REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN

CIPTAAN

Dalam rangka pelindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	:	EC00202021237, 6 Juli 2020
Pencipta		
Nama	:	**Zaenal Abidin, S.Si., M.Cs., Ph.D.**
Alamat	:	Jl. Sriwibowo No. 4, Kembangarum, Semarang Barat , Semarang, Jawa Tengah, 50148
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**Zaenal Abidin, S.Si., M.Cs., Ph.D.**
Alamat	:	Jl. Sriwibowo No. 4, Kembangarum, Semarang Barat , Semarang , Jawa Tengah, 50148
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Program Komputer**
Judul Ciptaan	:	**Aplikasi "Sistem Pengenalan Ekspresi Wajah Menggunakan Metode Fisherface Dengan Pendekatan Jaringan Syaraf Tiruan Backpropagation"**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	10 Desember 2010, di Yokyakarta
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	:	000193309

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

SISTEM PENGENALAN EKSPRESI WAJAH MENGGUNAKAN METODE FISHERFACE DENGAN PENDEKATAN JARINGAN SYARAF TIRUAN BACKPROPAGATION

Zaenal Abidin, S.Si., M.Cs., Ph.D

PRAKATA

Puji dan syukur penulis panjatkan kepada Allah SWT karena hanya dengan rahmat dan ridha-Nya penulis dapat menyelesaikan buku panduan ini. Shalawat serta salam juga penulis panjatkan kepada Nabi Muhammad SAW sebagai penuntun umat manusia di muka bumi ini, karena hanya dengan tuntunannya penulis dapat menjalani setiap langkah dalam kehidupannya dengan bijak dan penuh kesabaran.

Segala kekurangan dan ketidaksempurnaan yang dapat ditemukan dalam buku panduan ini, disadari sebagai keterbatasan yang dimiliki oleh penulis sendiri. Karena itulah penulis sangat mengharapkan adanya pengembangan lebih lanjut yang sangat mungkin dilakukan oleh siapa pun yang memiliki minat atau ketertarikan yang sama.

Dalam penyusunan buku panduan ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada seluruh pihak yang turut membantu dan memberikan dukungan dalam penulisan buku panduan ini.

Terakhir, mohon maaf dan terima kasih juga penulis sampaikan untuk semua pihak yang telah membantu pembuatan buku panduan ini, namun tidak dapat penulis sebutkan satu persatu. Semoga buku panduan ini dapat bermanfaat bagi pembaca.

Semarang, 10 Juni 2020

Penulis

SISTEM PENGENALAN EKSPRESI WAJAH MENGGUNAKAN METODE FISHERFACE DENGAN PENDEKATAN JARINGAN SYARAF TIRUAN BACKPROPAGATION

Sistem pengenalan ekspresi wajah diimplementasikan dengan menggunakan MATLAB 7.8.0. *User interface* dibuat dengan menggunakan fasilitas GUI yang ada di MATLAB. Sebelum sistem siap untuk melakukan proses pengenalan, sistem terlebih dahulu akan memuat data citra untuk dilatih.

1 Pelatihan Citra Ekspresi Wajah

Pelatihan citra dapat dilakukan dengan beberapa tahapan seperti dijelaskan pada sub bab berikut.

1.1 Input data pelatihan

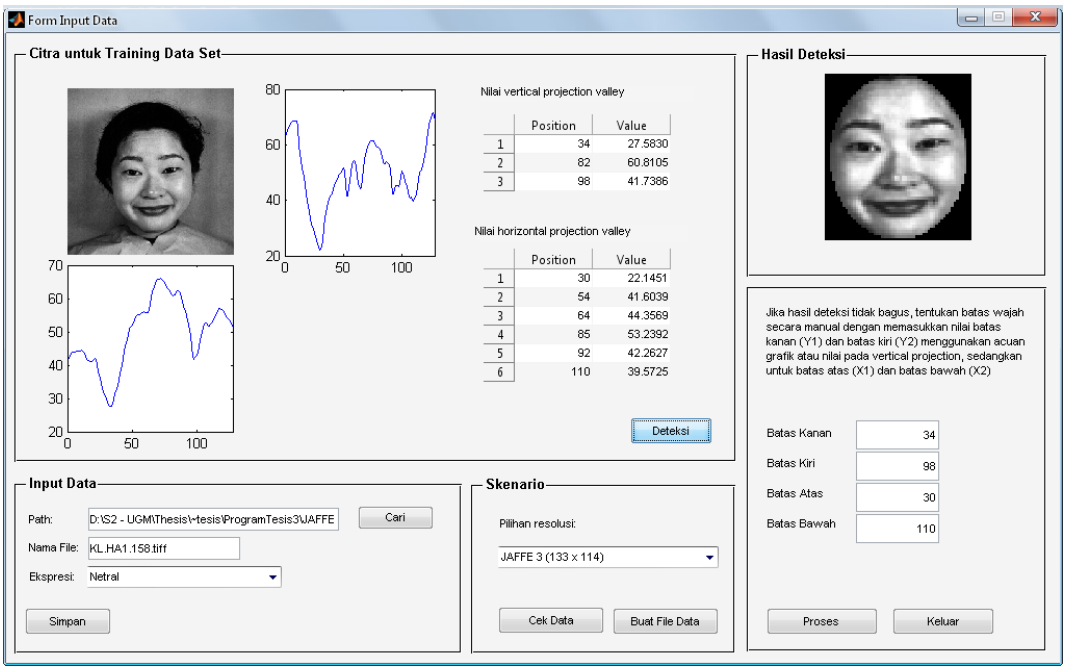
Sebelum melakukan konstruksi *fisherface*, maka terlebih dahulu disiapkan data citra yang akan dilatih. Implementasi program untuk mempersiapkan data citra latih dapat dilihat pada Gambar 1. Matriks representasi dari citra yang akan dilatih diubah menjadi vektor kolom, dan kelas ekspresi dari citra dipilih sesuai dengan ekspresi yang sesuai. Selanjutnya data setiap citra latih akan disimpan dalam sebuah file berekstensi *.mat. Tampilan antarmuka input data *training* dapat dilihat pada Gambar 2.

1.2 Konstruksi *fisherface*

Proses konstruksi *fisherface*, terdiri dari perhitungan PCA dan LDA. Di mana keduanya dijadikan dalam satu buah fungsi yang diberi nama “Fisherface”. Cuplikan kode program untuk perhitungan PCA dan LDA berturut-turut ditunjukkan pada Gambar 3 dan 4.

```
Imh=handles.citra;[m,n]=size(Imh);
vektor=double(reshape(Imh,m*n,1));
pilihan=get(handles.PopEkspresi,'Value');
switch pilihan
    case 1
        target = [0 0 0 0 0 0 1];
    case 2
        target = [0 0 0 0 0 1 0];
    case 3
        target = [0 0 0 0 1 0 0];
    case 4
        target = [0 0 0 1 0 0 0];
    case 5
        target = [0 0 1 0 0 0 0];
    case 6
        target = [0 1 0 0 0 0 0];
    case 7
        target = [1 0 0 0 0 0 0];
end;
savefile='A1_DataTraining.mat'; load A1_DataTraining.mat;
[m,n]=size(CA);
if isempty(CA{1,1}) && isempty(CA{1,2}) && isempty(CA{1,3})
      CA(1,:)={vektor,target,handles.pathfile};
else
      CA(m+1,:)={vektor,target,handles.pathfile};
end
save(savefile,'CA');
```

Gambar 1. Cuplikan program input data pelatihan



Gambar 2. Antarmuka halaman input data pelatihan

```
m_data = mean(T,2);
A = T - repmat(m_data,1,P);
C = A'*A;
[V D] = eig(C);
C_eig_vec = [];
for i = 1 : P-Jum_Kelas
    C_eig_vec = [C_eig_vec V(:,i)];
end

V_PCA = A * C_eig_vec;
ProjectedImages_PCA = [];
for i = 1 : P
    temp = V_PCA'*A(:,i);
    ProjectedImages_PCA = [ProjectedImages_PCA temp];
end
```

Gambar 3. Cuplikan program perhitungan PCA

```
    m_PCA = mean(ProjectedImages_PCA,2);
    m = zeros(P-Jum_Kelas,Jum_Kelas);
    Sw = zeros(P-Jum_Kelas,P-Jum_Kelas);
    Sb = zeros(P-Jum_Kelas,P-Jum_Kelas);

    for i = 1 : Jum_Kelas
        m(:,i) = mean( ( ProjectedImages_PCA(:,((i-
    1)*Pop_Kelas+1):i*Pop_Kelas) ), 2 )';

        S  = zeros(P-Jum_Kelas,P-Jum_Kelas);
        for j = ( (i-1)*Pop_Kelas+1 ) : ( i*Pop_Kelas )
            S = S + (ProjectedImages_PCA(:,j)-
    m(:,i))*(ProjectedImages_PCA(:,j)-m(:,i))';
        end

        Sw = Sw + S;
        Sb = Sb + (m(:,i)-m_PCA) * (m(:,i)-m_PCA)';
    end

    [J_eig_vec, J_eig_val] = eig(Sb,Sw);
    J_eig_vec = fliplr(J_eig_vec);

    for i = 1 : Jum_Kelas-1
        V_LDA(:,i) = J_eig_vec(:,i);
    end

    for i = 1 : Jum_Kelas*Pop_Kelas
       ProjectedImages_Fisher(:,i) = V_LDA' *
       ProjectedImages_PCA(:,i);
    end
```

Gambar 4. Cuplikan program perhitungan LDA

1.3 Pembelajaran Backpropagation

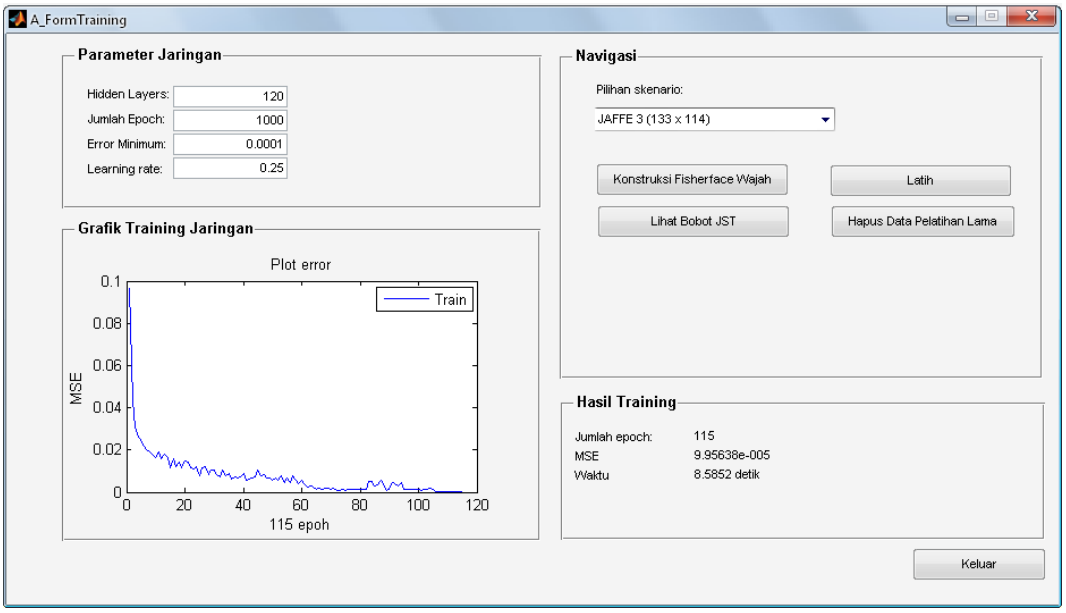
Sub bab ini membahas tentang bagian proses pembelajaran *backpropagation*. Masukkan untuk pembelajaran *backpropagation* adalah matriks *fisher* yang dihasilkan dari *konstruksi fisherface* sebelumnya. Kemudian pada saat akan dilakukan pembelajaran jaringan syaraf tiruan harus ditentukan dahulu parameter-parameter jaringan seperti jumlah *neuron hidden layer*, *learning rate*, maksimum *epoch* yang diinginkan, dan minimum *error* yang diharapkan. Fungsi aktivasi yang digunakan dari *input layer* ke *hidden layer* adalah fungsi sigmoid bipolar, sedangkan dari *hidden layer* ke *output layer* diaktivasi oleh fungsi sigmoid biner karena diharapkan output yang dihasilkan berada di antara interval 0 dan 1. Bobot awal dibangkitkan menggunakan inisialisasi Nguyen-Widrow. Dalam proses ini akan dihasilkan bobot jaringan syaraf tiruan yang nantikan akan digunakan pada tahap pengenalan. Cuplikan program pembelajaran *backpropagation* dapat dilihat pada Gambar 5.

```
load A1_DataTraining.mat;
load A1_DataFisher.mat;
hidden=get(handles.TxtHidden,'string');
alpha=get(handles.TxtAlpha,'string');
galat=get(handles.TxtGalat,'string');
epoch=get(handles.TxtEpoch,'string');
if isempty(hidden) || isempty(alpha) || isempty(galat) ||
isempty(epoch)
    msgbox('Data parameter JST harus diisi lengkap!','Pesan
Sistem','warn');
    return
end
hidden=str2double(hidden);
alpha=str2double(alpha);
galat=str2double(galat);
epoch=str2double(epoch);
% vektor target
t=[];
for i=1:length(CA)
    t=[t CA{i,2}.'];
end
[y,v,v0,w,w0,iterasi,MSE,p,MSEi,waktu] =
BackPropagation(ProjectedImages_Fisher,t,hidden,alpha,galat,epo
ch);
```

