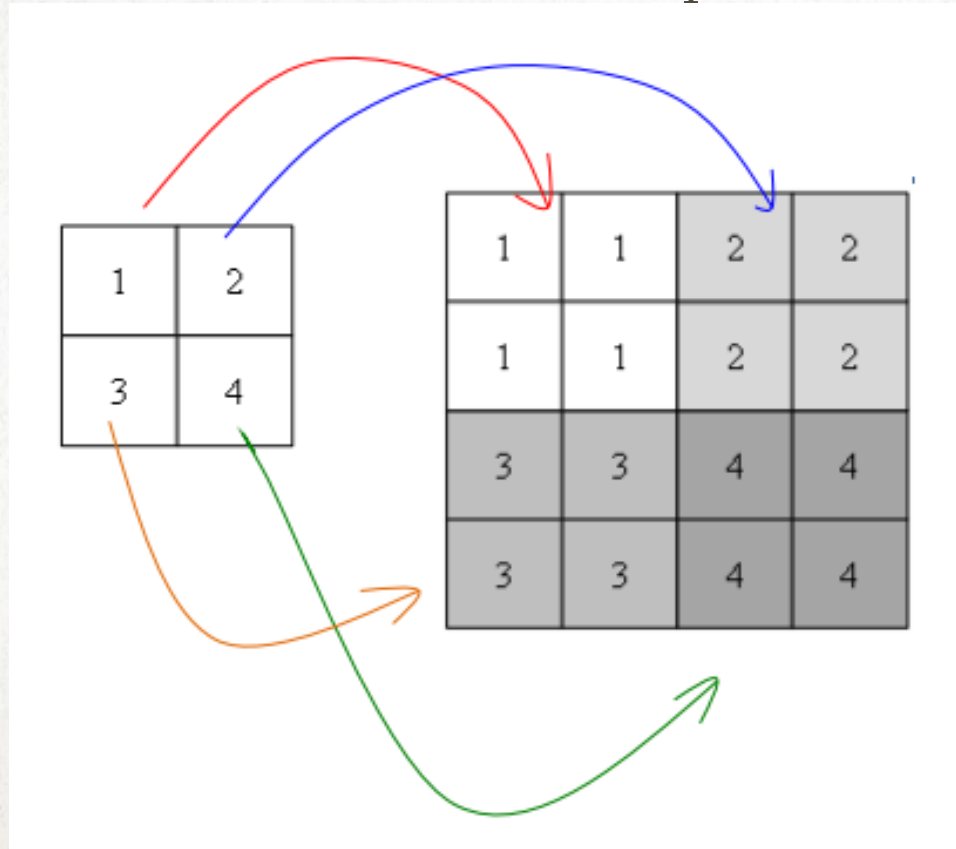


OPERASI GEOMETRIK

PERTEMUAN KE 9

Suatu citra dapat diperbesar dengan membuat setiap piksel menjadi beberapa piksel. Seperti ditunjukkan pada Gambar 8.13, memberikan contoh cara memperbesar citra.



Gambar 8.13 Cara memperbesar citra

MEMPERBE SAR CITRA



LISTING PROGRAM MEMPERBESAR CITRA:

```
function G = perbesar(berkas, sy, sx)
% PERBESAR Melakukan operasi pembesaran citra.
% Masukan: berkas = nama berkas image
% sy : skala pembesaran pada sumbu Y
% sx : skala pembesaran pada sumbu X
%
% Versi 1

F = imread(berkas);
[tinggi, lebar] = size(F);
tinggi_baru = tinggi * sy;
lebar_baru = lebar * sx;
F2 = double(F);
for y=1 : tinggi_baru
    y2 = ((y-1) / sy) + 1;
    for x=1 : lebar_baru
        x2 = ((x-1) / sx) + 1;
        G(y, x) = F(floor(y2), floor(x2));
    end
end
G = uint8(G);      % untuk mengatur nilai piksel
                   % agar berada pada jangkauan [0 -
                   % 255]
```

