1. *Form Utama (home.fig)*

```matlab
function varargout = home(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @home_OpeningFcn, ...
'gui_OutputFcn', @home_OutputFcn, ...
'gui_LayotFcn', [], ..., ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function home_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = home_OutputFcn(hObject, eventdata, handles)
citra=imread('cover.jpg');
axes(handles.axes1);
imshow(citra);
varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
    embed;
close(gcf);

function pushbutton2_Callback(hObject, eventdata, handles)
extract;
```
function pushbutton3_Callback(hObject, eventdata, handles) HELPS;
close(gcf);

function pushbutton4_Callback(hObject, eventdata, handles) ABOUTS;
close(gcf);

function pushbutton5_Callback(hObject, eventdata, handles)  

pilih = questdlg(["EXIT CRYSTOGRAPH 
"], ["CRYSTOGRAPH","YES","NO","YES");  
if strcmp(pilih,'NO')  
return;  
else  
    close(gcf);
end
2. Form Proses Enkripsi dan Penyisipan (ENCRYPT & EMBED.fig)

```matlab
function varargout = embed(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @embed_OpeningFcn, ...
'gui_OutputFcn', @embed_OutputFcn, ...
'gui_LayOutFcn', [], ..., ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function embed_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = embed_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles, varargin)
global citra ex map
%UNTUK MEMBUKA KOTAK DIALOG OPEN CITRA
[namaCitra, lokasi] = uigetfile({'*.bmp;*.png;*.gif','Image File (*.bmp,*.png,*.gif)'}, 'Browse Image');
tmp = imread(fullfile(lokasi, namaCitra));
[b1 k1 c1] = size(tmp);

[nama, ex] = strtok(namaCitra, '.');
if strcmp(ex, 'gif')
    [citra, map] = imread(fullfile(lokasi, namaCitra));
elseif c1<3
    [citra, map] = imread(fullfile(lokasi, namaCitra));
else
    citra = imread(fullfile(lokasi, namaCitra));
end
set(handles.fn, 'String', namaCitra);
set(handles.loc, 'String', lokasi);
```
%UNTUK MENAMPILKAN UKURAN DIMENSI DAN BIT DEPTH
[b k c]=size(citra);
set(handles.d1,'String',b);
set(handles.d2,'String',k);
bit=c*8;
set(handles.bd,'String',bit);

m = floor((b*k*c)/7);
if strcmp(ex,'.gif')
    m=m-2;
else
    m = m-1;
end
set(handles.max,'String',m);
set(handles.pushbutton2,'Enable','on');
set(handles.pushbutton3,'Enable','on');

function fn_Callback(hObject, eventdata, handles)

function fn_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function d1_Callback(hObject, eventdata, handles)

function d1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function d2_Callback(hObject, eventdata, handles)

function d2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function size_Callback(hObject, eventdata, handles)

function size_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function bd_Callback(hObject, eventdata, handles)

function bd_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function loc_Callback(hObject, eventdata, handles)

function loc_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function pushbutton2_Callback(hObject, eventdata, handles)
    set(hObject,'String','
    set(handles.d1,'String','
    set(handles.d2,'String','
    set(handles.bd,'String','
    set(handles.loc,'String','
    set(handles.max,'String','

function pushbutton3_Callback(hObject, eventdata, handles)
    set(handles.p,'Enable','on');
    set(handles.b,'Enable','on');
    set(handles.m,'Enable','on');
    set(handles.pushbutton4,'Enable','on');
    set(handles.pushbutton17,'Enable','on');

function p_Callback(hObject, eventdata, handles)

function p_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end
function max_Callback(hObject, eventdata, handles)

function max_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function c_Callback(hObject, eventdata, handles)

function c_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton4_Callback(hObject, eventdata, handles, varargin)
global indeksCipher ex
indeksCipher=0;
plainteks = get(handles.p,'String');

