

LAMPIRAN A
LISTING PROGRAM

Algoritma menu utama fcompher.m

```
clear all;
clc;
form1=figure(...
    'units','points',...
    'position',[100 150 303 306],...
    'color',[.8 .8 .9],...
    'menubar','none',...
    'resize','off',...
    'numbertitle','off',...
    'name','Software For Fingerprint Compression');
i = imread('front.bmp'); imshow(i,'truecolor');
menu1=uimenu('parent',form1,'Label','File');
menu11=uimenu('parent',menu1,'Label','Open','Callback','menuopen');
menu12=uimenu('parent',menu1,'Label','Close','Callback','close');
menu13=uimenu('parent',menu1,'Label','Exit','Callback','close');
menu2=uimenu('parent',form1,'Label','View');
menu3=uimenu('parent',form1,'Label','Tools');
menu31=uimenu('parent',menu3,'Label','Compress','Callback','kompres');
menu32=uimenu('parent',menu3,'Label','Decompress','Callback','dekompres');
menu4=uimenu('parent',form1,'Label','Help');
menu41=uimenu('parent',menu4,'Label','About');
```

Algoritma menu kompres.m

```
formkomp=figure('units','points','position',[120 250 450 150],'color',[.8 .8 .9],
    'menubar','none','resize','off','numbertitle','off','name','Compress');

label1=uicontrol('parent',formkomp,'units','points','position',[5 120 100 15],
    'backgroundcolor',[.8 .8 .9],'style','Text','string','Input File Name ',
    'fontname','arial','fontsize',10);

edit1=uicontrol('parent',formkomp,'units','points','position',[100 120 250 15],
    'backgroundcolor',[1 1 1],'style','Edit','string','', 'horizontalalignment','left',
    'fontname','arial','fontsize',10);

label2=uicontrol('parent',formkomp,'units','points','position',[5 90 100 15],
    'backgroundcolor',[.8 .8 .9],'style','Text','string','Output File Name',
    'fontname','arial','fontsize',10);

edit2=uicontrol('parent',formkomp,'units','points','position',[100 90 250 15],
    'backgroundcolor',[1 1 1],'style','Edit','string','', 'horizontalalignment','left',
    'fontname','arial','fontsize',10);

tombol_ok=uicontrol('parent',formkomp,'units','points','position',[70 40 60 15],
    'style','pushbutton','string','Ok','fontname','arial','fontsize',10,
    'callback','fungcomp');

tombol_cancel=uicontrol('parent',formkomp,'units','points',
    'position',[150 40 60 15],'style','pushbutton','string','Cancel','fontname','arial',
    'fontsize',10,'callback','close');
```

```

tombol_browse1=uicontrol('parent',formkomp,'units','points',
    'position',[360 120 50 15],'style','pushbutton','string','Browse','fontname','arial',
    'fontsize',10,'callback','pathload');

tombol_browse2=uicontrol('parent',formkomp,'units','points','position',[360 90 50
    15],'style','pushbutton','string','Browse','fontname','arial','fontsize',10,'callback','pathsa
    ve');

label3=uicontrol('parent',formkomp,'units','points','position',[5 60 650 15],
    'backgroundcolor',[.8 .8 .9],'style','Text','string','Level',
    'fontname','arial','fontsize',10);

popup1=uicontrol('parent',formkomp,'units','points','position',[355 60 40 15],
    'backgroundcolor',[1 1 1],'style','popupmenu','string',' 1| 2| 3',
    'fontname','arial','fontsize',10);

proses=uicontrol('parent',formkomp,'units','points','position',[0 0 650 15],
    'backgroundcolor',[.8 .8 .9],'style','Text','horizontalalignment','Left',
    'string','', 'fontname','arial','fontsize',10);

```

Algoritma menu Dekompres.m

```

formdekomp=figure('units','points', 'position',[120 250 450 150], 'color',[.8 .8 .9],
    'menubar','none','resize','off','numbertitle','off','name','Decompress');

label1=uicontrol('parent',formdekomp,'units','points','position',[5 120 100 15],...
    'backgroundcolor',[.8 .8 .9],'style','Text','string','Input File Name ',
    'fontname','arial','fontsize',10);

edit1=uicontrol('parent',formdekomp,'units','points', 'position',[100 120 250 15],
    'backgroundcolor',[1 1 1],'style','Edit', 'string','', 'horizontalalignment','left',
    'fontname','arial','fontsize',10);

label2=uicontrol('parent',formdekomp,'units','points','position',[5 90 100 15],
    'backgroundcolor',[.8 .8 .9],'style','Text','string','Output File Name',
    'fontname','arial','fontsize',10);

edit2=uicontrol('parent',formdekomp,'units','points','position',[100 90 250 15],
    'backgroundcolor',[1 1 1],'style','Edit','string','', 'horizontalalignment','left',
    'fontname','arial','fontsize',10);

tombol_ok=uicontrol('parent',formdekomp,'units','points','position',[70 40 60 15],
    'style','pushbutton','string','Ok','fontname','arial','fontsize',10,'callback','fungdecomp');

tombol_cancel=uicontrol('parent',formdekomp,'units','points','position',[150 40 60 15],
    'style','pushbutton','string','Cancel','fontname','arial','fontsize',10,'callback','close');

tombol_browse1=uicontrol('parent',formdekomp,'units','points',
    'position',[360 120 50 15],'style','pushbutton','string','Browse',
    'fontname','arial','fontsize',10,'callback','pathloaddekom');

```

```
tombol_browse2=uicontrol('parent',formdekom,'units','points',
    'position',[360 90 50 15], 'style','pushbutton','string','Browse',
    'fontname','arial','fontsize',10,'callback','pathsavedekom');

proses=uicontrol('parent',formdekom,'units','points','position',[0 0 650 15],
    'backgroundcolor',[.8 .8 .9], 'style','Text', 'horizontalalignment','Left',
    'string','', 'fontname','arial','fontsize',10);
```

Algoritma pathload.m

```
filenamex = "";
pathnamex = "";
[filenamex, pathnamex] = uigetfile( ...
    {'*.bmp','BMP Files (*.bmp)'}, 'Choose a Image File');
text1 = [pathnamex filenamex];
set(edit1,'string',text1);
```

Algoritma pathsave.m

```
filenamey = "";
pathnamey = "";
[filenamey, pathnamey, filterindex] = uiputfile( ...
    {'*.mat','Compressed Fingerprint Files (*.mat)'}, 'Save as');
text2=[pathnamey filenamey];
set(edit2,'string',text2);
```

Algoritma pathloaddekom.m

```
filenameeq = "";
pathnameeq = "";
[filenameeq, pathnameeq] = uigetfile( ...
    {'*.mat','Compressed Fingerprint Files (*.mat)'}, 'Choose a Fingerprint
    Compressed File');
text1 = [pathnameeq filenameeq];
set(edit1,'string',text1);
```

Algoritma pathsavedekom.m

```
filenamez = "";
pathnamez = "";
[filenamez, pathnamez, filterindex] = uiputfile( ...
    {'*.bmp','BMP Files (*.bmp)'}, 'Save as');
text2=[pathnamez filenamez];
set(edit2,'string',text2);
```

