\begin{bmatrix} * & * & * & * & * \\ * & * & * & \updownarrow & \updownarrow \\ * & * & \diagdown & * & * \\ * & \leftrightarrow & * & * & * \\ * & \leftrightarrow & * & * & * \end{bmatrix}$

Berdasarkan konvolusi dengan kedua *mask* tersebut, kita menghitung kekuatan tepi,  $G[f(x,y)]$ , yang merupakan magnitudo dari gradien, dan arah tepi,  $\alpha(x,y)$ , untuk setiap *pixel*:

$$G[f(x,y)] = \sqrt{G_x^2 + G_y^2} \quad (8.6)$$

$$\alpha(x,y) = \tan^{-1} \frac{G_y}{G_x} \quad (8.7)$$

Karena menghitung akar adalah persoalan rumit dan menghasilkan nilai riil, maka dalam praktek kekuatan tepi biasanya disederhanakan perhitungannya dengan menggunakan salah satu dari alternatif rumus berikut [DUL97]:

- (i)  $G[f(x,y)] = |G_x^2| + |G_y^2|$ , atau
- (ii)  $G[f(x,y)] = |G_x| + |G_y|$ , atau
- (iii)  $G[f(x,y)] = \max\{|G_x^2|, |G_y^2|\}$ , atau
- (iv)  $G[f(x,y)] = \max\{|G_x|, |G_y|\}$ .

Dalam praktek, persamaan (ii) dan (iv) biasanya lebih disukai dan lebih mudah dikerjakan karena mengandung jumlah operasi aritmetika yang lebih sedikit.

Hasil pendeteksian tepi adalah **citra tepi** (*edges image*)  $g(x,y)$ , yang nilai setiap *pixel*-nya menyatakan kekuatan tepi:

$$g(x,y) = G[f(x,y)]$$

Keputusan apakah suatu *pixel* merupakan tepi atau bukan tepi dinyatakan dengan operasi pengambangan berikut:

$$g(x,y) = \begin{cases} 1, & \text{jika } G[f(x,y)] \geq T \\ 0, & \text{lainnya} \end{cases} \quad (8.8)$$

yang dalam hal ini  $T$  adalah nilai ambang, *pixel* tepi dinyatakan putih sedangkan *pixel* bukan tepi dinyatakan hitam. Gambar 8.4 adalah contoh hasil deteksi semua tepi citra Lena, citra *Camera*, dan citra botol.



**Gambar 8.4** Deteksi semua tepi citra Lena, camera, dan botol

Selain operator gradien yang sudah disebutkan di atas, masih ada beberapa operator gradien pertama yang lain yang dapat digunakan untuk mendeteksi tepi di dalam citra, yaitu:

(a) Operator gradien selisih-terpusat (*center-difference*):

$$D_x(x, y) = \frac{\partial f(x, y)}{\partial x} = \frac{f(x+1, y) - f(x-1, y)}{2} \quad (8.9)$$

$$D_y(x, y) = \frac{\partial f(x, y)}{\partial y} = \frac{f(x, y+1) - f(x, y-1)}{2} \quad (8.10)$$

yang ekuivalen dengan *mask* berikut:

$$D_2(x) = [-1 \quad 0 \quad 1] \quad \text{dan} \quad D_2(y) = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

(b) Operator Sobel

Tinjau pengaturan *pixel* di sekitar *pixel*  $(x, y)$ :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x, y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Operator Sobel adalah magnitudo dari gradien yang dihitung dengan

$$M = \sqrt{s_x^2 + s_y^2}$$

yang dalam hal ini, turunan parsial dihitung dengan

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \quad (8.11)$$

$$s_y = (a_0 + ca_1 + a_{22}) - (a_6 + ca_5 + a_4) \quad (8.12)$$

dengan konstanta  $c = 2$ . Dalam bentuk *mask*,  $s_x$  dan  $s_y$  dapat dinyatakan sebagai