Gambar 5. Cuplikan program pembelajaran *backpropagation*

Pada Gambar 5 tentang cuplikan program pembelajaran *backpropagation*, terdapat pemanggilan fungsi “Backpropagation” dengan inputnya adalah vektor ciri, vektor target, jumlah *neuron hidden*, nilai *learning rate*, target *error* dan maksimum epoh yang diinginkan. Output proses pembelajaran disimpan dalam bentuk file. Dalam file ini, akan memuat arsitektur jaringan yang digunakan, dan nilai bobot JST serta nilai MSE-nya.

Tampilan antarmuka halaman pembelajaran sistem pengenalan ekspresi wajah ditunjukkan seperti pada Gambar 6.



Gambar 6. Antarmuka halaman pembelajaran pada saat dijalankan

2 Pengenalan Ekspresi Wajah

Dalam melakukan pengenalan ekspresi wajah, *user* pertama kali memasukkan citra wajah yang akan dikenali jenis ekspresinya. Selanjutnya dari citra wajah yang dimasukkan, akan dicari lokasi yang merupakan wajah melalui proses deteksi wajah.

1.4 Deteksi wajah

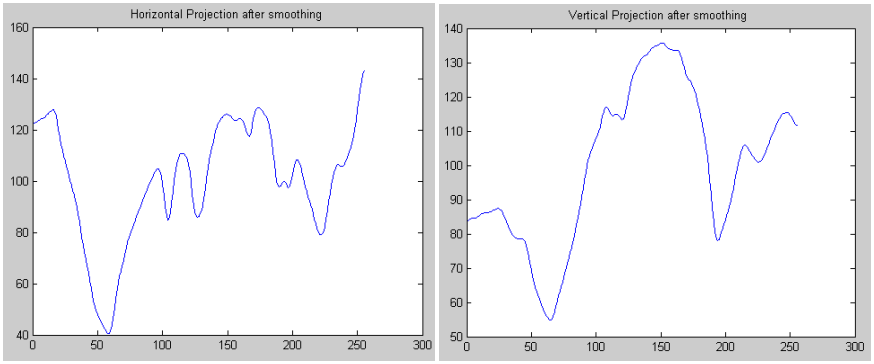
Sub bab ini menjelaskan tentang proses deteksi wajah. Metode yang digunakan dalam deteksi wajah adalah metode *integral projection*. Metode ini menjumlahkan piksel perbaris dan perkolomnya. Jadi setiap piksel dalam satu baris dijumlahkan sepanjang lebar dari citra, sebanyak tinggi citra dan selanjutnya disebut *horizontal integral projection*. Setiap piksel dalam satu kolom juga dijumlahkan sepanjang tinggi citra, sebanyak lebar citra, dan selanjutnya disebut dengan *vertikal integral projection*. Gambar 7 adalah cuplikan untuk mencari nilai dari *horizontal integral projection* dan *vertikal integral projection*.

```
XProj = sum(Imd,2);
YProj = sum(Imd,1);
```

Gambar 7. Cuplikan program mencari nilai *horizontal projection* dan *vertical projection*

```
h = fspecial('average', [5 5]);
XProj = imfilter(XProj,h,'replicate');
YProj = imfilter(YProj,h,'replicate');
plot(XProj);title('Horizontal Projection after smoothing');
plot(YProj);title('Vertical Projection after smoothing ');
```

(a)



(b)

Gambar 8. (a) Cuplikan program untuk menampilkan kurva *horizontal projection* dan *vertical projection* setelah *smoothing*
(b) Hasil *plotting*

Sebelum menentukan batas wajah, kurva *integral projection* terlebih dahulu di *smoothing* untuk memfilter lembah yang kecil dan mengeliminasi informasi yang menggangu. Penentuan batas kanan dan kiri wajah, digunakan *vertikal projection*, sedangkan batas atas dan batas bawah wajah, menggunakan *horizontal projection*. Penentuan lembah pada grafik *integral projection* digunakan fungsi puncaklembah. Fungsi ini akan mendaftar semua posisi titik minimum lokal (lembah grafik), dan maksimum lokal (puncak grafik).

```
function [maxtab, mintab]= puncaklembah(v, delta)
maxtab = []; mintab = [];
v = v(:); % vektor
if nargin < 3
  x = (1:length(v))';
else
    error('Jumlah input tidak benar');
end
if (length(delta(:)))>1
  error('Input argument DELTA harus skalar');
end
if delta <= 0
  error('Input argument DELTA harus positif');
end
mn = Inf; mx = -Inf;
mnpos = NaN; mxpos = NaN;
lookformax = 1;
for i=1:length(v)
  this = v(i);
  if this > mx, mx = this; mxpos = x(i); end
  if this < mn, mn = this; mnpos = x(i); end
  if lookformax
    if this < mx-delta
      maxtab = [maxtab ; mxpos mx];
      mn = this; mnpos = x(i);
      lookformax = 0;
    end
  else
    if this > mn+delta
      mintab = [mintab ; mnpos mn];
      mx = this; mxpos = x(i);
      lookformax = 1;
    end
  end
end
```

Gambar 9. Cuplikan program menentukan titik-titik yang merupakan titik minimum lokal dan titik maksimum lokal

Bentuk kurva *horizontal projection* dan *vertical projection* setelah mengalami *smoothing* dapat dilihat pada Gambar 8. sedangkan cuplikan program untuk fungsi puncak lembah dapat dilihat pada Gambar 9.

1.5 Konstruksi *fisherface* citra yang akan dikenali

Proses konstruksi *fisherface* citra yang akan dikenali ini merupakan bagian dari implementasi DFD level 1 proses pengenalan, setelah berhasil ditemukan daerah wajah dari citra yang akan dikenali. Dalam melakukan konstruksi *fisherface* di sini dibutuhkan *eigenvector* dari perhitungan PCA dan *eigenvector* dari perhitungan LDA, yang tersimpan dalam data file, dalam cuplikan kode pada Gambar 10 eigenvector tersimpan dalam file A1_DataFisher.mat. Output dari proses konstruksi *fisherface* ini adalah matriks proyeksi citra yang akan dikenali.

```
load A1_DataFisher.mat;
[m,n] = size(Imh);
vektor=double(reshape(Imh',m*n,1));
Difference = vektor - m_data;
ProjectedTestImage = V_LDA' * V_PCA' * Difference;
```

Gambar 10. Cuplikan program konstruksi *fisherface*

1.6 Simulasi jaringan syaraf tiruan *feedforward*

Proses simulasi jaringan syaraf tiruan *feedforward* digunakan untuk menghitung keluaran jaringan berdasarkan arsitektur, pola masukan dan fungsi aktivasi yang dipakai.

Jadi dengan pola masukan citra yang akan dikenali ini, diperoleh keluaran yang nantinya dapat digunakan untuk menentukan jenis kelas ekspresinya. Keluaran dari simulasi jaringan syaraf tiruan *feedforward* ini adalah vektor kolom berukuran 7×1. Nilai lebih dari atau sama dengan 0,6 akan dikonversi ke-1, sedangkan nilai kurang dari 0,6 akan dikonversi ke-0. Nilai 1 mengindikasikan bahwa ekspresi tertentu ada, sedangkan nilai 0 mengindikasikan ekspresi tertentu tidak ada. Cuplikan kode program untuk simulasi jaringan syaraf tiruan

feedforward dan sekaligus kode untuk menampilkan hasil *output* dari simulasi jaringan ditunjukkan pada Gambar 11.

```
y = FeedForward(ProjectedTestImage,p,v,v0,w,w0);
[baris,kolom]=find (y>=0.6);
[H,W]=size(y);
y2=zeros(H,W);
for i=1:length(y)
    y2(baris,kolom)=1;
end
set(handles.TxtJijik,'String',y2(1,1));
set(handles.TxtTakut,'String',y2(1,2));
set(handles.TxtTerkejut,'String',y2(1,3));
set(handles.TxtMarah,'String',y2(1,4));
set(handles.TxtSedih,'String',y2(1,5));
set(handles.TxtSenang,'String',y2(1,6));
set(handles.TxtNetral,'String',y2(1,7));
if y2 == [0 0 0 0 0 0 1]
    set(handles.TxtHasil,'String','Netral');
elseif y2 == [0 0 0 0 0 1 0]
    set(handles.TxtHasil,'String','Senang');
elseif y2 == [0 0 0 0 1 0 0]
    set(handles.TxtHasil,'String','Sedih');
elseif y2 == [0 0 0 1 0 0 0]
    set(handles.TxtHasil,'String','Marah');
elseif y2 == [0 0 1 0 0 0 0]
    set(handles.TxtHasil,'String','Terkejut');
elseif y2 == [0 1 0 0 0 0 0]
    set(handles.TxtHasil,'String','Takut');
elseif y2 == [1 0 0 0 0 0 0]
    set(handles.TxtHasil,'String','Jijik');
else
    set(handles.TxtHasil,'String','Tidak dikenali');
end
```

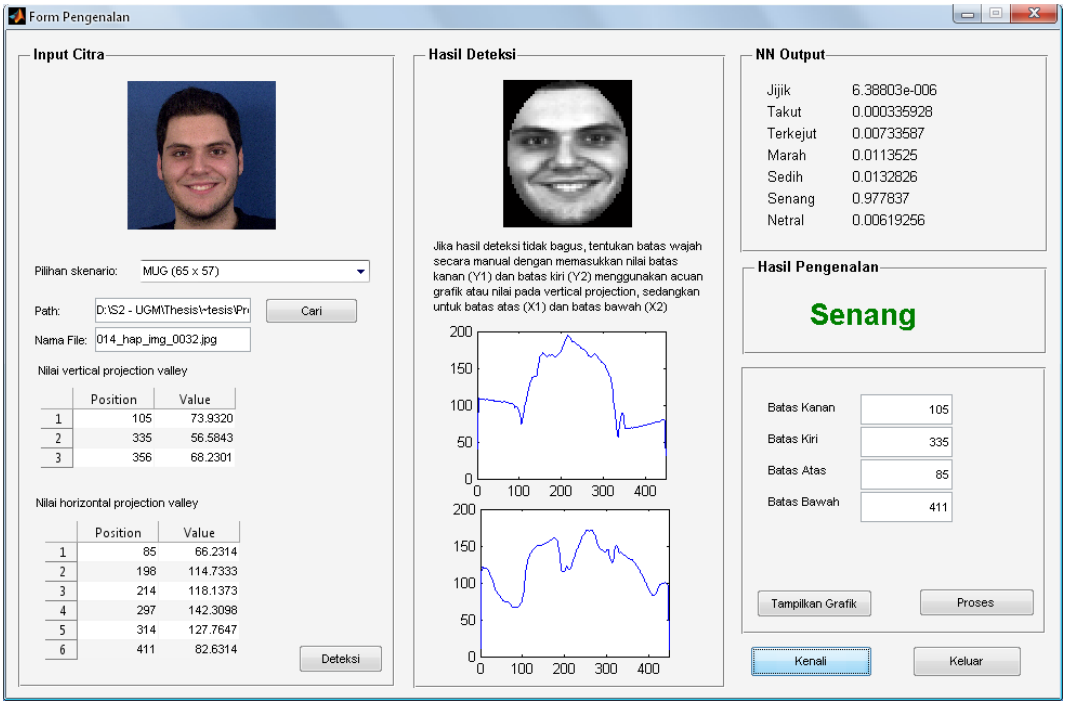
Gambar 11. Cuplikan program simulasi jaringan syaraf tiruan *feedforward*

Tampilan antarmuka pengenalan citra ekspresi wajah untuk user umum ditunjukkan seperti pada Gambar 12.