Algoritma fungkom.m

```
location1=get(edit1,'String');location2=get(edit2,'String');pilihan=get(popup1,'Value');
a=""; wname='bior5.5';
if (isequal(location1,a) || isequal(location2,a))
    disp('The Input and Output FileName must be Fill')
else
    set(proses,'backgroundcolor',[.5 .7 .9],...
    'string',' Processing.....');
    [i,map]=imread(location1);
    y=rgb2gray(i);    sx = size(y);
    switch pilihan
        case 1
            [cA1,cH1,cV1,cD1] = dwt2(y,wname);
            cA1_quant = unquant(cA1,10,1,255);
            cH1_quant = unquant(cH1,10,1,255);
            cV1_quant = unquant(cV1,10,1,255);
            cD1_quant = unquant(cD1,10,1,255);
            [ukuran1,hasil1] = huffcoding(cA1_quant);
            [ukuran2,hasil2] = huffcoding(cH1_quant);
            [ukuran3,hasil3] = huffcoding(cV1_quant);
            [ukuran4,hasil4] = huffcoding(cD1_quant);
            [hh,ww] = size(i); ukuran = int2str(hh*ww*8);
            figure('name','Original image (ukuran file(bits) = ' ukuran ')'),
            imshow(y,'trueSize'); namafile=[filenamey];
            file = [pathnamey namafile];
            save(file,'hasil1','hasil2','hasil3','hasil4','pilihan','map','sx');
        case 2
            [C,S] = wavedec2(y,2,wname);
            C_quant = unquant(C,10,1,255);
            [ukuran,hasil] = huffcoding(C_quant);
            [hh,ww] = size(i);
            ukuran2 = int2str(hh*ww*8);
            figure('name','Original image (ukuran file(bits) = ' ukuran2 ')'),
            imshow(y,'trueSize');
            namafile=[ filenamey];
            file = [pathnamey namafile];
            save(file,'hasil','pilihan','map','sx','S');
        case 3
            [C,S] = wavedec2(y,3,wname);
            C_quant = unquant(C,10,1,255);
            [ukuran,hasil] = huffcoding(C_quant);
            [hh,ww] = size(i);
            ukuran2 = int2str(hh*ww*8);
            figure('name','Original image (ukuran file(bits) = ' ukuran2 ')'),
            imshow(y,'trueSize');
            namafile=[ filenamey];
            file = [pathnamey namafile];
            save(file,'hasil','pilihan','map','sx','S');
    end;
end;
```

Algoritma fungdekomp.m

```
location1=get(edit1,'String');location2=get(edit2,'String');
a="";b=1;wname='bior5.5';
if (isequal(location1,a) || isequal(location2,a))
    disp('The Input and Output FileName must be Fill')
    b=2;
else
    set(proses,'backgroundcolor',[.5 .7 .9],'string',' Processing.....');
    load(location1);

    switch pilihan
    case 1
        decodehuf_1 = huffdecoding(hasil1);
        decodehuf_2 = huffdecoding(hasil2);
        decodehuf_3 = huffdecoding(hasil3);
        decodehuf_4 = huffdecoding(hasil4);
        dequant_1 = unquant(decodehuf_1,10,1);
        dequant_2 = unquant(decodehuf_2,10,1);
        dequant_3 = unquant(decodehuf_3,10,1);
        dequant_4 = unquant(decodehuf_4,10,1);
        X0 = idwt2(dequant_1,dequant_2,dequant_3,dequant_4,wname,sx);
        [hh,ww] = size(X0);
        ukuran2 = int2str(hh*ww*24);
        figure('name','Fingerprint after decoding (ukuran file(bits) = ' ukuran2)'),
        imshow(X0,map,'truecolor');
    case 2
        decodehuf = huffdecoding(hasil);
        C = unquant(decodehuf,10,1);
        X0 = waverec2(C,S,wname);
        [hh,ww] = size(X0);
        ukuran2 = int2str(hh*ww*24);
        figure('name','Fingerprint after decoding (ukuran file(bits) = ' ukuran2)'),
        imshow(X0,map,'truecolor');
    case 3
        decodehuf = huffdecoding(hasil);
        C = unquant(decodehuf,10,1);
        X0 = waverec2(C,S,wname);
        [hh,ww] = size(X0);
        ukuran2 = int2str(hh*ww*24);
        close;
        figure('name','Fingerprint after decoding (ukuran file(bits) = ' ukuran2)'),
        imshow(X0,map,'truecolor');
    end;
end;
if isequal(b,1)
    namafile=[filenamez '.bmp'];
    file = [pathnamez namafile];
    frame = getframe(gcf);
    imwrite(frame.cdata,file);
end;
```

Algoritma dwt2.m

```
function [a,h,v,d] = dwt2(x,varargin)

if ischar(varargin{1})
    [Lo_D,Hi_D] = wfilters(varargin{1},'d');
else
    Lo_D = varargin{1}; Hi_D = varargin{2};
end
sx = size(x); shift = 2;
sizeKEPT = 2*ceil(sx/2);
z = wconv('row',x,Lo_D);
a = convdown(z,Lo_D,sizeKEPT,shift);
h = convdown(z,Hi_D,sizeKEPT,shift);
z = wconv('row',x,Hi_D);
v = convdown(z,Lo_D,sizeKEPT,shift);
d = convdown(z,Hi_D,sizeKEPT,shift);

% Internal Function(s)
function y = convdown(x,f,sizeKEPT,shift)

y = wconv('col',x,f);
y = wkeep(y,sizeKEPT);
y = dyaddown(y,'col',shift);
y = dyaddown(y,'row',shift);
```

Algoritma idwt2.m

```
function x = idwt2(a,h,v,d,varargin)

if ischar(varargin{1})
    [Lo_R,Hi_R] = wfilters(varargin{1},'r');
else
    Lo_R = varargin{1}; Hi_R = varargin{2};
end
k=2;
while k<=length(varargin)
    if ischar(varargin{k})
        k = k+2;
    else
        sx = varargin{k}; k = k+1;
    end
end

x = upsacnv('2D',a,{Lo_R,Lo_R},sx)+ ... % Approximation.
    upsacnv('2D',h,{Hi_R,Lo_R},sx)+ ... % Horizontal Detail.
    upsacnv('2D',v,{Lo_R,Hi_R},sx)+ ... % Vertical Detail.
    upsacnv('2D',d,{Hi_R,Hi_R},sx);    % Diagonal Detail.
```

Algoritma wavedec2

```
function [c,s] = wavedec2(x,n,varargin)

if nargin==3
    [Lo_D,Hi_D] = wfilters(varargin{1},'d');
else
    Lo_D = varargin{1}; Hi_D = varargin{2};
end
s = [size(x)];
c = [];

for i=1:n
    [x,h,v,d] = dwt2(x,Lo_D,Hi_D); % decomposition
    c = [h(:)' v(:)' d(:)' c]; % store details
    s = [size(x);s]; % store size
end
% Last approximation.
c = [x(:)' c];
s = [size(x);s];
```

Algoritma waverec2.m

```
function x = waverec2(c,s,varargin)

x = appcoef2(c,s,varargin{:},0);
```

Algoritma appcoef2.m

```
function a = appcoef2(c,s,varargin)
rmax = size(s,1);
nmax = rmax-2;
if ischar(varargin{1})
    [Lo_R,Hi_R] = wfilters(varargin{1},'r');
else
    Lo_R = varargin{1}; Hi_R = varargin{2}
end
next = 2;
if nargin>=(2+next) , n = varargin{next}; else, n = nmax; end

nl = s(1,1);
nc = s(1,2);
a = zeros(nl,nc);
a(:) = c(1:nl*nc);

rm = rmax+1;
for p=nmax:-1:n+1
    [h,v,d] = detcoef2(c,s,p);
    a = idwt2(a,h,v,d,Lo_R,Hi_R,s(rm-p,:));
end
```

Algoritma detcoef2.m

```
function varargout = detcoef2(c,s,n)

k = size(s,1)-n;
first = s(1,1)*s(1,2)+3*sum(s(2:k-1,1).*s(2:k-1,2))+1;
add = s(k,1)*s(k,2);
last = first+add-1;
varargout{1} = reshape(c(first:last),s(k,:));
first = first+add; last = first+add-1;
varargout{2} = reshape(c(first:last),s(k,:));
first = first+add; last = first+add-1;
varargout{3} = reshape(c(first:last),s(k,:));
```

Algoritma uniquant.m

```
function Y = UniQuant(X, del, thr, ymax)
S=sign(X); X=abs(X);
if (nargin == 4) % kuantisasi dari X menjadi Y
    Y=floor((X-thr)/del)+1;
    ymax=floor(ymax); I=find(Y>ymax); Y(I)=ymax;
elseif (nargin == 3) % Dekuantisasi dari X menjadi Y
    Y=zeros(size(X)); I=find(X);
    Y(I)=(X(I)*del)+(thr-del/2);
else
    error('jumlah input tidak sesuai');
end
Y=Y.*S;
return
```

Algoritma huffcoding.m

```
function [ukuranbit,hasil] = huffcoding(x)
speed = 0;
level = 8;
doubX = double(x);
doubX = ceil(doubX);
[CompK,CompL]=size(doubX);
vecCompImg = mat2vec(doubX,level,CompK,CompL);
[y,Res]=huff06(vecCompImg,level,speed);
ukuranbit = int2str(Res(CompK+1,3));
hasil = y;
```

