

Aplikasi Matriks dalam Pengolahan Gambar

Adi Purnama (13514006)¹
 Program Studi Informatika
 Sekolah Teknik Elektro dan Informatika
 Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
¹adipurnama@s.itb.ac.id

Abstract— Matriks merupakan sebuah objek matematika yang terdiri dari susunan angka-angka berdasarkan baris dan kolom. Teori-teori mengenai matriks memiliki banyak manfaat untuk menyelesaikan berbagai masalah di dunia matematika, sains, bisnis & engineering. Di dalam makalah ini, penulis akan menjelaskan berbagai aplikasi matriks dalam pengolahan citra (*image processing*). Beberapa fungsi dasar yang dimiliki oleh perangkat lunak pengolahan citra seperti rotasi, translasi, *blur*, *sharpen*, dan lain lain akan dijelaskan dengan sudut pandang operasi matriks.

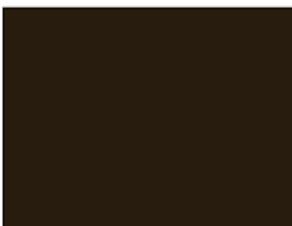
Keywords—Matriks, Pengolahan Citra, affine transformation, convolution matrix.

I. PENDAHULUAN

Sebuah gambar dapat didefinisikan sebagai fungsi dua dimensi $f(x,y)$ dimana x & y adalah koordinat bidang dan besar dari $f(x,y)$ adalah intensitas cahaya gambar pada sebuah titik (x,y) . [1] Sementara itu, gambar digital (*digital image*) merupakan sebuah gambar yang telah melalui proses *sampling* & *quantizing*. Pada awalnya, sebuah gambar bersifat kontinu. Dengan melakukan proses *sampling* & *quantizing*, gambar ini diubah menjadi bersifat diskrit.

Proses *sampling* adalah proses untuk mengubah koordinat gambar menjadi bersifat diskrit [1]. Hal ini dilakukan dengan membagi gambar menjadi petak-petak satuan yang disebut dengan piksel (*pixel*). Sementara itu, proses *quantizing* adalah proses pemberian nilai intensitas pada tiap-tiap piksel.

Setelah melalui kedua proses ini, gambar digital dapat dipandang sebagai sebuah matriks berukuran $m \times n$, dimana m adalah besarnya panjang gambar, n adalah besarnya lebar gambar dan elemen di dalam matriks merupakan intensitas warna pada setiap piksel di dalam gambar.

| | | | | |
|---|----------------------|----------------------|----------------------|----------------------|
|  | R:38 G:28 B:19 | R:41 G:30 B:17 | R:42 G:29 B:17 | R:39 G:30 B:13 |
| | R:40 G:26 B:20 | R:41 G:28 B:18 | R:40 G:28 B:16 | R:39 G:29 B:13 |
| | R:39 G:27 B:19 | R:37 G:28 B:18 | R:37 G:28 B:16 | R:39 G:30 B:15 |

Gambar 1. Sebuah gambar (kiri), dilakukan proses *sampling* (pemotongan gambar menjadi beberapa petak *pixel*) & *quantizing* (pemberian nilai intensitas pada setiap *pixel*)

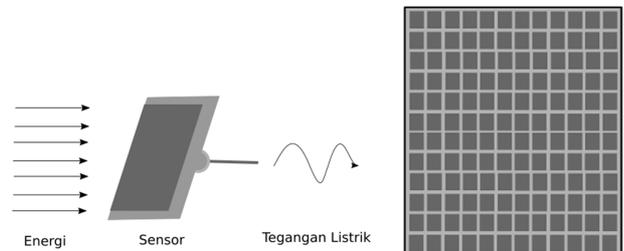
Gambar di atas, setelah diubah ke dalam bentuk gambar digital, dapat direpresentasikan dalam matriks G .

$$G = \begin{pmatrix} (38,28,19) & (41,30,17) & (39,30,13) & (39,30,13) \\ (40,26,20) & (41,28,18) & (40,28,16) & (39,29,13) \\ (39,27,19) & (37,28,18) & (37,28,16) & (39,30,15) \end{pmatrix}$$

Dengan memandang sebuah gambar dalam bentuk matriks, kita dapat melakukan pengolahan gambar dengan mengoperasikan nilai-nilai di dalam matriks ini.