1.7 Penghitungan *recognition rate* untuk seluruh data uji

Gambar 13 merupakan cuplikan program untuk melakukan proses mendapatkan hasil pengenalan secara keseluruhan dari data uji baik dari citra

yang telah dilatih maupun yang belum dilatih. Keseluruhan citra uji, sebelumnya telah disimpan dalam data file. Misalnya untuk data JAFFE tipe I, data citra yang akan diuji disimpan dalam file A1_DataTestAll.mat. Data citra tersebutlah yang akan diproses. Pada tahap konstruksi *fisherface* di sini dibutuhkan *eigenvector* dari perhitungan PCA dan *eigenvector* dari perhitungan LDA, yang tersimpan dalam file A1_DataFisher.mat. Setelah matriks proyeksi fisher didapatkan maka, proses akan dilanjutkan dengan simulasi jaringan syaraf tiruan *feedforward*, pada tahap simulasi ini dibutuhkan bobot jaringan syaraf tiruan hasil pelatihan yang telah dilakukan sebelumnya, yang tersimpan dalam file A1_DataBobotJST.mat. Hasil keluaran jaringan syaraf tiruan secara keseluruhan akan disimpan dalam file A1_DataRecognizingAll.mat.



Gambar 12. Antarmuka halaman pengenalan untuk user umum saat dijalankan

Untuk menampilkan hasilnya, tinggal memanggil data file A1_DataRecognizingAll.mat. Adapun cuplikan program menampilkan hasil *recognition rate* dan *false positive* dari hasil pengenalan dapat dilihat pada

Gambar 14. Tampilan antarmuka hasil pengenalan untuk user administrator saat dijalankan ini dapat dilihat pada Gambar 15.

```
load A1_DataFisher.mat;
load A1_DataBobotJST.mat;
load A1_DataRecognizingAll.mat;

JmlData = size(CA,1);
for i=1:1:JmlData
    NamaFile = CA{i,3};
    Label = CA{i,2};
    Difference = CA{i,1} - m_data;
    ProjectedTestImage = V_LDA' * V_PCA' * Difference;
    y = FeedForward(ProjectedTestImage,p,v,v0,w,w0);
    [baris,kolom]=find (y>=0.6);
    [H,W]=size(y);
    y2=zeros(H,W);
    for k=1:length(y)
      y2(baris,kolom)=1;
    end
    Output=y2;
    if isequal(Label, Output)
      Ketr = 1;
    else
      Ketr = 0;
    end
    savefile='A1_DataRecognizingAll.mat';
    [m,n]=size(Hasil);
    if isempty(Hasil{1,1}) && isempty(Hasil{1,2}) &&
isempty(Hasil{1,3}) && isempty(Hasil{1,4})
Hasil(1,:)={NamaFile,Label,Output,Ketr};
    else
      Hasil(m+1,:)={NamaFile,Label,Output,Ketr};
    end
    save(savefile,'Hasil');
end
```

Gambar 13. Cuplikan program untuk melakukan proses hasil pengenalan secara keseluruhan

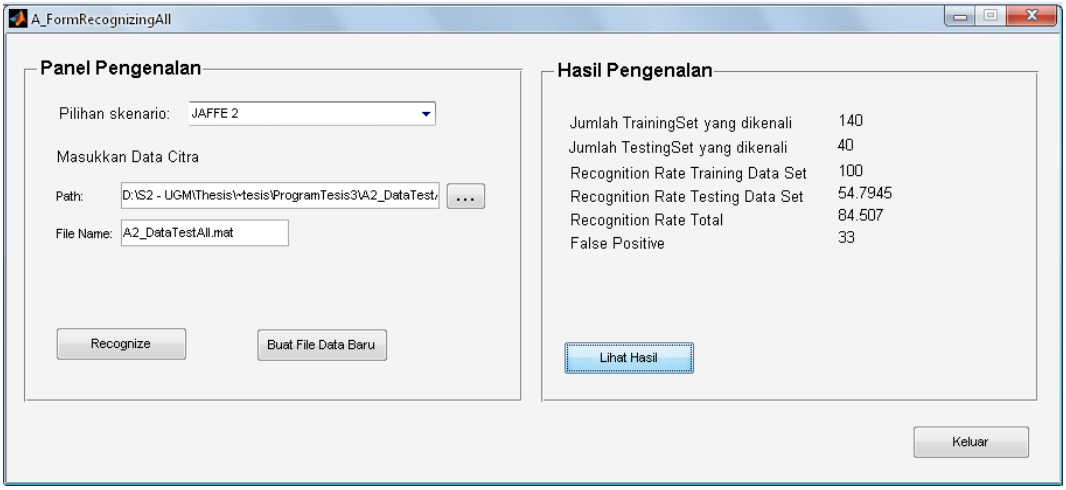
```
load A1_DataRecognizingAll.mat;
HasilAkhir=cell2mat(Hasil(:,4));
HasilTrain=HasilAkhir(1:140,:);
HasilTest=HasilAkhir(141:213,:);

indeks = find(HasilTrain==1);
JumSuksesTraining = length(indeks);
RecRateTraining = JumSuksesTraining/140*100;

indeks1 = find(HasilTest==1);
JumSuksesTesting = length(indeks1);
RecRateTesting = JumSuksesTesting/73*100;

JumSuksesTotal = JumSuksesTraining + JumSuksesTesting;
RecRateTotal = JumSuksesTotal/213*100;
JumGagal = 213 - JumSuksesTotal;
set(handles.TxtJumTrain,'string',JumSuksesTraining);
set(handles.TxtJumTest,'string',JumSuksesTesting);
set(handles.TxtRecTrain,'string',RecRateTraining);
set(handles.TxtRecTest,'string',RecRateTesting);
set(handles.TxtRecTot,'string',RecRateTotal);
set(handles.TxtFalse,'string',JumGagal);
```

Gambar 14. Cuplikan program untuk menampilkan hasil pengenalan secara keseluruhan pada citra JAFFE tipe I



Gambar 15. Tampilan antarmuka halaman pengenalan untuk user administrator

LAMPIRAN

Lampiran 1

Kode Fungsi Backpropagation

```
function [y,v,v0,w,w0,iterasi,MSE,p,MSEi,waktu] =
BackPropagation(x,t,p,alpha,e,epoch)
if (nargin < 6)
    disp('Jumlah argumen input tidak benar!');
end

%% ==Langkah_0==Inisialisasi==================================
%% Arsitektur
n = 6; %input
m = 7; %output

%% vektor Input / Target
x = x.';
t=t.';
trainRecords = size(x,1);

%% Inisialisasi Bobot
% v = randn(n,p);
% w = randn(p,m);
% w0 = randn(1,m);
% v0 = randn(1,p);
beta = 0.7 * p^(1/n);
v = randc(n,p);
w = randc(p,m);
w0 = randc(1,m);
vj = sqrt(sum(v.^2));
for j =1:p
    v(:,j) = (beta * v(:,j)) ./ vj(j);
end
v0 = rand(1,p)-beta*ones(1,p);
%% Hidden Layer
zin = zeros(trainRecords,p);
z = zeros(trainRecords,p);
din = zeros(trainRecords,p);
dj = zeros(trainRecords,p);
chv = zeros(n,p);
chv0 = zeros(1,p);

%% Output Layer
yin = zeros(trainRecords,m);
y = zeros(trainRecords,m);
d = zeros(trainRecords,m);
chw = zeros(p,m);
chw0 = zeros(1,m);
iterasi = 0;
er = 0; error = 0;
tic;

%%% == Langkah_1 == Selama kondisi berhenti adalah salah ==
lakukan Langkah 2-9 ===
```

```
while er==0

%%% == Langkah_2 == Untuk setiap pasangan pola pelatihan
=========== lakukan Langkah 3-8 ===
    for Tp=1:trainRecords
%% Feed forward:
%%% ==Langkah_3==Tiap unit input (X_i,i=1,2,3,?,n) menerima sinyal
masukan
%%% xi dan mengirim sinyal ini ke seluruh unit pada lapisan
berikutnya (lapisan tersembunyi)
%%% ==Langkah_4=================================================
        %%% Hidden Layer
        for j=1:p
            zin(Tp,j) = 0;
            for i=1:n
                zin(Tp,j) = zin(Tp,j) + x(Tp,i) * v(i,j);
            end
            zin(Tp,j) = v0(j) + zin(Tp,j);
             z(Tp,j) = (2/(1+exp(-zin(Tp,j))))-1; %Fungsi aktivasi
        end
%%% == Langkah_5
==================================================
        %%% Output Layer
        for k=1:m
            yin(Tp,k) = 0;
            for j=1:p
                yin(Tp,k) = yin(Tp,k) + z(Tp,j) * w(j,k);
            end
            yin(Tp,k) = w0(k) + yin(Tp,k);
             y(Tp,k) = 1/(1+exp(-yin(Tp,k))); %Fungsi aktivasi
        end
%% Backpropagation dari galat (Training Started)
%%% ==Langkah_6=================================================
        for k=1:m
        d(Tp,k) = 0;
    %%% Error Info = (t - y) * (Turunan fungsi aktivasi pada layer
kedua)
        d(Tp,k) = (t(Tp,k) - y(Tp,k)) * (y(Tp,k) *(1 - y(Tp,k)));
            for j=1:p
                chw(j,k) = alpha * d(Tp,k) * z(Tp,j);
            end
            chw0(k) = alpha * d(Tp,k);
        end
%%% ==Langkah_7=================================================
        for j=1:p
            din(Tp,j) = 0;
            for k=1:m
                din(Tp,j) = din(Tp,j) + d(Tp,k) * w(j,k);
            end
            dj(Tp,j) = 0;
%%% Error Info = din * (Turunan fungsi aktivasi pada layer
pertama)
           dj(Tp,j) = (din(Tp,j) * ((1/2) * (1 + z(Tp,j))* (1 -
z(Tp,j))));
            for i=1:n
                chv(i,j) = alpha * dj(Tp,j) * x(Tp,i);
```

```
            end
            chv0(j) = alpha * dj(Tp,j);
        end
%% ==Langkah_8==Update Bobot dan bias=======================
        for k=1:m
            for j=1:p
                w(j,k)=w(j,k)+chw(j,k);
            end
           w0(k)=w0(k)+chw0(k);
        end
        for j=1:p
            for i=1:n
                v(i,j)=v(i,j)+chv(i,j);
            end
           v0(j)=v0(j)+chv0(j);
        end
    end
%% ==Langkah_9==Test_Kondisi berhenti=========================

%     error = sqrt((t-y).^2);

    MSE=(1/(7*trainRecords))*(sum(sum((t-y).^2)));

    %%% Update variabel kondisi berhenti
    if MSE <= e % W/P = e, W = bobot,P =input, e = error
        er =1;
    else
        er = 0;
    end
    iterasi = iterasi + 1;
    MSEi(iterasi) = MSE;

    if (iterasi >= epoch)
        break
    end

end %% akhir loop while langkah 1
waktu = toc;
%
save('BobotJST.mat','y','v','v0','w','w0','iterasi','MSE','p','err
orMax')
% erLine = ones(1,size(errorMax,2))*e;
% clf('reset'), cla reset;
% plot(1:size(errorMax,2),errorMax,1:size(errorMax,2),erLine,'r');
% xlabel('Iterasi '), ylabel('error'), title('Plot error');
end %% akhir sintak fungsi
```