Algoritma huffdecoding.m

```
function decodehuf = huffcoding(x)
y=huff06(x);
uk = cell2mat(y(1));
matdecode = cell2mat(y);
[K,L] = size(uk); ecodehuf = vec2mat(matdecode,K);
```

Algoritma hufflen.m

```
function HL = HuffLen(S)

if (length(S)==0)
    HL=0;
    return;
end
I=find(S<0);
S(I)=0;
if (sum(S)==max(S))
    HL=zeros(size(S));
    return;
end
HL=zeros(size(S)); S=S(:);
Ip=find(S>0); Sp=S(Ip);
N=length(Sp); HLP=zeros(size(Sp));
C=[Sp(:);zeros(N-1,1)]; Top=1:N;
[So,Si]=sort(-Sp); last=N; next=N+1;
while (last > 1)
    C(next)=C(Si(last))+C(Si(last-1));
    I=find(Top==Si(last));
    HLP(I)=HLP(I)+1; Top(I)=next;
    I=find(Top==Si(last-1));
    HLP(I)=HLP(I)+1; Top(I)=next;
    last=last-1; Si(last)=next;
    next=next+1; count=last-1;
    while ((count> 0) & (C(Si(count+1)) >= C(Si(count))))
        temp=Si(count);
        Si(count)=Si(count+1);
        Si(count+1)=temp;
        count=count-1;
    end
end
HL(Ip)=HLP;
return;
```

Algoritma hufftablen.m

```
function HLlen = HuffTabLen(HL)

if (nargin ~= 1);
    error('Fungsi harus mempunyai satu input.');
```

```
end
if (nargout ~= 1);
    error('Fungsi harus mempunyai satu input.');
```

```
end
HL=HL(:);
Prev=2;HLlen=0;
ZeroCount=0;
L=length(HL);
```

```

for l=1:L
    if HL(l)==0
        ZeroCount=ZeroCount+1;
    else
        while (ZeroCount > 0)
            if ZeroCount<3; HLlen=HLlen+5*ZeroCount; ZeroCount=0;
            elseif ZeroCount<19; HLlen=HLlen+11; ZeroCount=0;
            elseif ZeroCount<275; HLlen=HLlen+15; ZeroCount=0;
            else HLlen=HLlen+15; ZeroCount=ZeroCount-274; end;
        end
        if HL(l)>16
            HLlen=HLlen+11;
        else
            Inc=HL(l)-Prev;
            if Inc<0; Inc=Inc+16; end;
            if (Inc==0); HLlen=HLlen+1;
            elseif (Inc==1); HLlen=HLlen+2;
            elseif (Inc==2); HLlen=HLlen+5;
            elseif (Inc==15); HLlen=HLlen+4;
            else HLlen=HLlen+7;
            end
            Prev=HL(l);
        end
    end
end
HLlen=HLlen+7;
return;

```

Algoritma huffcode.m

```

function HK = HuffCode(HL)

N=length(HL);
L=max(HL); HK=zeros(N,L);
[HLs,HLi] = sort(HL);
Code=zeros(1,L);
for n=1:N
    if (HLs(n)>0)
        HK(HLi(n),:) = Code;
        k = HLs(n);
        while (k>0)
            Code(k) = Code(k) + 1;
            if (Code(k)==2)
                Code(k) = 0;
                k=k-1;
            else
                break
            end
        end
    end
end
end

```

Algoritma hufftree.m

```
function Htree = HuffTree(HL,HK)

if nargin<1
    error('hufftree error. ');
end
if nargin<2
    HK = huffcode(HL);
end
N=length(HL);
Htree=zeros(N*2,3);
root=1; next=2;
for n=1:N
    if HL(n)>0
        pos=root;
        for k=1:HL(n)
            if ((Htree(pos,1)==0) & (Htree(pos,2)==0))
                Htree(pos,2)=next;
                Htree(pos,3)=next+1;
                next=next+2;
            end
            if HK(n,k)
                pos=Htree(pos,3);
            else
                pos=Htree(pos,2);
            end
        end
        Htree(pos,1)=1;
        Htree(pos,2)=n;
    end
end
if N==1
    Htree(1,3)=2;
end
return
```

Algoritma huff06.m

```
function varargout = Huff06(xC, ArgLevel, ArgSpeed)

global y Byte BitPos Speed Level
if (nargin < 1);
    error('Fungsi harus mempunyai input. ');
end
if (nargout < 1);
    error('Fungsi harus mempunyai output. ');
end

if (~iscell(xC))
    Encode=0;Decode=1;
```

```

    y=xC(:);
else
    Encode=1;Decode=0;
    Speed=0;
    Level=8;
    NumOfX = length(xC);
end

if Encode
    Res=zeros(NumOfX,4);
    y=zeros(10,1);
    Byte=0;BitPos=1;
    PutVLIC(NumOfX);
    Ltot=0;
    for num=1:NumOfX
        x=xC{num};
        x=full(x(:));
        L=length(x);Ltot=Ltot+L;
        y=[y(1:Byte);zeros(50+2*L,1)];
        maxx=max(x);
        minx=min(x);
        Negative=0;
        if ( (((maxx*4)>L) | (maxx>1023)) & (L>1) & (maxx>minx))
            LogCode=1;
        else
            LogCode=0;
        end
        PutBit(LogCode);
        PutBit(Negative);
        I=find(x);
        x=abs(x);
        if LogCode
            xa=x;
            x(I)=floor(log2(x(I)));
            xa(I)=xa(I)-2.^x(I);
            x(I)=x(I)+1;
        end
        [bits, ent]=EncodeVector(x);
        if LogCode
            for i=1:L
                for ii=(x(i)-1):(-1):1
                    PutBit(bitget(xa(i),ii));
                end
            end
            bits=bits+sum(x)-length(I);
            ent=ent+(sum(x)-length(I))/L;
        end
        if L>0; Res(num,1)=L; else Res(num,1)=1; end;
        Res(num,2)=ent;
        Res(num,3)=bits;
    end
end

```

```

y=y(1:Byte);
varargout(1) = {y};
if (nargout >= 2)
    if Ltot<1; Ltot=1; end;
    Res(NumOfX+1,3)=Byte*8;
    Res(NumOfX+1,1)=Ltot;
    Res(NumOfX+1,2)=sum(Res(1:NumOfX,1).*Res(1:NumOfX,2))/Ltot;
    Res(:,4)=Res(:,3)./Res(:,1);
    varargout(2) = {Res};
end
end

```

```

if Decode
    Byte=0;BitPos=1;
    NumOfX=GetVLIC;
    xC=cell(NumOfX,1);
    for num=1:NumOfX
        LogCode=GetBit;
        Negative=GetBit;
        x=DecodeVector;
        L=length(x);
        I=find(x);
        if Negative
            Sg=zeros(size(I));
            for i=1:length(I); Sg(i)=GetBit; end;
            Sg=Sg*2-1;
        else
            Sg=ones(size(I));
        end
        if LogCode
            xa=zeros(L,1);
            for i=1:L
                for ii=2:x(i)
                    xa(i)=2*xa(i)+GetBit;
                end
            end
            x(I)=2.^(x(I)-1);
            x=x+xa;
        end
        x(I)=x(I).*Sg;
        xC{num}=x;
    end
    varargout(1) = {xC};
end

```

```

return

```

```

function [bits, ent] = EncodeVector(x, bits, HL, Maxx, Meanx)
global y Byte BitPos Speed Level
Level = Level - 1;
MaxL=50000;

```

```

L=length(x);
if L==0
    PutBit(0);
    PutVLIC(L);
    PutBit(0);
    bits=2+6;
    ent=0;
    Level = Level + 1;
    return
end
if L==1
    PutBit(0);
    PutVLIC(L);
    PutVLIC(x(1));
    bits=1+2*6;
    if (x(1)>=16); bits=bits+4; end;
    if (x(1)>=272); bits=bits+4; end;
    if (x(1)>=4368); bits=bits+5; end;
    if (x(1)>=69904); bits=bits+5; end;
    if (x(1)>=1118480); bits=bits+4; end;
    ent=0;
    Level = Level + 1;
    return
end
if max(x)==min(x)
    PutBit(0);
    PutVLIC(L);
    for i=1:7; PutBit(1); end;
    PutVLIC(x(1));
    bits=1+6+7+6;
    if (x(1)>=16); bits=bits+4; end;
    if (x(1)>=272); bits=bits+4; end;
    if (x(1)>=4368); bits=bits+5; end;
    if (x(1)>=69904); bits=bits+5; end;
    if (x(1)>=1118480); bits=bits+4; end;
    if (L>=16); bits=bits+4; end;
    if (L>=272); bits=bits+4; end;
    if (L>=4368); bits=bits+5; end;
    if (L>=69904); bits=bits+5; end;
    if (L>=1118480); bits=bits+4; end;
    ent=0;
    Level = Level + 1;
    return
end
I=find(x);
if (L/2-length(I))>50
    Maxx=max(x);
    Hi=IntHist(x,0,Maxx);
    Hinz=nonzeros(Hi);
    ent=log2(L)-sum(Hinz.*log2(Hinz))/L;
    x2=x(I);

