

LAMPIRAN A

CITRA

A.1. Cover image yang digunakan

Lena512.bmp



Peppers512.bmp



Jet512.bmp



Boat512.bmp

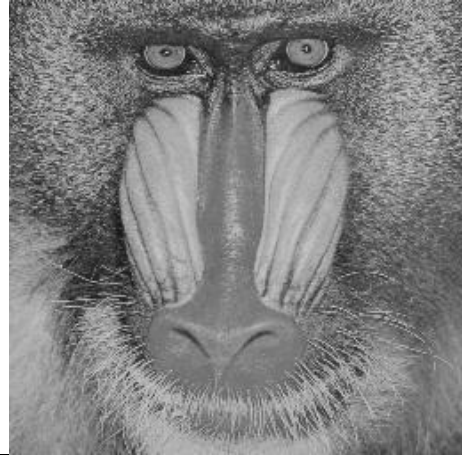


A.2. Secret image yang digunakan

Lena256.bmp



Baboon256.bmp



Bird256.bmp



A.3. Citra Pengujian Kebenaran Proses Penyisipan



A.4. Cover-image "Lena512.bmp" Sebelum dan Setelah Disisipkan Citra 'Lena256.bmp'

Citra asli Lena512.bmp



L_Lena_6.bmp



L_Lena_8.bmp



L_Lena_10.bmp



L_Lena_12.bmp



L_Lena_16.bmp



L_Lena_20.bmp



L_Lena_50.bmp



L_Lena_100.bmp



A.5 Cover-image "Lena512.bmp" Sebelum dan Setelah Disisipkan Citra 'Baboon256.bmp'

Citra asli Lena512.bmp



L_Baboon_6.bmp



L_Baboon_8.bmp



L_Baboon_10.bmp



L_Baboon_12.bmp



L_Baboon_16.bmp



L_Baboon_20.bmp



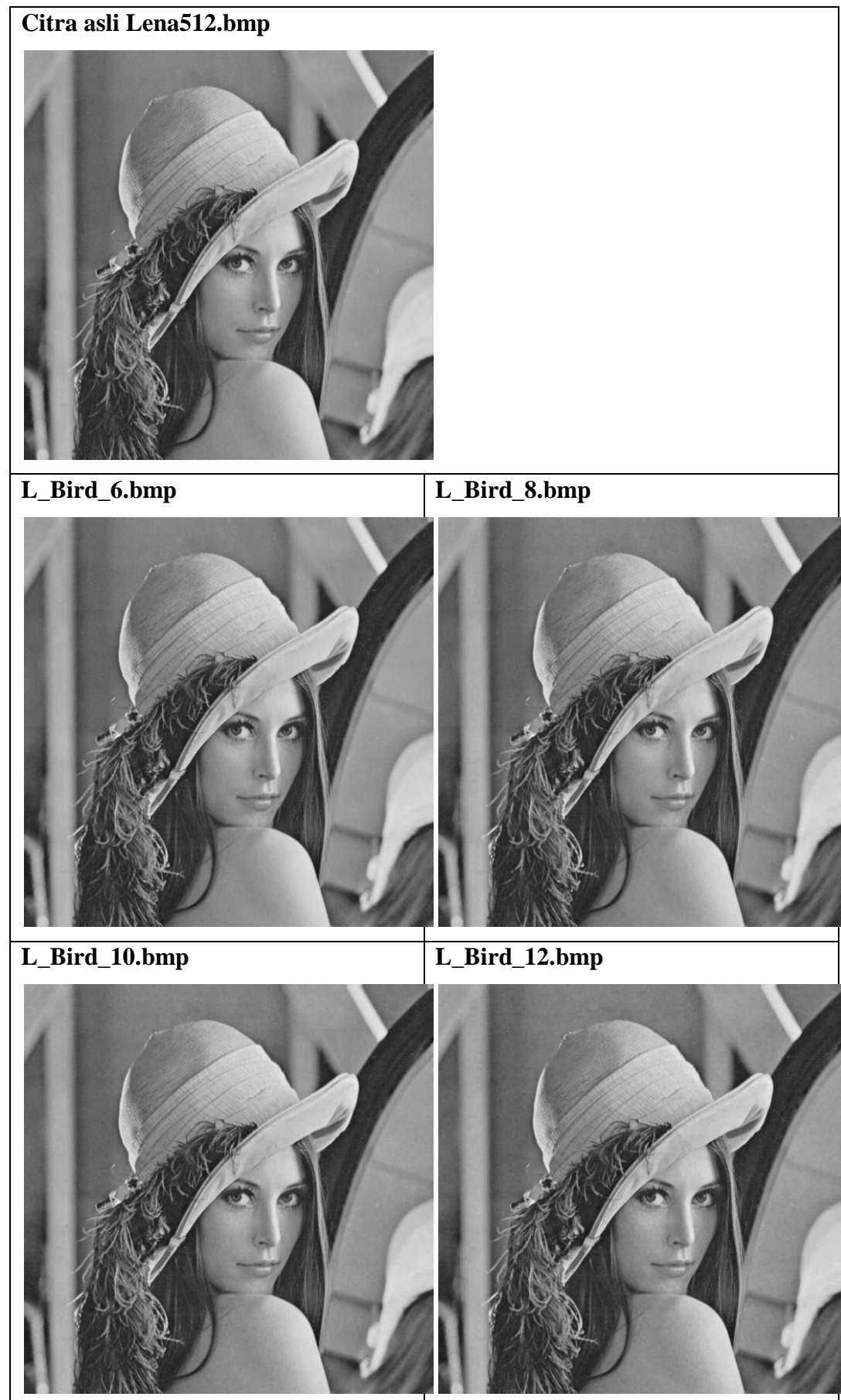
L_Baboon_50.bmp



L_Baboon_100.bmp



A.6 Cover-image "Lena512.bmp" Sebelum dan Setelah Disisipkan Citra 'Bird256.bmp'



L_Bird_16.bmp



L_Bird_20.bmp



L_Bird_50.bmp



L_Bird_100.bmp



A.7 Cover-image "Peppers512.bmp" Sebelum dan Setelah Disisipkan Citra 'Lena256.bmp'

