**EKSPLORASI OCTAVE**

Disusun untuk memenuhi tugas mata kuliah Pengelolaan Citra

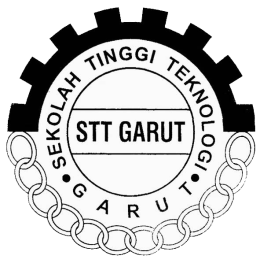
Oleh :

Arif Budiman (1406011)

Firda Frianda (1406090)

Rivan Mulyana (1406108)

Yulpan Paisal (1406130)



**SEKOLAH TINGGI TEKNOLOGI GARUT**

**PROGRAM STUDI INFORMATIKA**

**2017**

**EKSPLORASI OCTAVE**

1. **Geser**

**Source Code :**

F = imread('C:\Users\rivan\Pictures\teknologi.jpg');

[tinggi, lebar] = size(F);

sx = 45; sy = -35;

F2 = double(F);

G = zeros(size(F2));

for y=1 : tinggi

for x=1 : lebar

xlama = x-sx;

ylama = y-sy;

if(xlama>=1) && (xlama<=lebar) && (ylama>=1) && (ylama<=tinggi)

G(y,x) = F2(ylama, xlama);

else

G(y,x) = 0;

end end end

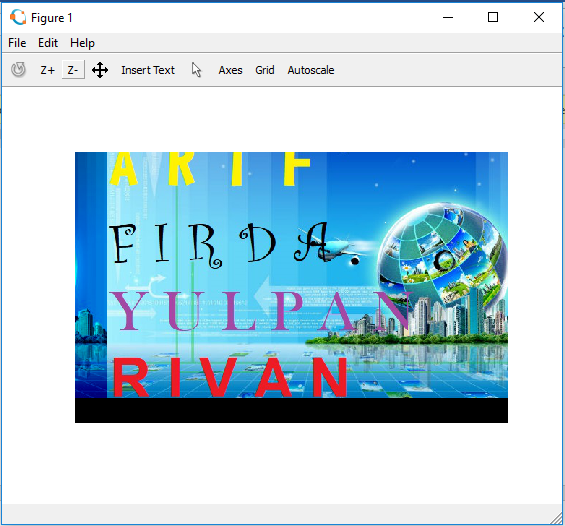
G = uint8(G);

figure(1);

imshow(G);

clear all;

**output**



1. **Rotasi**

**Source code :**

% ROTASI Melakukan Operasi pemutaran citra.

% Versi 1

% Menggunakan pendekatan pemetaan ke belakang

F = imread(‘C:\Users\rivan\Pictures\teknologi.jpg’);

[tinggi, lebar] = size(F);

sudut = 10; % Sudut pemutaran

rad = pi \* sudut/180;

cosa = cos(rad);

sina = sin(rad);

F2 = double(F);

for y=1 : tinggi

for x=1 : lebar

x2 = round(x \* cosa + y \* sina);

y2 = round(y \* cosa - x \* sina);

if (x2>=1) && (x2<=lebar) && ...

(y2>=1) && (y2<=tinggi)

G(y, x) = F2(y2, x2);

else

G(y, x) = 0;