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Arah tepi dihitung dengan persamaan

$$\mathbf{a}(x,y) = \tan^{-1} \left( \frac{S_y}{S_x} \right) \quad (8.13)$$

**Contoh 8.2.** Contoh berikut ini memeperlihatkan pendeteksian tepi dengan operator Sobel. Konvolusi pertama dilakukan terhadap *pixel* yang bernilai 1 (di titik pusat *mask*):

$$\begin{bmatrix} 3 & 4 & 2 & 5 & 1 \\ 2 & 1 & 6 & 4 & 2 \\ 3 & 5 & 7 & 1 & 3 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 3 & 2 \end{bmatrix}$$

(i) citra semula

$$\begin{bmatrix} * & * & * & * & * \\ * & 18 & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}$$

(ii) hasil konvolusi

Nilai 18 pada citra hasil konvolusi diperoleh dengan perhitungan berikut:

$$S_x = (3)(-1) + (2)(-2) + (3)(-1) + (2)(1) + (6)(2) + (7)(1) = 11$$

$$S_y = (3)(1) + (4)(2) + (2)(1) + (3)(-1) + (5)(-2) + (7)(-1) = -7$$

$$M = \sqrt{s_x^2 + s_y^2} = \sqrt{11^2 + (-7)^2} \cong |S_x| + |S_y| = |11| + |-7| = 18$$

Pada contoh ini, nilai  $M = \sqrt{s_x^2 + s_y^2}$  dihampiri dengan menghitung

$$M \cong |S_x| + |S_y|.$$

■

**Contoh 8.3.** Di bawah ini contoh lain pendeteksian tepi dengan operator Sobel, dimana hasil konvolusi dibandingkan dengan  $T = 12$ .

Citra:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 4 & 3 \\ 0 & 0 & 2 & 0 & 2 & 4 & 3 & 3 & 2 & 3 \\ 0 & 0 & 1 & 3 & 3 & 4 & 3 & 3 & 3 & 3 \\ 0 & 1 & 0 & 4 & 3 & 3 & 2 & 4 & 3 & 2 \\ 0 & 0 & 1 & 2 & 3 & 3 & 4 & 4 & 4 & 3 \end{bmatrix}$$

$|\text{gradien } -x| + |\text{gradien } -y|$ :

$$\begin{bmatrix} * & * & * & * & * & * & * & * & * & * \\ * & 4 & 6 & 4 & 10 & 14 & 12 & 14 & 4 & * \\ * & 6 & 8 & 10 & 20 & 16 & 12 & 6 & 0 & * \\ * & 4 & 10 & 14 & 10 & 2 & 4 & 2 & 4 & * \\ * & 2 & 12 & 12 & 2 & 2 & 4 & 6 & 8 & * \\ * & * & * & * & * & * & * & * & * & * \end{bmatrix}$$

Hasil pengambangan dengan  $T = 12$ :

$$\begin{bmatrix} * & * & * & * & * & * & * & * & * & * \\ * & 4 & 6 & 4 & 10 & 14 & 12 & 14 & 4 & * \\ * & 6 & 8 & 10 & 20 & 16 & 12 & 6 & 0 & * \\ * & 4 & 10 & 14 & 10 & 2 & 4 & 2 & 4 & * \\ * & 2 & 12 & 12 & 2 & 2 & 4 & 6 & 8 & * \\ * & * & * & * & * & * & * & * & * & * \end{bmatrix}$$

■

(c) Operator Prewitt

Persamaan gradien pada operator Prewitt sama seperti operator Sobel, tetapi menggunakan nilai  $c = 1$ :

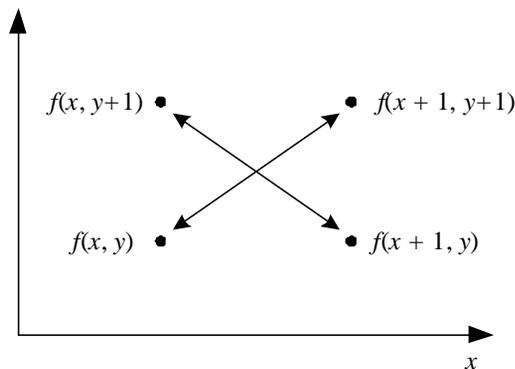
$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

