

# PENERAPAN METODE KONVOLUSI DALAM PENGOLAHAN CITRA DIGITAL

**Wikaria Gazali<sup>1</sup>; Haryono Soeparno<sup>2</sup>; Jenny Ohliati<sup>3</sup>**

<sup>1,3</sup> Mathematics & Statistics Department, School of Computer Science, Binus University

<sup>2</sup> Computer Science Department, School of Computer Science, Binus University

Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480

wikaria@binus.edu; haryono@binus.edu

## ABSTRACT

*A lot of photography studios today still perform image processing manually using installed the functions on the computer. As we know, manual image processing takes quite a long time. Therefore, they need an application program which is easier, user friendly and produces higher quality images. The application program created in this research uses metode konvolusi with several variations of image processing such as: smooth, Gaussian blur, sharpen, mean removal, emboss, and edge detection. The results of testing and evaluation of the program shows that image processing using the metode konvolusi is capable of producing better image quality.*

**Keywords:** *smooth, Gaussian blur, sharpen, mean removal, emboss, edge detection.*

## ABSTRAK

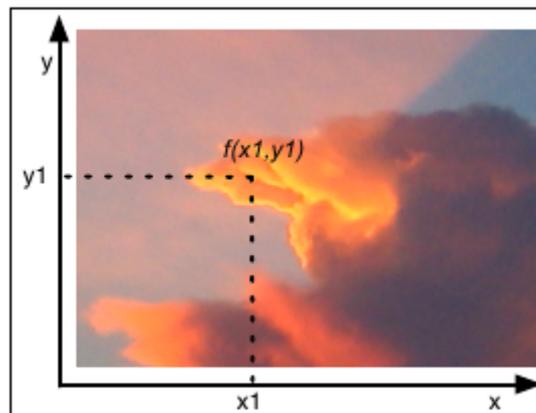
*Studio fotografi saat ini masih melakukan pengolahan citra secara manual menggunakan fungsi-fungsi yang sudah terpasang pada komputer. Pengolahan citra secara manual memakan waktu yang cukup lama. Oleh karena itu diperlukan program yang lebih mudah, user friendly dalam penggunaannya dan menghasilkan gambar yang lebih berkualitas. Metode yang digunakan dalam program aplikasi ini adalah metode konvolusi. Beberapa variasi pengolahan citra yang menggunakan metode ini yaitu: smooth, Gaussian blur, sharpen, mean removal, emboss, dan edge detection. Hasil uji coba dan evaluasi dari program tersebut menyatakan bahwa pengolahan citra menggunakan metode konvolusi berdasarkan data-data yang ada mampu menghasilkan kualitas citra yang lebih baik.*

**Kata kunci:** *pengolahan citra, smooth, Gaussian blur, sharpen, mean removal, emboss, edge detection*

## PENDAHULUAN

Citra atau gambar dalam bahasa latin *imago* adalah suatu representasi, kemiripan, atau imitasi dari suatu obyek atau benda. Citra dapat dikelompokkan menjadi citra tampak dan citra tidak tampak. Contoh citra tampak dalam kehidupan sehari-hari: foto, gambar, dan lukisan, sedangkan citra tidak tampak misalnya: data gambar dalam *file* (citra digital), dan citra yang direpresentasikan menjadi fungsi matematis. Di antara jenis citra tersebut, hanya citra digital yang dapat diolah menggunakan komputer. Jenis citra lain, jika hendak diolah dengan komputer, harus diubah dulu menjadi citra digital, misalnya foto *discan* dengan *scanner*, persebaran panas tubuh foto ditangkap dengan kamera infra merah dan diubah menjadi informasi numeris, informasi densitas dan komposisi bagian dalam tubuh manusia ditangkap dengan bantuan pesawat sinar- $x$  dan sistem deteksi radiasi menjadi informasi digital. Kegiatan untuk mengubah informasi citra fisik non digital menjadi digital disebut sebagai pencitraan (*imaging*).

Citra digital dapat didefinisikan sebagai fungsi dua variable,  $f(x,y)$ , di mana  $x$  dan  $y$  adalah koordinat spasial dan nilai  $f(x,y)$  adalah intensitas citra pada koordinat tersebut, hal tersebut diilustrasikan pada Gambar 1. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue* - RGB).