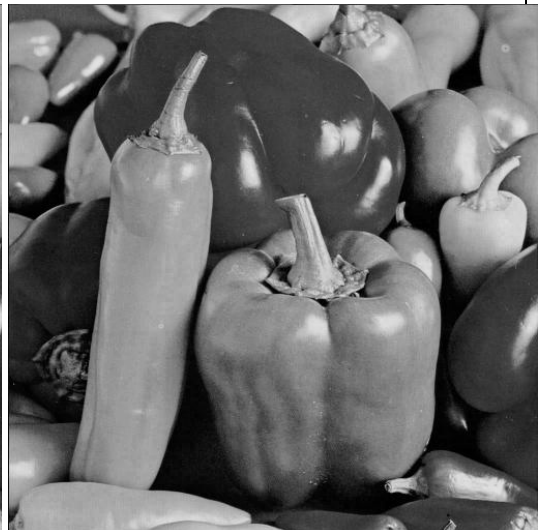
Citra asli Peppers512.bmp



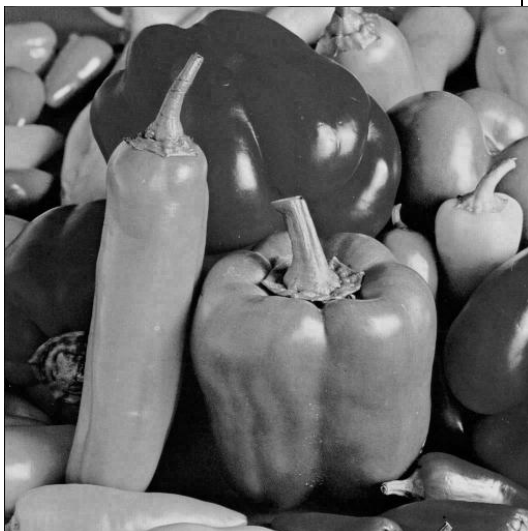
P_Lena_6.bmp



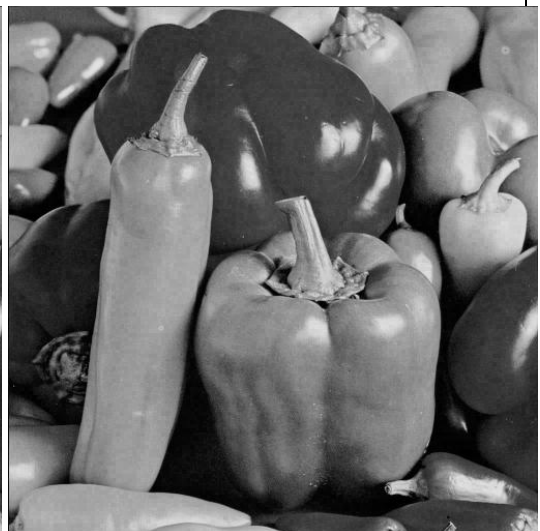
P_Lena_8.bmp



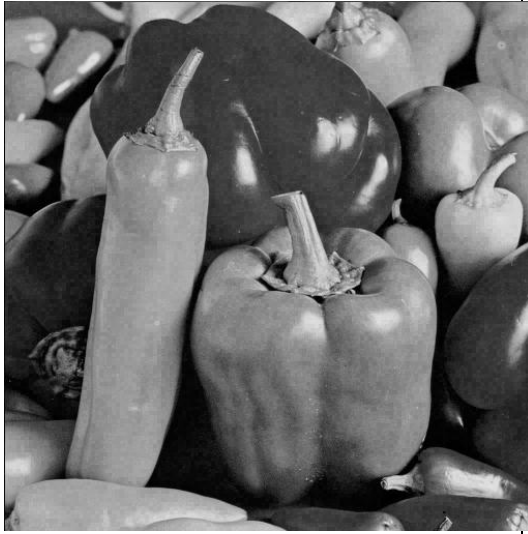
P_Lena_10.bmp



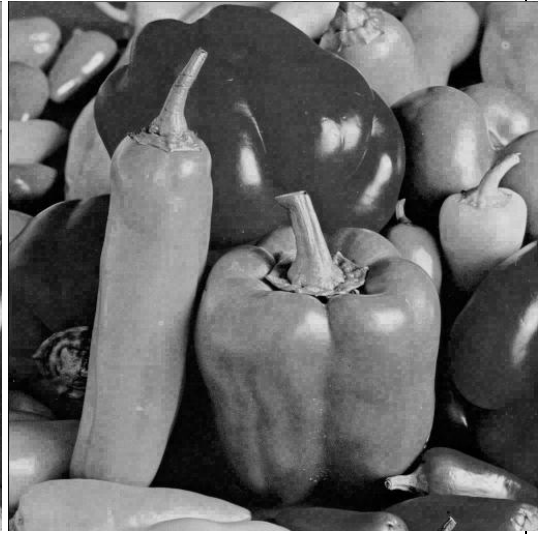
P_Lena_12.bmp



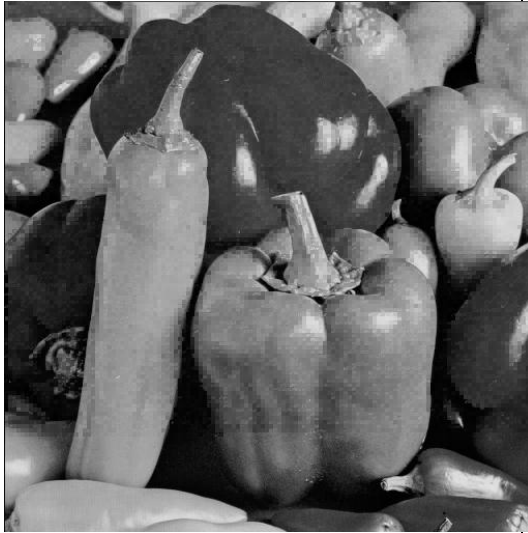
P_Lena_16.bmp



P_Lena_20.bmp



P_Lena_50.bmp



P_Lena_100.bmp



A.8 Cover-image "Peppers512.bmp" Sebelum dan Setelah Disisipkan Citra 'Baboon256.bmp'