(d) Operator Roberts

Operator Roberts sering disebut juga operator silang (gambar 8.5). Gradien Roberts dalam arah- $x$  dan arah- $y$  dihitung dengan rumus:

$$R_+(x, y) = f(x+1, y+1) - f(x, y) \quad (8.14)$$

$$R_-(x, y) = f(x, y+1) - f(x+1, y) \quad (8.15)$$



**Gambar 8.5** Operator silang

Operator  $R_+$  adalah hampiran turunan berarah dalam arah  $45^\circ$ , sedangkan  $R_-$  adalah hampiran turunan berarah dalam arah  $135^\circ$ .

Dalam bentuk *mask* konvolusi, operator Roberts adalah:

$$R_+ = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ dan } R_- = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Khusus untuk operator Roberts, arah tepi dihitung dengan rumus

$$\mathbf{a}(x, y) = \frac{\mathbf{p}}{4} + \tan^{-1}\left(\frac{R_-}{R_+}\right) \quad (8.17)$$

Sedangkan kekuatan tepi umumnya dihitung dengan rumus

$$G[f(x,y)] = |R_+| + |R_-|$$

**Contoh 8.4.** Contoh berikut ini memeperlihatkan pendeteksian tepi dengan operator Roberts.

$$\begin{bmatrix} 3 & 4 & 2 & 5 & 1 \\ 2 & 1 & 6 & 4 & 2 \\ 3 & 5 & 7 & 1 & 3 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 3 & 2 \end{bmatrix}$$

(i) citra semula

$$\begin{bmatrix} 4 & 3 & 3 & 6 & * \\ 5 & 7 & 8 & 2 & * \\ 2 & 5 & 4 & 4 & * \\ 1 & 1 & 8 & 7 & * \\ * & * & * & * & * \end{bmatrix}$$

(ii) hasil konvolusi

Nilai 4 pada pojok kiri atas pada citra hasil konvolusi diperoleh dengan perhitungan sebagai berikut:

$$f'[0,0] = |3 - 1| + |4 - 2| = 4$$

## 8.4 Pendeteksian Tepi dengan Operator Turunan Kedua

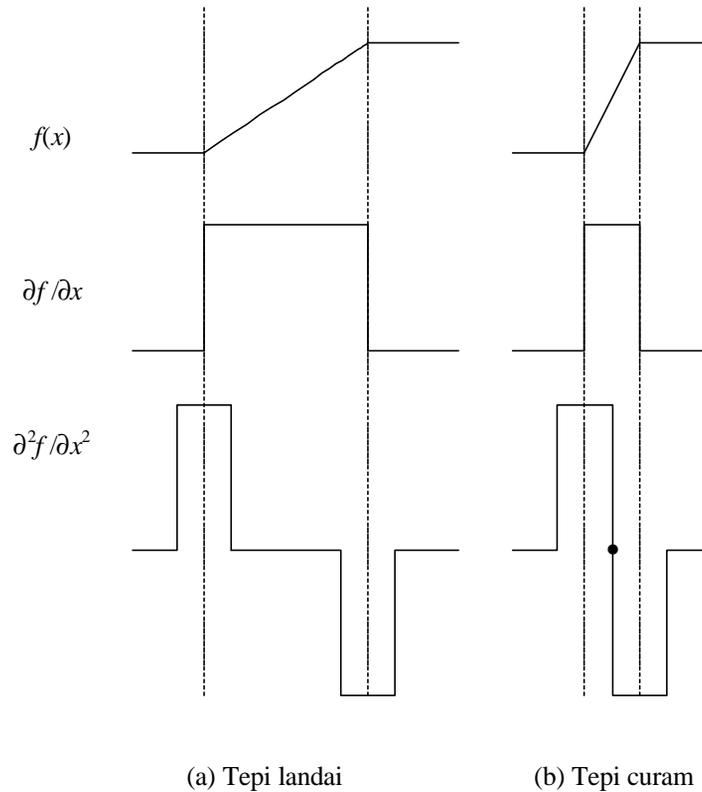
---

Operator turunan kedua disebut juga operator Laplace. Operator Laplace mendeteksi lokasi tepi lebih akurat khususnya pada tepi yang curam. Pada tepi yang curam, turunan keduanya mempunyai persilangan nol (*zero-crossing*), yaitu titik di mana terdapat pergantian tanda nilai turunan kedua (Gambar 8.6),

sedangkan pada tepi yang landai tidak terdapat persilangan nol. Persilangan nol merupakan lokasi tepi yang akurat.

Turunan kedua fungsi dengan dua peubah adalah:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (8.18)$$



**Gambar 8.6** Deteksi tepi dengan operator turunan kedua

Dengan menggunakan definisi hampiran selisih-mundur (*backward difference approximation*):

$$G_3(x) = \frac{\partial f(x, y)}{\partial x} = \frac{f(x, y) - f(x - \Delta x, y)}{\Delta x} \quad (8.19)$$

$$G_3(y) = \frac{\partial f(x, y)}{\partial y} = \frac{f(x, y) - f(x, y - \Delta y)}{\Delta y} \quad (8.20)$$

maka

$$\begin{aligned}
 \nabla^2 f &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\
 &= G_1(G_3(x)) + G_1(G_3(y)) \\
 &= \frac{1}{\Delta x} G_1(f(x, y)) - G_1(f(x - \Delta x, y)) + \frac{1}{\Delta y} G_1(f(x, y)) - G_1(f(x, y - \Delta y)) \\
 &= \frac{1}{\Delta x} \left\{ \frac{f(x + \Delta x, y) - f(x, y) - f(x, y) + f(x - \Delta x, y)}{\Delta x} \right\} \\
 &\quad + \frac{1}{\Delta y} \left\{ \frac{f(x, y + \Delta y) - f(x, y) - f(x, y) + f(x, y - \Delta y)}{\Delta y} \right\} \\
 &= \frac{f(x + \Delta x, y) - 2f(x, y) + f(x - \Delta x, y)}{(\Delta x)^2} \\
 &\quad + \frac{f(x, y + \Delta y) - 2f(x, y) + f(x, y - \Delta y)}{(\Delta y)^2}
 \end{aligned}$$

Dengan mengasumsikan  $\Delta x = \Delta y = 1$ , maka diperoleh:

$$\begin{aligned}
 \nabla^2 f(x, y) &= f(x + 1, y) - 2f(x, y) + f(x - 1, y) + f(x, y + 1) - 2f(x, y) + f(x, y - 1) \\
 &= f(x, y - 1) + f(x - 1, y) - 4f(x, y) + f(x + 1, y) + f(x, y + 1)
 \end{aligned} \tag{8.21}$$

atau dapat dinyatakan sebagai *mask*:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Selain *mask* di atas, masih ada dua hampiran operator Laplace yang lain, yaitu

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \text{ dan } \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Kadang-kadang diinginkan memberi bobot yang lebih pada *pixel* tengah di antara *pixel* tetangganya. Operator Laplace yang digunakan untuk tujuan ini adalah

$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

Operator Laplace termasuk ke dalam penapis lolos-tinggi sebab jumlah seluruh koefisiennya nol dan koefisiennya mengandung nilai negatif maupun positif.