Kode program halaman depan

```
function varargout = depan(varargin)
% DEPAN M-file for depan.fig
%      DEPAN, by itself, creates a new DEPAN or raises the
existing
```

```
%      singleton*.
%
%      H = DEPAN returns the handle to a new DEPAN or the handle
to
%      the existing singleton*.
%
%      DEPAN('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in DEPAN.M with the given input
arguments.
%
%      DEPAN('Property','Value',...) creates a new DEPAN or raises
the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before depan_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to depan_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 depan

% Last Modified by GUIDE v2.5 19-Oct-2010 18:41:36

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @depan_OpeningFcn, ...
                   'gui_OutputFcn',  @depan_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
% End initialization code - DO NOT EDIT

% --- Executes just before depan is made visible.
function depan_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 depan (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

handles.axes1=imshow('\gbrdepan.jpg');

% --- Outputs from this function are returned to the command line.
function varargout = depan_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 during object creation, after setting all
properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (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 axes1

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

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

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

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

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

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

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

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

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

% --------------------------------------------------------------------
function jaffe_Callback(hObject, eventdata, handles)
% hObject    handle to jaffe (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
A_FormRecognizing;

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

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

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

% --------------------------------------------------------------------
function keluar_Callback(hObject, eventdata, handles)
% hObject    handle to keluar (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
clear all;
close all;

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

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

% --------------------------------------------------------------------
function jaffeinput_Callback(hObject, eventdata, handles)
```

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

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

Kode Program Input Data

```
function varargout = A_FormInputData(varargin)
% A_FORMINPUTDATA M-file for A_FormInputData.fig
%      A_FORMINPUTDATA, by itself, creates a new A_FORMINPUTDATA
or raises the existing
%      singleton*.
%
%      H = A_FORMINPUTDATA returns the handle to a new
A_FORMINPUTDATA or the handle to
%      the existing singleton*.
%
%      A_FORMINPUTDATA('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in A_FORMINPUTDATA.M with the given
input arguments.
%
%      A_FORMINPUTDATA('Property','Value',...) creates a new
A_FORMINPUTDATA or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before A_FormInputData_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to A_FormInputData_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
A_FormInputData

% Last Modified by GUIDE v2.5 24-Nov-2010 22:13:52

% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
   'gui_Singleton',  gui_Singleton, ...
   'gui_OpeningFcn', @A_FormInputData_OpeningFcn, ...
   'gui_OutputFcn',  @A_FormInputData_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
% End initialization code - DO NOT EDIT


% --- Executes just before A_FormInputData is made visible.
function A_FormInputData_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 A_FormInputData (see
VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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


% --- Outputs from this function are returned to the command line.
function varargout = A_FormInputData_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;
```

```
function TxtPath_Callback(hObject, eventdata, handles)
% hObject    handle to TxtPath (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 TxtPath as text
%str2double(get(hObject,'String')) returns contents of TxtPath as
a double


% --- Executes during object creation, after setting all
properties.
function TxtPath_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtPath (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 TxtNama_Callback(hObject, eventdata, handles)
% hObject    handle to TxtNama (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 TxtNama as text
%str2double(get(hObject,'String')) returns contents of TxtNama as
a double


% --- Executes during object creation, after setting all
properties.
function TxtNama_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtNama (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 TomSimpan.
function TomSimpan_Callback(hObject, eventdata, handles)
% hObject    handle to TomSimpan (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

load mask.mat;
I = handles.citra;
[Img XProj YProj vmintab hmintab X1 X2 Y1 Y2] = deteksiwajah(I);
Imh1=imresize(histeq(Img),[33 29]).*uint8(imresize(mask,[33 29]));
Imh2=imresize(histeq(Img),[65 57]).*uint8(imresize(mask,[65 57]));
Imh3=imresize(histeq(Img),[130 114]).*uint8(imresize(mask,[130
114]));
imwrite(Imh1,strcat('D:\S2 -
UGM\Thesis\~tesis\ProgramTesis3\A1_All\',handles.nama));
imwrite(Imh2,strcat('D:\S2 -
UGM\Thesis\~tesis\ProgramTesis3\A2_All\',handles.nama));
imwrite(Imh3,strcat('D:\S2 -
UGM\Thesis\~tesis\ProgramTesis3\A3_All\',handles.nama));
[m1,n1] = size(Imh1);
[m2,n2] = size(Imh2);
[m3,n3] = size(Imh3);
vektor1=double(reshape(Imh1',m1*n1,1));%membuat flat vektor
vektor2=double(reshape(Imh2',m2*n2,1));
vektor3=double(reshape(Imh3',m3*n3,1));
pilihan=get(handles.PopEkspresi,'Value');
switch pilihan
    case 1
target = [0 0 0 0 0 0 1];
    case 2
target = [0 0 0 0 0 1 0];
    case 3
target = [0 0 0 0 1 0 0];
    case 4
target = [0 0 0 1 0 0 0];
    case 5
target = [0 0 1 0 0 0 0];
    case 6
target = [0 1 0 0 0 0 0];
    case 7
target = [1 0 0 0 0 0 0];
end;

%% simpan ke file *.mat
savefile='A1_DataTestAll.mat'; %nama file database yang akan
dibuat
load A1_DataTestAll.mat;
[m,n]=size(CA);
if isempty(CA{1,1}) && isempty(CA{1,2}) && isempty(CA{1,3})
    CA(1,:)={vektor1,target,handles.nama}; %menambahkan record
baru
else
```

```
    CA(m+1,:)={vektor1,target,handles.nama}; %menambahkan record
baru
end
save(savefile,'CA'); %menyimpan record

savefile='A2_DataTestAll.mat'; %nama file database yang akan
dibuat
load A2_DataTestAll.mat;
[m,n]=size(CA);
if isempty(CA{1,1}) && isempty(CA{1,2}) && isempty(CA{1,3})
    CA(1,:)={vektor2,target,handles.nama}; %menambahkan record
baru
else
    CA(m+1,:)={vektor2,target,handles.nama}; %menambahkan record
baru
end
save(savefile,'CA'); %menyimpan record

savefile='A3_DataTestAll.mat'; %nama file database yang akan
dibuat
load A3_DataTestAll.mat;
[m,n]=size(CA);
if isempty(CA{1,1}) && isempty(CA{1,2}) && isempty(CA{1,3})
    CA(1,:)={vektor3,target,handles.nama}; %menambahkan record
baru
else
    CA(m+1,:)={vektor3,target,handles.nama}; %menambahkan record
baru
end
save(savefile,'CA'); %menyimpan record


guidata(hObject,handles);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla(handles.axes1,'reset');
cla(handles.axes2,'reset');
cla(handles.axes5,'reset');
cla(handles.axes6,'reset');
set(handles.uitable3,'data','');
set(handles.uitable4,'data','');
set(handles.TxtKanan,'string','');
set(handles.TxtKiri,'string','');
set(handles.TxtAtas,'string','');
set(handles.TxtBawah,'string','');
[nama,path]=uigetfile('*.png;*.bmp;*.jpg;*.pgm;*.tiff;*.gif','Buka
Gambar','D:\S2 - UGM\Thesis\~tesis\ProgramTesis3\JAFFE -
TestDataSet');
handles.citra=imread(fullfile(path,nama));
handles.nama=nama;
```

```
handles.pathfile=fullfile(path,nama);
axes(handles.axes1);
imshow(handles.citra);
set(handles.TxtPath,'string',handles.pathfile);
set(handles.TxtNama,'string',nama);
guidata(hObject,handles);

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


% --- Executes on selection change in PopEkspresi.
function PopEkspresi_Callback(hObject, eventdata, handles)
% hObject    handle to PopEkspresi (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 PopEkspresi
contents as cell array
%contents{get(hObject,'Value')} returns selected item from
PopEkspresi


% --- Executes during object creation, after setting all
properties.
function PopEkspresi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PopEkspresi (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 on selection change in PopSkenario.
function PopSkenario_Callback(hObject, eventdata, handles)
% hObject    handle to PopSkenario (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 PopSkenario
contents as cell array
%contents{get(hObject,'Value')} returns selected item from
PopSkenario


% --- Executes during object creation, after setting all
properties.
function PopSkenario_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PopSkenario (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 on button press in TomCek.
function TomCek_Callback(hObject, eventdata, handles)
% hObject    handle to TomCek (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        p=exist('A1_DataTraining.mat');
        if p==2
            msgbox('File A1_DataTraining.mat sudah ada!','Pesan
Sistem','warn');
        else
            msgbox('File A1_DataTraining.mat belum ada, silakan
buat!','Pesan Sistem','warn');
        end;
    case 2
        p=exist('A2_DataTraining.mat');
        if p==2
            msgbox('File A2_DataTraining.mat sudah ada!','Pesan
Sistem','warn');
        else
            msgbox('File A2_DataTraining.mat belum ada, silakan
buat!','Pesan Sistem','warn');
        end;
    case 3
        p=exist('A3_DataTraining.mat');
        if p==2
            msgbox('File A3_DataTraining.mat sudah ada!','Pesan
Sistem','warn');
```

```
        else
            msgbox('File A3_DataTraining.mat belum ada, silakan
buat!','Pesan Sistem','warn');
        end;
    case 4
        p=exist('B2_DataTraining.mat');
        if p==2
            msgbox('File A2_DataTraining.mat sudah ada!','Pesan
Sistem','warn');
        else
            msgbox('File A2_DataTraining.mat belum ada, silakan
buat!','Pesan Sistem','warn');
        end;
end;

% --- Executes on button press in TomBuatFile.
function TomBuatFile_Callback(hObject, eventdata, handles)
% hObject    handle to TomBuatFile (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        savefile='A1_DataTraining.mat';
        vektor=[];
        target=[];
        namafile=[];
        CA={vektor,target,namafile};
        save(savefile,'CA');
    case 2
        savefile='A2_DataTraining.mat';
        vektor=[];
        target=[];
        namafile=[];
        CA={vektor,target,namafile};
        save(savefile,'CA');
    case 3
        savefile='A3_DataTraining.mat';
        vektor=[];
        target=[];
        namafile=[];
        CA={vektor,target,namafile};
        save(savefile,'CA');
   case 4
        savefile='B2_DataTraining.mat';
        vektor=[];
        target=[];
        namafile=[];
        CA={vektor,target,namafile};
        save(savefile,'CA');
end;


% --- Executes on button press in TomDeteksi.
function TomDeteksi_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to TomDeteksi (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
load mask.mat;
I = handles.citra;
[Img XProj YProj vmintab hmintab X1 X2 Y1 Y2] = deteksiwajah(I);
axes(handles.axes6);
plot(YProj);xlim([0 length(YProj)]);
axes(handles.axes5);
plot(XProj); xlim([0 length(XProj)]);
set(handles.uitable3,'data',vmintab);
set(handles.uitable4,'data',hmintab);
Img2=imresize(histeq(Img),[65 57]).*uint8(imresize(mask,[65 57]));
axes(handles.axes2);
imshow(Img2);
set(handles.TxtKanan,'string',Y1);
set(handles.TxtKiri,'string',Y2);
set(handles.TxtAtas,'string',X1);
set(handles.TxtBawah,'string',X2);
guidata(hObject,handles);

function TxtKanan_Callback(hObject, eventdata, handles)
% hObject    handle to TxtKanan (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 TxtKanan as
text
%        str2double(get(hObject,'String')) returns contents of
TxtKanan as a double



% --- Executes during object creation, after setting all
properties.
function TxtKanan_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtKanan (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 TxtKiri_Callback(hObject, eventdata, handles)
% hObject    handle to TxtKiri (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 TxtKiri as text
%        str2double(get(hObject,'String')) returns contents of
TxtKiri as a double


% --- Executes during object creation, after setting all
properties.
function TxtKiri_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtKiri (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 TxtAtas_Callback(hObject, eventdata, handles)
% hObject    handle to TxtAtas (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 TxtAtas as text
%        str2double(get(hObject,'String')) returns contents of
TxtAtas as a double


% --- Executes during object creation, after setting all
properties.
function TxtAtas_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtAtas (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 TxtBawah_Callback(hObject, eventdata, handles)
% hObject    handle to TxtBawah (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 TxtBawah as
text
%        str2double(get(hObject,'String')) returns contents of
TxtBawah as a double


% --- Executes during object creation, after setting all
properties.
function TxtBawah_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtBawah (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 TomProses.
function TomProses_Callback(hObject, eventdata, handles)
% hObject    handle to TomProses (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.TomSimpan,'visible','off');
set(handles.TomSimpan1,'visible','on');
load mask.mat;
I = handles.citra;
Y1 = str2num(get(handles.TxtKanan,'string'));
Y2 = str2num(get(handles.TxtKiri,'string'));
X1 = str2num(get(handles.TxtAtas,'string'));
X2 = str2num(get(handles.TxtBawah,'string'));
Img = deteksiwajahnew(I,X1,X2,Y1,Y2);
Img2=imresize(histeq(Img),[65 57]).*uint8(imresize(mask,[65 57]));
axes(handles.axes2);
imshow(Img2);
guidata(hObject,handles);


% --- Executes on button press in TomSimpan1.
function TomSimpan1_Callback(hObject, eventdata, handles)
% hObject    handle to TomSimpan1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
set(handles.TomSimpan,'visible','on');
set(handles.TomSimpan1,'visible','off');
load mask.mat;
I = handles.citra;
Y1 = str2num(get(handles.TxtKanan,'string'));
Y2 = str2num(get(handles.TxtKiri,'string'));
X1 = str2num(get(handles.TxtAtas,'string'));
X2 = str2num(get(handles.TxtBawah,'string'));
Img = deteksiwajahnew(I,X1,X2,Y1,Y2);
Imh1=imresize(histeq(Img),[33 29]).*uint8(imresize(mask,[33 29]));
Imh2=imresize(histeq(Img),[65 57]).*uint8(imresize(mask,[65 57]));
Imh3=imresize(histeq(Img),[130 114]).*uint8(imresize(mask,[130
114]));
imwrite(Imh1,strcat('D:\S2 -
UGM\Thesis\~tesis\ProgramTesis3\A1_All\',handles.nama));
imwrite(Imh2,strcat('D:\S2 -
UGM\Thesis\~tesis\ProgramTesis3\A2_All\',handles.nama));
imwrite(Imh3,strcat('D:\S2 -
UGM\Thesis\~tesis\ProgramTesis3\A3_All\',handles.nama));
[m1,n1] = size(Imh1);
[m2,n2] = size(Imh2);
[m3,n3] = size(Imh3);
vektor1=double(reshape(Imh1',m1*n1,1));%membuat flat vektor
vektor2=double(reshape(Imh2',m2*n2,1));
vektor3=double(reshape(Imh3',m3*n3,1));
pilihan=get(handles.PopEkspresi,'Value');
switch pilihan
    case 1
target = [0 0 0 0 0 0 1];
    case 2
target = [0 0 0 0 0 1 0];
    case 3
target = [0 0 0 0 1 0 0];
    case 4
target = [0 0 0 1 0 0 0];
    case 5
target = [0 0 1 0 0 0 0];
    case 6
target = [0 1 0 0 0 0 0];
    case 7
target = [1 0 0 0 0 0 0];
end;

%% simpan ke file *.mat
savefile='A1_DataTestAll.mat'; %nama file database yang akan
dibuat
load A1_DataTestAll.mat;
[m,n]=size(CA);
if isempty(CA{1,1}) && isempty(CA{1,2}) && isempty(CA{1,3})
    CA(1,:)={vektor1,target,handles.nama}; %menambahkan record
baru
else
    CA(m+1,:)={vektor1,target,handles.nama}; %menambahkan record
baru
end
```

```
save(savefile,'CA'); %menyimpan record

savefile='A2_DataTestAll.mat'; %nama file database yang akan
dibuat
load A2_DataTestAll.mat;
[m,n]=size(CA);
if isempty(CA{1,1}) && isempty(CA{1,2}) && isempty(CA{1,3})
    CA(1,:)={vektor2,target,handles.nama}; %menambahkan record
baru
else
    CA(m+1,:)={vektor2,target,handles.nama}; %menambahkan record
baru
end
save(savefile,'CA'); %menyimpan record

savefile='A3_DataTestAll.mat'; %nama file database yang akan
dibuat
load A3_DataTestAll.mat;
[m,n]=size(CA);
if isempty(CA{1,1}) && isempty(CA{1,2}) && isempty(CA{1,3})
    CA(1,:)={vektor3,target,handles.nama}; %menambahkan record
baru
else
    CA(m+1,:)={vektor3,target,handles.nama}; %menambahkan record
baru
end
save(savefile,'CA'); %menyimpan record
```