if strcmp(ex,'.gif')
    plainteks = strcat(plainteks,'u');
end

kunci = ['A' 0;
    'B' 1;
    'C' 2;
    'D' 3;
    'E' 4;
    'F' 5;
    'G' 6;
    'H' 7;
    'I' 8;
    'J' 9;
    'K' 10;
    'L' 11;
    'M' 12;
    'N' 13;
    'O' 14;
    'P' 15;
    'Q' 16;
    'R' 17;
    'S' 18;
    'T' 19;
    'U' 20;
    'V' 21;
    'W' 22;
    'X' 23;
    'Y' 24;
    'Z' 25;
    'a' 26;
    'b' 27;
%UNTUK MENDAPATKAN INDEKS PLAINTEKS
[baris bykChar] = size(plainteks);
for i = 1:1:bykChar
for j = 1:1:100
if plainteks(i)==kunci(j)
    indeksPlain(i)=double(kunci(j,2));
    break;
else
    j=j+1;
end
end
end

kunciB=str2num(get(handles.b,'String'));
kunciM=str2num(get(handles.m,'String'));

M1=kunciM;
n=100;
while M1~=0
    sisa=mod(n,M1);
    n=M1;
    M1=sisa;
end
hasil=n;

if (kunciB>99) || (kunciB<0)
    msgbox('B Key Must Be: 0<=b<100','CRYSTOGRAPH','error');
elseif hasil~=1 || kunciM>100
    msgbox('M Key Must Be Coprime With 100 and Less Than 100','CRYSTOGRAPH','error');
else
    for i=1:1:bykChar
        indeksCipher(i)=mod(((kunciM*indeksPlain(i))+kunciB),100);
    end
end

for i=1:1:bykChar
for j=1:1:100
if indeksCipher(i)==double(kunci(j,2));
    cipherteks(i)=kunci(j,1);
    break;
else
    j=j+1;
end
end
end

set(handles.c,'String',cipherteks);
function pushbutton10_Callback(hObject, eventdata, handles, varargin)
global indeksCipher citra stego
warna=double(citra);
%MENCARI BINER DARI CITRA
binCitra=dec2bin(warna);
[br bt] = size(binCitra);

%MENCARI BINER DARI INDEKS CIPHERTEKS
binIndeksCipher=dec2bin(indeksCipher,7);
binIndeksCipher=reshape(binIndeksCipher.',1,[]); %membuat matriks
dengan ukuran 1xjumlah binIndeksCipher
[baris kolom]=size(binIndeksCipher);

%PROSES PENYISIPAN PESAN DI BIT TERAKHIR
for i=1:1:kolom
    binCitra(i,8)=binIndeksCipher(i);
end

%KARAKTER PENANDA AKHIR PESAN (PADDING)
pad=127;
pad=dec2bin(pad,7);
pad=reshape(pad.',1,[]);
indx=1;

%PROSES PENYISIPAN PENANDA AKHIR PESAN(PADDING)
for i=kolom+1:1:kolom+6
    binCitra(i,8)=pad(indx);
    indx = indx+1;
end

%PROSES PENYISIPAN PESAN DI BIT TERAKHIR------
%PERUBAHAN BINER KE CITRA
stegano=bin2dec(binCitra);
[b kl c]=size(citra);
indeks=1;
for i=1:1:c
    for j=1:1:kl
        for k=1:1:b
            steganoImg(k,j,i)=stegano(indeks);
            indeks=indeks+1;
        end
    end
end
end

stego=double(steganoImg);
stego=uint8(stego);
warndlg('EMBEDDING FINISH...','CRYSTOGRAPH');
function pushbutton11_Callback(hObject, eventdata, handles, varargin)
global stego citra2 ex map

bit = str2num(get(handles.bd,'String'));
bit = bit/8;
hasil=stego;