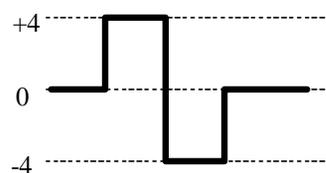
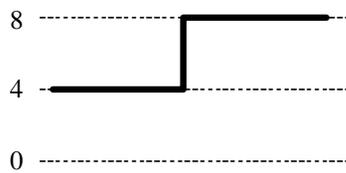
**Contoh 8.5.** [GAL90] Contoh berikut ini memperlihatkan pendeteksian tepi vertikal dengan operator Laplace:

$$\begin{bmatrix} 4 & 4 & 4 & | & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & | & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & | & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & | & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & | & 8 & 8 & 8 & 8 \end{bmatrix}$$

(i) Citra semula

$$\begin{bmatrix} * & * & * & | & * & * & * & * \\ * & 0 & +4 & | & -4 & 0 & 0 & * \\ * & 0 & +4 & | & -4 & 0 & 0 & * \\ * & 0 & +4 & | & -4 & 0 & 0 & * \\ * & * & * & | & * & * & * & * \end{bmatrix}$$

(ii) Hasil konvolusi



Satu baris dari hasil pendeteksian tepi:

$$0 \quad +4 \quad | \quad -4 \quad 0 \quad 0$$

Pada contoh di atas, persilangan nol bersesuaian dengan tepi pada citra semula, yang terdapat pada titik tengah antara dua buah *pixel* yang bersesuaian. *Pixel* tepi seharusnya ditandai secara konsisten, apakah *pixel* di sebelah kiri atau di sebelah kanan garis “|”. ■

**Contoh 8.6.** [GAL90] Pendeteksian tepi diagonal (miring) dengan operator Laplace:

$$\begin{bmatrix} 4 & 4 & 8 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 4 & 8 & 8 & 8 \\ 4 & 4 & 4 & 4 & 4 & 8 & 8 \\ 4 & 4 & 4 & 4 & 4 & 4 & 8 \end{bmatrix}$$

(i) Citra semula

$$\begin{bmatrix} * & * & * & * & * & * & * & * \\ * & 0 & +8 & -4 & 0 & 0 & 0 & * \\ * & 0 & 0 & +8 & -4 & 0 & 0 & * \\ * & 0 & 0 & 0 & +8 & -4 & 0 & * \\ * & * & * & * & * & * & * & * \end{bmatrix}$$

(ii) Hasil konvolusi

■

**Contoh 8.7.** Pendeteksian tepi landai dengan operator Laplace:

$$\begin{bmatrix} 2 & 2 & 2 & 5 & 8 & 8 & 8 & 8 \\ 2 & 2 & 2 & 5 & 8 & 8 & 8 & 8 \\ 2 & 2 & 2 & 5 & 8 & 8 & 8 & 8 \\ 2 & 2 & 2 & 5 & 8 & 8 & 8 & 8 \\ 2 & 2 & 2 & 5 & 8 & 8 & 8 & 8 \end{bmatrix}$$

(i) Citra semula

$$\begin{bmatrix} * & * & * & * & * & * & * & * \\ * & 0 & +3 & 0 & -3 & 0 & 0 & * \\ * & 0 & +3 & 0 & -3 & 0 & 0 & * \\ * & 0 & +3 & 0 & -3 & 0 & 0 & * \\ * & * & * & * & * & * & * & * \end{bmatrix}$$

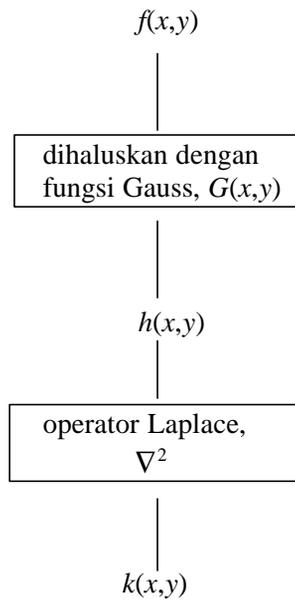
(ii) Hasil konvolusi

Satu baris dari hasil pendeteksian tepi:

$$0 \quad +3 \quad 0 \quad -3 \quad 0$$

Pada contoh di atas tidak terdapat persilangan nol; lokasi tepi yang sesungguhnya ditentukan secara interpolasi. ■

Kadangkala pendeteksian tepi dengan operator Laplace menghasilkan tepi-tepi palsu yang disebabkan oleh gangguan pada gambar [DUL97]. Untuk mengurangi kemunculan tepi palsu, citra disaring dulu dengan fungsi Gaussian (Gambar 8.7).



