

Program Simulasi *Eigenface*

```
% Pengenalan wajah menggunakan metode Eigenface

clear all
close all
clc
%Banyaknya jumlah gambar yang akan di training set.
M=4;

%Pemilihan standar deviasi dan rata-rata.
um=100;
ustd=80;

%Membaca dan menampilkan gambar.
S=[];
figure(1);
for i=1:M
str=strcat(int2str(i),'.bmp');
eval('gbr=imread(str);');
gbr=gbr(:,:,1);
subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
imshow(gbr)
if i==4
title('Training set','fontsize',18)
end
drawnow;
[irow icol]=size(gbr);
temp=reshape(gbr',irow*icol,1);
S=[S temp];
end

%Mengganti mean dan standar deviasi dari semua gambar
for i=1:size(S,2)
temp=double(S(:,i));
m=mean(temp);
st=std(temp);
S(:,i)=(temp-m)*ustd/st+um;
end

%Menampilkan gambar yang sudah dinormalisasi
figure(2);
for i=1:M
str=strcat(int2str(i),'.jpg');
gbr=reshape(S(:,i),icol,irow);
gbr=gbr';
eval('imwrite(gbr,str)');
subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
imshow(gbr)
drawnow;
if i==4
title('Training Set setelah di Normalisasi','fontsize',18)
end
end
```

```
end
```

```
%Gambar rata2  
m=mean(S,2);  
tmgbr=uint8(m);  
gbr=reshape(tmgbr,icol,irow);  
gbr=gbr';  
figure(3);  
imshow(gbr);  
title('Average Face','fontsize',18)
```

```
dbx=[];  
for i=1:M  
temp=double(S(:,i));  
dbx=[dbx temp];  
end
```

```
%Matriks kovarians C=A'A, L=AA'  
A=dbx';  
L=A*A';  
[vv dd]=eig(L);  
v=[];  
d=[];  
for i=1:size(vv,2)  
if(dd(i,i)>1e-4)  
v=[v vv(:,i)];  
d=[d dd(i,i)];  
end  
end
```

```
[B index]=sort(d);  
ind=zeros(size(index));  
dtemp=zeros(size(index));  
vtemp=zeros(size(v));  
pjpg=length(index);  
for i=1:pjpg  
dtemp(i)=B(pjpg+1-i);  
ind(i)=pjpg+1-index(i);  
vtemp(:,ind(i))=v(:,i);  
end  
d=dtemp;  
v=vtemp;
```

```
%Menormalisasikan vektor2 eigen  
for i=1:size(v,2)  
kk=v(:,i);  
temp=sqrt(sum(kk.^2));  
v(:,i)=v(:,i)./temp;  
end
```

```
%Vektor2 eigen dari matriks C  
u=[];  
for i=1:size(v,2)  
temp=sqrt(d(i));
```

```

u=[u (dbx*v(:,i))./temp];
end

for i=1:size(u,2)
kk=u(:,i);
temp=sqrt(sum(kk.^2));
u(:,i)=u(:,i)./temp;
end

%Menampilkan gambar2 eigenface
figure(4);
for i=1:size(u,2)
gbr=reshape(u(:,i),icol,irow);
gbr=gbr';
gbr=histeq(gbr,255);
subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
imshow(gbr)
drawnow;
if i==4
title('Eigenfaces','fontsize',18)
end
end

%Mencari berat dari masing2 wajah yang ada di training set.
omega = [];
for h=1:size(dbx,2)
WW=[];
for i=1:size(u,2)
t = u(:,i)';
Bobotgbr = dot(t,dbx(:,h)');
WW = [WW; Bobotgbr];
end
omega = [omega WW];
end

%Mengambil input gambar baru
Gbrinput = input('Tolong masukan input yang akan di coba dan
format filenya\n','s');
Gbrinput =
imread(strcat('D:\Franky\T.A\Software\Percobaan\',Gbrinput));
Gbrinput =Gbrinput(:,:,1);
figure(5)
subplot(1,2,1)
imagesc(Gbrinput); colormap('gray');title('Gambar
Input','fontsize',18)
Gbrmsk=reshape(double(Gbrinput)',irow*icol,1);
temp=Gbrmsk;
me=mean(temp);
st=std(temp);
temp=(temp-me)*ustd/st+um;
Gbrnorm = temp;
Selisih = temp-m;

```

```

p = [];
aa=size(u,2);
for i = 1:aa
ptg = dot(Gbrnorm,u(:,i));
p = [p; ptg];
end
GbrRekonstuksi = m + u(:,1:aa)*p;
GbrRekonstuksi = reshape(GbrRekonstuksi,icol,irow);
GbrRekonstuksi = GbrRekonstuksi';

%Menampilkan gambar rekonstruksi
subplot(1,2,2)
imagesc(GbrRekonstuksi); colormap('gray');
title('Gambar Rekonstruksi','fontsize',18)

BobotBaru = [];
for i=1:size(u,2)
t = u(:,i)';
BobotGbrInput = dot(t,Selisih');
BobotBaru = [BobotBaru; BobotGbrInput];
end

l1 = 1:M;
figure(6)
subplot(1,2,1)
stem(l1,BobotBaru)
title('Bobot Gambar Input','fontsize',14)

%Mencari jarak euklidis
e=[];
for i=1:size(omega,2)
q = omega(:,i);
SelisihBobot = BobotBaru-q;
mag = norm(SelisihBobot);
e = [e mag];
end

kk = 1:size(e,2);
subplot(1,2,2)
stem(kk,e)
title('Jarak Euklidis Gambar input','fontsize',14)

Jarak_euklidis_maksimum =max(e)
Jarak_euklidis_minimum =min(e)

```