**LAPORAN PENGELOLAAN CITRA**

*Operasi Citra Dengan Program Octave*

**Kelompok :**

Iwan Ridwanulloh 1306072

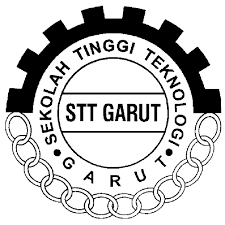
Firman Firmansyah 1406058

Muhammad Setyadi F.A 1406086

Taufik Gunawan 1406126

Nevia Yusuf 1506142

Sandi Sopiandi 1506145



**Teknik Informatika A**

**SEKOLAH TINGGI TEKNOLOGI GARUT**

Jl. Mayor Syamsu No.1 Telp.(0262)232773-FAX.(0262)232332 Tarogong Kidul Garut - 44151

**PROGRAM OCTAVE**

1. **Menggeser Citra**

F = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

[tinggi, lebar] = size(F);

sx = 45; % Penggeseran arah horisontal

sy = -45; % Penggeseran arah vertikal

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

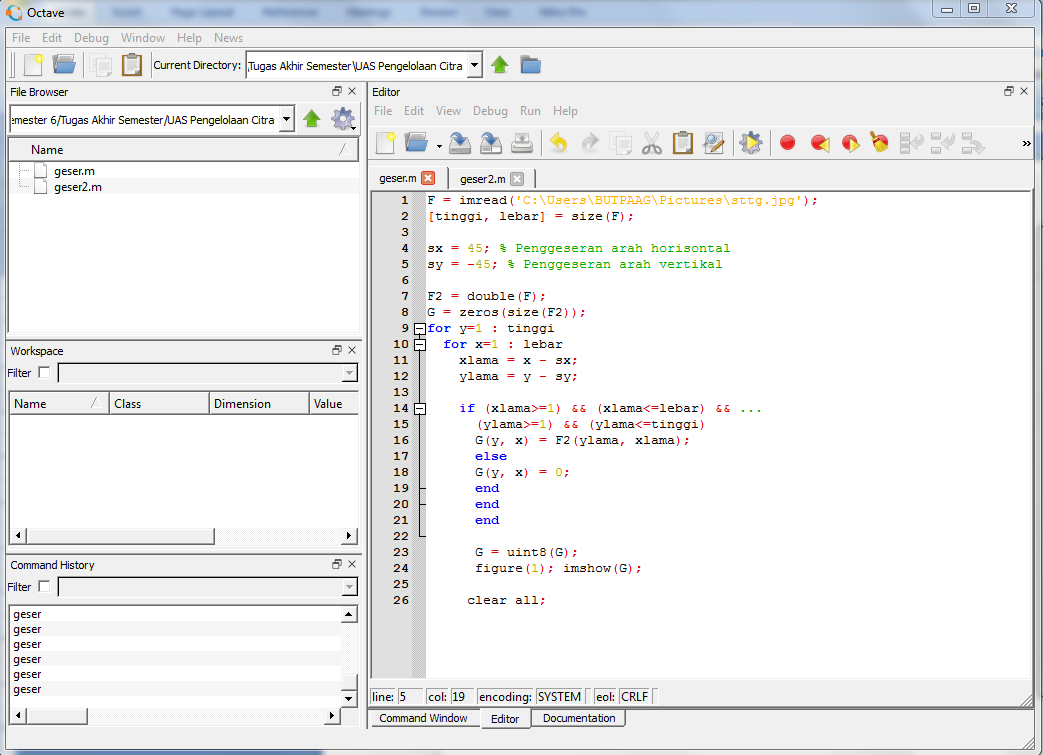
end

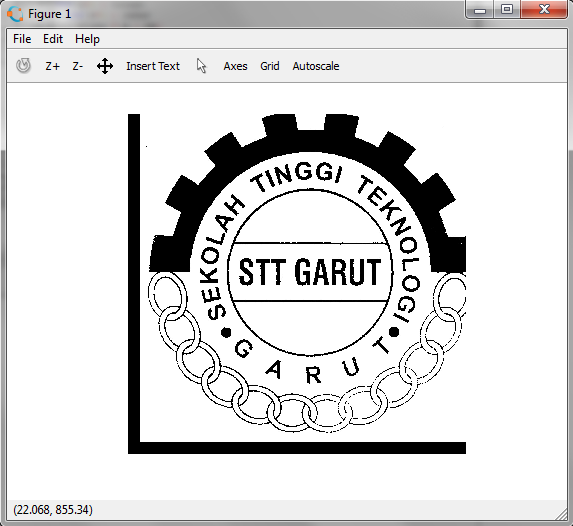
end

G = uint8(G);

figure(1); imshow(G);

clear all;





1. **Menggeser Citra 2**

F = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

[tinggi, lebar] = size(F);

sx = 45; % Penggeseran arah horisontal

sy = -35; % Penggeseran arah vertikal

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

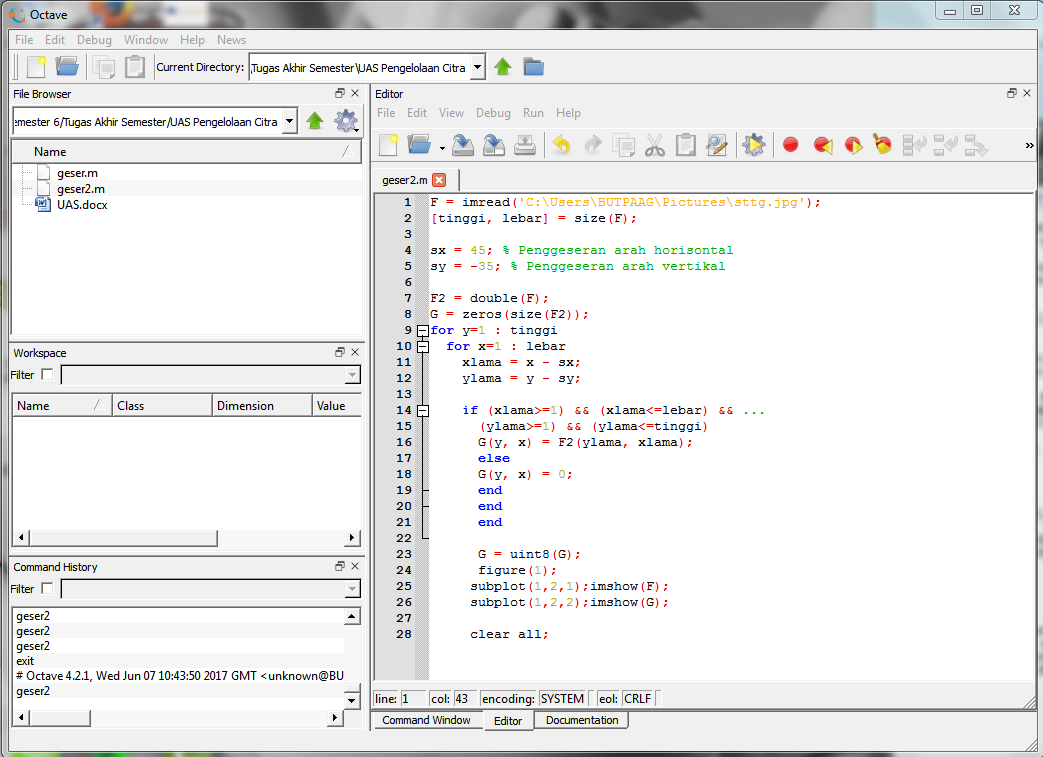
G = uint8(G);