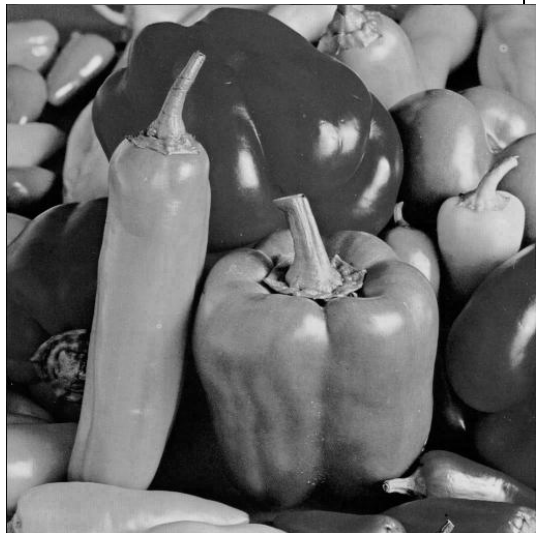
Citra asli Peppers512.bmp



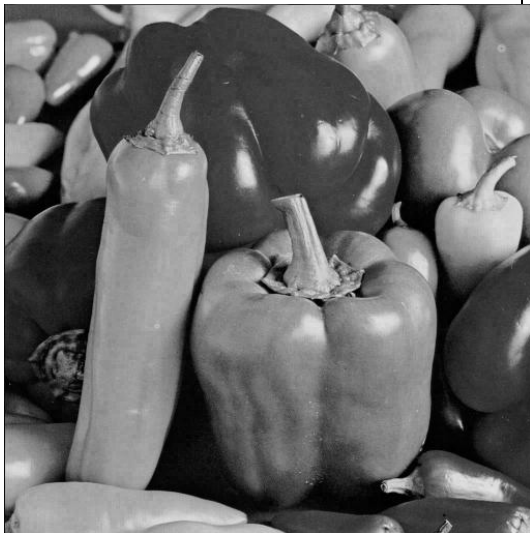
P_Baboon_6.bmp



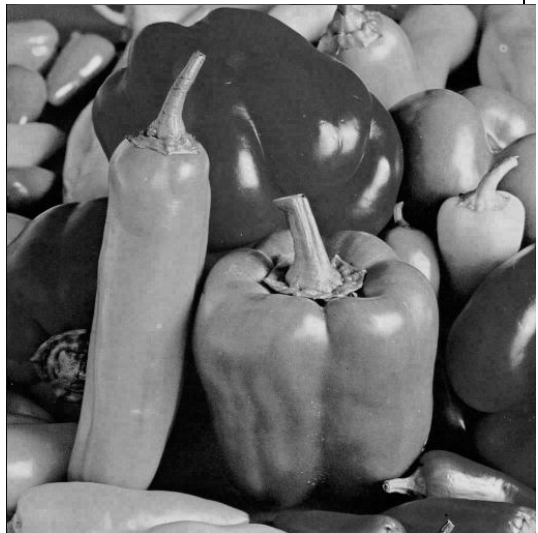
P_Baboon_8.bmp



P_Baboon_10.bmp



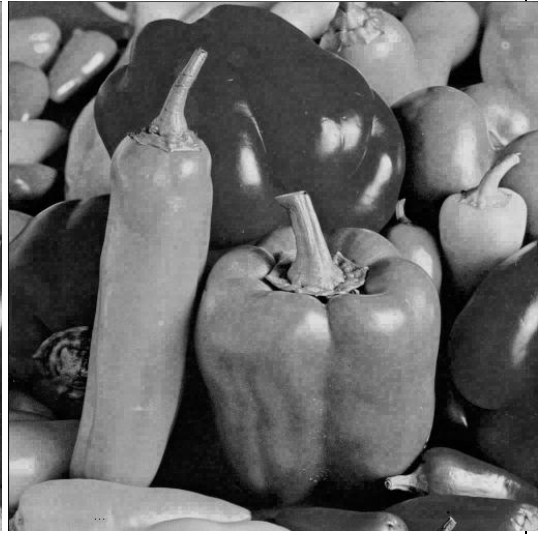
P_Baboon_12.bmp



P_Baboon_16.bmp



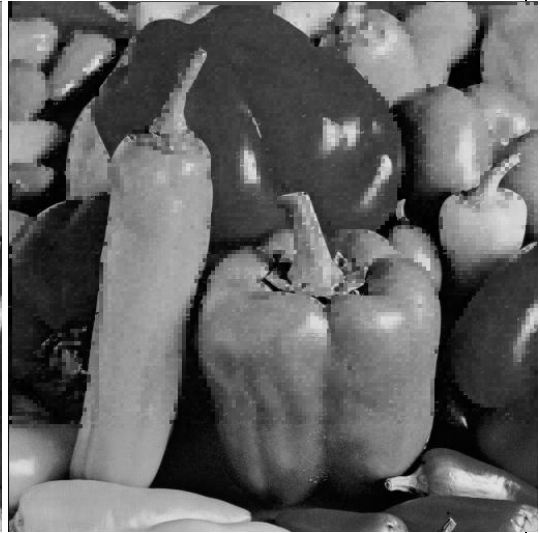
P_Baboon_20.bmp



P_Baboon_50.bmp



P_Baboon_100.bmp

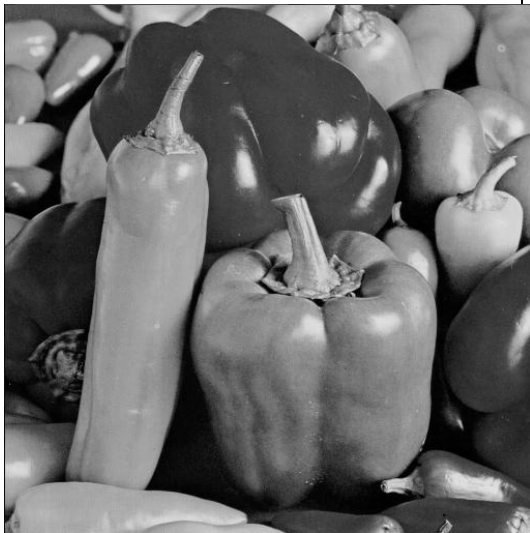


A.9 Cover-image "Peppers512.bmp" Sebelum dan Setelah Disisipkan Citra 'Bird256.bmp'