```

```

I=[I(:);L+1];
for i=length(I):-1:2; I(i)=I(i)-I(i-1); end;
x1=I-1; % the runs
if Speed
    Byte=Byte+1;
else
    PutBit(0);
    PutVLIC(0);
    PutBit(1);
end;
[bits1, temp] = EncodeVector(x1);
[bits2, temp] = EncodeVector(x2);
bits=bits1+bits2+8;
Level = Level + 1;
return
end

if (nargin==1)
    Maxx=max(x);
    Meanx=mean(x);
    Hi=IntHist(x,0,Maxx);
    Hinz=nonzeros(Hi);
    ent=log2(L)-sum(Hinz.*log2(Hinz))/L;
    HL=HuffLen(Hi);
    HLlen=HuffTabLen(HL);
    bits=6+HLlen+sum(HL.*Hi);
    if (L>=16); bits=bits+4; end;
    if (L>=272); bits=bits+4; end;
    if (L>=4368); bits=bits+5; end;
    if (L>=69904); bits=bits+5; end;
    if (L>=1118480); bits=bits+4; end;
else
    ent=0;
end
if (L>MaxL)
    L1=ceil(L/2);L2=L-L1;
    x1=x(1:L1);x2=x((L1+1):L);
elseif ((Level > 0) & (L>10))
    xm=median(x);
    x1=zeros(L,1);x2=zeros(L,1);
    x2(1)=x(1);i1=0;i2=1;
    for i=2:L
        if (x(i-1) <= xm)
            i1=i1+1; x1(i1)=x(i);
        else
            i2=i2+1; x2(i2)=x(i);
        end
    end
    x1=x1(1:i1);x2=x2(1:i2);
    L1=length(x1);L2=length(x2);
    Maxx1=max(x1);Maxx2=max(x2);

```

```

Meanx1=mean(x1);Meanx2=mean(x2);
Hi1=IntHist(x1,0,Maxx1);
Hi2=IntHist(x2,0,Maxx2);
HL1=hufflen(Hi1);HL2=hufflen(Hi2);
HLlen1=HuffTabLen(HL1);
HLlen2=HuffTabLen(HL2);
bits1=6+HLlen1+sum(HL1.*Hi1);
bits2=6+HLlen2+sum(HL2.*Hi2);
if (L1>=16); bits1=bits1+4; end;
if (L1>=272); bits1=bits1+4; end;
if (L1>=4368); bits1=bits1+5; end;
if (L1>=69904); bits1=bits1+5; end;
if (L1>=1118480); bits1=bits1+4; end;
if (L2>=16); bits2=bits2+4; end;
if (L2>=272); bits2=bits2+4; end;
if (L2>=4368); bits2=bits2+5; end;
if (L2>=69904); bits2=bits2+5; end;
if (L2>=1118480); bits2=bits2+4; end;
else
    bits1=bits;bits2=bits;
end
if (L>MaxL)
    if Speed
        BitPos=BitPos-1;
        if (~BitPos); Byte=Byte+1; BitPos=8; end;
    else
        PutBit(1);
    end;
    [bits1, temp] = EncodeVector(x1);
    [bits2, temp] = EncodeVector(x2);
    bits=bits1+bits2+1;
elseif ((bits1+bits2) < bits)
    if Speed
        BitPos=BitPos-1;
        if (~BitPos); Byte=Byte+1; BitPos=8; end;
    else
        PutBit(1);
    end;
    [bits1, temp] = EncodeVector(x1, bits1, HL1, Maxx1, Meanx1);
    [bits2, temp] = EncodeVector(x2, bits2, HL2, Maxx2, Meanx2);
    bits=bits1+bits2+1;
else
    bits=bits+1;
    if Speed
        Byte=Byte+floor(bits/8);
        BitPos=BitPos-mod(bits,8);
        if (BitPos<=0); BitPos=BitPos+8; Byte=Byte+1; end;
    else
        StartPos=Byte*8-BitPos;
        PutBit(0);
        PutVLIC(L);
    end;
end

```

```

PutHuffTab(HL);
HK=huffcode(HL);
for i=1:L;
    n=x(i)+1;
    for k=1:HL(n)
        PutBit(HK(n,k));
    end
end
BitsUsed=Byte*8-BitPos-StartPos;
end
end
Level = Level + 1;
return

```

```

function x = DecodeVector
global y Byte BitPos
MaxL=50000;
if GetBit
    x1=DecodeVector;
    x2=DecodeVector;
    L=length(x1)+length(x2);
    if (L>MaxL)
        x=[x1(:);x2(:)];
    else
        xm=median([x1;x2]);
        x=zeros(L,1);
        x(1)=x2(1);
        i1=0;i2=1;
        for i=2:L
            if (x(i-1) <= xm)
                i1=i1+1; x(i)=x1(i1);
            else
                i2=i2+1; x(i)=x2(i2);
            end
        end
    end
end
else
    L=GetVLIC;
    if (L>1)
        x=zeros(L,1);
        HL=GetHuffTab;
        if length(HL)
            Htree=HuffTree(HL);
            root=1;pos=root;
            l=0;
            while l<L
                if GetBit
                    pos=Htree(pos,3);
                else
                    pos=Htree(pos,2);
                end
            end
        end
    end
end

```

```

        if Htree(pos,1)
            l=l+1;
            x(l)=Htree(pos,2)-1;
            pos=root;
        end
    end
else
    x=x+GetVLIC;
end
elseif L==0
    if GetBit
        x1=DecodeVector;
        x2=DecodeVector;
        I=x1;
        I=I+1;
        L=length(I);
        for i=2:L;I(i)=I(i-1)+I(i); end;
        x=zeros(I(L)-1,1);
        x(I(1:(L-1)))=x2;
    else
        x=[];
    end
elseif L==1
    x=GetVLIC;
end
return

```

```

function PutHuffTab(HL)
global y Byte BitPos
HL=HL(:);
Prev=2;
ZeroCount=0;
L=length(HL);

for l=1:L
    if HL(l)==0
        ZeroCount=ZeroCount+1;
    else
        while (ZeroCount > 0)
            if ZeroCount<3
                for i=1:ZeroCount
                    PutBit(1);PutBit(1);PutBit(0);PutBit(1);PutBit(1);
                end
                ZeroCount=0;
            elseif ZeroCount<19
                PutBit(1);PutBit(1);PutBit(1);PutBit(0);PutBit(0);PutBit(0);PutBit(0);
                for (i=4:-1:1); PutBit(bitget(ZeroCount-3,i)); end;
                ZeroCount=0;
            elseif ZeroCount<275
                PutBit(1);PutBit(1);PutBit(1);PutBit(0);PutBit(0);PutBit(0);PutBit(1);
                for (i=8:-1:1); PutBit(bitget(ZeroCount-19,i)); end;
            end
        end
    end
end

```

```

        ZeroCount=0;
    else
        PutBit(1);PutBit(1);PutBit(1);PutBit(0);PutBit(0);PutBit(0);PutBit(1);
        for (i=8:-1:1); PutBit(1); end;
        ZeroCount=ZeroCount-274;
    end
end
if HL(l)>16
    PutBit(1);PutBit(1);PutBit(1);PutBit(0);PutBit(0);PutBit(1);PutBit(0);
    for (i=4:-1:1); PutBit(bitget(HL(l)-17,i)); end;
else
    Inc=HL(l)-Prev;
    if Inc<0; Inc=Inc+16; end;
    if (Inc==0)
        PutBit(0);
    elseif (Inc==1)
        PutBit(1);PutBit(0);
    elseif (Inc==2)
        PutBit(1);PutBit(1);PutBit(0);PutBit(1);PutBit(0);
    elseif (Inc==15)
        PutBit(1);PutBit(1);PutBit(0);PutBit(0);
    else
        PutBit(1);PutBit(1);PutBit(1);
        for (i=4:-1:1); PutBit(bitget(Inc,i)); end;
    end
    Prev=HL(l);
end
end
end
for (i=7:-1:1); PutBit(1); end;
return;