Kode program antarmuka halaman pelatihan jaringan syaraf

```
function varargout = A_FormTraining(varargin)
% A_FORMTRAINING M-file for A_FormTraining.fig
%      A_FORMTRAINING, by itself, creates a new A_FORMTRAINING or
raises the existing
%      singleton*.
%
%      H = A_FORMTRAINING returns the handle to a new
A_FORMTRAINING or the handle to
%      the existing singleton*.
%
%      A_FORMTRAINING('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in A_FORMTRAINING.M with the given
input arguments.
%
%      A_FORMTRAINING('Property','Value',...) creates a new
A_FORMTRAINING or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before A_FormTraining_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to A_FormTraining_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
A_FormTraining

% Last Modified by GUIDE v2.5 28-Nov-2010 17:01:47

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
   'gui_Singleton',  gui_Singleton, ...
   'gui_OpeningFcn', @A_FormTraining_OpeningFcn, ...
   'gui_OutputFcn',  @A_FormTraining_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
% End initialization code - DO NOT EDIT


% --- Executes just before A_FormTraining is made visible.
function A_FormTraining_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 A_FormTraining (see
VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

```
% --- Outputs from this function are returned to the command line.
function varargout = A_FormTraining_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 TomKeluar.
function TomKeluar_Callback(hObject, eventdata, handles)
% hObject    handle to TomKeluar (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;
clear all;

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


% --- Executes on button press in TomLatih.
function TomLatih_Callback(hObject, eventdata, handles)
% hObject    handle to TomLatih (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
%% Pelatihan untuk seluruh wajah


skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        load A1_DataTraining.mat;
        load A1_DataFisher.mat;
        hidden=get(handles.TxtHidden,'string');
        alpha=get(handles.TxtAlpha,'string');
        galat=get(handles.TxtGalat,'string');
        epoch=get(handles.TxtEpoch,'string');
        if isempty(hidden) || isempty(alpha) || isempty(galat) ||
isempty(epoch)
            msgbox('Data parameter JST harus diisi
lengkap!','Pesan Sistem','warn');
            return
        end
```

```
        hidden=str2double(hidden);
        alpha=str2double(alpha);
        galat=str2double(galat);
        epoch=str2double(epoch);
        % vektor target
        t=[];
        for i=1:length(CA)
            t=[t CA{i,2}.'];
        end
        [y,v,v0,w,w0,iterasi,MSE,p,MSEi,waktu] =
BackPropagation(ProjectedImages_Fisher,t,hidden,alpha,galat,epoch)
;



        % Keterangan hasil
        axes(handles.axes2);
        plot(1:size(MSEi,2),MSEi);
        xlabel(strcat(num2str(iterasi),' epoh')), ylabel('MSE'),
title('Plot error'),legend('Train');

save('A1_DataBobotJST.mat','y','v','v0','w','w0','iterasi','MSE','
p')
        set(handles.text8,'string','Jumlah epoch:  ');
        set(handles.text10,'string',iterasi);
        set(handles.text12,'string','MSE');
        set(handles.text13,'string',MSE);
        set(handles.text22,'string','Waktu');
        set(handles.text23,'string',strcat(num2str(waktu),'
detik'));
    case 2
        load A2_DataTraining.mat;
        load A2_DataFisher.mat;
        hidden=get(handles.TxtHidden,'string');
        alpha=get(handles.TxtAlpha,'string');
        galat=get(handles.TxtGalat,'string');
        epoch=get(handles.TxtEpoch,'string');
        if isempty(hidden) || isempty(alpha) || isempty(galat) ||
isempty(epoch)
            msgbox('Data parameter JST harus diisi
lengkap!','Pesan Sistem','warn');
            return
        end
        hidden=str2double(hidden);
        alpha=str2double(alpha);
        galat=str2double(galat);
        epoch=str2double(epoch);
        % vektor target
        t=[];
        for i=1:length(CA)
            t=[t CA{i,2}.'];
        end
        [y,v,v0,w,w0,iterasi,MSE,p,MSEi,waktu] =
BackPropagation(ProjectedImages_Fisher,t,hidden,alpha,galat,epoch)
;

        % Keterangan hasil
```

```
        axes(handles.axes2);
        plot(1:size(MSEi,2),MSEi);
        xlabel(strcat(num2str(iterasi),' epoh')), ylabel('MSE'),
title('Plot error'),legend('Train');

save('A2_DataBobotJST.mat','y','v','v0','w','w0','iterasi','MSE','
p')
        set(handles.text8,'string','Jumlah epoch:  ');
        set(handles.text10,'string',iterasi);
        set(handles.text12,'string','MSE');
        set(handles.text13,'string',MSE);
        set(handles.text22,'string','Waktu');
        set(handles.text23,'string',strcat(num2str(waktu),'
detik'));
    case 3
        load A3_DataTraining.mat;
        load A3_DataFisher.mat;
        hidden=get(handles.TxtHidden,'string');
        alpha=get(handles.TxtAlpha,'string');
        galat=get(handles.TxtGalat,'string');
        epoch=get(handles.TxtEpoch,'string');
        if isempty(hidden) || isempty(alpha) || isempty(galat) ||
isempty(epoch)
            msgbox('Data parameter JST harus diisi
lengkap!','Pesan Sistem','warn');
            return
        end
        hidden=str2double(hidden);
        alpha=str2double(alpha);
        galat=str2double(galat);
        epoch=str2double(epoch);
        % vektor target
        t=[];
        for i=1:length(CA)
            t=[t CA{i,2}.'];
        end
        [y,v,v0,w,w0,iterasi,MSE,p,MSEi,waktu] =
BackPropagation(ProjectedImages_Fisher,t,hidden,alpha,galat,epoch)
;