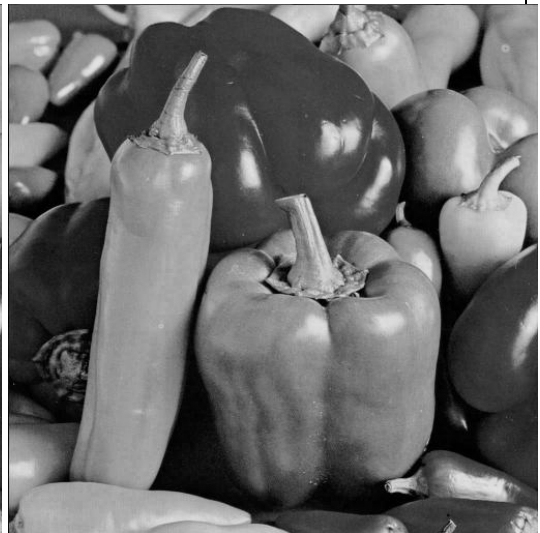
Citra asli Peppers512.bmp



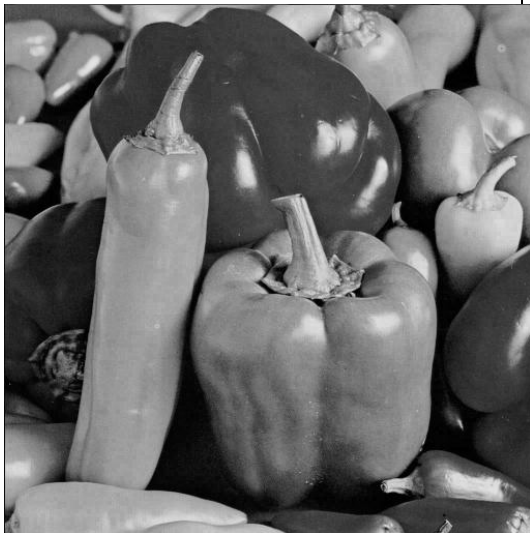
P_Bird_6.bmp



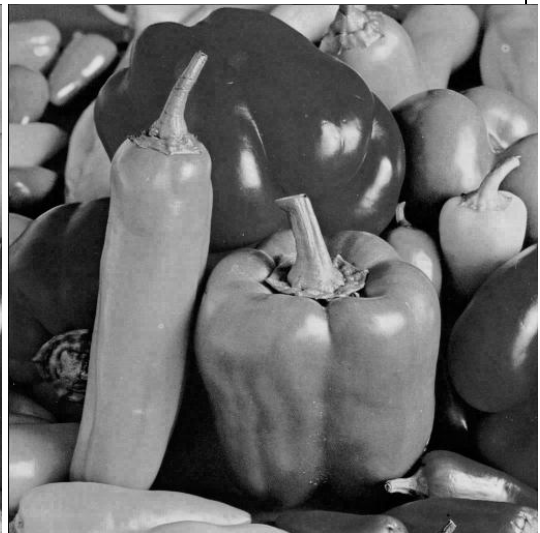
P_Bird_8.bmp



P_Bird_10.bmp



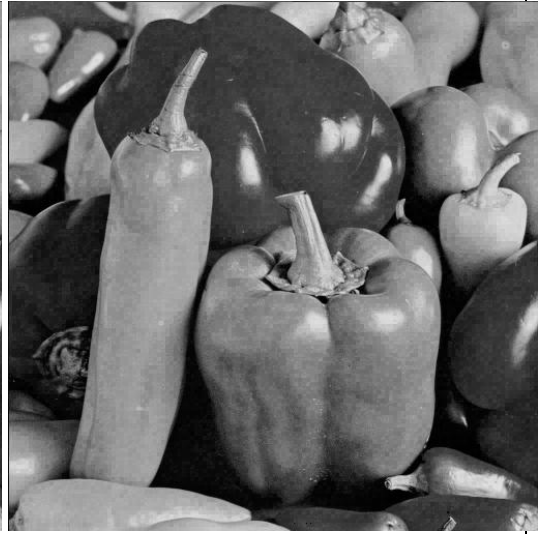
P_Bird_12.bmp



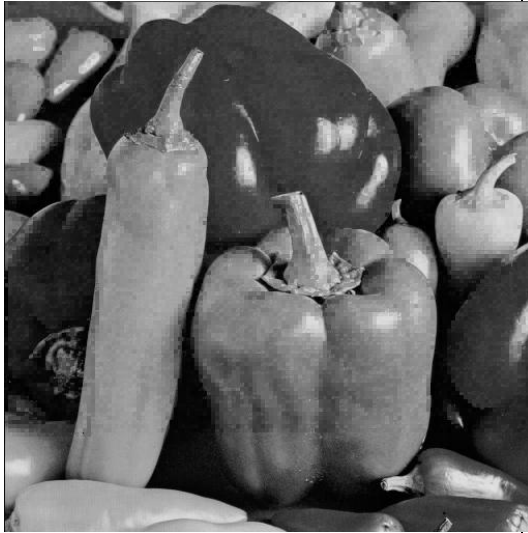
P_Bird_16.bmp



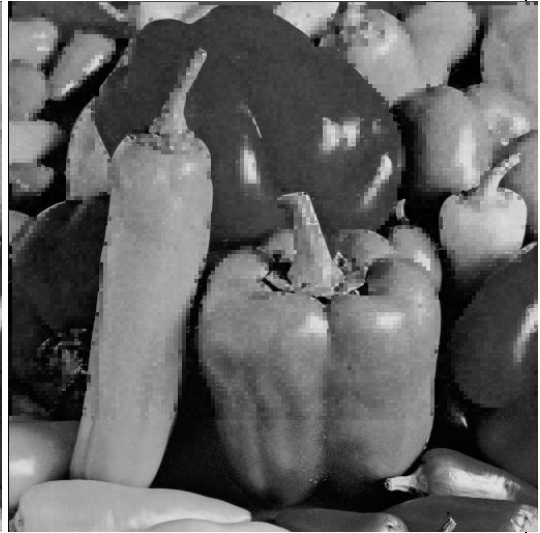
P_Bird_20.bmp



P_Bird_50.bmp



P_Bird_100.bmp



**A.10 Cover-image "Jet512.bmp" Sebelum dan Setelah Disisipkan Citra
'Lena256.bmp'**

Citra asli Jet512.bmp



J_Lena_6.bmp



J_Lena_8.bmp



J_Lena_10.bmp



J_Lena_12.bmp



J_Lena_16.bmp



J_Lena_20.bmp



J_Lena_50.bmp



J_Lena_100.bmp



A.11 Cover-image "Jet512.bmp" Sebelum dan Setelah Disisipkan Citra 'Baboon256.bmp'

Citra asli Jet512.bmp



J_Baboon_6.bmp



J_Baboon_8.bmp



J_Baboon_10.bmp



J_Baboon_12.bmp



J_Baboon_16.bmp



J_Baboon_20.bmp



J_Baboon_50.bmp



J_Baboon_100.bmp



A.12 Cover-image "Jet512.bmp" Sebelum dan Setelah Disisipkan Citra 'Bird256.bmp'

Citra asli Jet512.bmp



J_Bird_6.bmp



J_Bird_8.bmp



J_Bird_10.bmp



J_Bird_12.bmp



J_Bird_16.bmp



J_Bird_20.bmp



J_Bird_50.bmp



J_Bird_100.bmp



A.13 Cover-image "Boat512.bmp" Sebelum dan Setelah Disisipkan Citra 'Lena256.bmp'

Citra asli Boat512.bmp



B_Lena_6.bmp



B_Lena_8.bmp



B_Lena_10.bmp



B_Lena_12.bmp



B_Lena_16.bmp



B_Lena_20.bmp



B_Lena_50.bmp



B_Lena_100.bmp



A.14 Cover-image "Boat512.bmp" Sebelum dan Setelah Disisipkan Citra 'Baboon256.bmp'

Citra asli Boat512.bmp



B_Baboon_6.bmp



B_Baboon_8.bmp



B_Baboon_10.bmp



B_Baboon_12.bmp



B_Baboon_16.bmp



B_Baboon_20.bmp



B_Baboon_50.bmp



B_Baboon_100.bmp



A.15 Cover-image "Boat512.bmp" Sebelum dan Setelah Disisipkan Citra 'Bird256.bmp'

Citra asli Boat512.bmp



B_Bird_6.bmp



B_Bird_8.bmp



B_Bird_10.bmp



B_Bird_12.bmp



B_Bird_16.bmp



B_Bird_20.bmp



B_Bird_50.bmp



B_Bird_100.bmp



LAMPIRAN B

PENILAIAN SUBJEKTIF

B.1 Nilai Mean Opinion Score (MOS) citra Lena.bmp untuk nilai threshold yang berbeda-beda