**Gambar 8.7** Skema pendeteksian tepi untuk citra yang mengalami gangguan.

Berdasarkan skema pada Gambar 8.7:

$$k(x, y) = \nabla^2 h(x, y) \quad (8.22)$$

dan

$$h(x, y) = f(x, y) * G(x, y) \quad (8.23)$$

maka dapat dibuktikan bahwa

$$\nabla^2 [f(x, y) * G(x, y)] = f(x, y) * \nabla^2 G(x, y) \quad (8.24)$$

Jadi,

$$k(x, y) = f(x, y) * \nabla^2 G(x, y) \quad (8.25)$$

yang dalam hal ini,

$$\nabla^2 G(x, y) = \left( \frac{x^2 + y^2 - 2s^2}{s^4} \right) e^{-\frac{(x^2 + y^2)}{2s^2}} \quad (8.26)$$

Fungsi  $\nabla^2 G(x,y)$  merupakan turunan kedua dari fungsi Gauss, kadang-kadang disebut juga fungsi *Laplacian of Gaussian (LoG)* atau fungsi topi orang Mexico (*Mexican Hat*), karena bentuk kurvanya seperti topi Meksiko.

Jadi, untuk mendeteksi tepi dari citra yang mengalami gangguan, kita dapat melakukan salah satu dari dua operasi ekuivalen di bawah ini:

1. Konvolusi citra dengan fungsi Gauss  $G(x,y)$ , kemudian lakukan operasi Laplacian terhadap hasilnya, atau
2. Konvolusi citra dengan penapis *LoG*.

Contoh penapis *LoG* yang berukuran  $5 \times 5$ :

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

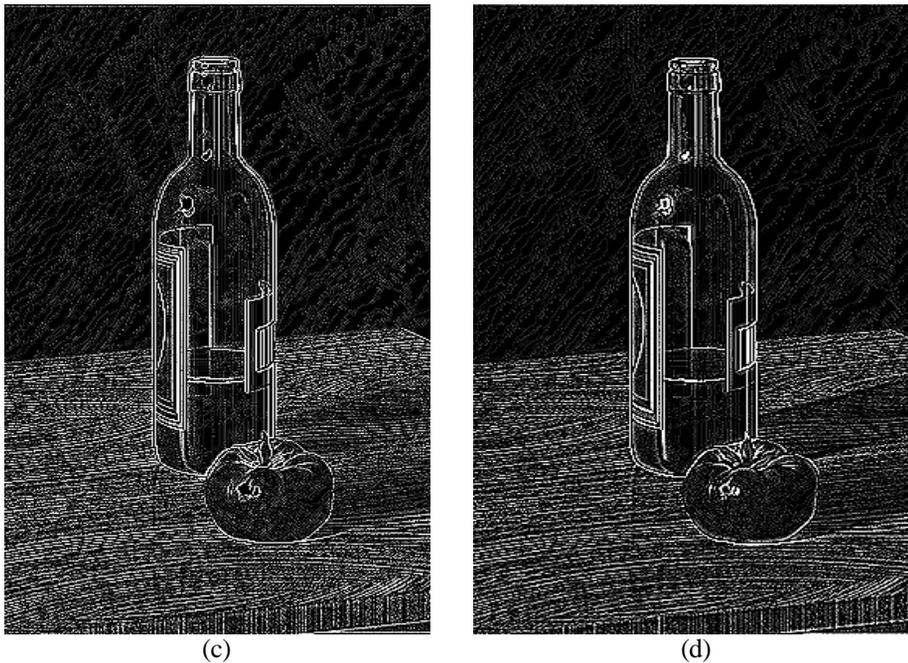
Hasil pendeteksian tepi dengan operator Laplace dan *Laplacian of Gaussian* diperlihatkan pada Gambar 8.8.