II. REPRESENTASI GAMBAR MENGGUNAKAN MATRIKS

Agar sebuah gambar bisa diolah oleh komputer, pertama-tama gambar tersebut harus diubah ke dalam bentuk digital (matriks). Energi cahaya yang berbentuk analog dikonversi menjadi tegangan listrik menggunakan sensor. Proses ini disebut juga dengan *sampling*. Besarnya tegangan listrik yang dihasilkan berbanding lurus dengan intensitas cahaya yang diterima oleh sensor.



Gambar 2. Sensor mengubah energi cahaya yang diterima menjadi sinyal tegangan listrik (kiri), sensor disusun sebagai petak-petak, nilai pembacaan intensitasnya mewakili elemen pada matriks di dalam gambar digital.

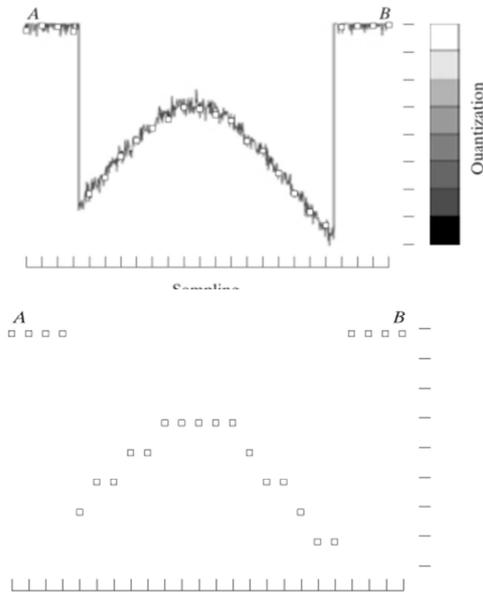
Sumber gambar : Gonzalez, Woods, "Digital Image Processing", Prentice Hall, 2001

Sensor-sensor tersebut disusun sebagai petak-petak berupa matriks. Nilai pembacaan tegangan listrik pada masing-masing sensor menggambarkan nilai intensitas cahaya pada setiap piksel. Jumlah sensor yang digunakan akan mempengaruhi kualitas gambar yang dihasilkan.

Setelah proses ini selesai dilakukan, kita akan mendapatkan nilai tegangan listrik yang dihasilkan oleh masing – masing sensor. Nilai tegangan listrik ini

menggambarakan besarnya intensitas cahaya yang diterima oleh sensor.

Namun, nilai intensitas cahaya ini masih berupa nilai yang kontinu . Oleh karena itu, nilai ini diubah menjadi nilai yang bersifat diskrit menggunakan proses kuantisasi

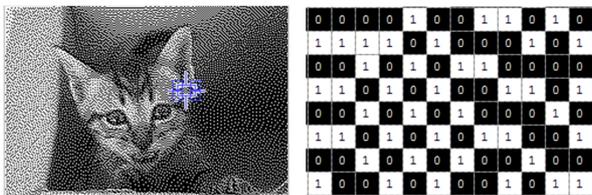


Gambar 3. Intensitas cahaya belum dikuantisasi (atas), intensitas cahaya sudah dikuantisasi menjadi struktur diskrit (bawah) Sumber gambar : Gonzalez, Woods , “Digital Image Processing”, Prentice Hall ,2001

Terdapat beberapa alternatif representasi gambar menggunakan matriks.[12]

A. Binary Image

Gambar disimpan dalam bentuk matriks yang elemen-elemennya berisi 0 dan 1 . Binary image hanya dapat menampilkan dua buah warna , yaitu hitam (0) dan putih (1)



Gambar 4 (Kiri) gambar direpresentasikan dalam bentuk gambar biner , (kanan) cuplikan isi matriks pada gambar

B. Grayscale Image

Gambar disimpan dalam bentuk matriks yang elemen elemennya berada di dalam rentang 0 (hitam) sampai 255(putih). Bilangan diantara 0 & 255 merepresentasikan warna abu-abu. Representasi gambar dalam bentuk

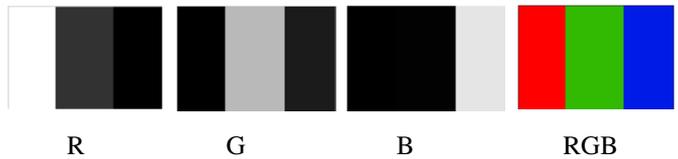
grayscale image sering digunakan dalam pemrosesan citra.