Gambar 1 Citra digital

Sebuah citra diubah ke bentuk digital agar dapat disimpan dalam memori komputer atau media lain. Proses mengubah citra ke bentuk digital bisa dilakukan dengan beberapa perangkat, misalnya *scanner*, *digital camera*, dan *handycam*. Ketika sebuah citra diubah ke dalam bentuk digital, bermacam-macam proses pengolahan citra dapat diperlakukan terhadap citra tersebut.

Pengolahan citra merupakan sebuah bentuk pemrosesan sebuah citra atau gambar dengan proses numerik dari gambar tersebut, dalam hal ini yang diproses adalah masing-masing pixel atau titik dari gambar tersebut. Salah satu teknik pemrosesan citra memanfaatkan komputer sebagai peranti lunak memproses masing-masing pixel dari sebuah gambar. Oleh karena itu muncul istilah pemrosesan citra secara digital atau *digital image processing*. *Digital image processing* diperkenalkan pertama kali di New York, USA pada awal tahun 1920-an. Pertama kalinya digunakan untuk meningkatkan kualitas gambar koran yang dikirimkan oleh kabel bawah laut yang terbentang antara London dan New York. Sampai tahun 1960-an perkembangannya tidaklah terlalu menggembirakan.

Namun pada akhir tahun 1960-an, di mana perkembangan komputer yang pesat dan mampu menawarkan kecepatan dan kapasitas yang lebih tinggi memacu perkembangan dari implementasi algoritma pemrosesan citra yang lebih pesat lagi. Untuk saat ini penggunaan dari pemrosesan citra telah melingkupi berbagai macam disiplin ilmu di antaranya bidang Arsitektur, Geografi, Ilmu Komputer, Kedokteran, Fotografi, Arkeologi, dan lain sebagainya.

Proses pengambilan citra menggunakan kamera terkadang mengalami penurunan mutu, misalnya mengandung derau (*noise*), kurang tajamnya, terjadi pengaburan (*blur*), dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasikan karena informasi yang disampaikan oleh citra tersebut berkurang. Pihak dari cuci cetak sendiri juga dapat melakukan pengolahan citra dengan piranti lunak dari komputer, tetapi biasanya cukup lama untuk menghasilkan citra yang diminta atau untuk bisa diinterpretasikan. Hal ini disebabkan proses yang dilakukan secara manual dengan penambahan warna dan penghalusan citra melalui piranti lunak komputer. Akan lebih cepat untuk pihak cuci cetak, apabila telah ada perangkat lunak yang secara otomatis menangani hal tersebut. Atas dasar itu, artikel ini membahas perancangan program untuk mengolah citra menggunakan Metode konvolusi, di mana terdiri dari penghalusan citra, Gaussian *blur*, mean removal, penajaman citra, emboss, dan edge detection.

## METODE

Artikel ini membahas perancangan program untuk mengolah citra menggunakan Metode konvolusi. Konvolusi adalah operator matematika yang penting untuk banyak operator dalam *image processing*. Konvolusi menyediakan cara untuk menggabungkan dua *array*, biasanya untuk ukuran *array* yang berbeda, tetapi untuk dimensi *array* yang sama, menghasilkan *array* ketiga yang mempunyai dimensi yang sama. Konvolusi dapat digunakan dalam *image processing* untuk menerapkan operator yang mempunyai nilai *output* dari piksel yang berasal dari kombinasi linear nilai *input* piksel tertentu.

Konvolusi citra adalah tehnik untuk menghaluskan suatu citra atau memperjelas citra dengan menggantikan nilai piksel dengan sejumlah nilai piksel yang sesuai atau berdekatan dengan piksel aslinya. Tetapi dengan adanya konvolusi, ukuran dari citra tetap sama, tidak berubah.

Konvolusi memiliki dua buah fungsi  $f(x)$  dan  $g(x)$  yang didefinisikan sebagai berikut:

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(a) \cdot g(x - a) da \quad (1)$$

yang dalam hal ini, tanda (\*) menyatakan operator konvolusi dan peubah (*variable*)  $a$  adalah peubah bantu.