```

```

function HL=GetHuffTab
global y Byte BitPos

```

```

Prev=2;
ZeroCount=0;
HL=zeros(10000,1);
HLi=0;
EndOfTable=0;

```

```

while ~EndOfTable
    if GetBit
        if GetBit
            if GetBit
                Inc=0;
                for (i=1:4); Inc=Inc*2+GetBit; end;
                if Inc==0
                    ZeroCount=0;
                    for (i=1:4); ZeroCount=ZeroCount*2+GetBit; end;
                    HLi=HLi+ZeroCount+3;

```

```

elseif Inc==1
    ZeroCount=0;
    for (i=1:8); ZeroCount=ZeroCount*2+GetBit; end;
    HLi=HLi+ZeroCount+19;
elseif Inc==2
    HLi=HLi+1;
    HL(HLi)=0;
    for (i=1:4); HL(HLi)=HL(HLi)*2+GetBit; end;
    HL(HLi)=HL(HLi)+17;
elseif Inc==15
    EndOfTable=1;
else
    Prev=Prev+Inc;
    if Prev>16; Prev=Prev-16; end;
    HLi=HLi+1;HL(HLi)=Prev;
end
else
    if GetBit
        if GetBit
            HLi=HLi+1;
        else
            Prev=Prev+2;
            if Prev>16; Prev=Prev-16; end;
            HLi=HLi+1;HL(HLi)=Prev;
        end
    else
        Prev=Prev-1;
        if Prev<1; Prev=16; end;
        HLi=HLi+1;HL(HLi)=Prev;
    end
end
else
    Prev=Prev+1;
    if Prev>16; Prev=1; end;
    HLi=HLi+1;HL(HLi)=Prev;
end
else
    HLi=HLi+1;HL(HLi)=Prev;
end
end
if HLi>0
    HL=HL(1:HLi);
else
    HL=[];
end

return;

function PutVLIC(N)
global y Byte BitPos
if (N<0)

```

```

    error('Huff06-PutVLIC: Number is negative.');
```

```
elseif (N<16)
    PutBit(0);PutBit(0);
    for (i=4:-1:1); PutBit(bitget(N,i)); end;
elseif (N<272)
    PutBit(0);PutBit(1);
    N=N-16;
    for (i=8:-1:1); PutBit(bitget(N,i)); end;
elseif (N<4368)
    PutBit(1);PutBit(0);
    N=N-272;
    for (i=12:-1:1); PutBit(bitget(N,i)); end;
elseif (N<69940)
    PutBit(1);PutBit(1);PutBit(0);
    N=N-4368;
    for (i=16:-1:1); PutBit(bitget(N,i)); end;
elseif (N<1118480)
    PutBit(1);PutBit(1);PutBit(1);PutBit(0);
    N=N-69940;
    for (i=20:-1:1); PutBit(bitget(N,i)); end;
elseif (N<17895696)
    PutBit(1);PutBit(1);PutBit(1);PutBit(1);
    N=N-1118480;
    for (i=24:-1:1); PutBit(bitget(N,i)); end;
else
    error('Huff06-PutVLIC: Number is too large.');
```

```
end
return
```

```
function N=GetVLIC
global y Byte BitPos
N=0;
if GetBit
    if GetBit
        if GetBit
            if GetBit
                for (i=1:24); N=N*2+GetBit; end;
                N=N+1118480;
            else
                for (i=1:20); N=N*2+GetBit; end;
                N=N+69940;
            end
        else
            for (i=1:16); N=N*2+GetBit; end;
            N=N+4368;
        end
    else
        for (i=1:12); N=N*2+GetBit; end;
        N=N+272;
    end
else
```

```

if GetBit
    for (i=1:8); N=N*2+GetBit; end;
    N=N+16;
else
    for (i=1:4); N=N*2+GetBit; end;
end
end
return

function PutBit(Bit)
global y Byte BitPos
BitPos=BitPos-1;
if (~BitPos); Byte=Byte+1; BitPos=8; end;
y(Byte) = bitset(y(Byte),BitPos,Bit);
return

function Bit=GetBit
global y Byte BitPos
BitPos=BitPos-1;
if (~BitPos); Byte=Byte+1; BitPos=8; end;
Bit=bitget(y(Byte),BitPos);
return;

function Hi=IntHist(W,i1,i2);
W=W(:);
L=length(W);
Hi=zeros(i2-i1+1,1);
if (i2-i1)>50
    for l=1:L
        i=W(l)-i1+1;
        Hi(i)=Hi(i)+1;
    end
else
    for i=i1:i2
        I=find(W==i);
        Hi(i-i1+1)=length(I);
    end
end
return;

```

LAMPIRAN B
CITRA ASLI
DAN
CITRA HASIL DEKOMPRESI



Citra asli finger_1.bmp



Citra hasil dekompresi tingkat 1



Citra hasil dekompres tingkat 2



Citra hasil dekompres tingkat 3



Citra asli finger_2.bmp



Citra hasil dekompresi tingkat 1



Citra hasil dekompres tingkat 2



Citra hasil dekompres tingkat 3



Citra asli finger_3.bmp



Citra hasil dekompresi tingkat 1



Citra hasil dekompres tingkat 2



Citra hasil dekompres tingkat 3



Citra asli finger_4.bmp



Citra hasil dekompresi tingkat 1



Citra hasil dekompres tingkat 2



Citra hasil dekompres tingkat 3



Citra asli finger_5.bmp



Citra hasil dekompresi tingkat 1



Citra hasil dekompres tingkat 2



Citra hasil dekompres tingkat 3



Citra asli finger_6.bmp



Citra hasil dekompresi tingkat 1



Citra hasil dekompres tingkat 2



Citra hasil dekompres tingkat 3



Citra asli finger_7.bmp



Citra hasil dekompresi tingkat 1



Citra hasil dekompres tingkat 2



Citra hasil dekompres tingkat 3



Citra asli finger_8.bmp



Citra hasil dekompresi tingkat 1



Citra hasil dekompres tingkat 2



Citra hasil dekompres tingkat 3



Citra asli finger_9.bmp



Citra hasil dekompresi tingkat 1



Citra hasil dekompres tingkat 2



Citra hasil dekompres tingkat 3



Citra asli finger_10.bmp



Citra hasil dekompresi tingkat 1



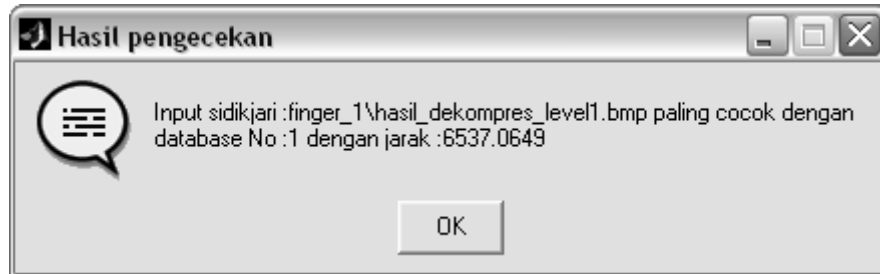
Citra hasil dekompres tingkat 2



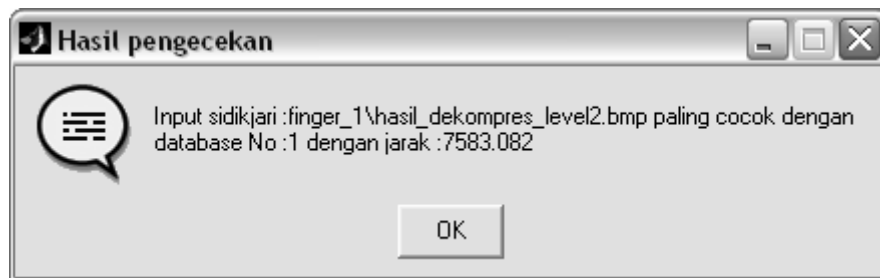
Citra hasil dekompres tingkat 3

LAMPIRAN C
TAMPILAN HASIL PENGECEKAN SIDIK JARI

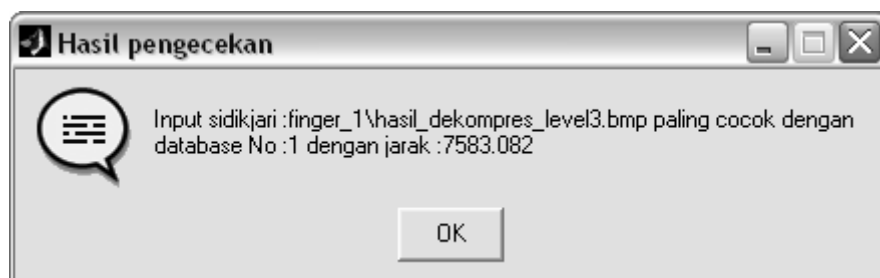
Hasil pengecekan dekompresi finger_1.bmp level 1 dengan fingerprint yang ada di database.