Gambar 5 (Kiri) gambar direpresentasikan dalam bentuk grayscale image , (kanan) cuplikan isi matriks pada gambar. Perhatikan bahwa nilai elemen-elemen matriks berada pada rentang 0 - 255

C. RGB

Gambar disimpan dalam bentuk matriks, yang elemen-elemennya berupa tripel (x1,x2,x3) dimana x1 adalah intensitas warna merah, x2 adalah warna hijau dan x3 adalah warna biru. Intensitas masing-masing warna berada dalam rentang 0 – 255.



Gambar 6. Gambar RGB dipisahkan menurut komponen-komponennya. Intensitas warna pada tiap komponen direpresentasikan dalam bentuk grayscale.

Perhatikan gambar di atas. Pada komponen R (merah) , bagian kiri gambar memiliki intensitas yang tinggi (ditunjukkan dengan grayscale image yang berwarna putih) . Ini menunjukkan bahwa intensitas warna merah paling tinggi berada di bagian kiri gambar.

Untuk mengolah gambar dalam format RGB, gambar dipisahkan terlebih dahulu menurut komponen-komponennya . Lalu , komponen-komponennya itu diproses dengan menganggap komponen tersebut sebagai grayscale. Setelah diproses , komponen itu digabungkan kembali menjadi gambar RGB yang utuh

D. Indexed Image

Gambar disimpan pada dua buah matriks. Matriks pertama memiliki ukuran yang sama dengan jumlah pixel pada gambar. Setiap elemen pada matriks ini adalah sebuah bilangan yang merupakan kode warna. Sementara itu, matriks kedua (disebut juga color map) menyimpan nilai intensitas warna yang bersesuaian dengan kode warna pada matriks pertama

$$G_{rgb} = \begin{pmatrix} (0,70,32) & (0,100,30) \\ (0,70,32) & (0,100,30) \end{pmatrix}$$

$$G_{ind} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \quad Map = \begin{pmatrix} 0 & 70 & 32 \\ 0 & 100 & 30 \end{pmatrix}$$

Matriks di atas menunjukkan konversi gambar yang direpresentasikan dengan RGB menjadi *indexed image*. Pada *indexed image*, kode warna 0 mengacu pada (0,70,30) dan kode warna 1 mengacu pada (0,100,30). Representasi gambar menggunakan *indexed image* dapat mengurangi besarnya ukuran penyimpanan data.

Dari beberapa alternatif di atas, representasi gambar menggunakan *grayscale* banyak digunakan dalam pengolahan citra.

III. TRANSFORMASI GAMBAR

Salah satu fungsi dasar yang dimiliki oleh perangkat lunak pengolah gambar adalah transformasi gambar. Dengan melakukan transformasi gambar, kita dapat mengubah ukuran, sudut, dimensi & perspektif dari gambar yang kita sedang olah. Transformasi gambar ini dapat direpresentasikan menggunakan perkalian matriks.

Salah satu cara sederhana untuk melakukan transformasi gambar adalah melakukan transformasi setiap titik pixel pada gambar mula-mula (x,y) ke koordinat baru (x',y'). Lalu, menempatkan kembali nilai intensitas pixel ke koordinat yang baru.

$$\begin{aligned} x' &= f(x, y) \\ y' &= g(x, y) \end{aligned}$$

Transformasi titik koordinat ini dapat dilakukan menggunakan persamaan *affine* (*affine transformation*). *Affine transformation* merupakan transformasi yang mempertahankan kesejajaran (titik titik dalam sebuah garis akan tetap berada di garis tersebut setelah dilakukan transformasi) dan rasio jarak objek yang dilakukan transformasi (titik tengah suatu garis akan tetap berada di tengah setelah dilakukan transformasi) [4]. Koordinat (x,y) setelah dilakukan transformasi dapat ditentukan menggunakan persamaan berikut. [6]

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} s_x & r_x & 0 \\ r_y & s_y & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

Keterangan

- x' : Koordinat x setelah ditransformasi
- y' : Koordinat y setelah ditransformasi
- x : Koordinat x mula-mula
- y : Koordinat y mula-mula
- s_x : Faktor pengali *scaling* sumbu x
- s_y : Faktor pengali *scaling* sumbu y