Untuk pengolahan citra, operasi yang dilakukan adalah diskrit karena nilai koordinat piksel merupakan nilai yang diskrit. Selanjutnya *filter* atau *mask* yang digunakan pada pengolahan citra biasanya berukuran terbatas, dalam artian bobot atau pengaruh dari titik-titik yang cukup jauh sudah tidak signifikan, sehingga dapat diabaikan (dianggap nol).

Bentuk diskrit dari operasi konvolusi satu dimensi pada pengolahan citra adalah:

$$h(x) = f(x) * g(x) = \sum_{a=-\infty}^{\infty} f(a) \cdot g(x - a) \quad (2)$$

Untuk fungsi dengan dua dimensi, operasi konvolusi didefinisikan sebagai berikut:

Untuk fungsi integral:

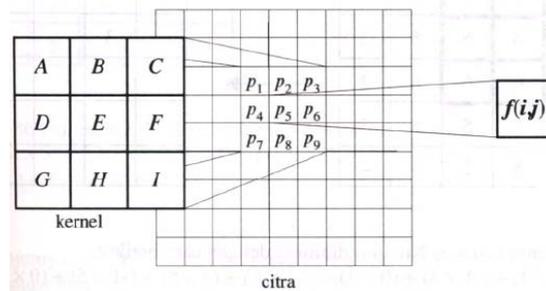
$$h(x, y) = f(x, y) * g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b) \cdot g(x - a, y - b) da db \quad (3)$$

Untuk fungsi diskrit:

$$h(x, y) = f(x, y) * g(x, y) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f(a, b) \cdot g(x - a, y - b) \quad (4)$$

Fungsi penapis  $g(x,y)$  disebut juga konvolusi *filter*, konvolusi *mask*, konvolusi *kernel*, atau *template*. Dalam bentuk diskrit konvolusi *kernel* dinyatakan dalam bentuk matriks (umumnya matriks 3x3). Ukuran matriks ini biasanya lebih kecil dari ukuran citra. Setiap elemen matriks disebut koefisien konvolusi.

Ilustrasi konvolusi ditunjukkan pada Gambar 2.



Gambar 2 Ilustrasi Konvolusi (Rinaldi Munir, 2004, p79).

$$f(i,j) = Ap_1 + Bp_2 + Cp_3 + Dp_4 + Ep_5 + Fp_6 + Gp_7 + Hp_8 + Ip_9$$

## HASIL DAN PEMBAHASAN

### Hasil Operasi Konvolusi

Operasi konvolusi dilakukan dengan menggeser konvolusi *kernel* piksel per piksel. Hasil konvolusi disimpan di dalam matriks baru.

Contoh 1. Misalkan citra  $f(x,y)$  yang berukuran 5x5 dan sebuah *kernel* atau *mask* yang berukuran 3x3 masing-masing adalah sebagai berikut:

$$f(x, y) = \begin{bmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 6 & 2 \\ 6 & 7 & 5 & 5 & 3 \\ 2 & 5 & 2 & 4 & 4 \end{bmatrix} \quad \text{dan} \quad g(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0.4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

(keterangan: tanda \* menyatakan posisi (0,0) dari *kernel*)

Operasi konvolusi antara citra  $f(x,y)$  dengan *kernel*  $g(x,y)$ , yaitu  $f(x,y) * g(x,y)$  dapat diilustrasikan sebagai berikut:

- (1) Tempatkan *kernel* pada sudut kiri atas, kemudian hitung nilai piksel pada posisi (0,0) dari *kernel*.

4	4	3	5	4
6	6	5	5	2
5	6	6	6	2
6	7	5	5	3
3	5	2	4	4

→

	3			

Hasil konvolusi = 3. Nilai ini dihitung dengan cara berikut:

$$(0 \times 4) + (-1 \times 4) + (0 \times 3) + (-1 \times 6) + (4 \times 6) + (-1 \times 5) + (0 \times 5) + (-1 \times 6) + (0 \times 6) = 3$$

- (2) Geser *kernel* satu piksel ke kanan, kemudian hitung nilai piksel pada posisi (0,0) dari *kernel*:

4	4	3	5	4
6	6	5	5	2
5	6	6	6	2
6	7	5	5	3
3	5	2	4	4

→

	3	0		

Hasil konvolusi = 0. Nilai ini dihitung dengan cara berikut:

$$(0 \times 4) + (-1 \times 3) + (0 \times 5) + (-1 \times 6) + (4 \times 5) + (-1 \times 5) + (0 \times 6) + (-1 \times 6) + (0 \times 6) = 0$$

- (3) Geser *kernel* satu piksel ke kanan, kemudian hitung nilai piksel pada posisi (0,0) dari *kernel*:

4	4	3	5	4
6	6	5	5	2
5	6	6	6	2
6	7	5	5	3
3	5	2	4	4

→

	3	0	2	

Hasil konvolusi = 2. Nilai ini dihitung dengan cara berikut:

$$(0 \times 3) + (-1 \times 5) + (0 \times 4) + (-1 \times 5) + (4 \times 5) + (-1 \times 2) + (0 \times 6) + (-1 \times 6) + (0 \times 2) = 2$$

- (4) Geser *kernel* satu piksel ke bawah, lalu mulai lagi melakukan konvolusi dari sisi kiri citra. Setiap kali konvolusi, geser *kernel* atau piksel ke kanan:

4	4	3	5	4
6	6	5	5	2
5	6	6	6	2
6	7	5	5	3
3	5	2	4	4

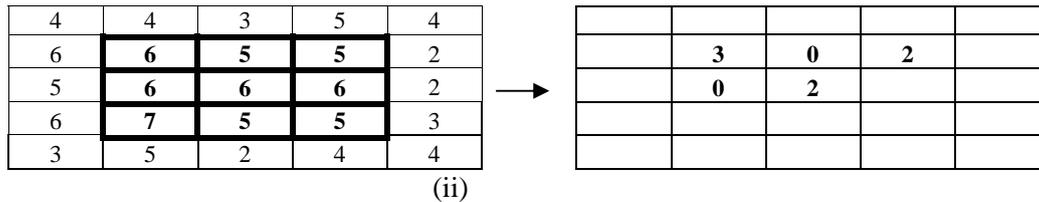
→

	3	0	2	
	0			

(i)

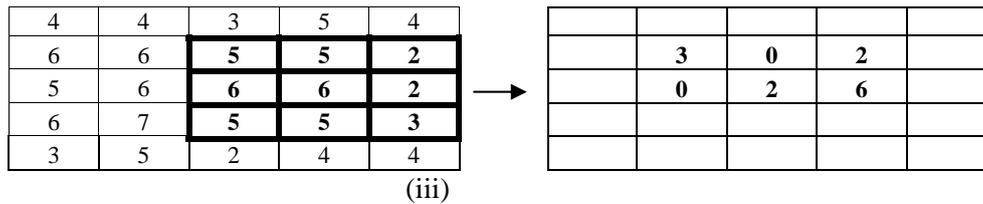
Hasil konvolusi = 0. Nilai ini dihitung dengan cara berikut:

$$(0 \times 6) + (-1 \times 6) + (0 \times 5) + (-1 \times 5) + (4 \times 6) + (-1 \times 6) + (0 \times 6) + (-1 \times 7) + (0 \times 5) = 0$$



Hasil konvolusi = 2. Nilai ini dihitung dengan cara berikut:

$$(0 \times 6) + (-1 \times 5) + (0 \times 5) + (-1 \times 6) + (4 \times 6) + (-1 \times 6) + (0 \times 7) + (-1 \times 5) + (0 \times 5) = 2$$



Hasil konvolusi = 6. Nilai ini dihitung dengan cara berikut:

$$(0 \times 5) + (-1 \times 5) + (0 \times 2) + (-1 \times 6) + (4 \times 6) + (-1 \times 2) + (0 \times 5) + (-1 \times 5) + (0 \times 3) = 6$$

Dengan cara yang sama, piksel-piksel pada baris ketiga di konvolusi sehingga menghasilkan:

	<b>3</b>	<b>0</b>	<b>2</b>	
	<b>0</b>	<b>2</b>	<b>6</b>	
	<b>6</b>	<b>0</b>	<b>2</b>	

Jika hasil konvolusi menghasilkan nilai piksel *negative*, nilai tersebut dijadikan 0. Sebaliknya jika hasil konvolusi menghasilkan nilai piksel lebih besar dari nilai keabuan maksimum (255), nilai tersebut dijadikan ke nilai keabuan maksimum.