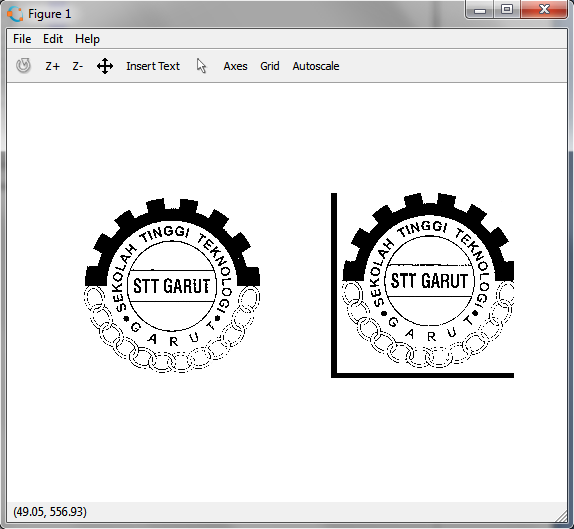
figure(1);

subplot(1,2,1);imshow(F);

subplot(1,2,2);imshow(G);

clear all;





1. **Merotasikan Citra**

F = imread('C:\Users\BUTPAAG\Pictures\sttg.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;

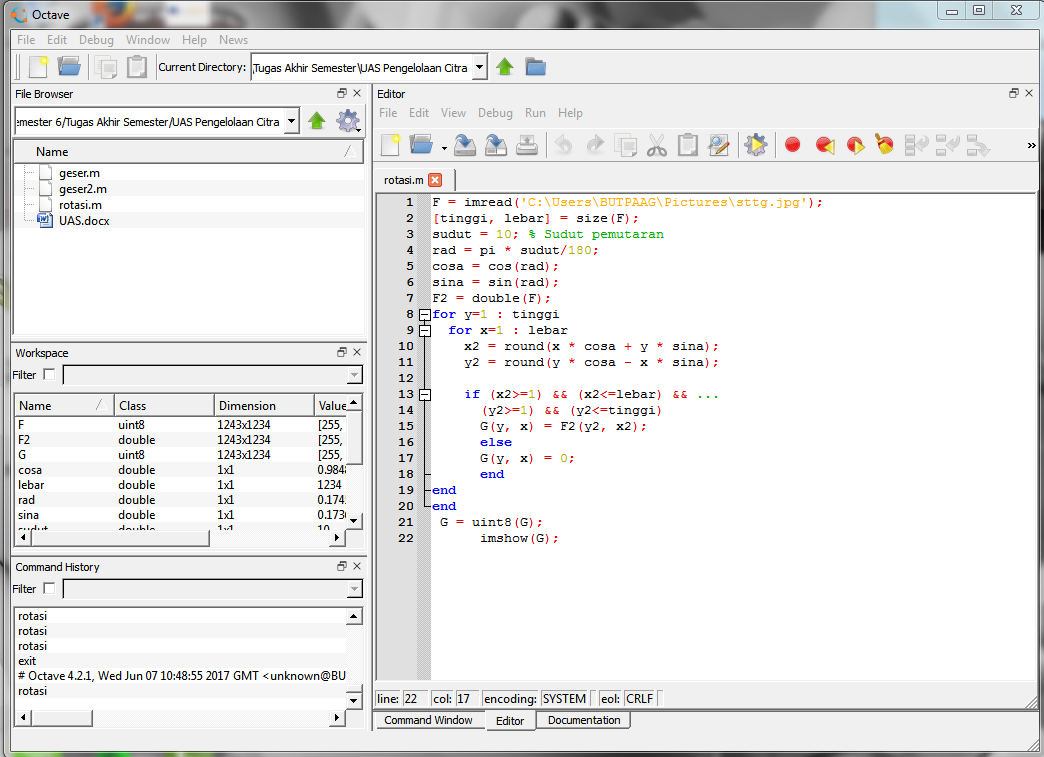
end

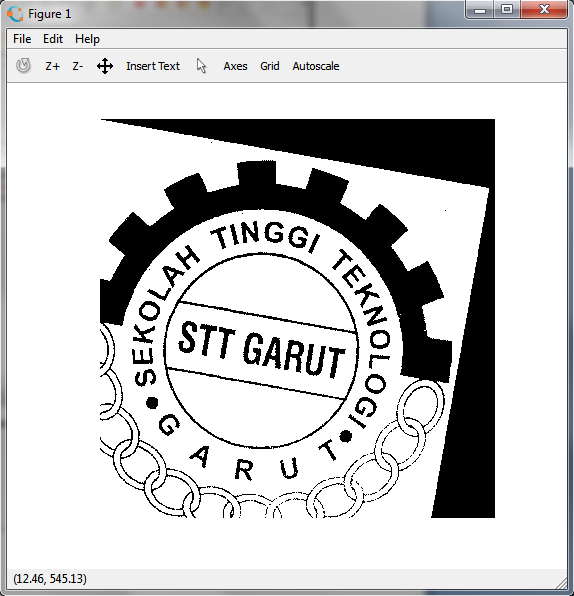
end

end

G = uint8(G);

imshow(G);





1. **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('C:\Users\BUTPAAG\Pictures\sttg.jpg');