LISTING PROGRAM MEMPERBESAR CITRA :

Berdasar fungsi memperbesar citra di atas, dapat diberikan perintah seperti berikut:

```
>> Img = perbesar('C:\Image\lena128.png', 3,  
3);
```

Selanjutnya, hasil perbesaran ditampilkan melalui:

```
>> imshow(Img);
```


CONTOH HASIL MEMPERBESAR GAMBAR



(a) Citra lena 128x128



(b) Pembesaran 3x tanpa interpolasi

Gambar 8.14 Pembesaran citra 3 kali tanpa interpolasi

CONTOH HASIL MEMPERBESAR GAMBAR DENGAN DIBERIKAN PERINTAH INTERPOLASI



(a) Citra lena 128x128



(b) Pembesaran 3x

Gambar 8.15 Pembesaran citra 3 kali dengan interpolasi

LISTING PROGRAM MEMPERBESAR :

```
function G = perbesar2(berkas, sy, sx)
% PERBESAR2 Melakukan operasi pembesaran
citra
% dengan interpolasi.
% Masukan: berkas = nama berkas image
% sy : skala pembesaran pada sumbu Y
% sx : skala pembesaran pada sumbu X

F = imread(berkas);
[tinggi, lebar] = size(F);
tinggi_baru = round(tinggi * sy);
lebar_baru = round(lebar * sx);
F2 = double(F);
for y=1 : tinggi_baru
    y2 = (y-1) / sy + 1;
    for x=1 : lebar_baru
        x2 = (x-1) / sx + 1;
```

LISTING PROGRAM MEMPERBESAR :

```
% Lakukan interpolasi bilinear
p = floor(y2);
q = floor(x2);
a = y2-p;
b = x2-q;
if (floor(x2)==lebar) || ...
    (floor(y2) == tinggi)
    G(y, x) = F(floor(y2), floor(x2));
else
    intensitas = (1-a)*((1-b)*F(p,q) + ...
        b * F(p, q+1)) + ...
        a *((1-b)* F(p+1, q) + ...
        b * F(p+1, q+1));
    G(y, x) = intensitas;
end
end
end
G = uint8(G);
```


LISTING PROGRAM MEMPERBESAR DENGAN INTERPOLASI:

Berdasar fungsi memperbesar citra dengan interpolasi di atas, dapat diberikan perintah seperti berikut:

```
>> Img = perbesar2('C:\Image\lena128.png',  
4, 4);
```

Selanjutnya, hasil perbesaran citra dengan interpolasi ditampilkan melalui:

```
>> imshow(Img);
```

Pengecilan citra berarti mengurangi jumlah piksel. Algoritma pembesaran tanpa interpolasi dan pembesaran dengan interpolasi, dapat digunakan untuk kepentingan memperkecil citra. Adapun cara yang digunakan adalah dengan memberikan s_y dan s_x bilangan pecahan, seperti $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, dan seterusnya.

MEMPERKECIL CITRA



LISTING PROGRAM MEMPERKECIL DENGAN INTERPOLASI:

Berdasar fungsi memperkecil citra di atas, dapat diberikan perintah seperti berikut:

```
>> Img = perbesar2('C:\Image\lena256.png',  
0.5, 0.5);
```

Selanjutnya, hasil memperkecil citra, ditampilkan melalui:

```
>> imshow(Img);
```

CONTOH HASIL MEMPERKECIL CITRA



(a) Citra lena 256x256

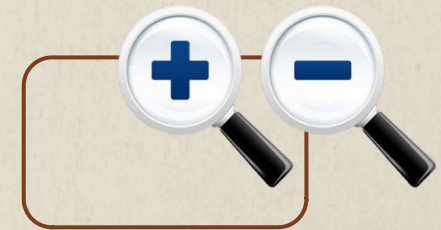


(b) Hasil pengecilan 0,5 x
pada arah vertikal dan horisontal

Gambar 8.16 Pengecilan citra lena256.png dengan interpolasi

Algoritma pembesaran tanpa interpolasi dan pembesaran dengan interpolasi, juga dapat digunakan untuk memperbesarkan/memperkecil citra, dengan skala horizontal dan vertikal yang berbeda. Adapun cara yang digunakan adalah dengan memberikan s_y dan s_x bilangan bilangan yang berbeda.