if strcmp(ex,'.gif')
    [name,loca]=uiputfile('SteganoResult.gif','Save');
citra2=fullfile(loca,name);
imwrite(hasil,map,citra2);
warndlg('SAVE FINISH...','CRYSTOGRAPH');
elseif strcmp(ex,'.png')
    if bit<3
        [name,loca]=uiputfile('SteganoResult.png','Save');
citra2=fullfile(loca,name);
imwrite(hasil,map,citra2);
warndlg('SAVE FINISH...','CRYSTOGRAPH');
    else
        [name,loca]=uiputfile('SteganoResult.png','Save');
citra2=fullfile(loca,name);
imwrite(hasil,citra2);
warndlg('SAVE FINISH...','CRYSTOGRAPH');
    end
else
    if bit<3
        [name,loca]=uiputfile('SteganoResult.bmp','Save');
citra2=fullfile(loca,name);
imwrite(hasil,map,citra2);
warndlg('SAVE FINISH...','CRYSTOGRAPH');
    else
        [name,loca]=uiputfile('SteganoResult.bmp','Save');
citra2=fullfile(loca,name);
imwrite(hasil,citra2);
warndlg('SAVE FINISH...','CRYSTOGRAPH');
    end
end
end

function pushbutton12_Callback(hObject, eventdata, handles)

function pushbutton13_Callback(hObject, eventdata, handles)

function pushbutton14_Callback(hObject, eventdata, handles)
pilih = questdlg(['BACK TO MAIN MENU?
''','[\'CRYSTOGRAPH\'],\'YES','\'NO','\'YES\']);
if strcmp(pilih,'NO')
    return;
else
    close(gcbf);
    home;
end

function pushbutton15_Callback(hObject, eventdata, handles)
set(handles.p,'String','');
set(handles.b,'String','');
set(handles.m,'String','');
set(handles.c,'String','');

function pushbutton16_Callback(hObject, eventdata, handles)
views;

function m_Callback(hObject, eventdata, handles)

function m_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function b_Callback(hObject, eventdata, handles)

function b_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton17_Callback(hObject, eventdata, handles,varargin)
maks=str2num(get(handles.max,'String'));
plaint=get(handles.p,'String');
[b kar]=size(plaint);
if kar>maks
    messagebox(['Cannot Continue, Your Message is : ',num2str(kar),'
Characters'],'CRYSTOGRAPH','error');
else
    messagebox(['Can Continue, Your Message is : ',num2str(kar),'
Characters'],'CRYSTOGRAPH','none');
end
3. Form Proses Ekstraksi dan Dekripsi (EXTRACT & DECRYPT.fig)

```matlab
function varargout = extract(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @extract_OpeningFcn, ...
'gui_OutputFcn', @extract_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function extract_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = extract_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function pushbutton4_Callback(hObject, eventdata, handles,varargin)
set(handles.cip,'String','');
set(handles.bk,'String','');
set(handles.mk,'String','');
set(handles.minver,'String','');
set(handles.plain,'String','');

function pushbutton5_Callback(hObject, eventdata, handles)
pilih = questdlg(['BACK TO MAIN MENU ?'],['CRYSTOGRAPH'],['YES','NO','YES']);
if strcmp(pilih,'NO')
    return;
else
    close(gcf);
    home;
end

function mk_Callback(hObject, eventdata, handles)

function mk_CreateFcn(hObject, eventdata, handles)
```

Universitas Sumatera Utara
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function bk_Callback(hObject, eventdata, handles)

function bk_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function minver_Callback(hObject, eventdata, handles)

function minver_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton6_Callback(hObject, eventdata, handles,varargin)
kunciM=str2num(get(handles.mk,'String'));
k=0;
while k>=0
    Minvers=(1+(double(k)*100))/double(kunciM);
    if(round(Minvers)-Minvers==0 && Minvers>0)
        break;
    end
    k=k+1;
end
set(handles.minver,'String',Minvers);
set(handles.pushbutton7,'Enable','on');

function pushbutton7_Callback(hObject, eventdata, handles,varargin)
global ex

kunci = ['A' 0;
'B' 1;
'C' 2;
'D' 3;
'E' 4;
'F' 5;
'G' 6;
'H' 7;
'I' 8;
'J' 9;
'K' 10;
ciph = get(handles.cip,'String');
Minv=str2num(get(handles.minver,'String'));
kunciB=str2num(get(handles.bk,'String'));