Masalah timbul bila piksel yang dikonvolusi adalah piksel pinggir, karena beberapa koefisien konvolusi tidak dapat diposisikan pada piksel-piksel citra, seperti contoh di bawah ini:

4	4	3	<b>5</b>	<b>4</b>	<b>?</b>
6	6	5	<b>5</b>	<b>2</b>	<b>?</b>
5	6	6	<b>6</b>	<b>2</b>	<b>?</b>
6	7	5	5	3	
3	5	2	4	4	

Solusi untuk masalah ini adalah (Rinaldi Munir, 2004, p83): (1) piksel-piksel pinggir diabaikan, tidak dikonvolusi, jadi nilai piksel pinggir sama dengan nilai pada citra semula; (2) duplikasi elemen citra, misalnya elemen kolom pertama disalin ke kolom M+1 dst.; (3) elemen bertanda “?” diasumsikan bernilai 0 atau konstanta lain, sehingga konvolusi pinggir-pinggir dapat dilakukan.

Solusi dengan ketiga pendekatan di atas mengasumsikan bagian pinggir citra lebarnya sangat kecil (hanya satu piksel) *relative* dibandingkan dengan ukuran citra, sehingga piksel-piksel pinggir tidak memperlihatkan efek yang kasat mata.

4	4	3	5	4
6	<b>3</b>	<b>0</b>	<b>2</b>	2
5	<b>0</b>	<b>2</b>	<b>6</b>	2
6	<b>6</b>	<b>0</b>	<b>2</b>	3
3	5	2	4	4

Gambar 3 Piksel-piksel pinggir (yang tidak diarsir) tidak dikonvolusi.  
(Rinaldi Munir, 2004, p84).

Dalam algoritma konvolusi citra  $N \times M$  dengan *mask* atau *kernel* yang berukuran  $3 \times 3$  piksel yang dikonvolusi adalah elemen  $(i,j)$ . Delapan buah piksel yang bertetangga dengan piksel  $(i,j)$  diperlihatkan pada Gambar 4.

$i-1, j-1$	$i-1, j$	$i-1, j+1$
$i, j-1$	$i, j$	$i, j+1$
$i+1, j-1$	$i+1, j$	$i+1, j+1$

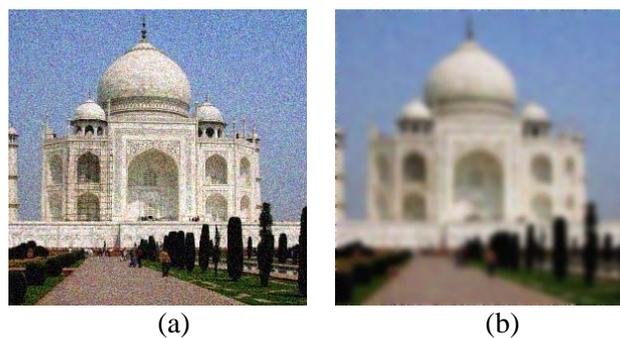
Gambar 4 Delapan piksel yang bertetangga dengan piksel  $(i,j)$ .

Dapat dilihat bahwa operasi konvolusi merupakan komputasi pada area lokal, karena komputasi untuk suatu piksel pada citra keluaran melibatkan piksel-piksel tetangga pada citra masukannya.

## Hasil Pengujian Konvolusi pada Citra

### Smooth

Berikut adalah perbandingan hasil pengolahan citra dengan menggunakan konvolusi *smooth* yaitu menghaluskan citra yang mengalami gangguan *noise* (Gambar 5).



Gambar 5 (a) Image asli yang mengalami *noise*; (b) Hasil konvolusi *smooth*.

Dapat dilihat dari gambar di atas bahwa citra asli *noise* berhasil dihaluskan dengan menggunakan konvolusi *smooth*. *Noise* pada citra dapat dihilangkan dan dihaluskan, tetapi, operasi *smooth* ini mempunyai efek pemerataan derajat keabuan, sehingga gambar yang diperoleh tampak lebih kabur kontrasnya. Efek pengaburan ini disebut efek *blurring*.