end

end

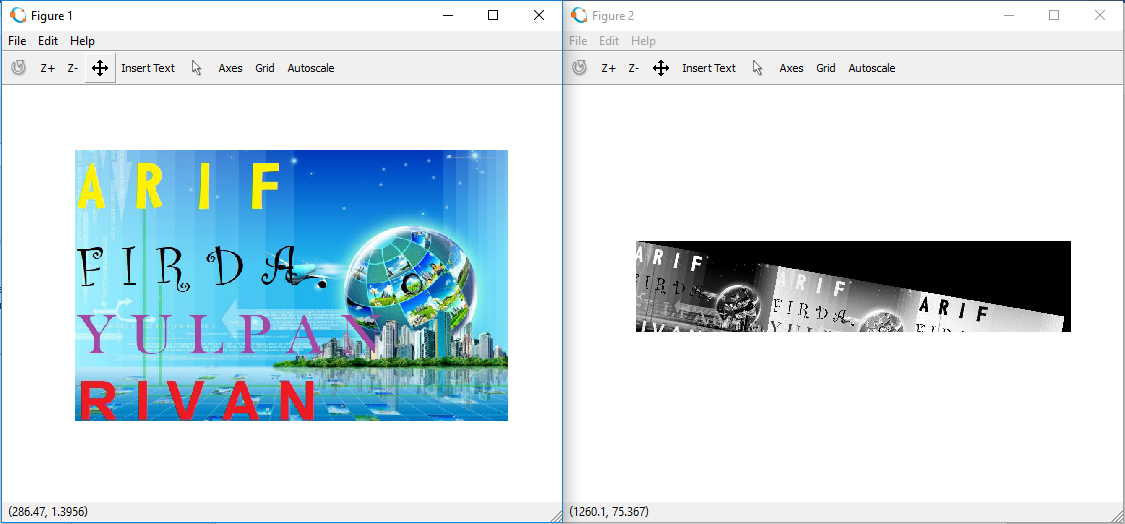
end

G = uint8(G);

figure(1);imshow(F);

figure(2);imshow(G);

**Output :**



1. **Perbesaran Citra**

**Source code :**

function G = perbesar(berkas, sy, sx)

F = imread(' C:\Users\rivan\Pictures\teknologi');

ukuran = size(F);

tinggi = ukuran(1);

lebar = ukuran(2);

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

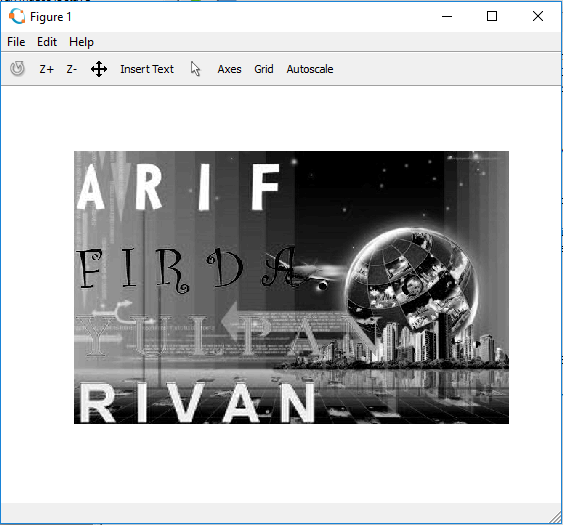
G = uint8(G);

Proses setelah di run ketik code ini di command window :

>> Img = perbesar(' C:\Users\rivan\Pictures\teknologi.jpg', 3, 3);

>> imshow(Img);

**Output :**



1. **Perkecil Citra**

**Source code :**

function G = perkecil(berkas, sy, sx)

F = imread(' C:\Users\rivan\Pictures\teknologi');

ukuran = size(F);

tinggi = ukuran(1);

lebar = ukuran(2);

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

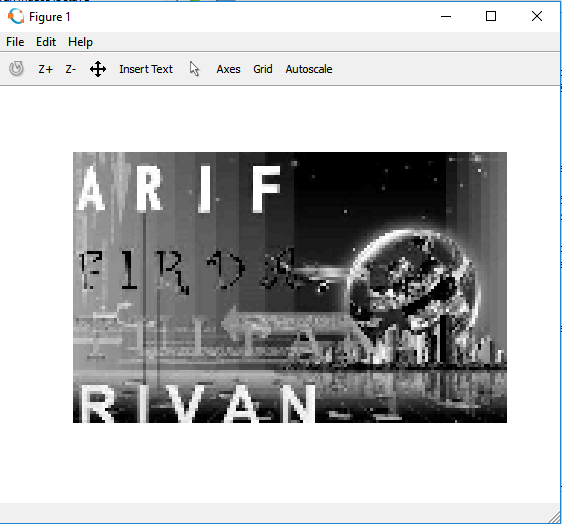
G = uint8(G);