(a)



(b)



**Gambar 8.8** (a) citra botol; (b) Laplace; (c) Laplace dengan bobot lebih; (d) Laplacian of Gaussian (LoG)

## 8.5 Pendeteksian Tepi dengan Operator Kompas

Operator kompas (*compass operator*) digunakan untuk mendeteksi semua tepi dari berbagai arah di dalam citra. Operator kompas yang dipakai untuk pendeteksian tepi menampilkan tepi dari 8 macam arah mata angin: Utara, Timur Laut, Timur, Tenggara, Selatan, Barat Daya, dan Barat Laut. Pendeteksian tepi dilakukan dengan mengkonvolusikan citra dengan berbagai *mask* kompas, lalu dicari nilai kekuatan tepi (*magnitude*) yang terbesar dan arahnya. Jika misalnya digunakan sebanyak  $p$  buah *mask* kompas dan nilai kekuatan tepi pada *pixel*  $(x, y)$  untuk semua *mask* adalah  $G_1[f(x,y)]$ ,  $G_2[f(x,y)]$ , ...,  $G_p[f(x,y)]$ , maka besar kekuatan tepi adalah:

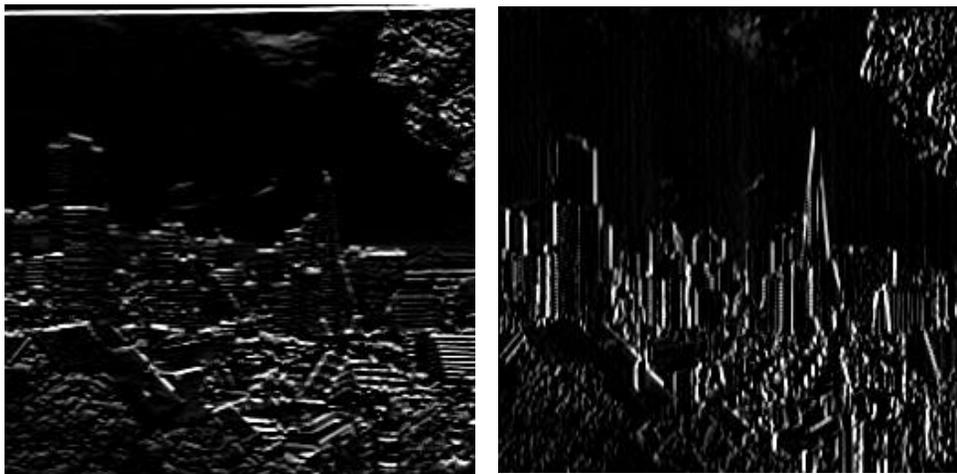
$$G[f(x, y)] = \max_i \{G_i[f(x, y)] \mid i = 1, 2, \dots, p\} \quad (8.27)$$

Jika *mask*  $k$  adalah *mask* yang memberikan kekuatan terbesar, maka arah tepi ditentukan dari *mask*  $k$  tersebut.