	Zen	Moris	Kristy	Dina	Febryan	Bimo	Juventus	Michael	David	Linda	MOS
L_Lena_6.bmp	5	5	5	5	5	5	5	5	5	5	5
L_Lena_8.bmp	4	4	4	4	4	4	4	4	5	5	4.2
L_Lena_10.bmp	4	4	4	4	5	4	3	4	5	5	4.2
L_Lena_12.bmp	4	4	4	5	5	4	3	3	5	5	4.2
L_Lena_16.bmp	4	4	3	5	3	4	3	4	4	4	3.8
L_Lena_20.bmp	4	3	3	5	3	3	3	3	4	5	3.6
L_Lena_50.bmp	3	3	2	3	2	2	2	2	3	3	2.5
L_Lena_100.bmp	3	2	1	2	1	1	1	1	2	2	1.6
L_Baboon_6.bmp	4	5	5	5	5	5	5	5	5	5	4.9
L_Baboon_8.bmp	4	5	4	5	5	5	5	4	4	5	4.6
L_Baboon_10.bmp	4	4	4	5	5	5	4	4	4	5	4.4
L_Baboon_12.bmp	4	4	4	5	4	4	4	5	5	5	4.4
L_Baboon_16.bmp	4	4	3	4	4	4	3	4	4	5	3.9
L_Baboon_20.bmp	3	3	3	4	3	3	3	3	3	4	3.2
L_Baboon_50.bmp	3	3	2	3	2	2	2	2	2	3	2.4
L_Baboon_100.bmp	3	2	1	2	1	1	1	1	1	2	1.5
L_Bird_6.bmp	5	5	5	5	5	5	5	5	5	5	5
L_Bird_8.bmp	5	5	4	5	5	4	4	4	4	5	4.5
L_Bird_10.bmp	5	5	4	5	5	4	3	4	4	5	4.4
L_Bird_12.bmp	5	5	4	4	4	4	3	3	3	5	4
L_Bird_16.bmp	4	4	4	5	4	4	3	4	4	5	4.1
L_Bird_20.bmp	3	3	3	4	3	3	2	3	3	4	3.1
L_Bird_50.bmp	3	3	2	3	2	2	2	2	2	3	2.4
L_Bird_100.bmp	3	2	1	2	1	1	1	1	1	2	1.5

B.2 Nilai *Mean Opinion Score* (MOS) citra Peppers.bmp untuk nilai threshold yang berbeda-beda

	Zen	Moris	Kristy	Dina	Febryan	Bimo	Juventus	Michael	David	Linda	MOS
P_Lena_6.bmp	5	5	5	5	5	5	5	5	5	5	5
P_Lena_8.bmp	5	5	5	5	5	5	5	5	5	5	5
P_Lena_10.bmp	4	5	5	5	5	5	5	5	5	5	4.9
P_Lena_12.bmp	4	5	5	4	5	5	4	4	5	5	4.6
P_Lena_16.bmp	4	4	4	5	4	4	3	4	5	5	4.2
P_Lena_20.bmp	4	4	4	4	4	4	3	3	3	4	3.7
P_Lena_50.bmp	3	3	3	3	2	3	2	3	2	4	2.8
P_Lena_100.bmp	3	3	1	2	1	1	1	2	1	3	1.8
P_Baboon_6.bmp	5	5	5	5	5	5	5	5	5	5	5
P_Baboon_8.bmp	5	5	5	5	5	5	5	5	5	5	5
P_Baboon_10.bmp	4	5	5	5	5	5	4	4	5	5	4.7
P_Baboon_12.bmp	4	5	5	5	5	5	4	4	5	5	4.7
P_Baboon_16.bmp	4	4	4	5	4	4	3	3	5	4	4
P_Baboon_20.bmp	3	4	4	4	4	4	2	2	4	4	3.5
P_Baboon_50.bmp	3	3	2	3	2	3	2	2	3	3	2.6
P_Baboon_100.bmp	3	2	1	2	1	1	1	1	2	2	1.6
P_Bird_6.bmp	5	5	5	5	5	5	5	5	5	5	5
P_Bird_8.bmp	4	5	5	5	5	5	5	5	5	5	4.9
P_Bird_10.bmp	4	5	5	5	5	5	4	4	5	5	4.7
P_Bird_12.bmp	4	5	5	5	5	5	4	4	5	5	4.7
P_Bird_16.bmp	4	5	4	4	4	4	3	3	4	4	3.9
P_Bird_20.bmp	3	4	3	4	3	3	3	3	3	4	3.3
P_Bird_50.bmp	3	4	2	3	2	2	2	2	2	3	2.5
P_Bird_100.bmp	3	3	2	2	1	1	1	1	1	2	1.7

B.3 Nilai *Mean Opinion Score* (MOS) citra Jet.bmp untuk nilai threshold yang berbeda-beda

	Zen	Moris	Kristy	Dina	Febryan	Bimo	Juventus	Michael	David	Linda	MOS
J_Lena_6.bmp	5	5	5	5	5	5	5	5	5	5	5
J_Lena_8.bmp	5	5	5	5	5	5	5	5	5	5	5
J_Lena_10.bmp	5	5	5	5	5	5	4	5	4	5	4.8
J_Lena_12.bmp	5	5	5	5	5	5	4	4	5	5	4.8
J_Lena_16.bmp	4	4	5	5	4	4	3	4	5	5	4.3
J_Lena_20.bmp	3	4	5	5	4	3	3	3	3	5	3.8
J_Lena_50.bmp	3	4	4	4	2	2	2	2	3	5	3.1
J_Lena_100.bmp	3	3	3	3	1	1	1	1	1	4	2.1
J_Baboon_6.bmp	5	5	5	5	5	5	5	5	5	5	5
J_Baboon_8.bmp	4	5	5	5	5	5	5	5	5	5	4.9
J_Baboon_10.bmp	4	5	5	5	5	5	4	4	5	5	4.7
J_Baboon_12.bmp	4	5	5	5	5	5	4	4	5	5	4.7
J_Baboon_16.bmp	3	4	4	5	5	5	3	3	4	5	4.1
J_Baboon_20.bmp	3	4	4	5	4	4	3	3	4	5	3.9
J_Baboon_50.bmp	3	4	3	4	3	4	2	2	4	4	3.3
J_Baboon_100.bmp	2	3	2	3	2	2	1	1	2	3	2.1
J_Bird_6.bmp	5	5	5	5	5	5	5	5	5	5	5
J_Bird_8.bmp	5	5	5	5	5	5	5	5	5	5	5
J_Bird_10.bmp	5	5	5	5	5	5	4	4	4	5	4.7
J_Bird_12.bmp	4	5	5	5	5	5	4	4	4	5	4.6
J_Bird_16.bmp	4	4	5	5	4	4	3	3	4	5	4.1
J_Bird_20.bmp	4	4	4	5	4	4	3	3	3	5	3.9
J_Bird_50.bmp	3	4	3	4	3	3	2	2	3	4	3.1
J_Bird_100.bmp	2	3	2	3	2	2	2	1	1	3	2.1

B.4 Nilai *Mean Opinion Score* (MOS) citra Boat.bmp untuk nilai threshold yang berbeda-beda