MEMPERBESAR CITRA DENGAN SKALA VERTIKAL DAN HORIZONTAL BERBEDA



LISTING PROGRAM MEMPERBESAR CITRA DENGAN SKALA VERTIKAL DAN HORIZONTAL BERBEDA:

Berdasar fungsi perbesar di atas, dapat diberikan perintah seperti berikut:

```
>> Img = perbesar2('C:\Image\gedung.png',  
0.5, 2.5);
```

Selanjutnya, hasil perbesaran ditampilkan melalui:

```
>> imshow(Img);
```


CONTOH HASIL MEMPERBESAR CITRA DENGAN SKALA VERTIKAL DAN HORIZONTAL BERBEDA



Gambar 8.17 Gedung di perkecil $\frac{1}{2}$ kali pada arah vertikal dan 2,5 kali pada arah horizontal

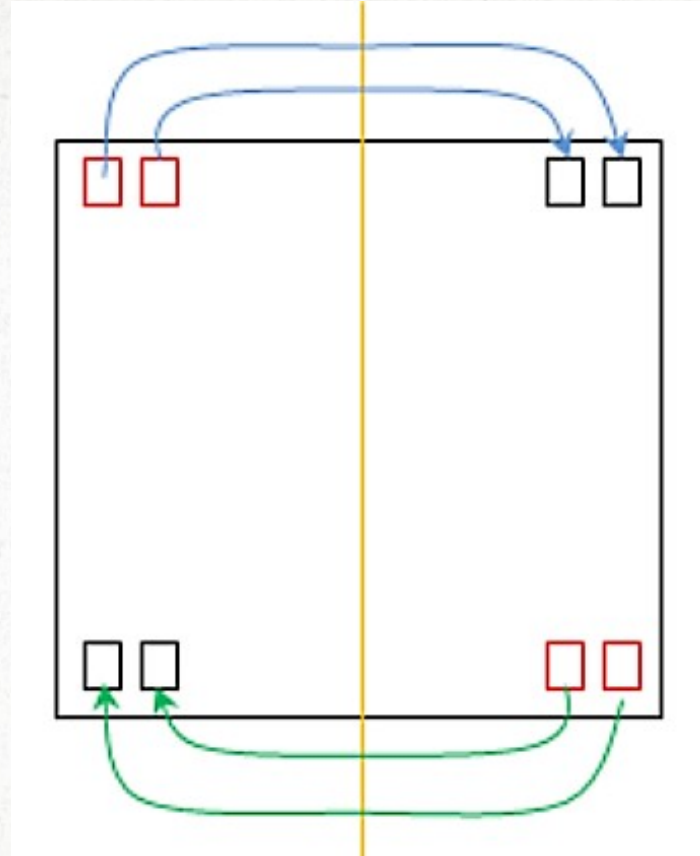
Pada proses pencerminan yang umum dilakukan adalah pencerminan secara horisontal dan pencerminan secara vertikal.

Pencerminan secara horisontal dilakukan dengan menukar dua piksel yang berseberangan kiri – kanan. Pencerminan secara horisontal ditunjukkan pada Gambar 8.17.