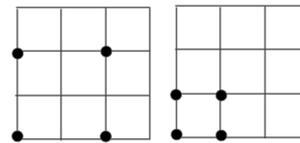
- r_x : Faktor pengali *shearing* sumbu x
- r_y : Faktor pengali *shearing* sumbu y
- t_x : Nilai translasi sumbu x
- t_y : Nilai translasi sumbu y

Sebagai contoh, andaikan kita mempunyai empat buah titik yang membentuk bangun persegi, direpresentasikan dengan matriks pada persamaan *affine*.

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

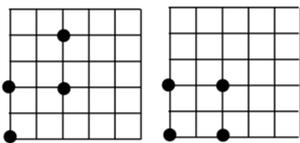
Untuk meningkatkan dimensi tinggi dan lebar persegi ini sebesar dua kali lipat, beri nilai s_x dan s_y sebagai 2. Sehingga matriks koordinat ini menjadi sebagai berikut

$$\begin{bmatrix} 0 & 0 & 1 \\ 2 & 0 & 1 \\ 0 & 2 & 1 \\ 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



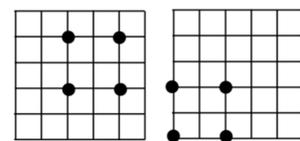
Untuk melakukan *shearing*, beri nilai pada r_x (*horizontal shear*) atau r_y (*vertical shear*). Beri nilai s_x & s_y sebesar 1 (penskalaan 1 kali lipat).

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



Untuk melakukan translasi, ubah nilai pada t_x & t_y, dengan mempertahankan nilai s_x & s_y sebesar 1.

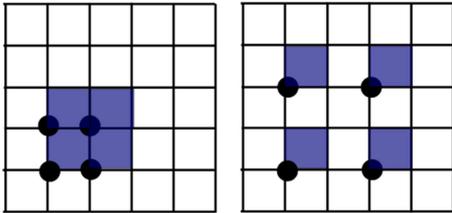
$$\begin{bmatrix} 2 & 2 & 1 \\ 4 & 2 & 1 \\ 2 & 4 & 1 \\ 4 & 4 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 2 & 0 & 1 \\ 0 & 2 & 1 \\ 2 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$



Untuk melakukan rotasi, nilai elemen pada matriks dapat dihitung menggunakan rumus

$$\begin{aligned} s_x &= \cos(\alpha) & r_x &= \sin(\alpha) \\ r_y &= -\sin(\alpha) & s_y &= \cos(\alpha) \end{aligned}$$

Namun, pada beberapa kasus, operasi transformasi gambar melibatkan perubahan jumlah piksel. Sebagai contoh, apabila kita memperbesar ukuran gambar (*scaling*), gambar yang telah diperbesar itu memiliki jumlah piksel yang lebih banyak daripada sebelumnya. Oleh karena itu, diperlukan sebuah metode untuk mengisi kekurangan piksel yang ada



Gambar 7 Empat buah piksel (kiri) ditransformasikan menjadi gambar sebelah kanan. Perhatikan bahwa ada piksel kosong pada gambar.

Ada beberapa prosedur untuk mengisi kekosongan pada piksel ini. Prosedur ini disebut juga dengan interpolasi gambar (*image interpolation*).[9]

A. Nearest Neighbor Interpolation

Nilai intensitas warna pada piksel yang kosong diisi oleh piksel terdekatnya. Interpolasi gambar jenis ini mengakibatkan gambar menjadi berukuran lebih besar terlihat terkotak-kotak (*jagged*)

B. Bilinear Interpolation

Nilai intensitas warna pada piksel yang kosong merupakan nilai rata-rata dari empat (piksel 2x2) intensitas warna di sekitar piksel kosong tersebut. Gambar yang dihasilkan oleh interpolasi jenis ini lebih halus daripada metode *nearest neighbor*.

C. Bicubic Interpolation

Nilai intensitas warna pada piksel yang kosong merupakan nilai rata-rata dari 16 (piksel 4x4) intensitas warna di sekitar piksel kosong tersebut.