Proses setelah di run ketik code ini di command window :

>> Img = perkecil(' C:\Users\rivan\Pictures\teknologi.jpg ', 0.25, 0.25);

>> imshow(Img);

**Output :**

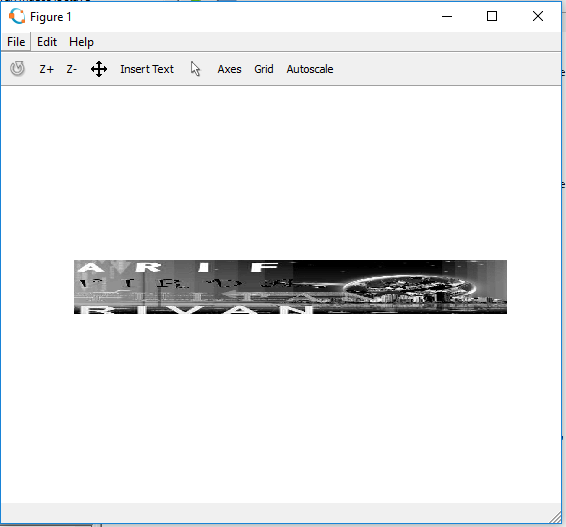


1. **perbesaran dengan skala vertical dan horizontal berbeda**

berikan perintah sebagai berikut :

>> Img = perbesar(' C:\Users\rivan\Pictures\teknologi.jpg ', 0.5, 2.5);

>> imshow(Img);



1. **Cerminh**

**Source code**

function G = cerminh(F)

% CERMINH berfungsi untuk mencerminkan citra

% secara vertikal

% 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

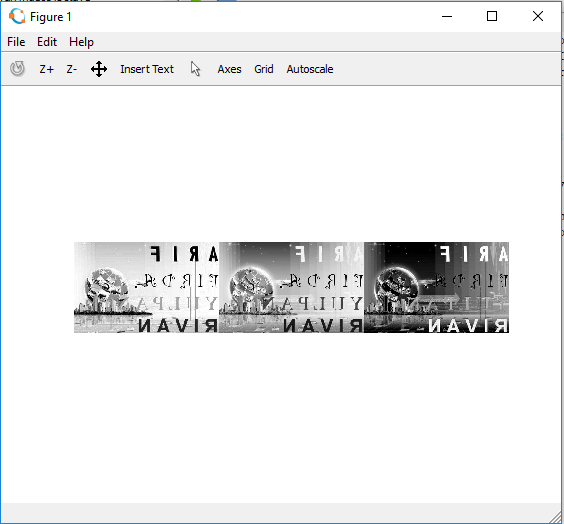
G = uint8(G);

Proses setelah di run ketik code ini di command window :

F = imread('C:\Users\rivan\Pictures\teknologi.jpg');

G=cerminh(F);

Imshow(G);



1. **Cerminv**

**Source code**

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

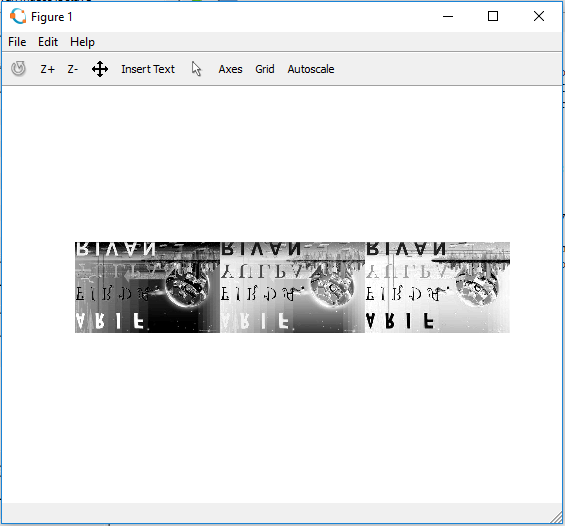
G = uint8(G);