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

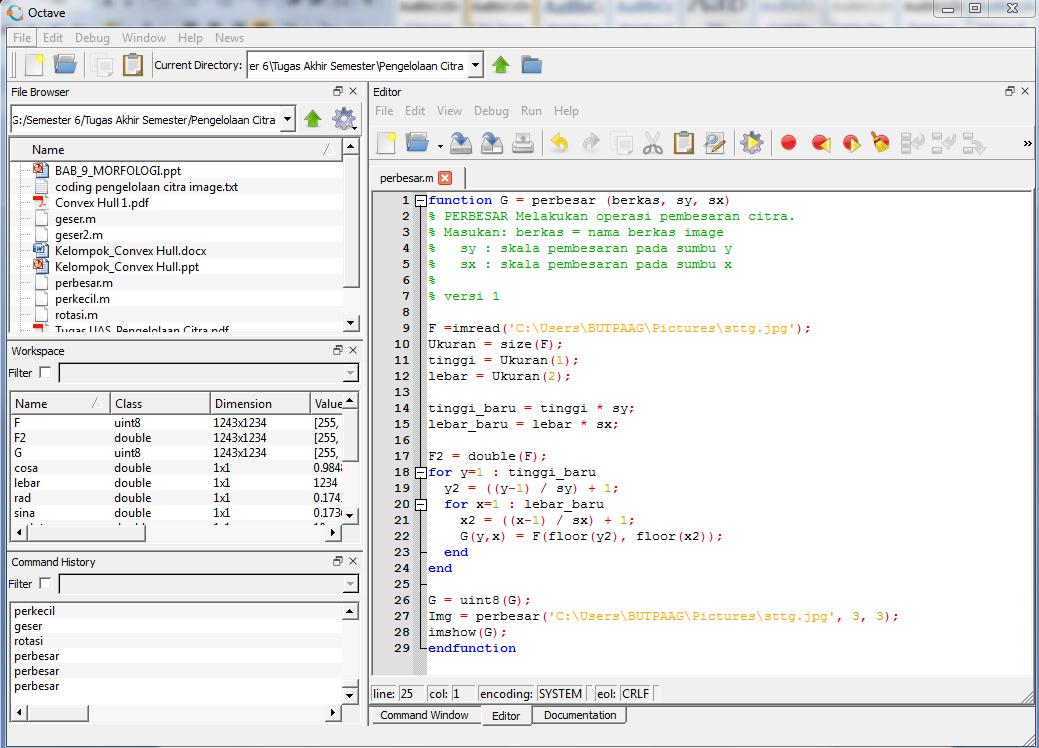
end

G = uint8(G);

Img = perbesar('C:\Users\BUTPAAG\Pictures\sttg.jpg', 3, 3);

imshow(G);

endfunction



function G = perbesar2(berkas, sy, sx)

% PERBESAR Melakukan operasi pembesaran citra

% dengan interpolasi.

% Masukan: berkas = nama berkas image

% sy : skala pembesaran pada sumbu y

% sx : skala pembesaran pada sumbu x

% versi 2

F = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

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;

% 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) + ...

G(y,x) = intensitas;

end

end

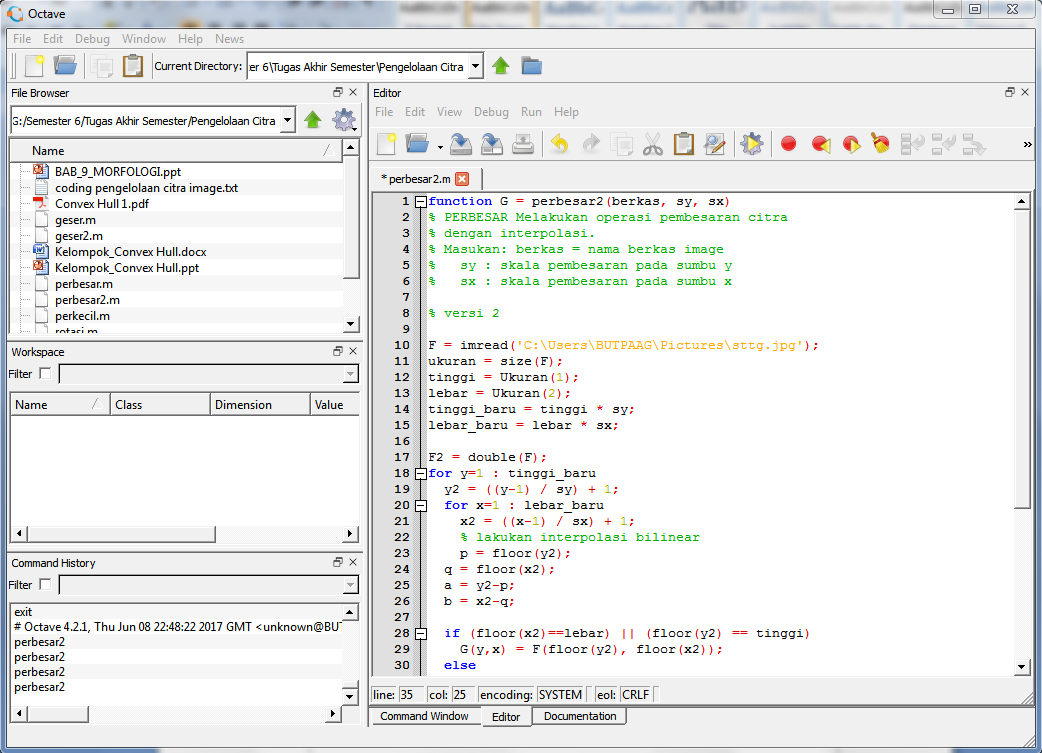
end

G = uint8(G);

Img = perbesar('C:\Users\BUTPAAG\Pictures\sttg.jpg', 4, 4);

imshow(G);

endfunction



1. **Memperkecil Citra**

function G = Memperkecil2(F, sy, sx)

% Memperkecil2 Melakukan operasi Memperkecil citra

% dengan interpolasi.

% Masukan: berkas = nama berkas image

% sy : skala pembesaran pada sumbu Y

% sx : skala pembesaran pada sumbu X

%

% Versi 2

Ukuran = size(F);

tinggi = Ukuran(1);

lebar = Ukuran(2);

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;