Prosedur lain untuk mengatasi masalah ini adalah dengan melakukan transformasi balik. [10] Andaikan (x',y') adalah koordinat gambar setelah ditransformasi. Kita akan mencari (x,y) yang bersesuaian dengan (x',y') , lalu menempatkan nilai intensitas warna (x,y) ke koordinat (x',y') .

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} s_x & r_x & 0 \\ r_y & s_y & 0 \\ t_x & t_y & 1 \end{bmatrix}^{-1} = \begin{bmatrix} x & y & 1 \end{bmatrix}$$

Nilai (x,y) yang didapat dari hasil perkalian matriks di atas mungkin bukan merupakan bilangan bulat. Apabila

demikian, cukup bulatkan nilai (x,y) ke satuan terdekat.[10]

Pada beberapa kasus, operasi transformasi gambar justru melibatkan pengurangan jumlah piksel pada gambar. Salah satu prosedur untuk mengatasi masalah ini adalah dengan merata-ratakan beberapa piksel yang bertetangga.[11] Andaikan sebuah gambar yang ukurannya diperkecil sebesar 50%, maka setiap 2x2 blok piksel akan diubah menjadi sebuah piksel yang memiliki intensitas warna sebesar rata-rata 4 piksel tersebut.

Berikut adalah hasil pemrosesan gambar menggunakan perkalian matriks *affine*.

| Matriks <i>Affine</i> | Hasil Transformasi |
|--|--------------------|
| $\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | |
| $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | |
| $\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | |
| $\begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | |
| $\begin{bmatrix} 0.86 & 0.5 & 0 \\ -0.5 & 0.86 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | |

IV. EFEK FILTER GAMBAR

Selain transformasi gambar, fitur utama yang dimiliki oleh perangkat lunak pengolah gambar adalah fitur efek filter. Dengan fitur ini, kita dapat meningkatkan ketajaman gambar (*contrast*), mengatur pencahayaan (*brightness*), memberi efek buram (*blur*) dan lain lain. Proses pemberian efek gambar ini disebut juga dengan *image convolution*.

Proses konvolusi matriks berbeda dengan perkalian matriks biasa. Operasi konvolusi (dinotasikan dengan simbol $*$) dapat digambarkan sebagai berikut

$$V = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Operasi ini akan menghasilkan

$$V = (a*1) + (b*2) + (c*3) + (d*4) + (e*5) + (f*6) + (g*7) + (h*8) + (i*9)$$

Di dalam pemrosesan gambar, matriks

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

menggambarkan nilai intensitas cahaya pada tiap piksel gambar. Sementara itu, matriks

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

disebut juga dengan *kernel matrix*, sebuah matriks yang berfungsi sebagai efek filter yang akan diaplikasikan pada gambar.

Sebagai contoh, andaikan kita akan mengaplikasikan efek *blur* pada sebuah gambar. Kernel efek blur adalah sebagai berikut

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Andaikan matriks gambar yang akan diberi efek blur adalah sebagai berikut

$$\begin{bmatrix} 92 & 46 & 78 \\ 172 & \mathbf{110} & 31 \\ 173 & 162 & 71 \end{bmatrix}$$

Hitung nilai konvolusi matriks tersebut

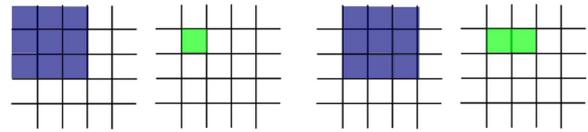
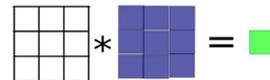
$$V = \begin{bmatrix} 92 & 46 & 78 \\ 172 & \mathbf{110} & 31 \\ 173 & 162 & 71 \end{bmatrix} * \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$V = 103$$

Hasil perhitungan ini merupakan nilai intensitas warna pada piksel yang sudah diberi efek buram. Maka, matriks gambar yang sudah diberi efek buram adalah sebagai berikut

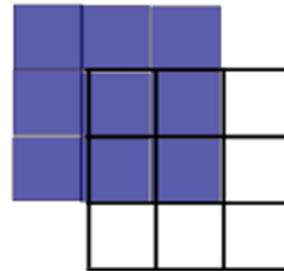
$$\begin{bmatrix} - & - & - \\ - & \mathbf{103} & - \\ - & - & - \end{bmatrix}$$

Proses ini baru memproses memburamkan sebuah piksel pada gambar. Lakukan proses ini berulang-ulang hingga seluruh piksel pada gambar telah dihitung.