PENCERMINAN CITRA



PENCERMINAN SECARA HORIZONTAL



Gambar 8.18 Pencermian secara horizontal

ALGORITMA PENCERMINKAN GAMBAR SECARA HORIZONTAL

Masukan:

- $f(M, N)$: Citra masukan berukuran M baris dan N kolom

Keluaran:

- $g(M, N)$: Hasil citra yang telah dicerminkan secara horizontal

1. FOR baris \leftarrow 1 TO M

2. FOR kolom \leftarrow 1 TO N

3. $g(\text{baris}, \text{kolom}) \leftarrow f(N - \text{baris} + 1, \text{kolom})$

4. END-FOR

5. END-FOR

LISTING PROGRAM PECERMINAN CITRA SECARA HORISONTAL:

```
function G = cerminh(F)
% CERMINH Berfungsi untuk mencerminkan citra
% secara horizontal
% Masukan: F = Citra berskala keabuan
[tinggi, lebar] = size(F);
for y=1 : tinggi
    for x=1 : lebar
        x2 = lebar - x + 1;
        y2 = y;
        G(y, x) = F(y2, x2);
    end
end
G = uint8(G);    % untuk mengatur nilai piksel
                  agar berada pada jangkauan [0 -
                  255]
```

LISTING PROGRAM PECERMINAN SECARA HORIZONTAL CITRA:

Berdasar fungsi pencerminan secara horisontal di atas, dapat diberikan perintah seperti berikut:

```
>> F = imread('C:\Image\boneka.png');  
>> G = cerminh(F);
```

Selanjutnya, hasil pencerminan horisontal citra ditampilkan melalui:

```
>> imshow(Img);
```


CONTOH HASIL PENCERMINAN CITRA SECARA HORIZONTAL



(a) Citra boneka.tif



(b) Pencermianan secara horizontal

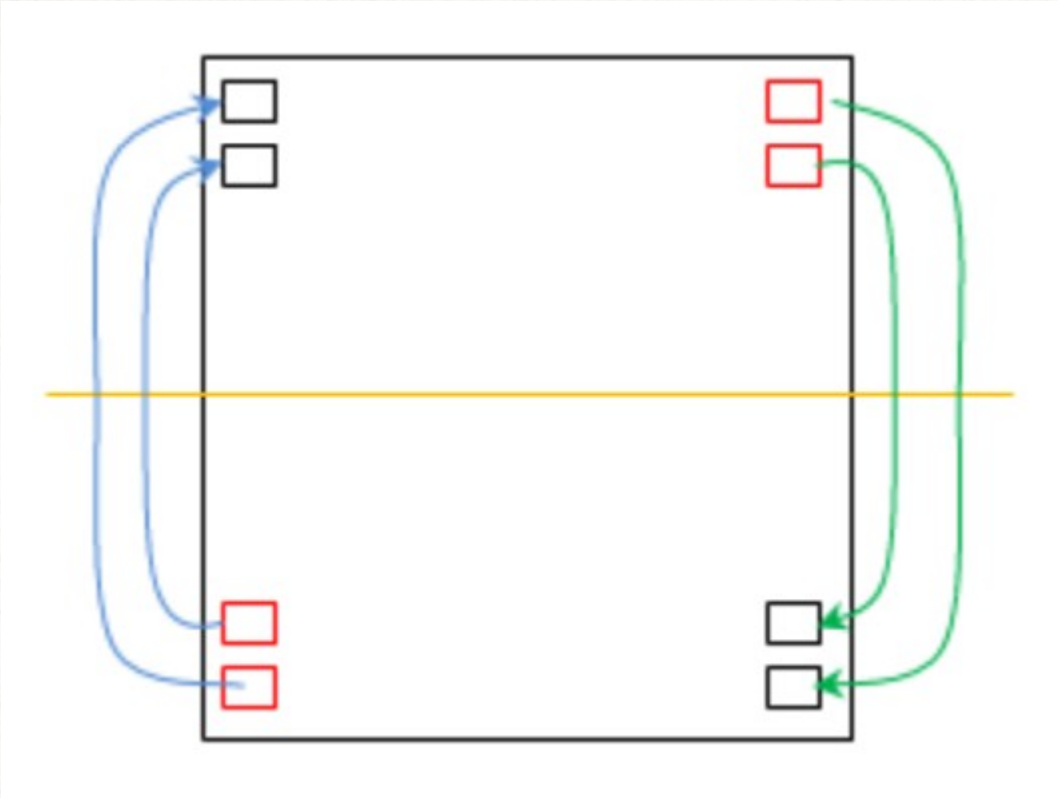
Gambar 8.19 Contoh hasil pencerminan secara horizontal

Pencerminan secara vertikal dilakukan dengan menukar dua piksel yang berseberangan atas – bawah. Pencerminan secara horisontal ditunjukkan pada Gambar 8.20.