% 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);

end

clc;

clear;

im = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

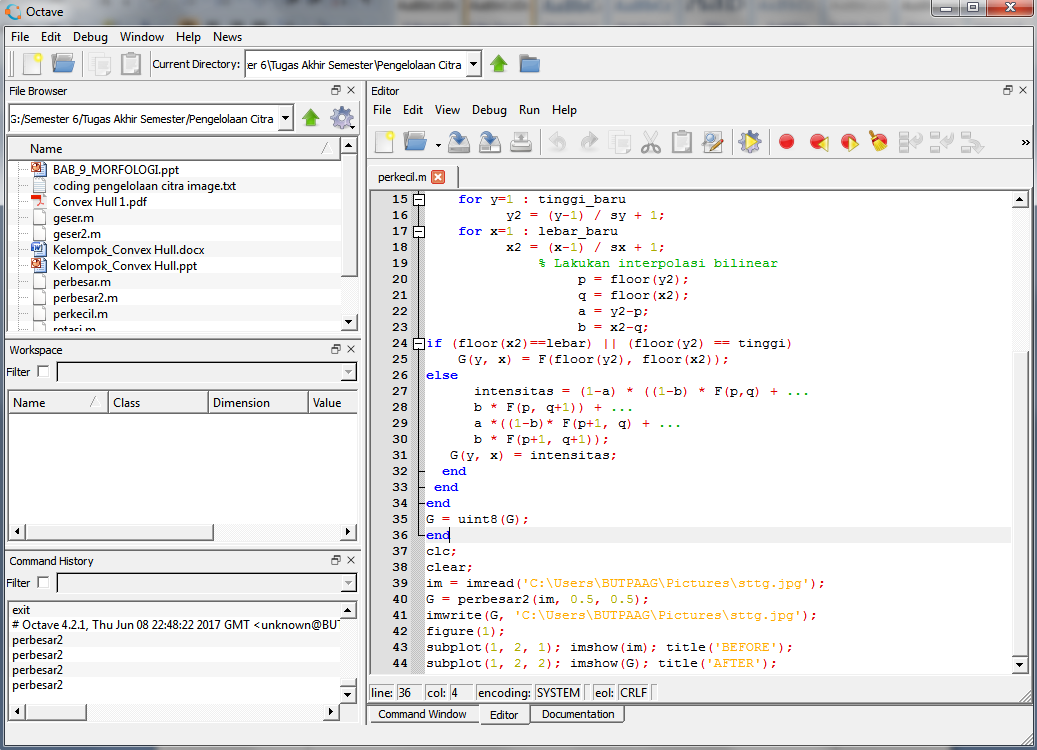
G = perbesar2(im, 0.5, 0.5);

imwrite(G, 'C:\Users\BUTPAAG\Pictures\sttg.jpg');

figure(1);

subplot(1, 2, 1); imshow(im); title('BEFORE');

subplot(1, 2, 2); imshow(G); title('AFTER');



1. **Pencerminan Citra**

function G = cerminh(F)

% CERMINH Berfungsi untuk mencerminkan citra

% secara horizontal

% Masukan: F = Citra berskala keabuan

F = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

[tinggi, lebar] = size(F);

for y=1 : tinggi

for x=1 : lebar

x2 = lebar - x + 1;

y2 = y;

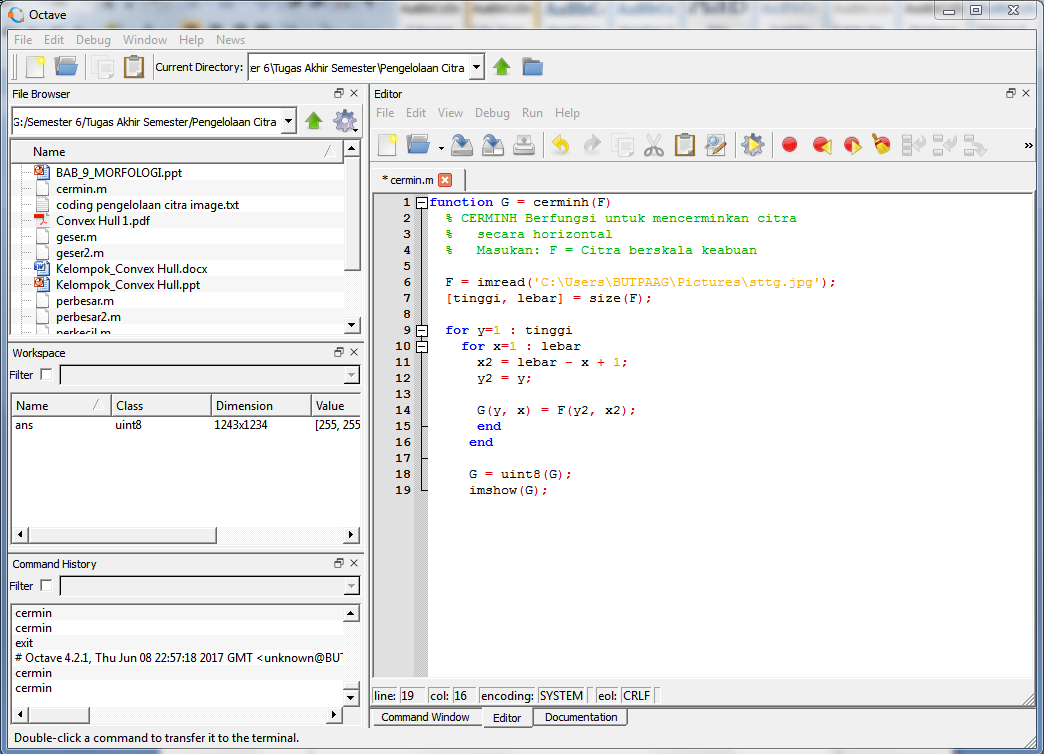
G(y, x) = F(y2, x2);

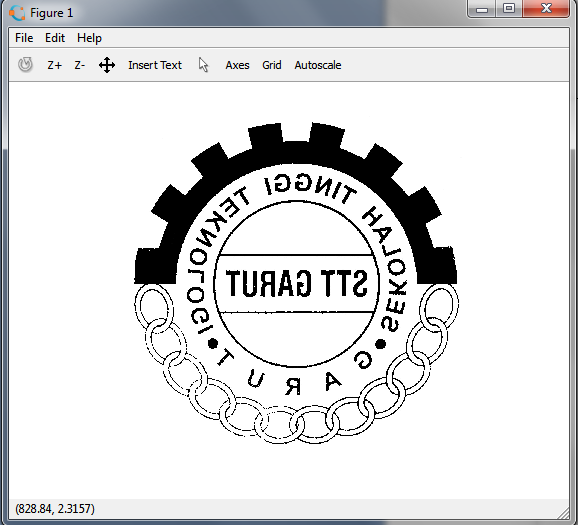
end

end

G = uint8(G);

imshow(G);





function G = cerminv(F)

% CERMINV Berfungsi untuk mencerminkan citra

% secara vertikal

% Masukan: F = Citra berskala keabuan

F = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

[tinggi, lebar] = size(F);

for y=1 : tinggi

for x=1 : lebar

x2 = x;

y2 = tinggi - y + 1;