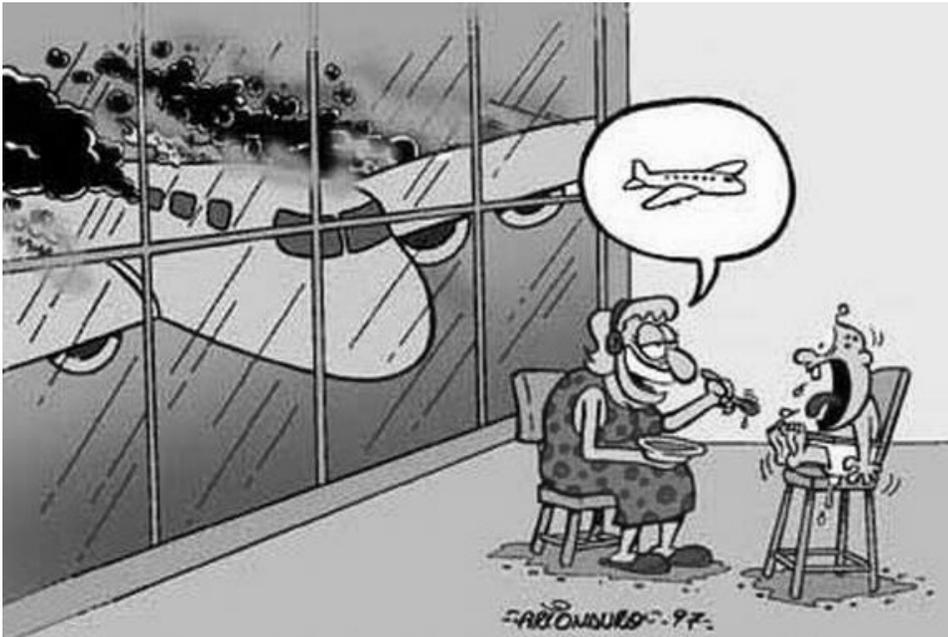
Operator kompas yang dipakai untuk pendeteksian tepi menampilkan tepi dari 8 macam arah mata angin:

Utara $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	Timur Laut $\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	Timur $\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	Tenggara $\begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
Selatan $\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	Barat Daya $\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	Barat $\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	Barat Laut $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$

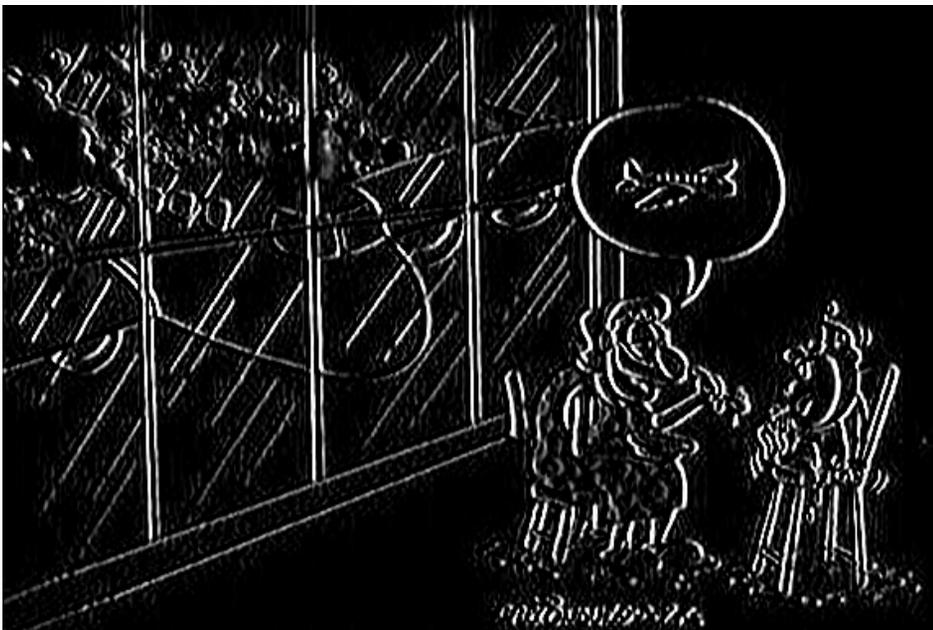
Operator kompas yang tersedia dapat juga digunakan untuk mendeteksi tepi dalam arah tertentu saja. Misalnya diinginkan mendeteksi tepi dalam arah horizontal dan vertikal, seperti hasil pendeteksian tepi citra San Fransisco (Gambar 8.9) dan citra WTC1109.



**Gambar 8.9** Deteksi tepi horizontal dan vertikal dari citra San Fransisco



Gambar 8.10. Citra WTC1109



Tepi vertikal



Tepi horizontal

*Gambar 8.10 (lanjutan).*