Proses setelah di run ketik code ini di command window :

F = imread('C:\Users\rivan\Pictures\teknologi.jpg');

G=cerminv(F);

Imshow(G);



1. **Transformasi Taffline**

**Source code :**

function G = taffline(F, a11, a12, a21, a22, tx, ty)

[tinggi, lebar] = size(F);

for y=1 : tinggi

for x=1 : lebar

x2 = a11 \* x + a12 \* y + tx;

y2 = a21 \* x + a22 \* y + ty;

if (x2>=1) && (x2<=lebar) && ...

(y2>=1) && (y2<=tinggi)

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 else

G(y, x) = 0;

end end end

G = uint8(G);

Proses setelah di run ketik code ini di command window :

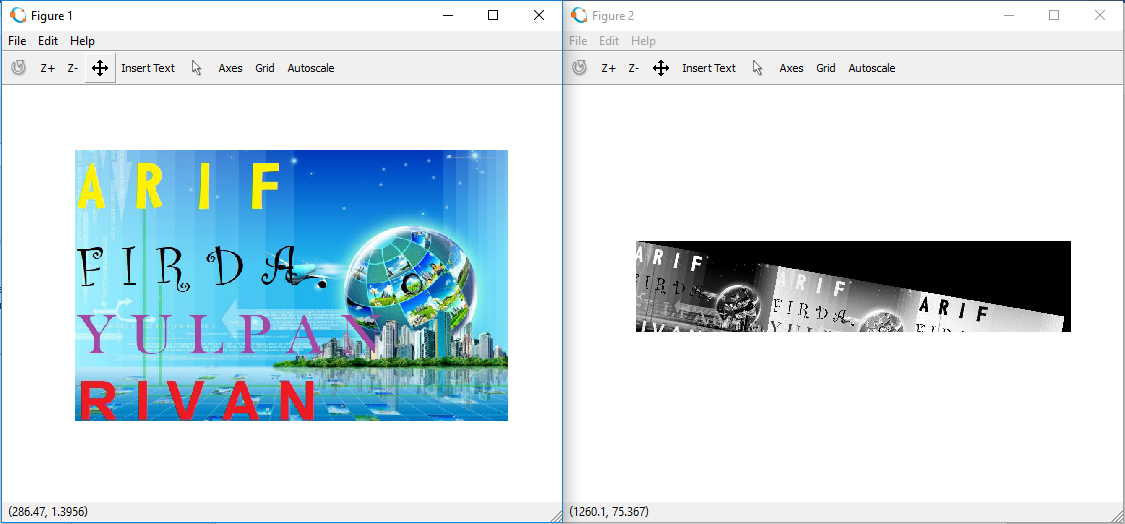
>> rad = 10 \* pi / 180;

>> G = taffline(F, cos(rad), sin(rad),...

-sin(rad), cos(rad), 0, 0);

>> imshow(G);

**Output :**



1. **Ripple**

**Source code :**

function G = ripple(F, ax, ay, tx, ty)

dimensi = size(F);

tinggi = dimensi(1);

lebar = dimensi(2);

for y=1 : tinggi

for x=1 : lebar

x2 = x + ax \* sin(2 \* pi \* y / tx);

y2 = y + ay \* sin(2 \* pi \* y / ty);

if (x2>=1) && (x2<=lebar) ...