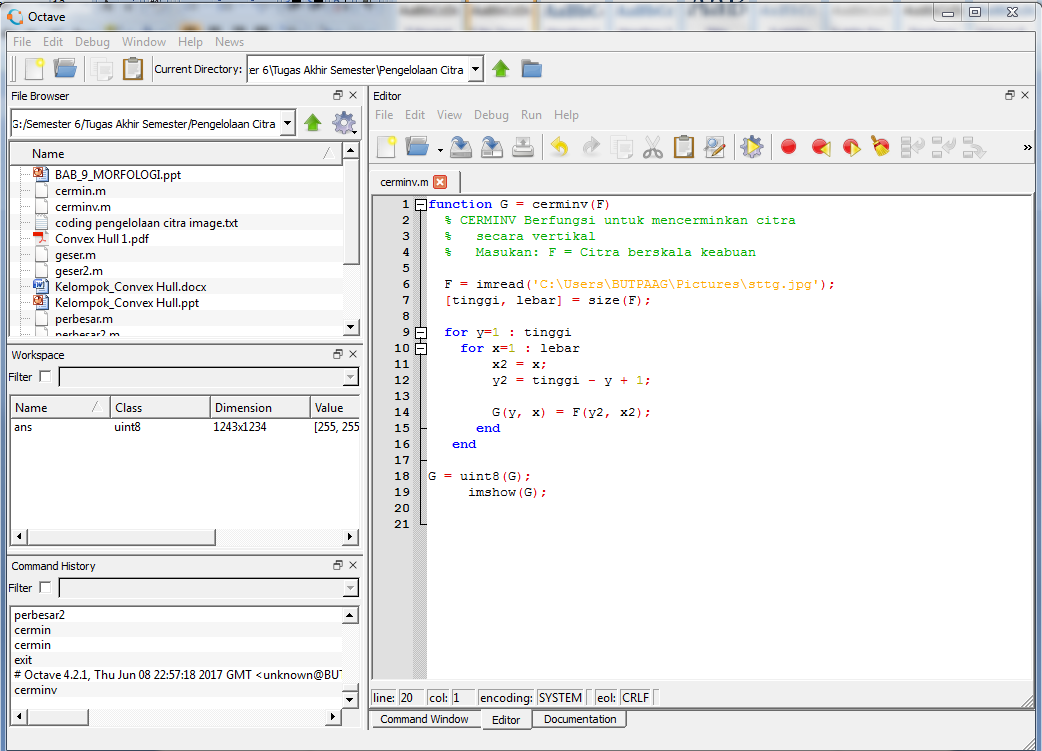
G(y, x) = F(y2, x2);

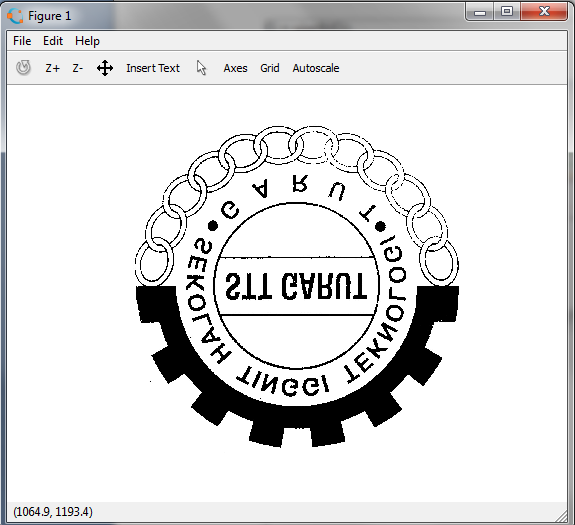
end

end

G = uint8(G);

imshow(G);





1. **Taffine**

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

% TAFFINE Digunakan untuk melakukan transformasi affine.

% Masukan: F = Citra berskala keabuan

% a11, a12, a21, a22, tx, ty = mengatur

% transformasi affine

F = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

[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)

% Lakukan interolasi 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

G(y, x) = 0;

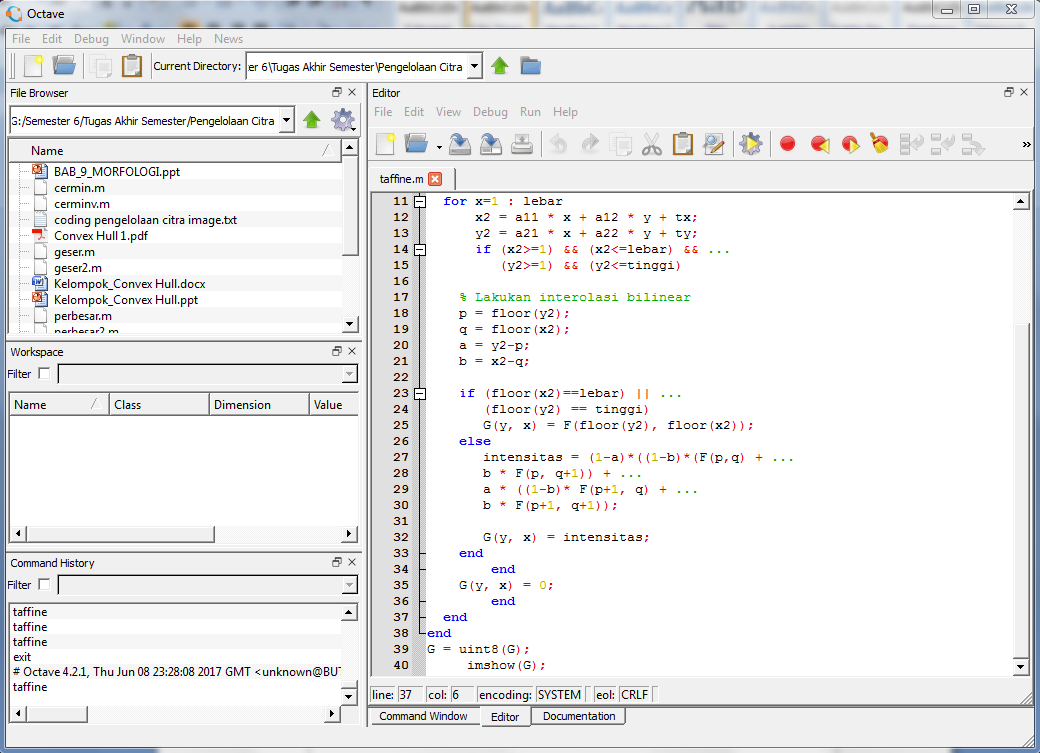
end

end

end

G = uint8(G);

imshow(G);



1. **Ripple**

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

% RIPPLE Berfungsi untuk melakukan transformasi 'ripple'

F = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

[tinggi, lebar] = size(F);

dimensi = size(F);

tinggi = dimensi(1);

lebar = dimensi(2);

for y=1 : tinggi

for x=1 : lebar

x2 = x + ax \* sin(2\* p1 \* y / tx);

y2 = y + ay \* sin(2\* p1 \* x / ty);