        % Keterangan hasil
        axes(handles.axes2);
        plot(1:size(MSEi,2),MSEi);
        xlabel(strcat(num2str(iterasi),' epoh')), ylabel('MSE'),
title('Plot error'),legend('Train');

save('A3_DataBobotJST.mat','y','v','v0','w','w0','iterasi','MSE','
p')
        set(handles.text8,'string','Jumlah epoch:  ');
        set(handles.text10,'string',iterasi);
        set(handles.text12,'string','MSE');
        set(handles.text13,'string',MSE);
        set(handles.text22,'string','Waktu');
        set(handles.text23,'string',strcat(num2str(waktu),'
detik'));
    case 4
```

```
        load B2_DataTraining.mat;
        load B2_DataFisher.mat;
        hidden=get(handles.TxtHidden,'string');
        alpha=get(handles.TxtAlpha,'string');
        galat=get(handles.TxtGalat,'string');
        epoch=get(handles.TxtEpoch,'string');
        if isempty(hidden) || isempty(alpha) || isempty(galat) ||
isempty(epoch)
            msgbox('Data parameter JST harus diisi
lengkap!','Pesan Sistem','warn');
            return
        end
        hidden=str2double(hidden);
        alpha=str2double(alpha);
        galat=str2double(galat);
        epoch=str2double(epoch);
        % vektor target
        t=[];
        for i=1:length(CA)
            t=[t CA{i,2}.'];
        end
         [y,v,v0,w,w0,iterasi,MSE,p,MSEi,waktu] =
BackPropagation(ProjectedImages_Fisher,t,hidden,alpha,galat,epoch)
;

        % Keterangan hasil
        axes(handles.axes2);
        plot(1:size(MSEi,2),MSEi);
        xlabel(strcat(num2str(iterasi),' epoh')), ylabel('MSE'),
title('Plot error'),legend('Train');

save('B2_DataBobotJST.mat','y','v','v0','w','w0','iterasi','MSE','
p')
        set(handles.text8,'string','Jumlah epoch:  ');
        set(handles.text10,'string',iterasi);
        set(handles.text12,'string','MSE');
        set(handles.text13,'string',MSE);
        set(handles.text22,'string','Waktu');
        set(handles.text23,'string',strcat(num2str(waktu),'
detik'));
end
guidata(hObject,handles);

function TxtEpoch_Callback(hObject, eventdata, handles)
% hObject    handle to TxtEpoch (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 TxtEpoch as
text
%str2double(get(hObject,'String')) returns contents of TxtEpoch as
a double
```

```
% --- Executes during object creation, after setting all
properties.
function TxtEpoch_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtEpoch (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 TxtGalat_Callback(hObject, eventdata, handles)
% hObject    handle to TxtGalat (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 TxtGalat as
text
%str2double(get(hObject,'String')) returns contents of TxtGalat as
a double


% --- Executes during object creation, after setting all
properties.
function TxtGalat_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtGalat (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 TxtAlpha_Callback(hObject, eventdata, handles)
% hObject    handle to TxtAlpha (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 TxtAlpha as
text
%str2double(get(hObject,'String')) returns contents of TxtAlpha as
a double


% --- Executes during object creation, after setting all
properties.
function TxtAlpha_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtAlpha (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 TomHapusLatih.
function TomHapusLatih_Callback(hObject, eventdata, handles)
% hObject    handle to TomHapusLatih (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        delete ('A1_DataBobotJST.mat');
    case 2
        delete ('A2_DataBobotJST.mat');
    case 3
        delete ('A3_DataBobotJST.mat');
    case 4
        delete ('B2_DataBobotJST.mat');
end

function TxtHidden_Callback(hObject, eventdata, handles)
% hObject    handle to TxtHidden (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 TxtHidden as
text
%str2double(get(hObject,'String')) returns contents of TxtHidden
as a double


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

```
function TxtHidden_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtHidden (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 TomFisherWajah.
function TomFisherWajah_Callback(hObject, eventdata, handles)
% hObject    handle to TomFisherWajah (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        load A1_DataTraining.mat;
        T=[];
        for i=1:length(CA)
            T=[T CA{i,1}];
        end
        [m_data V_PCA V_LDA ProjectedImages_PCA
ProjectedImages_Fisher]=Fisherface(T);

save('A1_DataFisher.mat','m_data','V_PCA','V_LDA','ProjectedImages
_PCA','ProjectedImages_Fisher');
    case 2
        load A2_DataTraining.mat;
        T=[];
        for i=1:length(CA)
            T=[T CA{i,1}];
        end
        [m_data V_PCA V_LDA ProjectedImages_PCA
ProjectedImages_Fisher]=Fisherface(T);

save('A2_DataFisher.mat','m_data','V_PCA','V_LDA','ProjectedImages
_PCA','ProjectedImages_Fisher');
    case 3
        load A3_DataTraining.mat;
        T=[];
        for i=1:length(CA)
            T=[T CA{i,1}];
        end
        [m_data V_PCA V_LDA ProjectedImages_PCA
ProjectedImages_Fisher]=Fisherface(T);

save('A3_DataFisher.mat','m_data','V_PCA','V_LDA','ProjectedImages
_PCA','ProjectedImages_Fisher');
```

```
    case 4
        load B2_DataTraining.mat;
        T=[];
        for i=1:length(CA)
            T=[T CA{i,1}];
        end
        [m_data V_PCA V_LDA ProjectedImages_PCA
ProjectedImages_Fisher]=Fisherface(T);

save('B2_DataFisher.mat','m_data','V_PCA','V_LDA','ProjectedImages
_PCA','ProjectedImages_Fisher');
end




% --- Executes on button press in TomFisherFiturWAjah.


function TxtMomentum_Callback(hObject, eventdata, handles)
% hObject    handle to TxtMomentum (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 TxtMomentum as
text
%        str2double(get(hObject,'String')) returns contents of
TxtMomentum as a double


% --- Executes during object creation, after setting all
properties.
function TxtMomentum_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtMomentum (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 selection change in PopSkenario.
function PopSkenario_Callback(hObject, eventdata, handles)
% hObject    handle to PopSkenario (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 PopSkenario
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
PopSkenario


% --- Executes during object creation, after setting all
properties.
function PopSkenario_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PopSkenario (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



function text13_Callback(hObject, eventdata, handles)
% hObject    handle to text13 (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 text13 as text
%        str2double(get(hObject,'String')) returns contents of
text13 as a double



function text23_Callback(hObject, eventdata, handles)
% hObject    handle to text23 (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 text23 as text
%        str2double(get(hObject,'String')) returns contents of
text23 as a double
```