## Gaussian Blur

Pengolahan citra dengan menggunakan konvolusi *gaussian blur* menyebabkan suatu citra menjadi kabur sehingga sudut-sudut tajam pada citra akan menjadi lebih halus. Pengolahan citra ini dapat berdampak suatu citra menjadi semakin baik, tapi bisa juga menjadi semakin buruk. Hasil konvolusi Gaussian *blur* dapat dilihat pada Gambar 6 di bawah ini, di mana hasil yang diberikan citra terlihat lebih halus dari citra aslinya.



Gambar 6 (a) Citra asli; (b) citra yang telah mengalami *Gaussian Blur*

## Sharpen

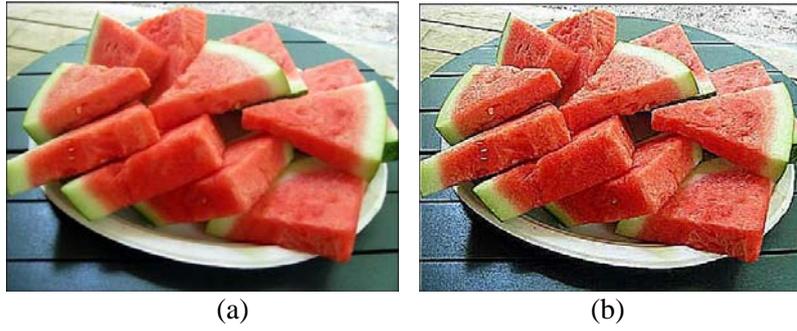
*File* gambar yang mengalami sharpen akan mengalami perubahan di mana warna-warna menjadi lebih tajam. Konvolusi *sharpen* sangat berguna untuk citra yang terlihat halus atau *blur* di mana berguna untuk memperjelas interpretasi citra itu sendiri dan hasilnya juga bisa nampak lebih baik dari citra sebelumnya. Hasil evaluasi citra yang telah diolah dengan konvolusi *sharpen* dapat dilihat pada Gambar 7. Dapat dilihat bahwa warna yang dihasilkan tampak lebih tajam dan terang.



Gambar 7 (a) Citra asli; (b) citra yang telah mengalami *sharpening*

## Mean Removal

Konvolusi *mean removal* memberikan ketajaman lebih pada citra. Konvolusi *mean removal* berbeda dengan konvolusi *sharpen* walaupun sama-sama mempertajam citra. Perbedaan itu terletak pada *mask* konvolusi yang digunakan. Ketajaman citra yang diberikan pada *mean removal* lebih tajam daripada ketajaman *sharpen*. Namun, *user* dapat menggunakan keduanya sesuai dengan kebutuhan yang diperlukan. Hasil evaluasi *mean removal* dapat dilihat pada Gambar 8. Citra yang dihasilkan terlihat lebih tajam dan nampak nyata. Pewarnaan yang dihasilkan lebih baik dari citra aslinya.



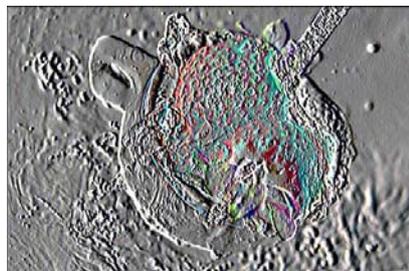
Gambar 8 (a) citra asli; (b) citra yang telah mengalami *mean removal*

### Emboss

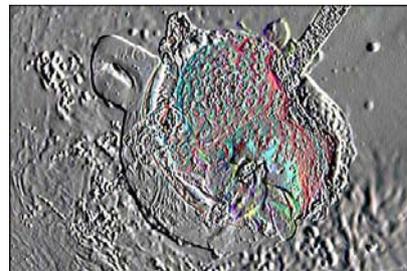
Hasil pengujian konvolusi *emboss* diberikan dalam Gambar 9, dengan arah kiri, kanan, atas, dan bawah.