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

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

% Lakukan interpolasi bilineaar

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;

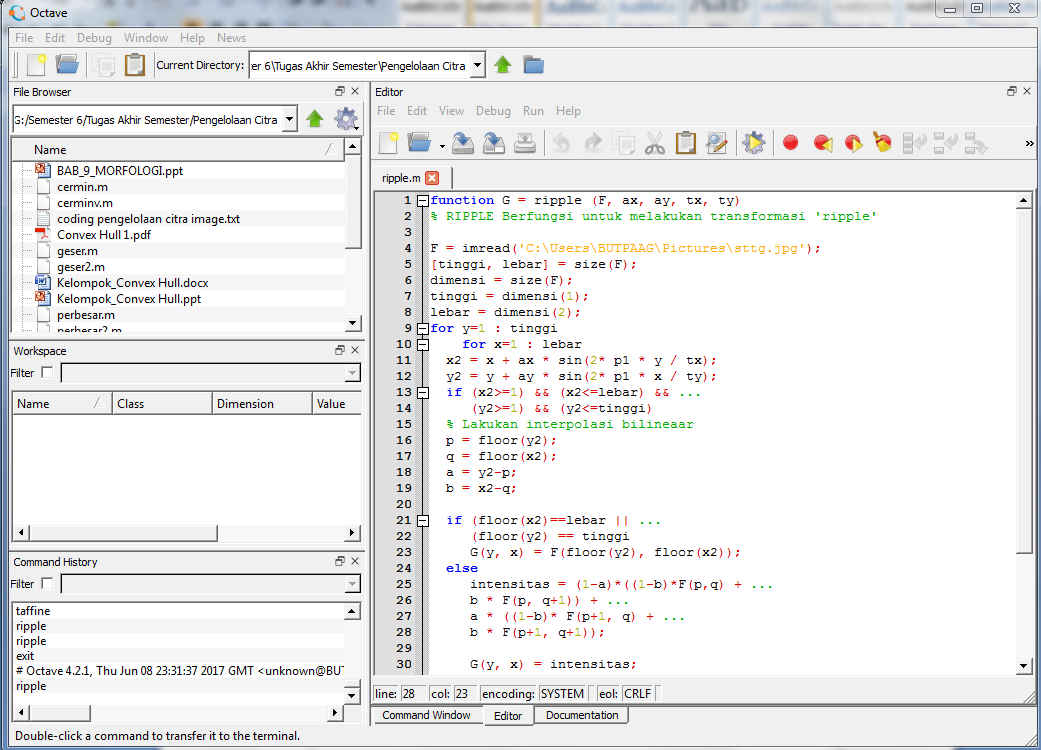
end

end

end

G = uint8(G);

imshow(G);



1. **Nalar**

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

% kombinasinya.

Lingkaran = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

Persegi = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

close all;

Citra1 = Lingkaran;

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

title('A');

Citra2 = Persegi;

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

Citra3 = not (Lingkaran);

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

Citra4 = and(Lingkaran, Persegi);

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

Citra5 = xor(Lingkaran, Persegi);

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

Citra6 = or(Lingkaran, Persegi);

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

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

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

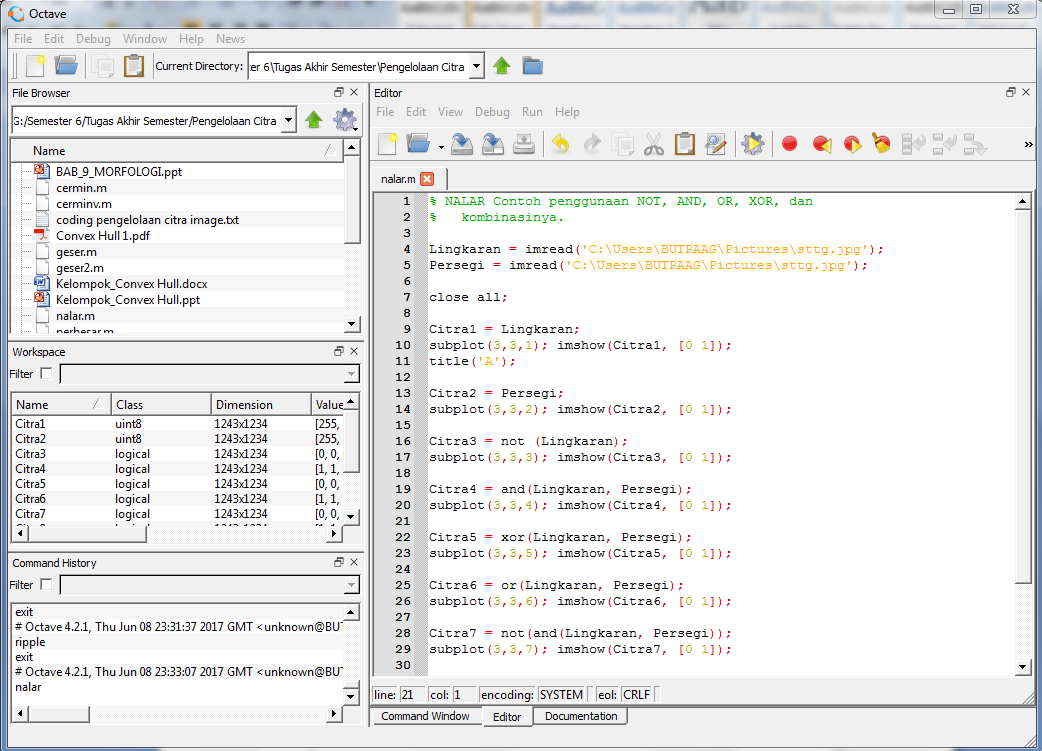
Citra8 = not(xor(Lingkaran, Persegi));

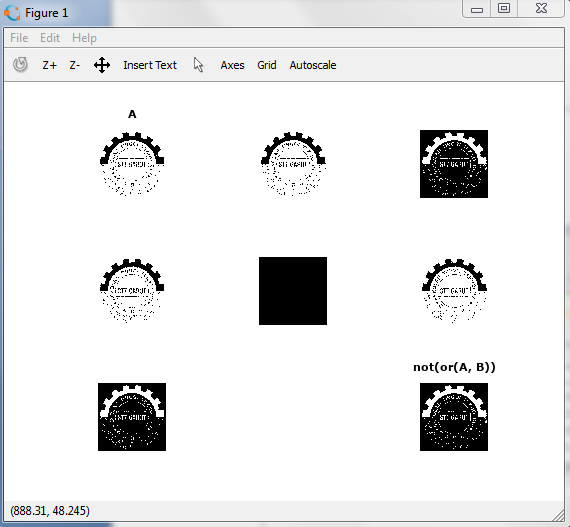
subplot(3,3,8); imshow(Citra8, [0 1]);