Kode program antarmuka halaman pengenalan untuk user umum

```
function varargout = A_FormRecognizing(varargin)
% A_FORMRECOGNIZING M-file for A_FormRecognizing.fig
```

```
%      A_FORMRECOGNIZING, by itself, creates a new
A_FORMRECOGNIZING or raises the existing
%      singleton*.
%
%      H = A_FORMRECOGNIZING returns the handle to a new
A_FORMRECOGNIZING or the handle to
%      the existing singleton*.
%
%      A_FORMRECOGNIZING('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in A_FORMRECOGNIZING.M with the
given input arguments.
%
%      A_FORMRECOGNIZING('Property','Value',...) creates a new
A_FORMRECOGNIZING or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before A_FormRecognizing_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to
A_FormRecognizing_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
A_FormRecognizing

% Last Modified by GUIDE v2.5 24-Nov-2010 21:33:54

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
   'gui_Singleton',  gui_Singleton, ...
   'gui_OpeningFcn', @A_FormRecognizing_OpeningFcn, ...
   'gui_OutputFcn',  @A_FormRecognizing_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
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before A_FormRecognizing is made visible.
function A_FormRecognizing_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 A_FormRecognizing (see
VARARGIN)

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

% Update handles structure
guidata(hObject, handles);
kondisi = 0;
% UIWAIT makes A_FormRecognizing wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = A_FormRecognizing_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 TomBuka.
function TomBuka_Callback(hObject, eventdata, handles)
% hObject    handle to TomBuka (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
kondisi = 0;
cla(handles.axes1,'reset');
cla(handles.axes2,'reset');
cla(handles.axes3,'reset');
cla(handles.axes4,'reset');
set(handles.uitable1,'data','');
set(handles.uitable2,'data','');
set(handles.TxtPath,'string','');
set(handles.TxtNama,'string','');
set(handles.TxtKanan,'string','');
set(handles.TxtKiri,'string','');
set(handles.TxtAtas,'string','');
set(handles.TxtBawah,'string','');
set(handles.TxtJijik,'string','');
```

```
set(handles.TxtTakut,'string','');
set(handles.TxtTerkejut,'string','');
set(handles.TxtMarah,'string','');
set(handles.TxtSedih,'string','');
set(handles.TxtSenang,'string','');
set(handles.TxtNetral,'string','');
set(handles.TxtHasil,'string','');
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
[nama,path]=uigetfile('*.bmp;*.jpg;*.pgm;*.tiff;*.gif','Buka
Gambar','D:\S2 - UGM\Thesis\~tesis\ProgramTesis3\JAFFE -
TestDataSet\');
    case 2
[nama,path]=uigetfile('*.bmp;*.jpg;*.pgm;*.tiff;*.gif','Buka
Gambar','D:\S2 - UGM\Thesis\~tesis\ProgramTesis3\JAFFE -
TestDataSet\');
    case 3
[nama,path]=uigetfile('*.bmp;*.jpg;*.pgm;*.tiff;*.gif','Buka
Gambar','D:\S2 - UGM\Thesis\~tesis\ProgramTesis3\JAFFE -
TestDataSet\');
    case 4
[nama,path]=uigetfile('*.bmp;*.jpg;*.pgm;*.tiff;*.gif','Buka
Gambar','D:\S2 - UGM\Thesis\~tesis\ProgramTesis3\MUG Dataset\');
end
handles.citra=imread(fullfile(path,nama));
pathfile=fullfile(path,nama);
axes(handles.axes1);
imshow(handles.citra);
set(handles.TxtPath,'string',pathfile);
set(handles.TxtNama,'string',nama);
guidata(hObject,handles);



function TxtPath_Callback(hObject, eventdata, handles)
% hObject    handle to TxtPath (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 TxtPath as text
%str2double(get(hObject,'String')) returns contents of TxtPath as
a double



% --- Executes during object creation, after setting all
properties.
function TxtPath_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtPath (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 TomKenal.
function TomKenal_Callback(hObject, eventdata, handles)
% hObject    handle to TomKenal (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
load mask.mat;
I = handles.citra;
[Img XProj YProj vmintab hmintab X1 X2 Y1 Y2] = deteksiwajah(I);
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        load A1_DataFisher.mat;
        load A1_DataBobotJST.mat;
        Imh = imresize(histeq(Img),[33
29]).*imresize(uint8(mask),[33 29]);
    case 2
        load A2_DataFisher.mat;
        load A2_DataBobotJST.mat;
        Imh = imresize(histeq(Img),[65
57]).*imresize(uint8(mask),[65 57]);
    case 3
        load A3_DataFisher.mat;
        load A3_DataBobotJST.mat;
        Imh = imresize(histeq(Img),[130
114]).*imresize(uint8(mask),[130 114]);
    case 4
        load B2_DataFisher.mat;
        load B2_DataBobotJST.mat;
        Imh = imresize(histeq(Img),[65
57]).*imresize(uint8(mask),[65 57]);
end

% Proses konstruksi fisherface wajah
[m,n] = size(Imh);
vektor=double(reshape(Imh',m*n,1));
Difference = vektor - m_data;
ProjectedTestImage = V_LDA' * V_PCA' * Difference;
y = FeedForward(ProjectedTestImage,p,v,v0,w,w0);

[baris,kolom]=find (y>=0.6);
[H,W]=size(y);
y2=zeros(H,W);
for i=1:length(y)
    y2(baris,kolom)=1;
end

set(handles.TxtJijik,'String',y2(1,1));
```

```
set(handles.TxtTakut,'String',y2(1,2));
set(handles.TxtTerkejut,'String',y2(1,3));
set(handles.TxtMarah,'String',y2(1,4));
set(handles.TxtSedih,'String',y2(1,5));
set(handles.TxtSenang,'String',y2(1,6));
set(handles.TxtNetral,'String',y2(1,7));
if y2 == [0 0 0 0 0 0 1]
    set(handles.TxtHasil,'String','Netral');
elseif y2 == [0 0 0 0 0 1 0]
    set(handles.TxtHasil,'String','Senang');
elseif y2 == [0 0 0 0 1 0 0]
    set(handles.TxtHasil,'String','Sedih');
elseif y2 == [0 0 0 1 0 0 0]
    set(handles.TxtHasil,'String','Marah');
elseif y2 == [0 0 1 0 0 0 0]
    set(handles.TxtHasil,'String','Terkejut');
elseif y2 == [0 1 0 0 0 0 0]
    set(handles.TxtHasil,'String','Takut');
elseif y2 == [1 0 0 0 0 0 0]
    set(handles.TxtHasil,'String','Jijik');
else
    set(handles.TxtHasil,'String','Tidak dikenali');
end


guidata(hObject,handles);

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


function TxtNama_Callback(hObject, eventdata, handles)
% hObject    handle to TxtNama (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 TxtNama as text
%str2double(get(hObject,'String')) returns contents of TxtNama as
a double


% --- Executes during object creation, after setting all
properties.
function TxtNama_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtNama (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 selection change in PopSkenario.
function PopSkenario_Callback(hObject, eventdata, handles)
% hObject    handle to PopSkenario (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 PopSkenario
contents as cell array
%contents{get(hObject,'Value')} returns selected item from
PopSkenario


% --- Executes during object creation, after setting all
properties.
function PopSkenario_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PopSkenario (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



function TxtKanan_Callback(hObject, eventdata, handles)
% hObject    handle to TxtKanan (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 TxtKanan as
text
%str2double(get(hObject,'String')) returns contents of TxtKanan as
a double
```

```
% --- Executes during object creation, after setting all
properties.
function TxtKanan_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtKanan (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 TxtKiri_Callback(hObject, eventdata, handles)
% hObject    handle to TxtKiri (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 TxtKiri as text
%str2double(get(hObject,'String')) returns contents of TxtKiri as
a double


% --- Executes during object creation, after setting all
properties.
function TxtKiri_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtKiri (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 TxtAtas_Callback(hObject, eventdata, handles)
% hObject    handle to TxtAtas (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 TxtAtas as text
%str2double(get(hObject,'String')) returns contents of TxtAtas as
a double


% --- Executes during object creation, after setting all
properties.
function TxtAtas_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtAtas (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 TxtBawah_Callback(hObject, eventdata, handles)
% hObject    handle to TxtBawah (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 TxtBawah as
text
%str2double(get(hObject,'String')) returns contents of TxtBawah as
a double


% --- Executes during object creation, after setting all
properties.
function TxtBawah_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtBawah (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 TomProses.
function TomProses_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to TomProses (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.TomKenal1,'visible','on');
set(handles.TomKenal,'visible','off');
load mask.mat;
I = handles.citra;
Y1 = str2num(get(handles.TxtKanan,'string'));
Y2 = str2num(get(handles.TxtKiri,'string'));
X1 = str2num(get(handles.TxtAtas,'string'));
X2 = str2num(get(handles.TxtBawah,'string'));
Img = deteksiwajahnew(I,X1,X2,Y1,Y2);
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        Imh=imresize(histeq(Img),[33 29]).*uint8(imresize(mask,[33
29]));
    case 2
        Imh=imresize(histeq(Img),[65 57]).*uint8(imresize(mask,[65
57]));
    case 3
        Imh=imresize(histeq(Img),[130
114]).*uint8(imresize(mask,[130 114]));
    case 4
        Imh=imresize(histeq(Img),[65 57]).*uint8(imresize(mask,[65
57]));
end
axes(handles.axes2);
imshow(Imh);
guidata(hObject,handles);

% --- Executes on button press in TomDeteksi.
function TomDeteksi_Callback(hObject, eventdata, handles)
% hObject    handle to TomDeteksi (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
guidata(hObject,handles);
load mask.mat;
I = handles.citra;
[Img XProj YProj vmintab hmintab X1 X2 Y1 Y2] = deteksiwajah(I);
axes(handles.axes3);
plot(YProj);xlim([0 length(YProj)]);
axes(handles.axes4);
plot(XProj); xlim([0 length(XProj)]);
set(handles.uitable1,'data',vmintab);
set(handles.uitable2,'data',hmintab);
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        Imh=imresize(histeq(Img),[33 29]).*uint8(imresize(mask,[33
29]));
    case 2
        Imh=imresize(histeq(Img),[65 57]).*uint8(imresize(mask,[65
57]));
```

```
    case 3
        Imh=imresize(histeq(Img),[130
114]).*uint8(imresize(mask,[130 114]));
    case 4
        Imh=imresize(histeq(Img),[65 57]).*uint8(imresize(mask,[65
57]));
end
axes(handles.axes2);
imshow(Imh);
set(handles.TxtKanan,'string',Y1);
set(handles.TxtKiri,'string',Y2);
set(handles.TxtAtas,'string',X1);
set(handles.TxtBawah,'string',X2);
handles.Imgg = handles.axes2;



% --- Executes on button press in TomTampilGrafik.
function TomTampilGrafik_Callback(hObject, eventdata, handles)
% hObject    handle to TomTampilGrafik (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
I = handles.citra;
[XProj,YProj,hmintab,vmintab]=integralprojection(I);
axes(handles.axes3);
plot(YProj);xlim([0 length(YProj)]);
axes(handles.axes4);
plot(XProj); xlim([0 length(XProj)]);



% --- Executes on button press in TomKenal1.
function TomKenal1_Callback(hObject, eventdata, handles)
% hObject    handle to TomKenal1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.TomKenal,'visible','on');
set(handles.TomKenal1,'visible','off');
load mask.mat;
Y1 = str2num(get(handles.TxtKanan,'string'));
Y2 = str2num(get(handles.TxtKiri,'string'));
X1 = str2num(get(handles.TxtAtas,'string'));
X2 = str2num(get(handles.TxtBawah,'string'));
I = handles.citra;
Img = deteksiwajahnew(I,X1,X2,Y1,Y2);
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        load A1_DataFisher.mat;
        load A1_DataBobotJST.mat;
        Imh = imresize(histeq(Img),[33
29]).*imresize(uint8(mask),[33 29]);
    case 2
        load A2_DataFisher.mat;
```

```
        load A2_DataBobotJST.mat;
        Imh = imresize(histeq(Img),[65
57]).*imresize(uint8(mask),[65 57]);
    case 3
        load A3_DataFisher.mat;
        load A3_DataBobotJST.mat;
        Imh = imresize(histeq(Img),[130
114]).*imresize(uint8(mask),[130 114]);
    case 4
        load B2_DataFisher.mat;
        load B2_DataBobotJST.mat;
        Imh = imresize(histeq(Img),[65
57]).*imresize(uint8(mask),[65 57]);
end

% Proses konstruksi fisherface wajah
[m,n] = size(Imh);
vektor=double(reshape(Imh',m*n,1));
Difference = vektor - m_data;
ProjectedTestImage = V_LDA' * V_PCA' * Difference;
y = FeedForward(ProjectedTestImage,p,v,v0,w,w0);

[baris,kolom]=find (y>=0.6);
[H,W]=size(y);
y2=zeros(H,W);
for i=1:length(y)
    y2(baris,kolom)=1;
end
set(handles.TxtJijik,'String',y2(1,1));
set(handles.TxtTakut,'String',y2(1,2));
set(handles.TxtTerkejut,'String',y2(1,3));
set(handles.TxtMarah,'String',y2(1,4));
set(handles.TxtSedih,'String',y2(1,5));
set(handles.TxtSenang,'String',y2(1,6));
set(handles.TxtNetral,'String',y2(1,7));
if y2 == [0 0 0 0 0 0 1]
    set(handles.TxtHasil,'String','Netral');
elseif y2 == [0 0 0 0 0 1 0]
    set(handles.TxtHasil,'String','Senang');
elseif y2 == [0 0 0 0 1 0 0]
    set(handles.TxtHasil,'String','Sedih');
elseif y2 == [0 0 0 1 0 0 0]
    set(handles.TxtHasil,'String','Marah');
elseif y2 == [0 0 1 0 0 0 0]
    set(handles.TxtHasil,'String','Terkejut');
elseif y2 == [0 1 0 0 0 0 0]
    set(handles.TxtHasil,'String','Takut');
elseif y2 == [1 0 0 0 0 0 0]
    set(handles.TxtHasil,'String','Jijik');
else
    set(handles.TxtHasil,'String','Tidak dikenali');
end
guidata(hObject,handles);
```