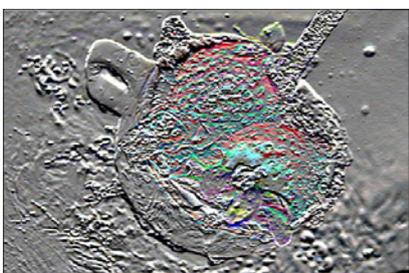
(a)



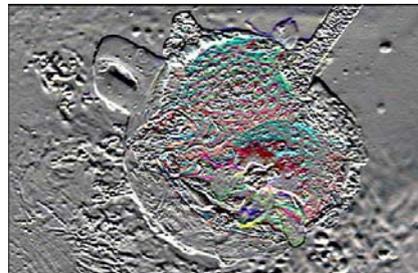
(b)



(c)



(d)



(e)

Gambar 9 (a) citra asli; (b) emboss dari arah kiri; (c) emboss dari arah kanan; (d) emboss dari arah atas; (e) emboss dari arah bawah.

## Edge Detection

Tepi suatu obyek dalam citra dinyatakan sebagai titik yang nilai keabuannya berbeda cukup besar dengan titik yang ada di sebelahnya. Hasil pengujian konvolusi *edge detection* dapat dilihat pada Gambar 10.



Gambar 10. Hasil operasi *edge detection filter* (a) Citra asli; (b) Citra yang telah dilakukan *edge detection*

## SIMPULAN

Dari hasil uji coba dan evaluasi, dapat disimpulkan bahwa: (1) *metode konvolusi* dapat digunakan dalam proses pengolahan citra seperti: penghalusan citra (*smooth*), *Gaussian blur*, *sharpen*, *mean removal*, *emboss* (memberi efek timbul pada citra), dan *edge detection* (pendeteksian tepi pada citra); (2) Konvolusi kernel yang digunakan berbentuk matriks 3x3, maka pengolahan citra yang dilakukan memberi efek kecil namun masih terlihat perbedaan antara citra asli dengan citra yang sudah diolah. Untuk mendapatkan hasil yang diinginkan user dapat melakukan proses berulang kali; (3) dalam melakukan penghalusan citra terhadap *noise* akan memberikan efek *blur* pada citra yang diolah, maka dari itu perlu dilakukan penajaman citra; (4) Fungsi konvolusi *mean removal* sama dengan konvolusi *sharpen* yaitu melakukan penajaman citra, tetapi konvolusi *mean removal* memberi ketajaman lebih pada citra. Hal tersebut dikarenakan oleh kernel yang digunakan oleh konvolusi *sharpen* hanya bekerja pada garis horizontal dan vertikal saja, sedangkan konvolusi *mean removal* juga bekerja pada garis diagonal; (5) berdasarkan data-data yang ada pengolahan citra menggunakan metode konvolusi mampu menghasilkan kualitas citra yang lebih baik.; (6) aplikasi pengolahan citra yang telah dibuat dapat diterapkan untuk kebutuhan studio foto, di mana penggunaannya lebih mudah daripada menggunakan *software* yang ada yang dilakukan secara manual atau menggunakan *software* khusus yang memerlukan keahlian khusus.

## DAFTAR PUSTAKA

- Achmad, B., Firdausy, K., *Teknik Pengolahan Citra Digital Menggunakan Delphi*. Yogyakarta: Ardi Publishing.
- Crane, Randy. (1997). *A Simplified Approach to Image Processing*. New Jersey: Prentice Hall.
- Jahne, Bernd. (2005). *Digital Image Processing* (6th ed.). Jerman: Springer.

- Joshi, B. K. (1973). *The Inversion of a Konvolusi Transform*. Department of Mathematics, Government College of Engineering and Technology, Raipur.
- Maria, P., Bosdogianni, P. (1999). *Image Processing: The Fundamental*. England: John Wiley & Sons.
- Munir, R. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.
- Pathmanabhan, A., Dinesh, S. (2007). The Effect of Gaussian Blurring on Extraction of Peaks and Pits from Digital Elevation Models. *Discrete Dynamics in Nature and Socity*.
- Zhang, M. Z., Ngo, H.T., Livingston, A. R., Asari, V. K. (2008). A High Performance Architecture for Implementation of 2-D Konvolusi With Quadrant Symmetric Kernels. *International Journal of Computers and Applications*, 30: 4.