[baris bykChar] = size(ciph);
if strcmp(ex,'.gif')
    bykChar=bykChar-1;
end

for i = 1:1:bykChar
    for j = 1:1:100
        if ciph(i)==kunci(j)
            indekPlain(i)=double(kunci(j,2));
            break;
        else
            j=j+1;
        end
    end
end

for i=1:1:bykChar
    plainteks(i)=mod((Minv*(indekPlain(i)-kunciB)),100);
end

for i=1:1:bykChar
    for j=1:1:100
        if plainteks(i)==double(kunci(j,2));
            break;  % This line seems to have an extra semicolon.
HasilPlain(i)=kunci(j,1);
break;
else
    j=j+1;
end
end
d
set(handles.plain,'String',HasilPlain);

set(handles.pushbutton4,'Enable','on');
function plain_Callback(hObject, eventdata, handles)
function plain_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function cip_Callback(hObject, eventdata, handles)
function cip_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton1_Callback(hObject, eventdata, handles, varargin)
global citra ex
[namaCitra,lokasi]=uigetfile({'*.bmp;*.png;*.gif','Image File
(*.bmp,.png,.gif)'},'Browse Image');
tmp = imread(fullfile(lokasi,namaCitra));
[b1 k1 c1]=size(tmp);
[nama,ex]=strtok(namaCitra,'.');
if strcmp(ex,'.gif')
    [citra,map] = imread(fullfile(lokasi,namaCitra));
elseif c1<3
    [citra,map] = imread(fullfile(lokasi,namaCitra));
else
    citra = imread(fullfile(lokasi,namaCitra));
end
end
d
set(handles.f,'String',namaCitra);
set(handles.l,'String',lokasi);

%UNTUK MENAMPILKAN UKURAN DIMENSI DAN BIT DEPTH
[b k c]=size(citra);
set(handles.d3,'String',b);
set(handles.d4,'String',k);
bit=c*8;
set(handles.b,'String',bit);

warna=double(citra);
%MENCARI BINER DARI CITRA
binCitra=dec2bin(warna);
[br bt] = size(binCitra);

function f_Callback(hObject, eventdata, handles)
function f_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function d3_Callback(hObject, eventdata, handles)
function d3_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function d4_Callback(hObject, eventdata, handles)
function d4_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function s_Callback(hObject, eventdata, handles)
function s_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function b_Callback(hObject, eventdata, handles)
function b_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
                get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function l_Callback(hObject, eventdata, handles)

function l_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
                get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton2_Callback(hObject, eventdata, handles)
set(handles.f,'String','');
set(handles.d3,'String','');
set(handles.d4,'String','');
set(handles.b,'String','');
set(handles.l,'String','');

function pushbutton3_Callback(hObject, eventdata, handles,varargin)

global citra ex
padding=127;
padding=dec2bin(padding);
warna=double(citra);
binCitra=dec2bin(warna);
[br bt] = size(binCitra);

%PROSES EKSTRAKSI CIPHER
j=1;
for i=1:1:br
    binCipher(i)=binCitra(i,8);
    cek(j)=binCipher(i);
if j==7
    if strcmp(cek,padding)
        break;
    else
        j=1;
    end
else
    j=j+1;
end
[b k] = size(binCipher);
k=k-7;
for i=1:1:k
    binCipherNew(i)=binCipher(i);
end
global hasil

%UBAH BINER KE DESIMAL
binCipherNew=reshape(binCipherNew,7,[]).';
[brs klm]=size(binCipherNew);
kali=[64 32 16 8 4 2 1];
temp=0;
for i=1:1:brs
    for j=1:1:klm
        temp=temp+(kali(j))*str2num(binCipherNew(i,j));
    end
    hasil(i)=temp;
    temp=0;
end