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

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

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





1. **Dilasi**

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

F = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

[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\_annggota + 1;

Xh(jum\_anggota) = hotx + kolom;

Yh(jum\_anggota) = hoty + baris;

end

end

end

G = zeros(tf, if); % 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;

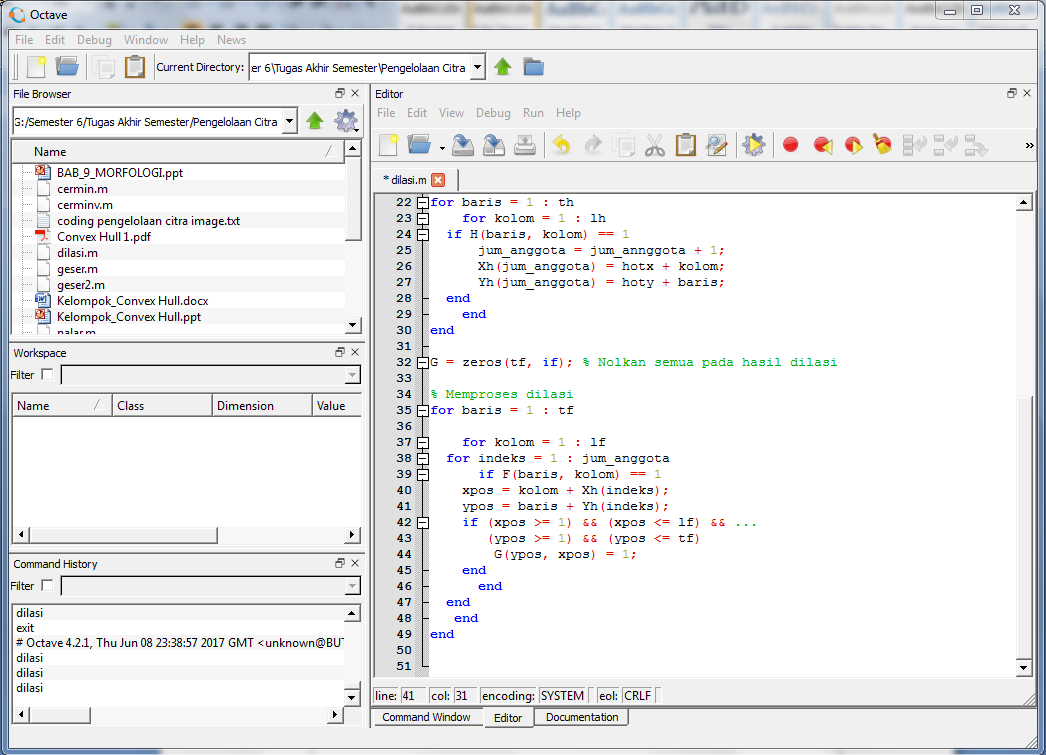
end

end

end

end

end



1. **Erosi**

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

F = imread('C:\Users\BUTPAAG\Pictures\sttg.jpg');

[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

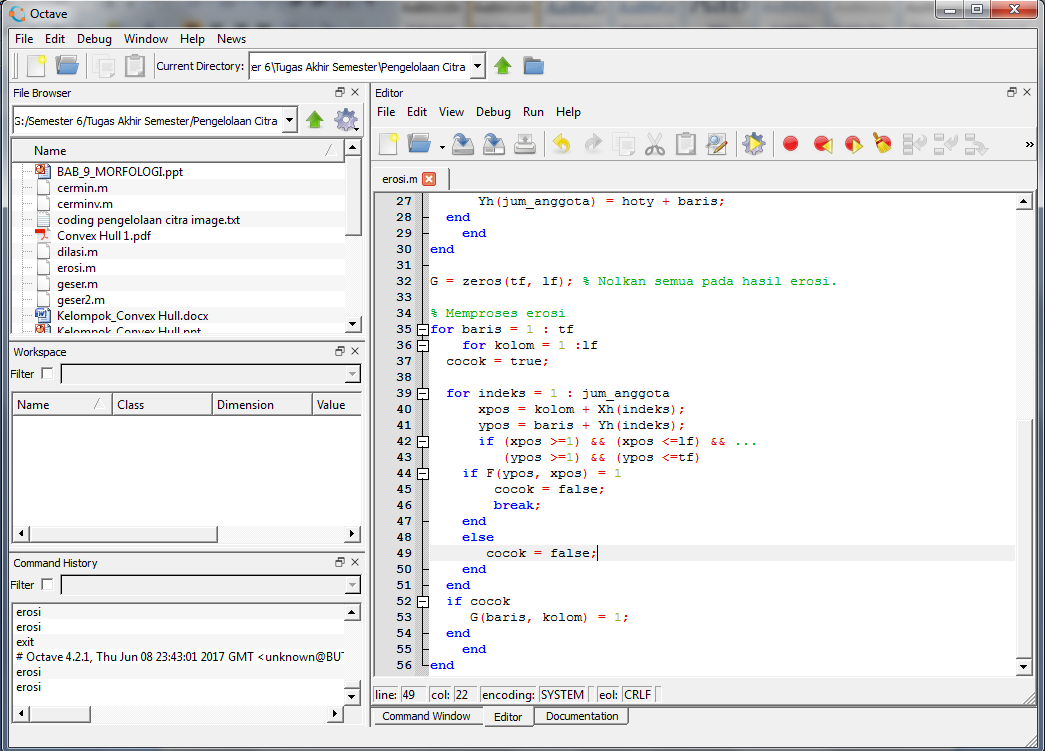
if cocok

G(baris, kolom) = 1;