Kode program antarmuka halaman pengenalan untuk user administrator

```
function varargout = A_FormRecognizingAll(varargin)
% A_FORMRECOGNIZINGALL M-file for A_FormRecognizingAll.fig
%      A_FORMRECOGNIZINGALL, by itself, creates a new
A_FORMRECOGNIZINGALL or raises the existing
%      singleton*.
%
%      H = A_FORMRECOGNIZINGALL returns the handle to a new
A_FORMRECOGNIZINGALL or the handle to
%      the existing singleton*.
%
%
A_FORMRECOGNIZINGALL('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in A_FORMRECOGNIZINGALL.M with the
given input arguments.
%
%      A_FORMRECOGNIZINGALL('Property','Value',...) creates a new
A_FORMRECOGNIZINGALL or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before A_FormRecognizingAll_OpeningFcn
gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to
A_FormRecognizingAll_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
A_FormRecognizingAll

% Last Modified by GUIDE v2.5 28-Nov-2010 17:09:12

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
   'gui_Singleton',  gui_Singleton, ...
   'gui_OpeningFcn', @A_FormRecognizingAll_OpeningFcn, ...
   'gui_OutputFcn',  @A_FormRecognizingAll_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
% End initialization code - DO NOT EDIT


% --- Executes just before A_FormRecognizingAll is made visible.
function A_FormRecognizingAll_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 A_FormRecognizingAll (see
VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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


% --- Outputs from this function are returned to the command line.
function varargout = A_FormRecognizingAll_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 TomBuka.
function TomBuka_Callback(hObject, eventdata, handles)
% hObject    handle to TomBuka (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[nama,path]=uigetfile('*.mat','Buka Gambar','D:\S2 -
UGM\Thesis\~tesis\ProgramTesis3\');
pathfile=fullfile(path,nama);
set(handles.TxtPath,'string',pathfile);
set(handles.TxtNama,'string',nama);
guidata(hObject,handles);
```

```
function TxtPath_Callback(hObject, eventdata, handles)
% hObject    handle to TxtPath (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 TxtPath as text
%str2double(get(hObject,'String')) returns contents of TxtPath as
a double


% --- Executes during object creation, after setting all
properties.
function TxtPath_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtPath (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 TomKenal.
function TomKenal_Callback(hObject, eventdata, handles)
% hObject    handle to TomKenal (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        load A1_DataTestAll.mat;
        load A1_DataFisher.mat;
        load A1_DataBobotJST.mat;
        load A1_DataRecognizingAll.mat;
        %% Pengenalan semua

        % Proses konstruksi fisherface wajah
        JmlData = size(CA,1);
        for i=1:1:JmlData
            NamaFile = CA{i,3};
            Label = CA{i,2};
            Difference = CA{i,1} - m_data;
            ProjectedTestImage = V_LDA' * V_PCA' * Difference;
            y = FeedForward(ProjectedTestImage,p,v,v0,w,w0);
            [baris,kolom]=find (y>=0.6);
            [H,W]=size(y);
            y2=zeros(H,W);
            for k=1:length(y)
```

```
                y2(baris,kolom)=1;
            end
            Output=y2;
            if isequal(Label, Output)
                Ketr = 1;
            else
                Ketr = 0;
            end
            % simpan data per citra
            savefile='A1_DataRecognizingAll.mat'; %nama file
database yang akan dibuat
            [m,n]=size(Hasil);
            if isempty(Hasil{1,1}) && isempty(Hasil{1,2}) &&
isempty(Hasil{1,3}) && isempty(Hasil{1,4})
                Hasil(1,:)={NamaFile,Label,Output,Ketr};
%menambahkan record baru
            else
                Hasil(m+1,:)={NamaFile,Label,Output,Ketr};
%menambahkan record baru
            end
            save(savefile,'Hasil'); %menyimpan record
        end
    case 2
        load A2_DataTestAll.mat;
        load A2_DataFisher.mat;
        load A2_DataBobotJST.mat;
        load A2_DataRecognizingAll.mat;
        %% Pengenalan semua

        % Proses konstruksi fisherface wajah
        JmlData = size(CA,1);
        for i=1:1:JmlData
            NamaFile = CA{i,3};
            Label = CA{i,2};
            Difference = CA{i,1} - m_data;
            ProjectedTestImage = V_LDA' * V_PCA' * Difference;
            y = FeedForward(ProjectedTestImage,p,v,v0,w,w0);
            [baris,kolom]=find (y>=0.6);
            [H,W]=size(y);
            y2=zeros(H,W);
            for k=1:length(y)
                y2(baris,kolom)=1;
            end
            Output=y2;
            if isequal(Label, Output)
                Ketr = 1;
            else
                Ketr = 0;
            end
            % simpan data per citra
            savefile='A2_DataRecognizingAll.mat'; %nama file
database yang akan dibuat
            [m,n]=size(Hasil);
            if isempty(Hasil{1,1}) && isempty(Hasil{1,2}) &&
isempty(Hasil{1,3}) && isempty(Hasil{1,4})
```

```
                Hasil(1,:)={NamaFile,Label,Output,Ketr};
%menambahkan record baru
            else
                Hasil(m+1,:)={NamaFile,Label,Output,Ketr};
%menambahkan record baru
            end
            save(savefile,'Hasil'); %menyimpan record
        end
    case 3
        load A3_DataTestAll.mat;
        load A3_DataFisher.mat;
        load A3_DataBobotJST.mat;
        load A3_DataRecognizingAll.mat;
        %% Pengenalan semua

        % Proses konstruksi fisherface wajah
        JmlData = size(CA,1);
        for i=1:1:JmlData
            NamaFile = CA{i,3};
            Label = CA{i,2};
            Difference = CA{i,1} - m_data;
            ProjectedTestImage = V_LDA' * V_PCA' * Difference;
            y = FeedForward(ProjectedTestImage,p,v,v0,w,w0);
            [baris,kolom]=find (y>=0.6);
            [H,W]=size(y);
            y2=zeros(H,W);
            for k=1:length(y)
                y2(baris,kolom)=1;
            end
            Output=y2;
            if isequal(Label, Output)
                Ketr = 1;
            else
                Ketr = 0;
            end
            % simpan data per citra
            savefile='A3_DataRecognizingAll.mat'; %nama file
database yang akan dibuat
            [m,n]=size(Hasil);
            if isempty(Hasil{1,1}) && isempty(Hasil{1,2}) &&
isempty(Hasil{1,3}) && isempty(Hasil{1,4})
                Hasil(1,:)={NamaFile,Label,Output,Ketr};
%menambahkan record baru
            else
                Hasil(m+1,:)={NamaFile,Label,Output,Ketr};
%menambahkan record baru
            end
            save(savefile,'Hasil'); %menyimpan record
        end
        case 4
        load B2_DataTestAll.mat;
        load B2_DataFisher.mat;
        load B2_DataBobotJST.mat;
        load B2_DataRecognizingAll.mat;
        %% Pengenalan semua
```

```
        % Proses konstruksi fisherface wajah
        JmlData = size(CA,1);
        for i=1:1:JmlData
            NamaFile = CA{i,3};
            Label = CA{i,2};
            Difference = CA{i,1} - m_data;
            ProjectedTestImage = V_LDA' * V_PCA' * Difference;
            y = FeedForward(ProjectedTestImage,p,v,v0,w,w0);
            [baris,kolom]=find (y>=0.6);
            [H,W]=size(y);
            y2=zeros(H,W);
            for k=1:length(y)
                y2(baris,kolom)=1;
            end
            Output=y2;
            if isequal(Label, Output)
                Ketr = 1;
            else
                Ketr = 0;
            end
            % simpan data per citra
            savefile='B2_DataRecognizingAll.mat'; %nama file
database yang akan dibuat
            [m,n]=size(Hasil);
            if isempty(Hasil{1,1}) && isempty(Hasil{1,2}) &&
isempty(Hasil{1,3}) && isempty(Hasil{1,4})
                Hasil(1,:)={NamaFile,Label,Output,Ketr};
%menambahkan record baru
            else
                Hasil(m+1,:)={NamaFile,Label,Output,Ketr};
%menambahkan record baru
            end
            save(savefile,'Hasil'); %menyimpan record
        end
end
guidata(hObject,handles);

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



function TxtNama_Callback(hObject, eventdata, handles)
% hObject    handle to TxtNama (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 TxtNama as text
%str2double(get(hObject,'String')) returns contents of TxtNama as
a double
```

```
% --- Executes during object creation, after setting all
properties.
function TxtNama_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TxtNama (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 TomView.
function TomView_Callback(hObject, eventdata, handles)
% hObject    handle to TomView (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        load A1_DataRecognizingAll.mat;
        HasilAkhir=cell2mat(Hasil(:,4));
        HasilTrain=HasilAkhir(1:140,:);
        HasilTest=HasilAkhir(141:213,:);

        indeks = find(HasilTrain==1);
        JumSuksesTraining = length(indeks);
        RecRateTraining = JumSuksesTraining/140*100;

        indeks1 = find(HasilTest==1);
        JumSuksesTesting = length(indeks1);
        RecRateTesting = JumSuksesTesting/73*100;

        JumSuksesTotal = JumSuksesTraining + JumSuksesTesting;
        RecRateTotal = JumSuksesTotal/213*100;
        JumGagal = 213 - JumSuksesTotal;
        set(handles.TxtJumTrain,'string',JumSuksesTraining);
        set(handles.TxtJumTest,'string',JumSuksesTesting);
        set(handles.TxtRecTrain,'string',RecRateTraining);
        set(handles.TxtRecTest,'string',RecRateTesting);
        set(handles.TxtRecTot,'string',RecRateTotal);
        set(handles.Txt####,'string',JumGagal);
    case 2
        load A2_DataRecognizingAll.mat;
        HasilAkhir=cell2mat(Hasil(:,4));
        HasilTrain=HasilAkhir(1:140,:);
        HasilTest=HasilAkhir(141:213,:);
```

```
    indeks = find(HasilTrain==1);
    JumSuksesTraining = length(indeks);
    RecRateTraining = JumSuksesTraining/140*100;

    indeks1 = find(HasilTest==1);
    JumSuksesTesting = length(indeks1);
    RecRateTesting = JumSuksesTesting/73*100;

    JumSuksesTotal = JumSuksesTraining + JumSuksesTesting;
    RecRateTotal = JumSuksesTotal/213*100;
    JumGagal = 213 - JumSuksesTotal;
    set(handles.TxtJumTrain,'string',JumSuksesTraining);
    set(handles.TxtJumTest,'string',JumSuksesTesting);
    set(handles.TxtRecTrain,'string',RecRateTraining);
    set(handles.TxtRecTest,'string',RecRateTesting);
    set(handles.TxtRecTot,'string',RecRateTotal);
    set(handles.Txt####,'string',JumGagal);
case 3
    load A3_DataRecognizingAll.mat;
    HasilAkhir=cell2mat(Hasil(:,4));
    HasilTrain=HasilAkhir(1:140,:);
    HasilTest=HasilAkhir(141:213,:);

    indeks = find(HasilTrain==1);
    JumSuksesTraining = length(indeks);
    RecRateTraining = JumSuksesTraining/140*100;

    indeks1 = find(HasilTest==1);
    JumSuksesTesting = length(indeks1);
    RecRateTesting = JumSuksesTesting/73*100;

    JumSuksesTotal = JumSuksesTraining + JumSuksesTesting;
    RecRateTotal = JumSuksesTotal/213*100;
    JumGagal = 213 - JumSuksesTotal;
    set(handles.TxtJumTrain,'string',JumSuksesTraining);
    set(handles.TxtJumTest,'string',JumSuksesTesting);
    set(handles.TxtRecTrain,'string',RecRateTraining);
    set(handles.TxtRecTest,'string',RecRateTesting);
    set(handles.TxtRecTot,'string',RecRateTotal);
    set(handles.Txt####,'string',JumGagal);
case 4
    load B2_DataRecognizingAll.mat;
    HasilAkhir=cell2mat(Hasil(:,4));
    HasilTrain=HasilAkhir(1:504,:);
    HasilTest=HasilAkhir(505:630,:);

    indeks = find(HasilTrain==1);
    JumSuksesTraining = length(indeks);
    RecRateTraining = JumSuksesTraining/504*100;

    indeks1 = find(HasilTest==1);
    JumSuksesTesting = length(indeks1);
    RecRateTesting = JumSuksesTesting/126*100;
```

```
        JumSuksesTotal = JumSuksesTraining + JumSuksesTesting;
        RecRateTotal = JumSuksesTotal/630*100;
        JumGagal = 630 - JumSuksesTotal;
        set(handles.TxtJumTrain,'string',JumSuksesTraining);
        set(handles.TxtJumTest,'string',JumSuksesTesting);
        set(handles.TxtRecTrain,'string',RecRateTraining);
        set(handles.TxtRecTest,'string',RecRateTesting);
        set(handles.TxtRecTot,'string',RecRateTotal);
        set(handles.Txt####,'string',JumGagal);
end
guidata(hObject,handles);


% --- Executes on button press in TomMatrix.
function TomMatrix_Callback(hObject, eventdata, handles)
% hObject    handle to TomMatrix (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
load DataRecognizingAll.mat;
% Netral
cellq=Hasil(:,2:4);


% --- Executes on selection change in PopSkenario.
function PopSkenario_Callback(hObject, eventdata, handles)
% hObject    handle to PopSkenario (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 PopSkenario
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
PopSkenario


% --- Executes during object creation, after setting all
properties.
function PopSkenario_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PopSkenario (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 on button press in TomBuatFile.
function TomBuatFile_Callback(hObject, eventdata, handles)
% hObject    handle to TomBuatFile (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.TxtJumTrain,'string','');
set(handles.TxtJumTest,'string','');
set(handles.TxtRecTrain,'string','');
set(handles.TxtRecTest,'string','');
set(handles.TxtRecTot,'string','');
set(handles.Txt####,'string','');
skenario=get(handles.PopSkenario,'Value');
switch skenario
    case 1
        savefile='A1_DataRecognizingAll.mat';
        NamaFile=[];
        Label=[];
        Output=[];
        Ketr=[];
        Hasil={NamaFile,Label,Output,Ketr};
        save(savefile,'Hasil');
    case 2
        savefile='A2_DataRecognizingAll.mat';
        NamaFile=[];
        Label=[];
        Output=[];
        Ketr=[];
        Hasil={NamaFile,Label,Output,Ketr};
        save(savefile,'Hasil');
    case 3
        savefile='A3_DataRecognizingAll.mat';
        NamaFile=[];
        Label=[];
        Output=[];
        Ketr=[];
        Hasil={NamaFile,Label,Output,Ketr};
        save(savefile,'Hasil');
   case 4
        savefile='B2_DataRecognizingAll.mat';
        NamaFile=[];
        Label=[];
        Output=[];
        Ketr=[];
        Hasil={NamaFile,Label,Output,Ketr};
        save(savefile,'Hasil');
end
```