	Zen	Moris	Kristy	Dina	Febryan	Bimo	Juventus	Michael	David	Linda	MOS
B_Lena_6.bmp	5	5	5	5	5	5	5	5	5	5	5
B_Lena_8.bmp	4	5	5	5	5	5	4	5	5	5	4.8
B_Lena_10.bmp	4	5	5	5	5	5	4	4	5	5	4.7
B_Lena_12.bmp	4	5	5	5	5	5	3	3	5	5	4.5
B_Lena_16.bmp	4	5	5	5	4	4	3	3	4	4	4.1
B_Lena_20.bmp	4	4	4	5	4	3	3	3	3	3	3.6
B_Lena_50.bmp	3	4	4	4	2	2	2	2	3	3	2.9
B_Lena_100.bmp	3	3	3	3	1	1	1	1	1	2	1.9
B_Baboon_6.bmp	4	5	5	5	5	5	5	5	5	5	4.9
B_Baboon_8.bmp	4	5	5	5	5	5	5	4	5	5	4.8
B_Baboon_10.bmp	4	5	5	5	5	5	4	4	4	4	4.5
B_Baboon_12.bmp	4	5	5	5	5	5	4	3	4	4	4.4
B_Baboon_16.bmp	4	4	4	5	4	4	3	3	4	4	3.9
B_Baboon_20.bmp	3	4	4	5	4	4	3	3	3	3	3.6
B_Baboon_50.bmp	3	3	3	3	3	3	3	2	2	2	2.7
B_Baboon_100.bmp	3	2	2	2	2	2	2	1	1	1	1.8
B_Bird_6.bmp	5	5	5	5	5	5	5	5	5	5	5
B_Bird_8.bmp	5	5	5	5	5	5	5	4	4	4	4.7
B_Bird_10.bmp	5	5	5	5	5	5	5	4	4	4	4.7
B_Bird_12.bmp	5	5	5	5	5	5	4	4	4	4	4.6
B_Bird_16.bmp	4	4	5	5	5	4	4	3	5	4	4.3
B_Bird_20.bmp	3	4	5	5	5	4	3	3	3	5	4
B_Bird_50.bmp	3	4	4	4	4	2	2	2	2	3	3
B_Bird_100.bmp	3	3	3	3	3	1	1	1	1	2	2.1

LAMPIRAN C

PERANGKAT LUNAK

```

function varargout = Coba280313(varargin)
% COBA280313 MATLAB code for Coba280313.fig
%     COBA280313, by itself, creates a new COBA280313 or raises
the existing
%     singleton*.
%
%     H = COBA280313 returns the handle to a new COBA280313 or
the handle to
%     the existing singleton*.
%
%     COBA280313('CALLBACK',hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in COBA280313.M with the given
input arguments.
%
%     COBA280313('Property','Value',...) creates a new COBA280313
or raises the
%     existing singleton*. Starting from the left, property
value pairs are
%     applied to the GUI before Coba280313_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to Coba280313_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Coba280313

% Last Modified by GUIDE v2.5 29-Mar-2013 15:06:19

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @Coba280313_OpeningFcn, ...
                  'gui_OutputFcn',  @Coba280313_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Coba280313 is made visible.
function Coba280313_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output      args, see OutputFcn.

```

```

% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to Coba280313 (see VARARGIN)

% Choose default command line output for Coba280313
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Coba280313 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Coba280313_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in RadioButton_GetImage.
function RadioButton_GetImage_Callback(hObject, eventdata,
handles)
% hObject      handle to RadioButton_GetImage (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
%Mengatur RadioButton_HideImage tidak aktif saat
RadioButton_GetImage
%dipilih
set(handles.RadioButton_HideImage, 'value', 0);
% Hint: get(hObject, 'Value') returns toggle state of
RadioButton_GetImage

function T_Callback(hObject, eventdata, handles)
% hObject      handle to T (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of T as text
%         str2double(get(hObject, 'String')) returns contents of T
as a double

% --- Executes during object creation, after setting all
properties.
function T_CreateFcn(hObject, eventdata, handles)
% hObject      handle to T (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of
MATLAB
% handles empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in PushButton_HideImage.
function PushButton_HideImage_Callback(hObject, eventdata,
handles)

% hObject handle to PushButton_HideImage (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
mode=handles.mode;
CoverImage=handles.CoverImage;
SecretImage=handles.SecretImage;
T = eval(get(handles.T, 'String'));
if mode==1
    clc;
    stat = 'Loading...';
    set(handles.stat,'string',stat);
    pause(0.001);
    x=CoverImage;
    x=double(x);
    blok=4;
    [c d]=size(x);
    jum_blok = (round(c/blok)^2);

    i=0;j=0;k=0;
    while(1)
        k=k+1;
        for q=1:blok
            for p=1:blok
                blok_value(q,p,k)= x(j+q,i+p);
            end
        end
        i=i+blok;
        if i==c;
            i=0;
            j=j+blok;
        end
        if j==d;
            break;
        end
    end

    for i=1:jum_blok
        min_blok(i)=min(min(blok_value(:, :, i)));
        max_blok(i)=max(max(blok_value(:, :, i)));
        bk(i)=(max_blok(i) - min_blok(i) + 1);
        L(i) = floor(blok*blok*(log2(bk(i))));
    end
end

```

```

secret_im = SecretImage;
secret_im = double(round(secret_im./16));
[f g]=size(secret_im);
[secret_im_bin]=kebiner(secret_im);

temp3=0;
for i=1:f
    for j=1:g
        pixel = ke4bit(secret_im_bin(i,j));
        for k=1:4
            temp3=temp3+1;
            secret_im_bit_stream(temp3)=pixel(k);
        end
    end
end
[a jum_bit_stream]=size(secret_im_bit_stream);

temp=0;
k=0;
adj_bk=0;
loop=0;
flag = 0 ;
for i=1:jum_blok
    loop=loop+1;
    bin2des=0;
    length_Li = L(i);
    value_bk=bk(i);
    index_bk=i;

    if ((value_bk >= 2) && (value_bk <= T))
        if temp> (jum_bit_stream-19)
            break;
        end
        temp=temp+length_Li;
        for j=1:length_Li
            k=k+1;
            sisipan = double(secret_im_bit_stream(k));
            bin2des = bin2des + sisipan*2^(length_Li-j);
            if (k==jum_bit_stream)
                bin2des=0;
                k=k-(temp-length_Li);
                for z=1:(temp-length_Li)
                    k=k+1;
                    sisipan = double(secret_im_bit_stream(k));
                    bin2des = bin2des + sisipan*(2^(temp-z));
                end
                break;
            end
        end
        S=basis_bk(bin2des,bk(i));
        value_basis = S;
        [u v]=size(value_basis);
        cc=0;
        for aa=1:blok
            for bb=1:blok

```

```

        cc=cc+1;
        if cc<=v
            value_blok_baru(aa,bb,i)= uint8(min_blok(i) +
value_basis(v-cc+1));
        else
            value_blok_baru(aa,bb,i)=
uint8(min_blok(i));
        end
    end
end

min_blok_adj(i)=min(min(value_blok_baru(:, :, i)));
max_blok_adj(i)=max(max(value_blok_baru(:, :, i)));
ck=max_blok_adj(i) - min_blok_adj(i) + 1;

while (ck~=bk(i))
    adj_bk= adj_bk+1;
    length_Li=length_Li;
    temp=temp-length_Li;
    k=k-length_Li ;

    if ((bk(i)<=T) && (flag==0))
        flag=0;
        bk(i)=bk(i) + 1;
    else
        flag=1;
        bk(i)=bk(i) - 1;
        if bk(i)==1
            bk(i)=bk(i) + 1;
            break;
        end
    end
end

L(i) = floor(blok*blok*(log2(bk(i))));
length_Li=L(i);
temp=temp+length_Li;
for j=1:length_Li
    k=k+1;
    sisipan = double(secret_im_bit_stream(k));
    bin2des = bin2des + sisipan*2^(length_Li-j);
    if (k==jum_bit_stream)
        bin2des=0;
        k=k-(temp-length_Li);
        for z=1:(temp-length_Li)
            k=k+1;
            sisipan =
double(secret_im_bit_stream(k));
            bin2des = bin2des + sisipan*(2^(temp-
z));

            end
            break;
        end
    end
end
S=basis_bk(bin2des,bk(i));
value_basis = S;
[u v]=size(value_basis);
cc=0;
for aa=1:blok