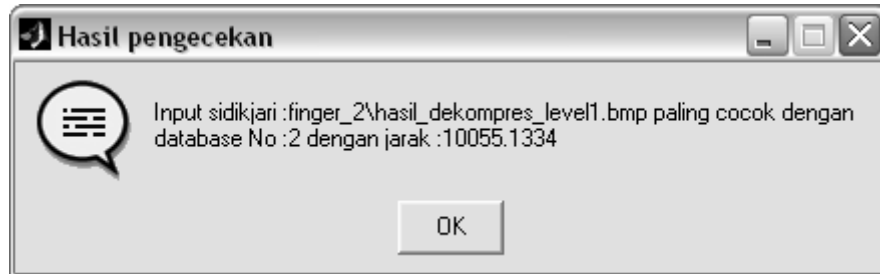
Hasil pengecekan dekompresi finger_1.bmp level 2 dengan fingerprint yang ada di database.



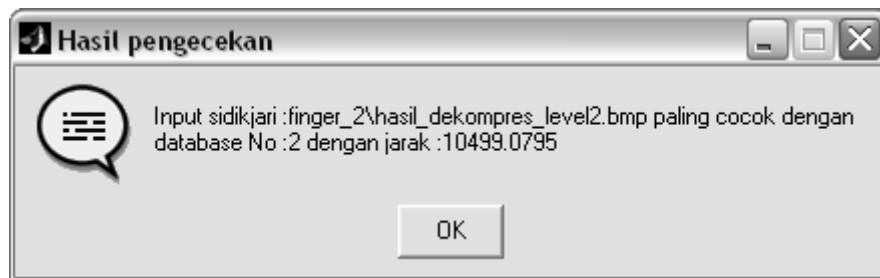
Hasil pengecekan dekompresi finger_1.bmp level 3 dengan fingerprint yang ada di database.



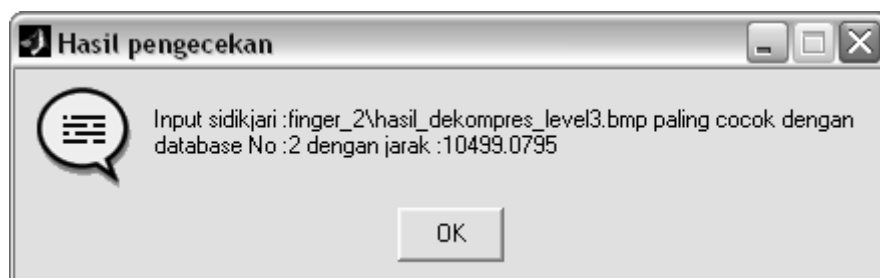
Hasil pengecekan dekompresi finger_2.bmp level 1 dengan fingerprint yang ada di database.



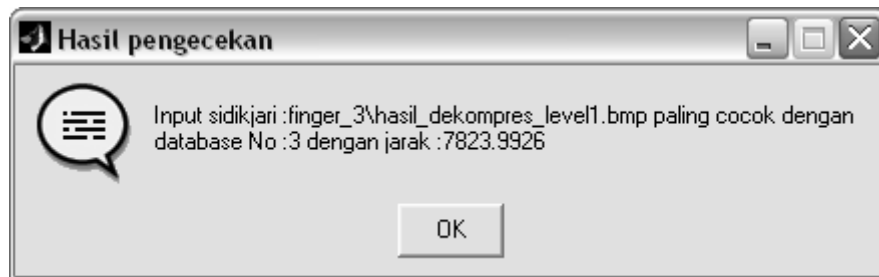
Hasil pengecekan dekompresi finger_2.bmp level 2 dengan fingerprint yang ada di database.



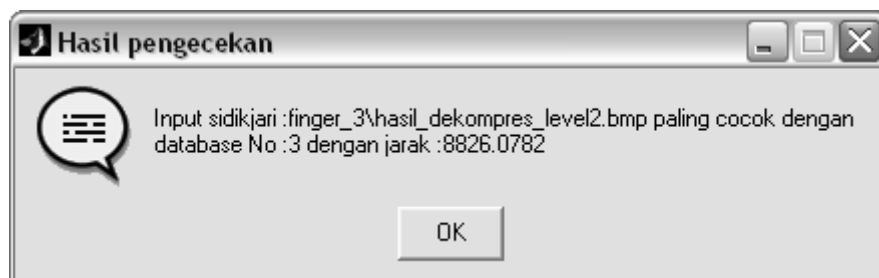
Hasil pengecekan dekompresi finger_2.bmp level 3 dengan fingerprint yang ada di database.



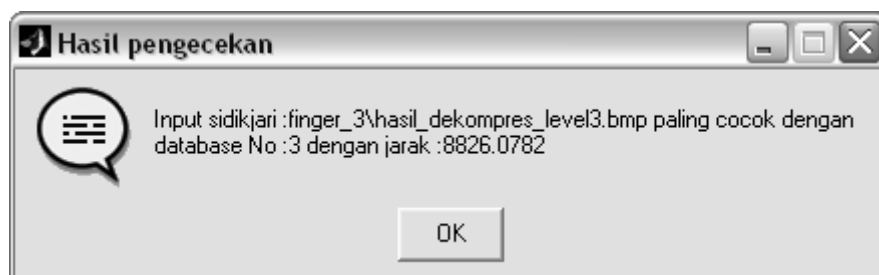
Hasil pengecekan dekompresi finger_3.bmp level 1 dengan fingerprint yang ada di database.



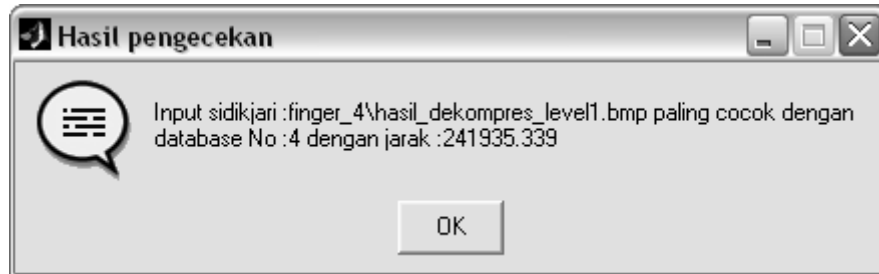
Hasil pengecekan dekompresi finger_3.bmp level 2 dengan fingerprint yang ada di database.



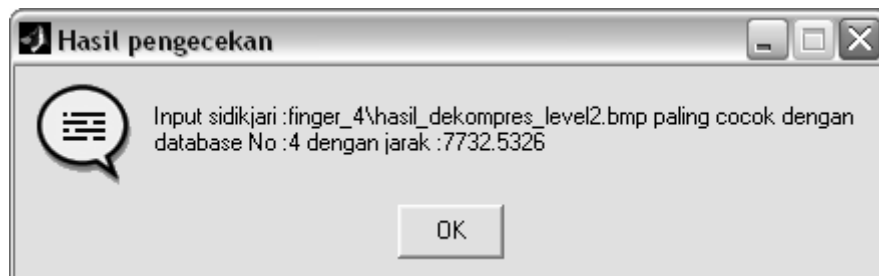
Hasil pengecekan dekompresi finger_3.bmp level 3 dengan fingerprint yang ada di database.



