

LAMPIRAN A

```

clear;
close all;
clc;
%-----%
% Membaca file dari database          %
%-----%

jum_kelas=1;
daya_tot_log=zeros(2,4);
jum_rasio_silent=zeros(2,4);
while jum_kelas<=3

    %-----%
    [file_list,file_path] = uigetfile('*.*','Silahkan Pilih File Database');
    fprintf(1,'\nFilelist yang dipilih : %s',file_list);
    %-----%

    %-----%
    % Menghitung jumlah baris yang terdapat pada filelist
    %-----%

    fid = fopen(file_list,'r');
    if fid < 0
        bg = 0;
    else
        bg = 0;
        while 1
            br = fgetl(fid);
            if ~isstr(br)
                break;
            end;
            bg = bg + 1;
        end;
        fclose(fid);
    end;
end;

```

```

end;

%-----%
% Proses loading sinyal suara ke dalam matriks speech
%-----%

fid = fopen(file_list,'r');

if fid < 0 | bg < 1
    error(['List Sinyal Suara Tidak Ditemukan "' file_list, '"']);
end;

for i = 1:bg
    nama_suara = fgetl(fid);
    if ~isstr(nama_suara)
        break;
    end;

    fprintf(1,'\nProses Loading Sinyal Suara : %s',nama_suara);
    disp(' ');
    [suara {i},fs,nbit] = wavread(nama_suara);

    x=0.96*suara {i};

    jpg_frame=round(length(x)/256);

%-----%
% Proses membagi sinyal suara menjadi beberapa frame
%-----%

%-----%
% Klasifikasi frame berdasarkan silent / non-silent

```

```

%-----%

syms u v w y z;

frame_j=zeros(pjg_frame*256,1);

silent=0;
non_silent=0;

w_ham=hamming(256);

frame{1}=x(1:256);

fft_frame{1}=fft(frame{1});
fft_kuadrat_frame{1}=(abs(fft_frame{1}))^2;
fft_kuadrat_frame{1}=sum(fft_kuadrat_frame{1});

temp_daya_total_silent=0;
temp_daya_total_non_silent=0;

sinyal_terwindow{1}=sum((w_ham.*frame{1}).^2);

if sinyal_terwindow{1}<1
    silent=silent+1;

%-----%
% Menghitung daya total frame ke-1 jika merupakan frame silent
%-----%

daya_total_silent=fft_kuadrat_frame{1}*u;
hasil_silent=int(daya_total_silent,u,0,4000);
temp_daya_total_silent=temp_daya_total_silent+hasil_silent;

```

```

temp_daya_total_silent(1)=double(temp_daya_total_silent);

else
    non_silent=non_silent+1;

    %-----%
    % Menghitung daya total frame ke-1 jika merupakan frame non-silent
    %-----%

    daya_total_non_silent=fft_kuadrat_frame{1}*u;
    hasil_non_silent=int(daya_total_non_silent,u,0,4000);

temp_daya_total_non_silent=temp_daya_total_non_silent+hasil_non_silent;
    temp_daya_total_non_silent{1}=double(temp_daya_total_non_silent);
end;

frame_j(1:256)=frame{1};

for j=2:pjg_frame
    frame{j}=x(192*(j-1)+1:192*(j-1)+256);
    frame_j(256*(j-1)+1:256*j)=frame{j};
    sinyal_terwindow{j}=sum((w_ham.*frame{j}).^2);

    fft_frame{j}=fft(frame{j});
    fft_kuadrat_frame{j}=(abs(fft_frame{j})).^2;
    fft_kuadrat_frame{j}=sum(fft_kuadrat_frame{j});

    if sinyal_terwindow{j}<1 % Ini harus dicek nilainya (tergantung
percobaan)
        silent=silent+1;

        %-----%

```

```

% Menghitung daya total frame ke-2 sd terakhir jika merupakan frame
silent
%-----%

daya_total_silent=fft_kuadrat_frame{j}*u;
hasil_silent=int(daya_total_silent,u,0,4000);
temp_daya_total_silent=temp_daya_total_silent+hasil_silent;
temp_daya_total_silent=double(temp_daya_total_silent);

else
non_silent=non_silent+1;

%-----%
% Menghitung daya total frame ke-2 sd terakhir jika merupakan frame
non-silent
%-----%

daya_total_non_silent=fft_kuadrat_frame{j}*u;
hasil_non_silent=int(daya_total_non_silent,u,0,4000);

temp_daya_total_non_silent=temp_daya_total_non_silent+hasil_non_silent;
temp_daya_total_non_silent=double(temp_daya_total_non_silent);

end;
end;

silent_ratio{i}=silent/pjg_frame*100;
jum_rasio_silent(jum_kelas,i)=silent_ratio{i};
disp(['Nilai silent ratio = ',num2str(silent_ratio{i}),'%']);

%-----%
% Klasifikasi frame berdasarkan pitch / non-pitch