PENCERMINAN CITRA



PENCERMINAN SECARA VERTIKAL



Gambar 8.20 Pencerminkan secara vertikal

ALGORITMA PENCERMINKAN GAMBAR SECARA VERTIKAL

Masukan:

- $f(M, N)$: Citra masukan berukuran M baris dan N kolom

Keluaran:

- $g(M, N)$: Hasil citra yang telah dicerminkan secara vertikal

1. FOR baris ~~1~~ TO M
2. FOR kolom ~~1~~ TO N
3. $g(\text{baris}, \text{kolom}) \leftarrow f(\text{baris}, N - \text{kolom} + 1)$
4. END-FOR
5. END-FOR

LISTING PROGRAM PECERMINAN CITRA SECARA VERTIKAL:

```
function G = cerminv(F)
% CERMINV Berfungsi untuk mencerminkan citra
% secara vertikal
% Masukan: F = Citra berskala keabuan
[tinggi, lebar] = size(F);
for y=1 : tinggi
    for x=1 : lebar
        x2 = x;
        y2 = tinggi - y + 1;
        G(y, x) = F(y2, x2);
    end
end
G = uint8(G);    % untuk mengatur nilai piksel
                  agar berada pada jangkauan [0 -
                  255]
```

LISTING PROGRAM PECERMINAN SECARA VERTIKAL CITRA:

Berdasar fungsi pencerminan secara vertikal di atas, dapat diberikan perintah seperti berikut:

```
>> F = imread('C:\Image\boneka.png');  
>> G = cerminv(F);
```

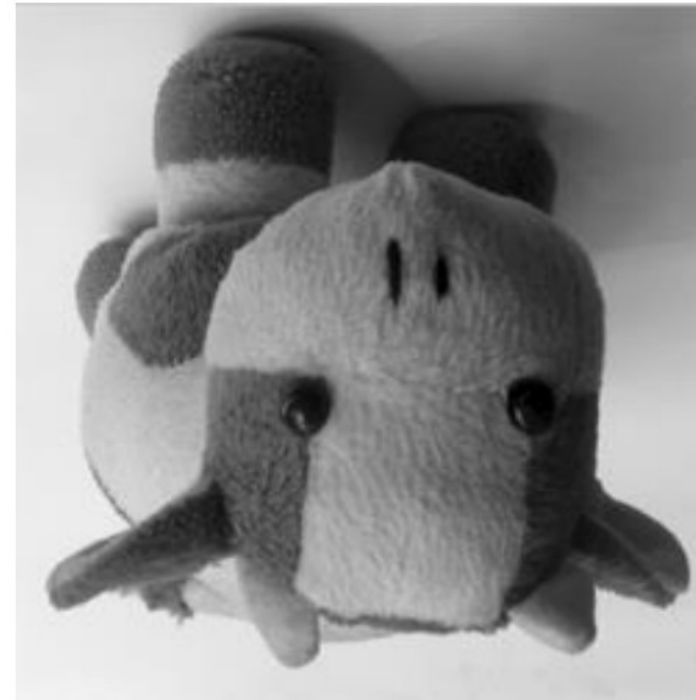
Selanjutnya, hasil pencerminan vertikal citra ditampilkan melalui:

```
>> imshow(Img);
```


CONTOH HASIL PENCERMINAN CITRA SECARA VERTIKAL



(a) Citra boneka.tif



(b) Pencermianan secara vertikal

Gambar 8.19 Contoh hasil pencerminan secara vertikal



Terimakasih