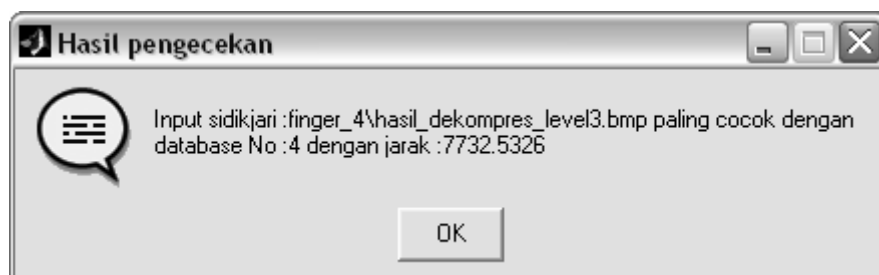
Hasil pengecekan dekompresi finger_4.bmp level 1 dengan fingerprint yang ada di database.



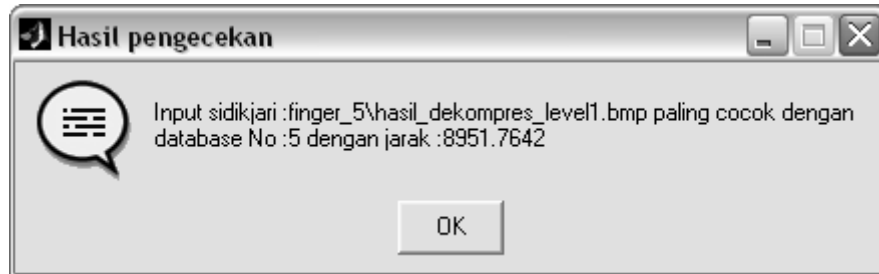
Hasil pengecekan dekompresi finger_4.bmp level 2 dengan fingerprint yang ada di database.



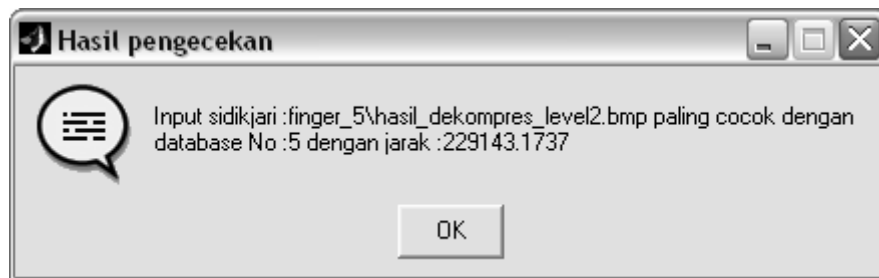
Hasil pengecekan dekompresi finger_4.bmp level 3 dengan fingerprint yang ada di database.



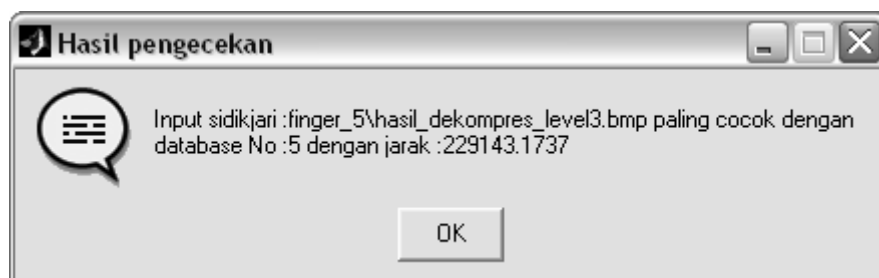
Hasil pengecekan dekompresi finger_5.bmp level 1 dengan fingerprint yang ada di database.



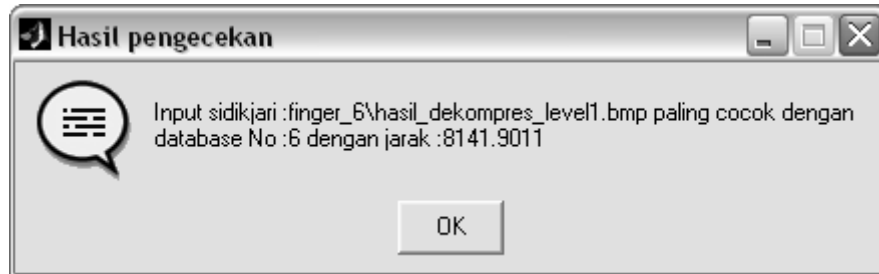
Hasil pengecekan dekompresi finger_5.bmp level 2 dengan fingerprint yang ada di database.



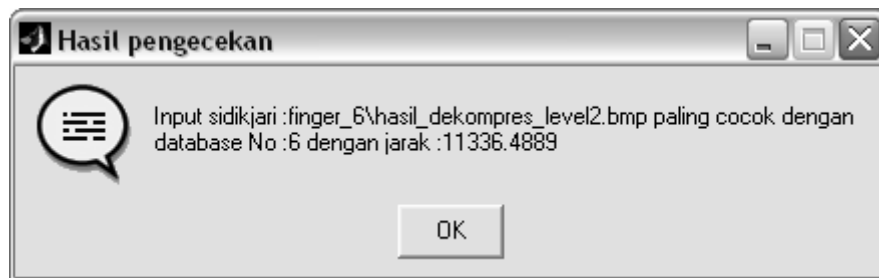
Hasil pengecekan dekompresi finger_5.bmp level 3 dengan fingerprint yang ada di database.



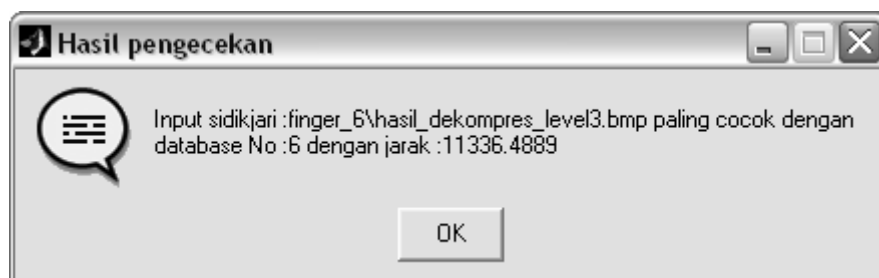
Hasil pengecekan dekompresi finger_6.bmp level 1 dengan fingerprint yang ada di database.



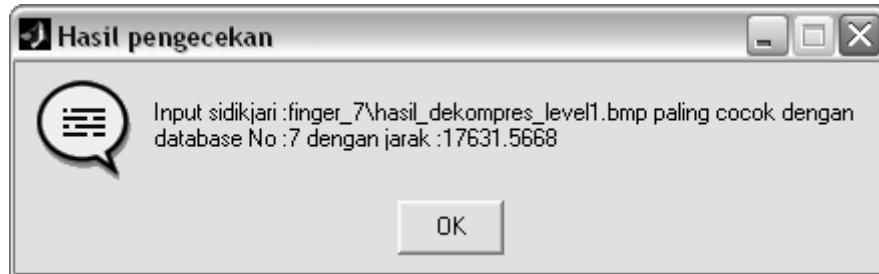
Hasil pengecekan dekompresi finger_6.bmp level 2 dengan fingerprint yang ada di database.



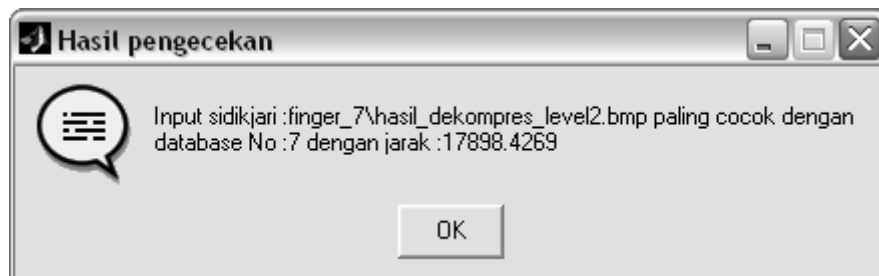
Hasil pengecekan dekompresi finger_6.bmp level 3 dengan fingerprint yang ada di database.