```

```

%-----%

pitch=0;
non_pitch=0;

autokorelasi_frame{1}=xcorr(frame{1},'coeff');

if autokorelasi_frame{1}>0.65
    pitch=pitch+1;
else
    non_pitch=non_pitch+1;
end;

for j=2:pjg_frame
    autokorelasi_frame{j}=xcorr(frame{j});
    if autokorelasi_frame{j}<0.15
        pitch=pitch+1;
    else
        non_pitch=non_pitch+1;
    end;
end;

pitch_ratio{i}=pitch/pjg_frame*100;
jum_rasio_pitch(jum_kelas,i)=pitch_ratio{i};
disp(['Nilai pitch ratio = ',num2str(pitch_ratio{i}),'%']);

%-----%
% Menghitung daya sub-band
%-----%

temp_sub_band_satu=0;

daya_sub_band_satu=fft_x_kuadrat*v;

```

```
hasil_sub_band_satu=int(daya_sub_band_satu,v,0,500);
temp_sub_band_satu=temp_sub_band_satu+hasil_sub_band_satu;
out_sub_band_satu(i)=double(temp_sub_band_satu);
daya_log_sub_band_satu(i)=log10(out_sub_band_satu(i));
```

```
temp_sub_band_dua=0;
```

```
daya_sub_band_dua=fft_x_kuadrat*w;
hasil_sub_band_dua=int(daya_sub_band_dua,w,500,1000);
temp_sub_band_dua=temp_sub_band_dua+hasil_sub_band_dua;
out_sub_band_dua(i)=double(temp_sub_band_dua);
daya_log_sub_band_dua(i)=log10(out_sub_band_dua(i));
```

```
temp_sub_band_tiga=0;
```

```
daya_sub_band_tiga=fft_x_kuadrat*y;
hasil_sub_band_tiga=int(daya_sub_band_tiga,y,1000,2000);
temp_sub_band_tiga=temp_sub_band_tiga+hasil_sub_band_tiga;
out_sub_band_tiga(i)=double(temp_sub_band_tiga);
daya_log_sub_band_tiga(i)=log10(out_sub_band_tiga(i));
```

```
temp_sub_band_empat=0;
```

```
daya_sub_band_empat=fft_x_kuadrat*z;
hasil_sub_band_empat=int(daya_sub_band_empat,z,2000,4000);
temp_sub_band_empat=temp_sub_band_empat+hasil_sub_band_empat;
out_sub_band_empat(i)=double(temp_sub_band_empat);
daya_log_sub_band_empat(i)=log10(out_sub_band_empat(i));
```

```
end;
```

```
daya_tot_log(jum_kelas,:)=daya_total_log;
```



```

    jum_kelas=jum_kelas+1;

end;

    %
=====
=====

    %
    % Menghitung brightness
    %
=====
=====

    fft_brg=fft(x);
    fft_brg_kuadrat=(abs(fft_brg)).^2;
    temp=0;
    for j=1:length(fft_brg_kuadrat)
    S1(j)=2*pi*fs*fft_brg_kuadrat(j)*y;
    hasil1(j)=int(S1(j),y,0,5500);
    temp1=temp1+hasil(j);
    fft_brg2=fft(x);
    fft_brg2_kuadrat=(abs(fft_brg)).^2;
    fft_brg2=fft(x);
    fft_brg_kuadrat=(abs(fft_brg)).^2;
    temp2=0;
    temp2=temp2+hasil2(j);
    brg = HASIL1(J)/HASIL2(J);
    end;

    %
=====
=====

    %
    % Menghitung bandwidth

```

```
%
```

```
=====
=====
for j=1 : length(fft(x))
bdw = 0 ;
bdw_1 = fft(x);
bdw1 = ((j*2*fs*pi-brg)* (bdw + (abs(bdw_1).^2))^0.5
for j=1:length(fft_x_kuadrat)
Sx(j)=fft_x_kuadrat(j)*y;
hasilx(j)=int(Sx(j),y,0,5500);
bdw2=bdw2+hasilx(j);
bdw = (bdw1/bdw2).^0.5
```

```
%=====
% addnfl
%=====

train.Path = train_dir;
di = dir(fullfile(train.Path, '*.wav'));
if isempty(di)
return
end
Info = imfinfo(fullfile(train.Path, di(1).name));
M = length(di);
train.suara = uint8(zeros(Nx, Ny, M));
for i = 1:M
[qx] = fft(waveread(fullfile(train.Path, di(i).name), 'wav'));
atrain.suara(:, :, i) = q2;
train>NamaFile {i} = di(i).name(1:end-4);
for i=5 :120
n = n +1
k = k + 1
Cn= cn + 2/k*log10 k*cos(i(k-0.5)*pi/k);
```

```
end
m = (y2-y1/x2-x1)
c = y2-(m*x2)
q2 = m*qx + c
jarak = abs((q2-qx).^2)
```

.....

```
for a=5 : 120
    jl = jl + match(jarak)
```

```
disp(' ');
fclose(fid);
```