Gambar 8. Proses yang dilakukan dalam image convolution, bagian tengah pada matriks kernel akan mengunjungi setiap piksel pada gambar secara traversal hingga seluruh piksel gambar selesai diproses.

Kasus khusus terjadi apabila matriks kernel mengunjungi bagian tepi matriks. Pada kasus ini, ada beberapa elemen matriks yang tidak terdefinisi nilainya.



Terdapat beberapa alternatif solusi untuk menyelesaikan persoalan ini[5], yaitu

1. Bagian tepi matriks tidak diproses sama sekali.
2. Bagian tepi matriks diperluas hingga seluruh nilai yang dibutuhkan untuk operasi konvolusi didapatkan.
3. Bagian tepi matriks dibuang dari gambar.

Perhatikan lagi matriks kernel pada efek buram

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Proses yang dilakukan oleh kernel efek buram ini adalah merata-ratakan nilai intensitas warna piksel beserta piksel di sekelilingnya.

$$V = (a+b+c+d+e+f+g+h+i)/9$$

Dengan mengubah nilai intensitas tiap piksel dengan nilai rata-rata intensitas piksel disekelilingnya, maka perbedaan intensitas warna antar piksel-piksel yang bertetangga tidak terlalu besar. Ini mengakibatkan gambar menjadi terlihat buram.

Selain efek buram (*blur*), berikut adalah matriks kernel untuk beberapa efek filter.

A. *Sharpen*

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Matriks kernel ini meningkatkan intensitas warna piksel di tengah berdasarkan perbedaan intensitas warna piksel di sekelilingnya (atas, bawah, kiri dan kanan). Semakin besar perbedaan intensitas warna piksel dengan sekelilingnya, semakin tinggi peningkatan intensitas warna piksel tersebut. Ini mengakibatkan efek peningkatan ketajaman gambar.

Untuk menghasilkan gambar yang lebih tajam, kita dapat menggunakan kernel matriks yang memperhitungkan seluruh piksel yang bertetangga dengan piksel tersebut.

B. *Emboss*

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

Dengan matriks ini, nilai intensitas piksel yang akan diberikan efek akan cenderung mendekati nilai intensitas piksel tetangganya di bagian kanan-bawah & cenderung menjauhi nilai intensitas piksel tetangganya di bagian kiri-atas. Hal ini menghasilkan efek pencahayaan yang datang dari sudut kiri atas gambar.

C. *Lighten & Darken*

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad x > 1 \text{ (lighten)}, x < 1 \text{ (darken)}$$

Untuk mengubah tingkat pencahayaan gambar, kita cukup meningkatkan nilai intensitas cahaya pada tiap piksel. Agar nilai intensitas cahaya piksel meningkat, kalikan nilainya dengan x , dimana $x > 1$. Sebaliknya, untuk mengurangi nilai intensitas, kalikan dengan x , dimana $x < 1$. Apabila $x = 1$, maka nilai intensitas cahaya piksel tersebut tidak berubah.

D. *Edge Detection*

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

Piksel tepi objek gambar cenderung memiliki warna yang berbeda dengan piksel di sekelilingnya. Sementara itu, piksel yang bukan merupakan tepi objek gambar cenderung memiliki warna yang sama dengan piksel di sekelilingnya. Dengan menggunakan filter *edge detection*, kita dapat mendeteksi tepi objek gambar dengan cara mengubah warna piksel yang bukan merupakan tepi objek gambar menjadi warna gelap (hitam).

Dengan menggunakan kernel matriks ini, apabila nilai

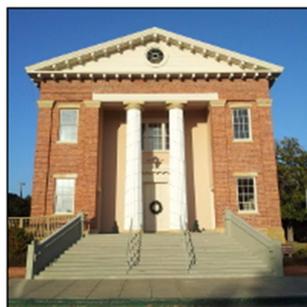
intensitas cahaya piksel cenderung sama dengan intensitas cahaya piksel di sekelilingnya, maka intensitas cahaya piksel tersebut akan cenderung mendekati nol (gelap).

$$a = b = c = d = e$$

$$a + b + c + d - 4e = 0$$

Sebaliknya, apabila intensitas cahaya piksel berbeda dengan sekelilingnya, ini menandakan bahwa piksel tersebut merupakan tepi suatu objek pada gambar. Piksel tersebut akan memiliki intensitas yang bukan nol.