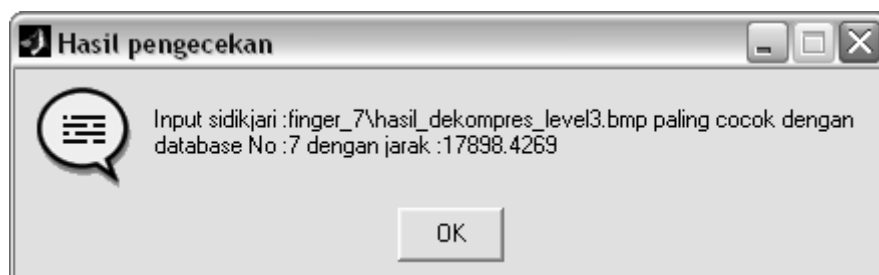
Hasil pengecekan dekompresi finger_7.bmp level 1 dengan fingerprint yang ada di database.



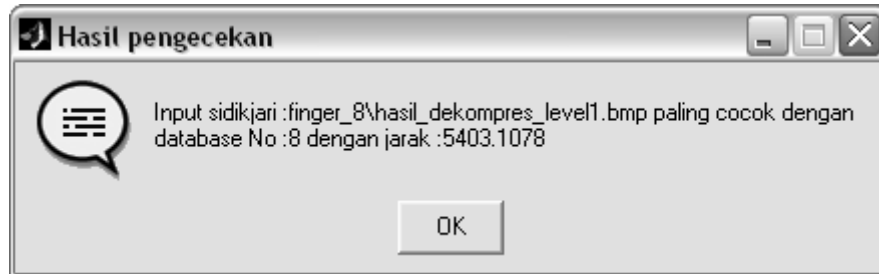
Hasil pengecekan dekompresi finger_7.bmp level 2 dengan fingerprint yang ada di database.



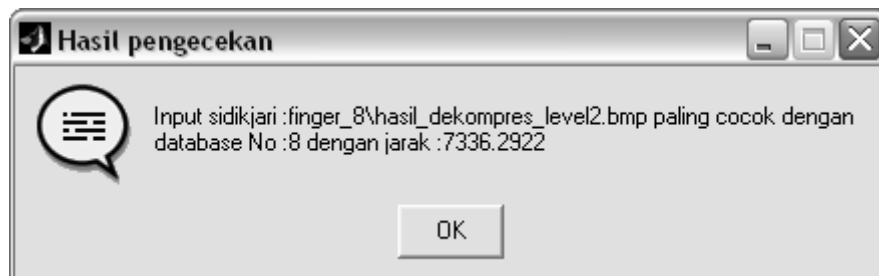
Hasil pengecekan dekompresi finger_7.bmp level 3 dengan fingerprint yang ada di database.



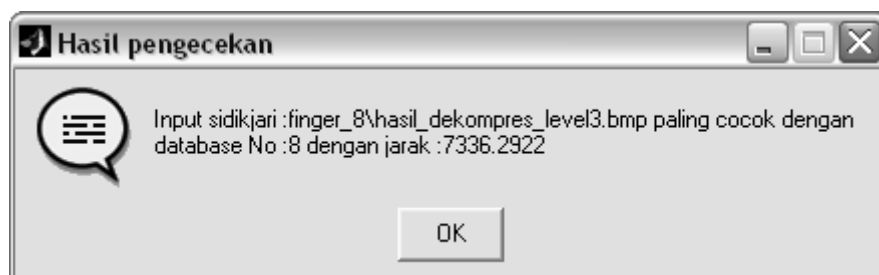
Hasil pengecekan dekompresi finger_8.bmp level 1 dengan fingerprint yang ada di database.



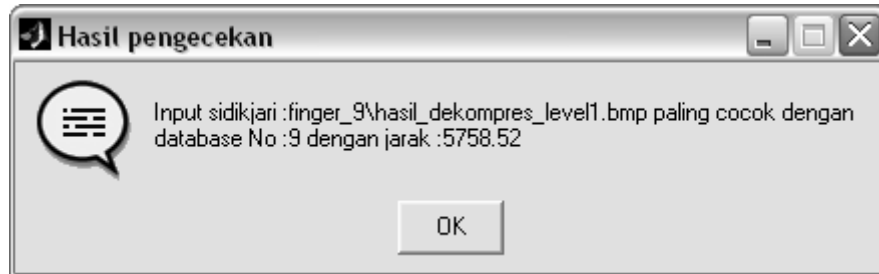
Hasil pengecekan dekompresi finger_8.bmp level 2 dengan fingerprint yang ada di database.



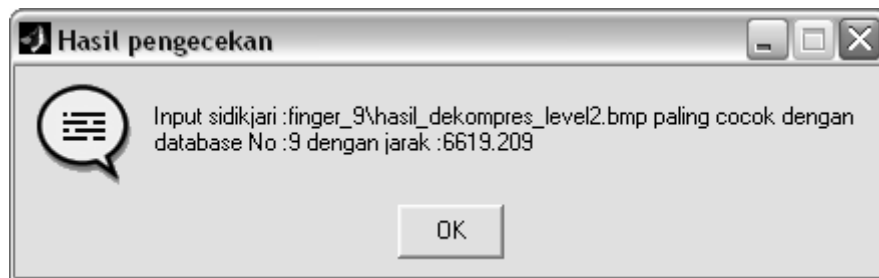
Hasil pengecekan dekompresi finger_8.bmp level 3 dengan fingerprint yang ada di database.



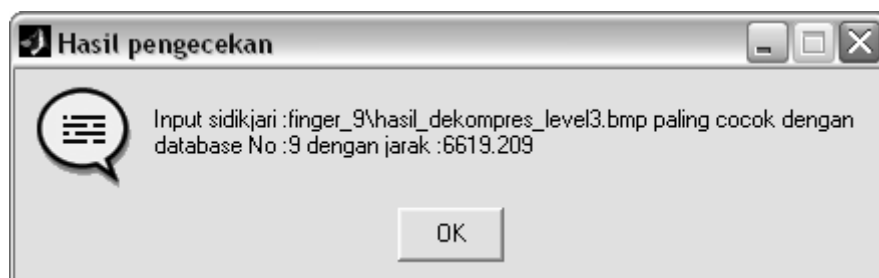
Hasil pengecekan dekompresi finger_9.bmp level 1 dengan fingerprint yang ada di database.



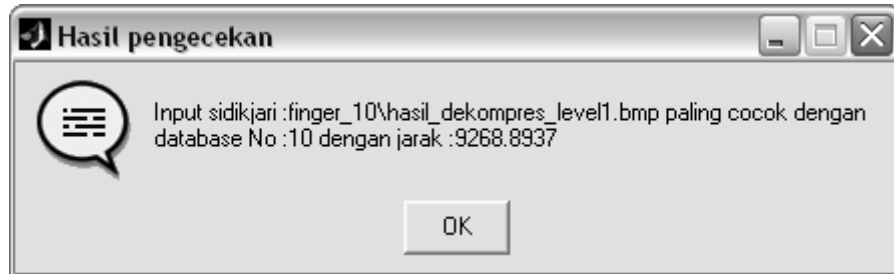
Hasil pengecekan dekompresi finger_9.bmp level 2 dengan fingerprint yang ada di database.



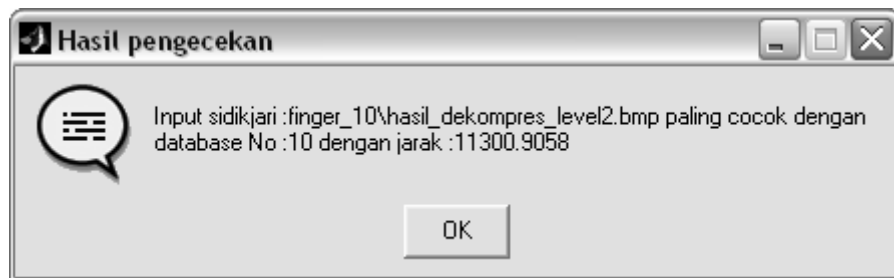
Hasil pengecekan dekompresi finger_9.bmp level 3 dengan fingerprint yang ada di database.



Hasil pengecekan dekompresi finger_10.bmp level 1 dengan fingerprint yang ada di database.



Hasil pengecekan dekompresi finger_10.bmp level 2 dengan fingerprint yang ada di database.



Hasil pengecekan dekompresi finger_10.bmp level 3 dengan fingerprint yang ada di database.