end

end

end

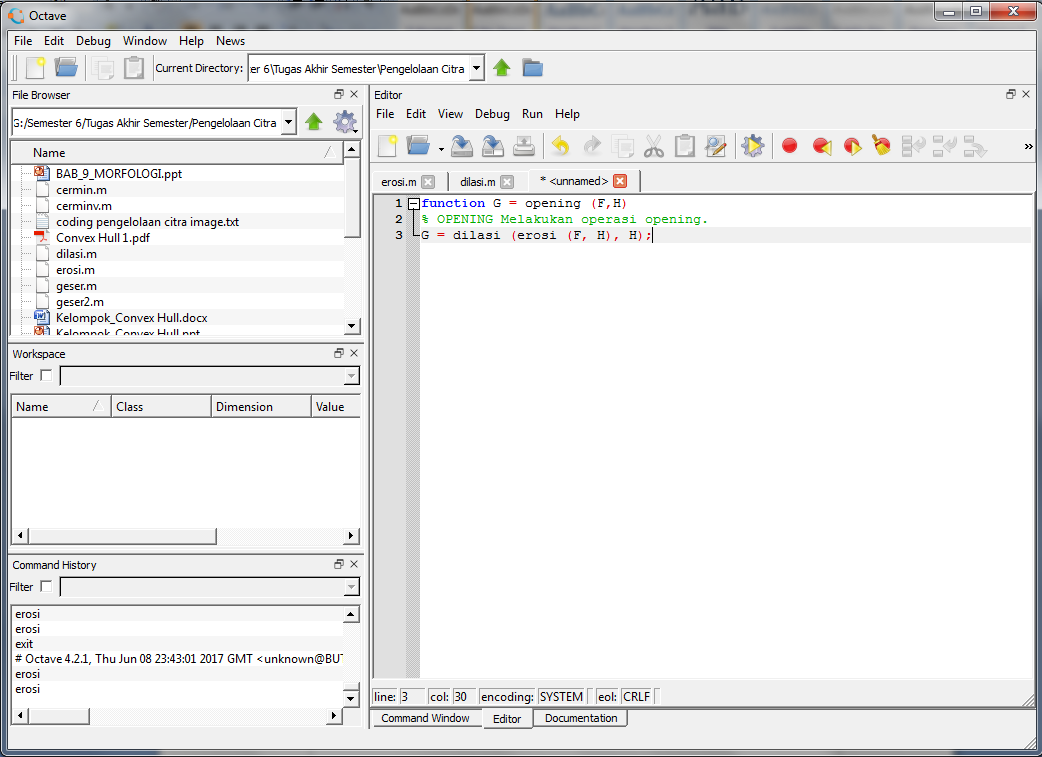


1. **Opening**

function G = opening (F,H)

% OPENING Melakukan operasi opening.

G = dilasi (erosi (F, H), H);



1. **Closing**

function G = closing (F, H)

% CLOSING Melakukan operasi opening.

G = erosi (dilasi (F, H), H);

chain\_code.m

function [kode\_rantai,xawal,yawal] = chain\_code (U)

% CHAIN\_CODE Digunakan untuk mendapatkan titik awal (x, y) dan

% kode rantai dari kontur U yang datanya telah terurutkan

% misalnya melalui get\_contour

% kode 1 2 3 4 5 6 7 8 9

kode =['3', '2', '1', '4', '0', '0', '5', '6', '7'];

xawal = U (1, 2);

yawal = U (1, 1);

kode\_rantai = ' ';

for p-2: length (U)

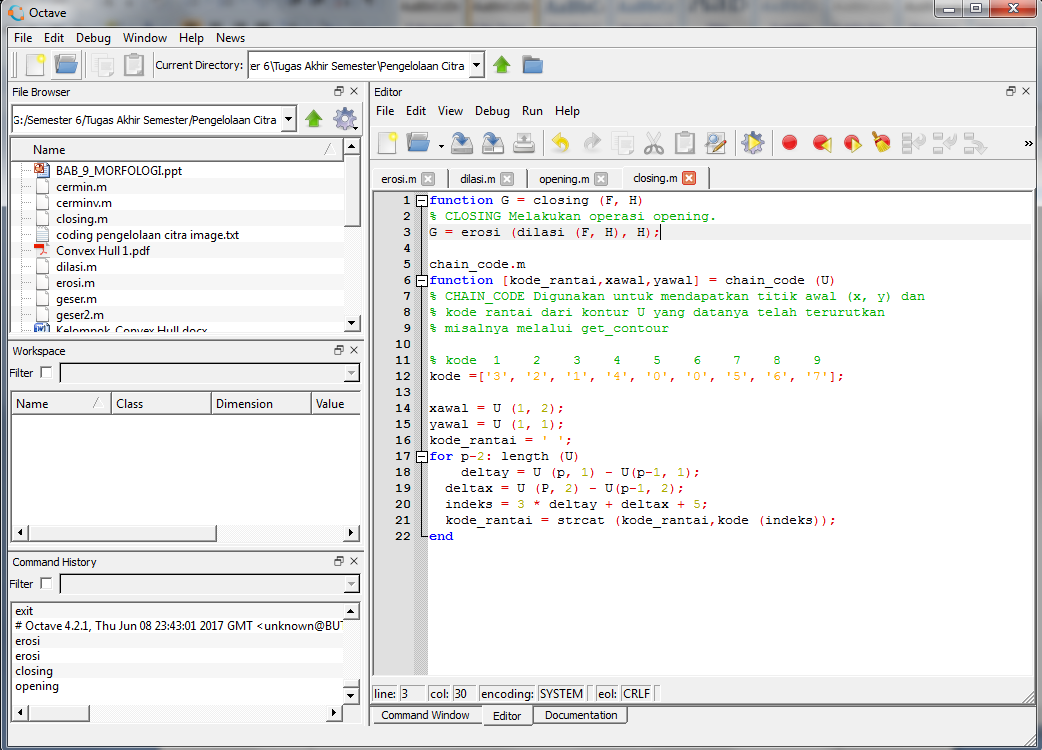
deltay = U (p, 1) - U(p-1, 1);

deltax = U (P, 2) - U(p-1, 2);

indeks = 3 \* deltay + deltax + 5;

kode\_rantai = strcat (kode\_rantai,kode (indeks));

end



1. **Show\_Contour**

show\_contour.m

function show\_contour(x, y, rantai\_code)

% SHOW\_CONTOUR Contoh untuk menggambarkan kontur

% melalui rantai kode

% Masukan fungsi ini adalah hasil

% fungsi chain\_code

% 0 1 2 3 4 5 6 7

Dx = [ +1, +1, 0, -1, -1, -1, 0, +1];

Dy = [ 0, -1, -1, -1, 0, +1, +1, +1];

U = zeros(1,2);

U(1,1) = y;

U(1,2) = x;

for p=2: length(rantai\_kode)

bilangan = rantai\_kode(p) -48;

posx = U(p-1, 2) + Dx(bilangan + 1);

posy = U(p-1, 1) + Dy(bilangan + 1);

U(p, 1) = posy;

U(p, 2) = posy;

end

% Membentuk gambar kontur

maks\_x = max(U(p,2));

maks\_y = max(U(p,1));

D = zeros(maks\_y, maks\_x);

for p=1: length(U)

D(U(p,1), U(p,2)) - 1;

end

imshow(D);