Berikut adalah hasil pengujian efek gambar terhadap beberapa matriks kernel.



Gambar 9. Sumber Gambar : <http://beej.us/blog/data/convolution-image-processing/>
Diakses tanggal 15 Desember 2015

| | |
|--|--|
| | <p style="text-align: center;"><i>Sharpen</i></p> $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ |
| | <p style="text-align: center;"><i>Blur</i></p> $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ |
| | <p style="text-align: center;"><i>Emboss</i></p> $\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$ |

| | |
|--|--|
|  | <p><i>Edge Detect</i></p> $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ |
|  | <p><i>Lighten</i></p> $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ |
|  | <p><i>Darken</i></p> $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ |

REFERENSI

- [1] Gonzalez, Woods , “*Digital Image Processing*”, Prentice Hall ,2001
- [2] Beej Jorgensen , “Image Processing Convolutions” <http://beej.us/blog/data/convolution-image-processing/> , Diakses tanggal 15 Desember 2015
- [3] Design Stacks, “Convoluton Matrix” , <http://designstacks.net/convolution-matrix> , Diakses tanggal 15 Desember 2015
- [4] Harvey Rhody , “Geometric Image Transformations” , http://www.cis.rit.edu/class/simg782/lectures/lecture_02/lec782_05_02.pdf , Diakses tanggal 12 Desember 2015
- [5] Jamie Ludwig, “Image Convolution” , http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig_ImageConvolution.pdf . Diakses tanggal 11 Desember 2015
- [6] Anthony Thyssen, “Affline Matrix Transforms” , <http://www.imagemagick.org/Usage/distorts/affine/> . Diakses tanggal 12 Desember 2015
- [7] Victor Powell , “Image Kernels Explained Visually”, <http://setosa.io/ev/image-kernels/> Diakses tanggal 12 Desember 2015
- [8] Neil Dogson , “Computer Graphics & Image Processing” , <http://www.cl.cam.ac.uk/teaching/2003/Graphics/CGIP.pdf> . Diakses tanggal 11 Desember 2015
- [9] Sean McHugh , “Digital Image Interpolation” , <http://www.cambridgeincolour.com/tutorials/image-interpolation.htm> . Diakses tanggal 12 Desember 2015
- [10] Thayer Watkins , “The Geometric Transformation of Images “ , <http://www.sjsu.edu/faculty/watkins/transimage.htm> . Diakses tanggal 14 Desember 2015
- [11] Anthony Thyssen , “Rezise Or Scaling (General Techniques) , <http://www.imagemagick.org/Usage/resize/#distort> . Diakses 15 Desember 2015
- [12] Mathworks , “Image Types” , http://www.mathworks.com/help/matlab/creating_plots/image-types.html . Diakses 12 Desember 2015

V. KESIMPULAN

Matriks memiliki aplikasi yang cukup luas pada bidang *computer graphics* . Salah satunya adalah aplikasinya pada aplikasi pengolah gambar. Sebuah objek gambar dapat direpresentasikan sebagai matriks yang elemen-elemennya merupakan nilai intensitas warna pada masing-masing piksel pada matriks. Karena objek gambar dapat direpresentasikan dalam bentuk matriks, pengolahan gambar dapat direpresentasikan dalam bentuk operasi-operasi pada matriks. Pada makalah ini, telah dijelaskan bahwafungsi transformasi gambar dapat direpresentasikan sebagai perkalian matriks koordinat piksel dengan matriks *affine* . Sementara itu, pemberian efek filter pada gambar dapat direpresentasikan sebagai operasi konvolusi antara matriks intensitas cahaya gambar dengan matriks *kernel*.

VII. UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Allah S.W.T , karena atas rahmat-Nya , penulis dapat menyelesaikan makalah ini. Selain itu , penulis juga mengucapkan terima kasih kepada Bapak Rinaldi Munir dan Bapak Judhi Santoso selaku dosen pengampu kuliah IF 2123 Aljabar Geometri yang telah memberikan ilmu yang bermanfaat kepada penulis.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 15 Desember 2015



Adi Purnama (13514006)