(y2>=1) && (y2<=tinggi)

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

else

G(y, x) = 0;

end

end

end

G = uint8(G);

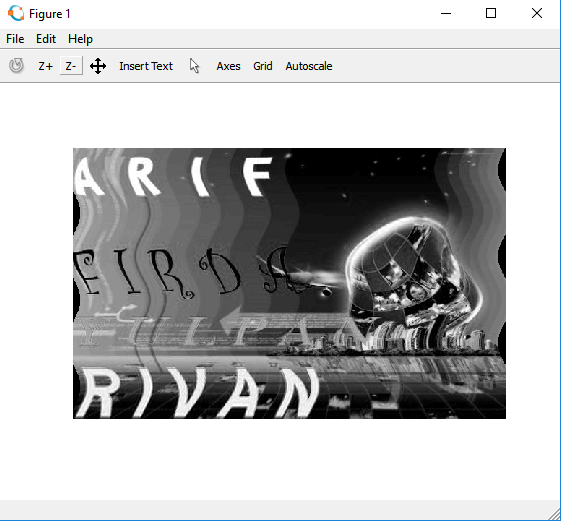
Proses setelah di run ketik code ini di command window :

>> F = imread(' C:\Users\rivan\Pictures\teknologi.jpg ');

>> G = ripple(F, 10, 15, 120, 250);

>> imshow(G);

**Output :**



1. **Nalar**

**Source code :**

%NALAR Contoh penggunaan NOT, AND, OR, XOR dan kombinasinya.

Lingkaran = imread('E:\lingkaran.png');

Persegi = imread('E:\persegi.png');

close all;

Citra1 = Lingkaran;

subplot(3,3,1); imshow(Citra1, [0 1]);

tittle ('A');

Citra2 = Persegi;

subplot(3,3,2); imshow(Citra2, [0 1]);

tittle ('B');

Citra3 = not(Lingkaran);

subplot(3,3,3); imshow(Citra3, [0 1]);

tittle ('not(A)');

Citra4 = and(Lingkaran, Persegi);

subplot(3,3,4); imshow(Citra4, [0 1]);

tittle ('and(A, B)');

Citra5 = xor(Lingkaran, Persegi);

subplot(3,3,5); imshow(Citra5, [0 1]);

tittle ('xor(A, B)');

Citra6 = or(Lingkaran, Persegi);

subplot(3,3,6); imshow(Citra6, [0 1]);

tittle ('or(A, B)');

Citra7 = not(and(Lingkaran, Persegi));

subplot(3,3,7); imshow(Citra7, [0 1]);

tittle ('not(and(A, B))');

Citra8 = not(xor(Lingkaran, Persegi));

subplot(3,3,8); imshow(Citra8, [0 1]);

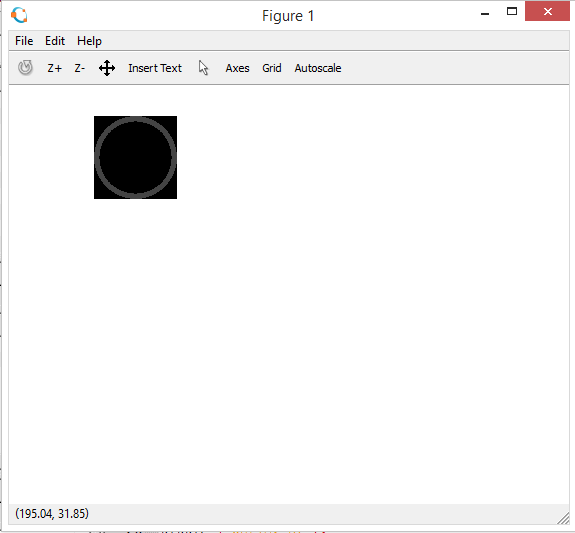
tittle ('not(xor(A, B))');

Citra9 = not(or(Lingkaran, Persegi));

subplot(3,3,9); imshow(Citra9, [0 1]);

tittle ('not(or(A, B))');

**Output :**



1. **Dilasi**

**Source code :**

function G = dilasi(F, H, hotx, hoty)

% DILASI Berguna untuk melaksanakan operasi dilasi.