```

```

        for bb=1:blok
            cc=cc+1;
            if cc==v
                value_blok_baru(aa,bb,i)= uint8(min_blok(i) +
value_basis(v-cc+1));
            else
                value_blok_baru(aa,bb,i)=
uint8(min_blok(i));
            end
        end
    end
    min_blok_adj(i)=min(min(value_blok_baru(:,:,i)));
    ck=(max(max(blok_value(:,:,i)))-
min_blok_adj(i)+1);
end

else

value_blok_baru(1:blok,1:blok,i)=uint8(blok_value(:,:,i));
    if (k==jum_bit_stream)
        break;
    end
end

    if ((k==jum_bit_stream) || (temp>= (jum_bit_stream-19)))
        break;
    end
end

for i=1:index_bk
    value_blok_stego_im(:,:,i)=value_blok_baru(:,:,i);
end

for i=(index_bk+1):jum_blok
    value_blok_stego_im(:,:,i)=blok_value(:,:,i);
end

mm=0;
nn=0;
qq=0;

for i=1:jum_blok

    for ii=1:blok
        for jj=1:blok
            rekontruksi_stego_im(mm+ii,nn+jj) =
value_blok_stego_im(ii,jj,i);
        end
    end

    if (nn==(d-blok))
        mm=mm+blok;nn=0;
    else
        nn=nn+blok;
    end
end
end

```



```

MSE=sum(sum(rekontruksi_stego_im-CoverImage).^2)/(512*512);
PSNR= 10*(log10((255^2)/MSE));
MSE=num2str(MSE);
PSNR=num2str(PSNR);
set(handles.NilaiMSE,'string',MSE);
set(handles.NilaiPSNR, 'string', PSNR);
pause(0.001);
adj_bk
index_bk
secret_im_bit_stream
size(value_blok_stego_im);
size(rekontruksi_stego_im);
axes(handles.Axes_StegoImage);
imshow(rekontruksi_stego_im);
pause(0.001);
stat = 'Selesai';
set(handles.stat,'string',stat);

handles.index_bk=index_bk;
handles.value_blok_stego_im=value_blok_stego_im;
handles.blok=blok;
handles.secret_im_bit_stream=secret_im_bit_stream;
handles.bk=bk;
handles.rekontruksi_stego_im=rekontruksi_stego_im;
guidata(hObject, handles);

end

function EditText_MSE_Callback(hObject, eventdata, handles)
% hObject      handle to EditText_MSE (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of EditText_MSE as
text
%      str2double(get(hObject,'String')) returns contents of
EditText_MSE as a double

% --- Executes during object creation, after setting all
properties.
function EditText_MSE_CreateFcn(hObject, eventdata, handles)
% hObject      handle to EditText_MSE (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function PSNR_Callback(hObject, eventdata, handles)
% hObject      handle to NilaiPSNR (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NilaiPSNR as
text
%          str2double(get(hObject,'String')) returns contents of
NilaiPSNR as a double

% --- Executes during object creation, after setting all
properties.
function NilaiPSNR_CreateFcn(hObject, eventdata, handles)
% hObject      handle to NilaiPSNR (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in PushButton_SelectCoverImage.
%Memilih citra CoverImage
function PushButton_SelectCoverImage_Callback(hObject, eventdata,
handles)
[filename, pathname] = uigetfile({'*.bmp','All Image Files';...
    '*..*','All Files' }, 'Pilih Cover Image');
if isequal([filename,pathname],[0,0])
    return
else
    handles.img=imread(fullfile(pathname, filename));
    guidata(hObject, handles);
    axes(handles.Axes_CoverImage);
    imshow(handles.img);
    CoverImage = handles.img;
    handles.CoverImage=CoverImage;
    guidata(hObject, handles);
end
% hObject      handle to PushButton_SelectCoverImage (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in PushButton_SelectSecretImage.
function PushButton_SelectSecretImage_Callback(hObject, eventdata,
handles)
[filename, pathname] = uigetfile({'*.bmp','All Image Files';...
    '*..*','All Files' }, 'Pilih Secret Image');
if isequal([filename,pathname],[0,0])
    return
else
    handles.img=imread(fullfile(pathname, filename));

```

```

        guidata(hObject, handles);
        axes(handles.Axes_SecretImage);
        imshow(handles.img);
        SecretImage = handles.img;
        handles.SecretImage = SecretImage;
        guidata(hObject, handles);
    end
% hObject      handle to PushButton_SelectSecretImage (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in RadioButton_HideImage.
function RadioButton_HideImage_Callback(hObject, eventdata,
handles)
set(handles.RadioButton_GetImage, 'value', 0);
% hObject      handle to RadioButton_HideImage (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject, 'Value') returns toggle state of
RadioButton_HideImage

% --- Executes on button press in PushButton_SelectStegoImage.
function PushButton_SelectStegoImage_Callback(hObject, eventdata,
handles)
% hObject      handle to PushButton_SelectStegoImage (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
[filename, pathname] = uigetfile({'*.bmp', 'All Image Files';...
    '*..*', 'All Files' }, 'Pilih Stego Image');
if isequal([filename, pathname], [0,0])
    return
else
    handles.img=imread(fullfile(pathname, filename));
    guidata(hObject, handles);
    axes(handles.Axes_StegoImage);
    imshow(handles.img);
    StegoImage = handles.img;
    handles.StegoImage=StegoImage;
    guidata(hObject, handles);
end

% --- Executes during object creation, after setting all
properties.
function PushButton_SelectStegoImage_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to PushButton_SelectStegoImage (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% --- Executes on button press      in PushButton_GetImage.