%UBAH KE KARAKTER
kunci = ['A' 0;
    'B' 1;
    'C' 2;
    'D' 3;
    'E' 4;
    'F' 5;
    'G' 6;
    'H' 7;
    'I' 8;
    'J' 9;
    'K' 10;
    'L' 11;
    'M' 12;
    'N' 13;
    'O' 14;
    'P' 15;
    'Q' 16;
    'R' 17;
    'S' 18;
    'T' 19;
    'U' 20;
    'V' 21;
    'W' 22;
    'X' 23;
    'Y' 24;
    'Z' 25;
    'a' 26;
    'b' 27;
    'c' 28;
    'd' 29;
    'e' 30;
    'f' 31;
    'g' 32;
    'h' 33;
    'i' 34;
    'j' 35;
    'k' 36;
    'l' 37;
    'm' 38;
    'n' 39;
    'o' 40;
    'p' 41;]
[brs bykChar]=size(hasil);

for i=1:1:bykChar
    for j=1:1:100
        if hasil(i)==double(kunci(j,2));
            cipherteks(i)=kunci(j,1);
            break;
        else
            j=j+1;
        end
    end
end

set(handles.cip,'String',cipherteks);

set(handles.pushbutton6,'Enable','on');
set(handles.mk,'Enable','on');
set(handles.bk,'Enable','on');
4. FormView (VIEWS.fig)

```matlab
function varargout = VIEWS(varargin)

    gui_Singleton = 1;
    gui_State = struct('gui_Name',       mfilename, ...
                      'gui_Singleton',  gui_Singleton, ...
                      'gui_OpeningFcn', @VIEWS_OpeningFcn, ...
                      'gui_OutputFcn',  @VIEWS_OutputFcn, ...
                      'gui_LayoutFcn',  [],  ...
                      'gui_Callback',   []);
    if nargin && ischar(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
    end

    if nargout
        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end

function VIEWS_OpeningFcn(hObject, eventdata, handles, varargin)

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

function varargout = VIEWS_OutputFcn(hObject, eventdata, handles, varargin)

    global citra ex map citra2
    axes(handles.axes2);
    if strcmp(ex,'.gif')
        imshow(citra);
    else
        imshow(citra);
    end
    axes(handles.axes3);
    if strcmp(ex,'.gif')
        imshow(citra2);
    else
        imshow(citra2);
    end
    varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
    close(gcf);

function axes2_CreateFcn(hObject, eventdata, handles)
```

Universitas Sumatera Utara
5. FormHelp (HELPS.fig)

```matlab
function varargout = HELPS(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @HELPS_OpeningFcn, ...
'gui_OutputFcn', @HELPS_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function HELPS_OpeningFcn(hObject, eventdata, handles, varargin)

temp = hObject;
handles.output = temp;

guidata(hObject, handles);

function varargout = HELPS_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function edit2_Callback(hObject, eventdata, handles)

function edit2_CreateFcn(hObject, eventdata, handles)

    if ispc && isequal(get(hObject,'BackgroundColor'),
                      get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function pushbutton2_Callback(hObject, eventdata, handles)

pilih = questdlg(['BACK TO MAIN MENU ?'],'CRYSTOGRAPH','YES','NO','YES');

if strcmp(pilih,'NO')
    return;
else
    close(gca);
    home;
end
```
6. FormAbout (ABOUTS.fig)

```matlab
function varargout = ABOUTS(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @ABOUTS_OpeningFcn, ...
    'gui_OutputFcn', @ABOUTS_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function ABOUTS_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = ABOUTS_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

function pushbutton1_Callback(hObject, eventdata, handles)
pilih = questdlg(['BACK TO MAIN MENU ?'], ['CRYSTOGRAPH'], 'YES', 'NO', 'YES');
if strcmp(pilih,'NO')
    return;
else
    close(gcbf);
    home;
end
```