% Masukan:

% F = citra yang akan dikenai dilasi

% H = elemen pentruksur

% (hy, hx) koordinat pusat piksel

[th, lh]=size(H);

[tf, lf]=size(F);

if nargin < 3

hotx = round(lh/2);

hoty = round(th/2);

end

Xh = [];

Yh = [];

jum\_anggota = 0;

% Menentukan koordinat piksel bernilai 1 pada H

for baris = 1 : th

for kolom = 1 : lh

if H(baris, kolom) == 1

jum\_anggota = jum\_anggota + 1;

Xh(jum\_anggota) = -hotx + kolom;

Yh(jum\_anggota) = -hoty + baris;

end

end

end

G = zeros(tf, lf); % Nolkan semua pada hasil dilasi

% Memproses dilasi

for baris = 1 : tf

for kolom = 1 : lf

for indeks = 1 : jum\_anggota

if F(baris, kolom) == 1

xpos = kolom + Xh(indeks);

ypos = baris + Yh(indeks);

if (xpos >= 1) && (xpos <= lf) && ...

(ypos >= 1) && (ypos <= tf)

G(ypos, xpos) = 1;

end

end

end

end

end

Proses setelah di run ketik code ini di command window :

>> F = [ 0 0 0 0 0;

0 1 1 1 0;

0 1 1 1 0;

0 1 0 0 0;

0 0 0 0 0];

>> H = [ 0 1 0;

0 1 0;

0 1 0];

>> G = dilasi(F,H)

**Output :**

G =

0 1 1 1 0

0 1 1 1 0

0 1 1 1 0

0 1 1 1 0

0 1 0 0 0

1. **Erosi**

**Source code :**

function G = erosi(F, H, hotx, hoty)

% EROSI Berguna untuk melaksanakan operasi erosi.

% Masukan:

% F = citra yang akan dikenai dilasi

% H = elemen pentruksur

% (hy, hx) koordinat pusat piksel

[th, lh]=size(H);

[tf, lf]=size(F);

if nargin < 3

hotx = round(lh/2);

hoty = round(th/2);

end

Xh = [];

Yh = [];

jum\_anggota = 0;

% Menentukan koordinat piksel bernilai 1 pada H

for baris = 1 : th

for kolom = 1 : lh

if H(baris, kolom) == 1

jum\_anggota = jum\_anggota + 1;

Xh(jum\_anggota) = -hotx + kolom;

Yh(jum\_anggota) = -hoty + baris;

end

end

end

G = zeros(tf, lf); % Nolkan semua pada hasil erosi

% Memproses erosi

for baris = 1 : tf

for kolom = 1 : lf

cocok = true;

for indeks = 1 : jum\_anggota

xpos = kolom + Xh(indeks);

ypos = baris + Yh(indeks);

if (xpos >= 1) && (xpos <= lf) && ...

(ypos >= 1) && (ypos <= tf)

if F(ypos, xpos) ~= 1

cocok = false;

break;

end

else

cocok = false;

end

end

if cocok

G(baris, kolom) = 1;

end

end

end

Proses setelah di run ketik code ini di command window :

>> F = [ 0 0 0 0 0;

0 1 1 1 0;

0 1 1 1 0;

0 1 0 0 0;

0 0 0 0 0];

>> H = [ 0 1 0;

0 1 0;

0 1 0];

>> G = erosi(F,H)

**Output :**

G =

0 0 0 0 0

0 0 0 0 0

0 1 0 0 0

0 0 0 0 0

0 0 0 0 0

**11. Opening dan Closing**

Dalam melakukan opening dan closing harus ada source code gerosi dan gdilasi.

**gerosi**

Source code :

function G = gerosi(F, H, hotx, hoty)

% GEROSI Berguna untuk melaksanakan operasi dilasi

% citra beraras keabuan.