```

```
function PushButton_GetImage_Callback(hObject, eventdata, handles)
% hObject      handle to PushButton_GetImage (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

```
% --- Executes on selection change in mode.
```

```
function mode_Callback(hObject, eventdata, handles)
% hObject      handle to mode (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns mode contents as
cell array
%          contents{get(hObject,'Value')} returns selected item from
mode
mode_noise=get(hObject,'Value');
switch mode_noise
    case 1
        mode = 1;
        handles.mode=mode;
        guidata(hObject, handles);
    case 2
        mode = 2;
        handles.mode=mode;
        guidata(hObject, handles);
end
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function mode_CreateFcn(hObject, eventdata, handles)
% hObject      handle to mode (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called
```

```
% Hint: popupmenu controls usually have a white background on
Windows.
```

```
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function Axes_CoverImage_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Axes_CoverImage (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called
```

```

% Hint: place code in OpeningFcn to populate Axes_CoverImage

% --- Executes during object creation, after setting all
properties.
function stat_CreateFcn(hObject, eventdata, handles)
% hObject    handle to stat (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% --- Executes on button press in PushButton_EkstrakImage.
function PushButton_EkstrakImage_Callback(hObject, eventdata,
handles)
% hObject    handle to PushButton_EkstrakImage (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
mode=handles.mode;
StegoImage=handles.StegoImage;
T = eval(get(handles.T, 'String'));

if mode==2
    clc;
    stat = 'Loading...';
    set(handles.stat, 'string', stat);
    pause(0.001);
    x=StegoImage;
    x=double(x);
    blok=4;
    [c d]=size(x);
    jum_blok = (round(c/blok)^2);

    i=0;j=0;k=0;
    while(1)
        k=k+1;
        for q=1:blok
            for p=1:blok
                value_blok_stego_im(q,p,k)= x(j+q,i+p);
            end
        end
        i=i+blok;
        if i==c;
            i=0;
            j=j+blok;
        end
        if j==d;
            break;
        end
    end

    for i=1:jum_blok
        clc
        min_blok_stego(i)=min(min(value_blok_stego_im(:, :, i)));
    end
end

```

```

        max_blok_stego(i)=max(max(value_blok_stego_im(:,:,i)));
        ck(i)=max_blok_stego(i) - min_blok_stego(i) + 1;
        L_stego(i) = floor(blok*blok*(log2(ck(i))));
    end

    temp6=0;
    for i=1:jum_blok
        length_Li = L_stego(i);
        value_bk=ck(i);
        index_bk_stego=i;

        temp5=0;
        if ((value_bk >= 2) && (value_bk <= T))
            for j=1:blok
                for k=1:blok
                    temp5=temp5+1;
                    bit_blok_stego(temp5) = value_blok_stego_im(j,k,i) -
min_blok_stego(i);
                end
            end
            value_blok_stego=0;
            for j=1:(blok*blok)
                value_blok_stego = value_blok_stego +
bit_blok_stego(j)*ck(i)^(j-1);
            end
            Si=des2bin(value_blok_stego);
            [ff gg]=size(Si);
            temp6=0;
            for j=1:gg
                temp6=temp6+1;
                bit_stream_stego(temp6)=Si(j);
            end
        end
    end
    bit_stream_stego=secret_im_bit_stream;

    [a b]=size(bit_stream_stego);
    b=b/4;

    temp=0;
    value_ekstraksi_im=0;
    for i=1:b
        for j=1:4
            temp=temp+1;
            value_ekstraksi_im=value_ekstraksi_im +
bit_stream_stego(temp)*(2^(4-j));
        end
        des_ekstraksi_im(i) = value_ekstraksi_im;
        value_ekstraksi_im = 0;
    end

    c= sqrt(b);

```

```

temp=0;
for i=1:c
    for j=1:c
        temp=temp+1;
        ekstraksi_im(i,j)=des_ekstraksi_im(temp);
    end
end

ekstraksi_im=ekstraksi_im.*16;
ekstraksi_im=uint8(ekstraksi_im)
size(ekstraksi_im)
axes(handles.Axes_HasilEkstraksi);
imshow(ekstraksi_im);

pause(0.001);
stat = 'Selesai';
set(handles.stat,'string',stat);

handles.ekstraksi_im = ekstraksi_im;
guidata(hObject, handles);

end

% --- Executes on button press in PushButton_SimpanStegoImage.
function PushButton_SimpanStegoImage_Callback(hObject, eventdata, handles)
% hObject    handle to PushButton_SimpanStegoImage (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
rekontruksi_stego_im=handles.rekontruksi_stego_im;
[FileName,PathName]=uiputfile('*.bmp','Save As Stego Image')
imwrite(rekontruksi_stego_im,FileName,'bmp');

% --- Executes on button press in PushButton_Keluar.
function PushButton_Keluar_Callback(hObject, eventdata, handles)
% hObject    handle to PushButton_Keluar (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;

% --- Executes during object creation, after setting all
properties.
function Axes_HasilEkstraksi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Axes_HasilEkstraksi (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: place code in OpeningFcn to populate Axes_HasilEkstraksi

% --- Executes on button press in PushButton_SimpanHasilEkstraksi.
function PushButton_SimpanHasilEkstraksi_Callback(hObject,

```

```

eventdata, handles)
% hObject    handle to PushButton_SimpanHasilEkstraksi (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)
ekstraksi_im=handles.ekstraksi_im;
[FileName,PathName]=uiputfile('*.bmp','Save As Secret Image hasil
ekstraksi')
imwrite(ekstraksi_im,FileName,'bmp');

```

Fungsi kebiner.m

```

function [matrik2]=kebiner(matrik1)
    matrik1=double(matrik1);
    [a c]=size(matrik1);
    matrik2=zeros(a,c);
    n=0;
    for i=1:a
        for j=1:c
            matrik1a=matrik1(i,j);
            matrik2a=0;
            if (matrik1a) < 2
                matrik2a=matrik1a;
            else
                while (matrik1a) >= 2
                    bil=rem(matrik1a,2);
                    b=bil*(10^n);
                    n=n+1;
                    matrik2a=matrik2a+b;
                    bagi=matrik1a/2;
                    matrik1a=floor(bagi);
                end
                b=matrik1a*(10^n);
                matrik2a=matrik2a+b;
                n=0;
            end
            matrik2(i,j)=matrik2a;
        end
    end
end

```

Fungsi ke4bit.m

```

function [bitke]=ke4bit(bilbiner)
    bitke=zeros(1,4);
    n=1;
    R=bilbiner;
    if (R) < 10
        n=1;
    else
        while (R) >= 10
            R=R/10;
            n=n+1;
        end
    end
end

```



```

    if (n) == 1
        bitke(1,4)=bilbiner;
    else
        while n >= 1
            sa=10^(n-1);
            bagi=floor(bilbiner/sa);
            f=4-n+1;
            bitke(1,f)=bagi;
            kur=bagi*sa;
            bilbiner=bilbiner-kur;
            n=n-1;
        end
    end
end

```

Fungsi basis_bk.m

```

function [S]=basis_bk(bilangan,bk)
    n=0;
    if (bilangan) < bk
        S(1) = bilangan;
    else
        while (bilangan) >= bk
            n=n+1;
            bil=mod(bilangan,bk);
            S(n)=bil;
            bagi=bilangan/bk;
            bilangan=floor(bagi);
        end
        n=n+1;
        S(n)=bilangan;
        S_temp=S;
    end

    for p=1:n
        S(p)=S_temp(n-p+1);
    end
end

```

Fungsi des2bin.m

```

function [X]=des2bin(bilangan)
    n=0;
    S=0;
    if (bilangan) < 2
        S(1)=bilangan;
        temp7=1;
    else
        while (bilangan) >= 2
            n=n+1;
            S(n)=rem(bilangan,2);
            bagi=bilangan/2;
            bilangan=floor(bagi);
        end
        n=n+1;
        temp7=n;
    end

```

```
        S(n)=bilangan;  
    end  
  
    for iii=1:temp7  
        X(iii) = S(temp7+1-iii);  
    end  
end
```