% Masukan:

% F = citra yang akan dikenai erosi

% H = elemen pentruksur

% (hy, hx) koordinat pusat piksel

[th, lh]=size(H);

[tf, lf]=size(F);

if nargin < 3

hotx = round(lh/2);

hoty = round(th/2);

end

G = zeros(tf, lf); % Nolkan semua pada hasil erosi

% Memproses erosi

for baris = 1 : tf

for kolom = 1 : lf

terkecil = 255;

for p=1:th

for q=1:lh

ypos = baris + p - hoty;

xpos = kolom + q - hotx;

if (xpos >= 1) && (xpos <= lf) && ...

(ypos >= 1) && (ypos <= tf)

nilai = F(ypos, xpos) + H(p, q);

if terkecil > nilai

terkecil = nilai;

end

end

end

end

% Berikan nilai ke G

if terkecil < 0

terkecil = 0;

end

G(baris, kolom) = terkecil;

end

end

G = uint8(G);

**gdilasi**

Source code :

function G = gdilasi(F, H, hotx, hoty)

% GDILASI Berguna untuk melaksanakan operasi dilasi pada

% citra beraras keabuan.

% Masukan:

% F = citra yang akan dikenai dilasi

% H = elemen pentruksur

% (hy, hx) koordinat pusat piksel

[th, lh]=size(H);

[tf, lf]=size(F);

if nargin < 3

hotx = round(lh/2);

hoty = round(th/2);

end

G = zeros(tf, lf); % Nolkan semua pada hasil dilasi

% Memproses dilasi

for baris = 1 : tf

for kolom = 1 : lf

terbesar = 0;

for p=1:th

for q=1:lh

ypos = baris - (p - hoty);

xpos = kolom- (q - hotx);

if (xpos >= 1) && (xpos <= lf) && ...

(ypos >= 1) && (ypos <= tf)

nilai = F(ypos, xpos) + H(p, q);

if terbesar < nilai

terbesar = nilai;

end

end

end

end

% Potong nilai terbesar kalau melebihi 255

if terbesar > 255

terbesar = 255;

end

% Berikan nilai terbesar ke G

G(baris, kolom) = terbesar;

end

end

G = uint8(G);

**gclosing**

Source code :

function G = gclosing(F, H)

% GCLOSING Berguna untuk melaksanakan operasi closing

% citra beraras keabuan.

% Masukan:

% F = citra yang akan dikenai erosi

% H = elemen pentruksur

G = gerosi(gdilasi(F, H), H);

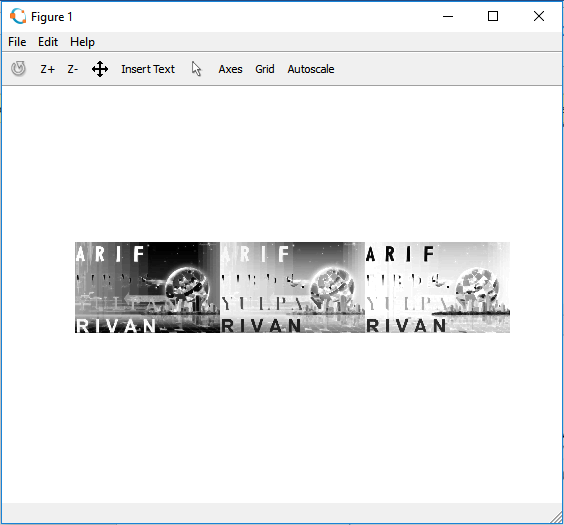
Proses setelah di run ketik code ini di command window :

>> Img = imread(' C:\Users\rivan\Pictures\teknologi.jpg ');

>> G = gclosing(Img, ones(5));

>> imshow(G)

**Output :**



**gopening**

**Source code :**

function G = gopening(F, H)

% GOPENING berguna untuk melaksanakan operasi opening

% citra beraras keabuan

% Masukan:

% F = citra yang

% H = elemen pentruksur

G = gdilasi(gerosi(F, H), H);

Proses setelah di run ketik code ini di command window :

>> Img = imread(' C:\Users\rivan\Pictures\teknologi.jpg ');

>> G = gopening(Img, ones(5));

>> imshow(G)

**Output :**

