

LAMPIRAN A
LISTING PROGRAM

```

% -----
% Program Tugas Akhir
% Peningkatan Kualitas Sinyal Suara
% Menggunakan Metode Estimasi Magnituda Spektral
% -----

clear;
close all;
clc;

format compact;more off;

% Baca sinyal input

[sig f_sam] = wavread('t0.wav'); %Signal file
sig=sig(:,1);
asli=sig;
disp('Menambahkan noise');
disp(' ');
pengali=input('Masukkan faktor pengali (coba mulai 0.0185) = ');
noise=pengali*randn(length(sig),1);
sig=sig+noise;
disp(' ');

fs = 16000; % Sampling rate
window = 10; % Window 10 milidetik
window_samp = window * (fs/1000); % Jumlah sampel per window
start_overlap = window_samp/2; % Titik awal untuk half-overlapped window

pjpg_sig = length(sig); % Panjang sinyal
st_locs = 1: start_overlap: pjpg_sig; % Awal lokasi
end_locs = st_locs + window_samp; % Akhir lokasi
end_locs = min(pjpg_sig , end_locs);
fft_points = 512; % Jumlah titik FFT
first_50ms = 5 ; % 50 milidetik pertama

```

```

fft_vec = [];
speech = [];
speech_arnr = [];
spch_detect=[];
cln_speech = [];
sig_vec = [];
pro_speech = [];
attn_speech=[];
sig_hann = [];
recon_sig = [];
phase_vec = [];
spec_sub=[];

% Memotong data dari sinyal speech dengan window dengan half-overlapped data
window 10 milidetik
loc_diff=end_locs(1)-st_locs(1);
std_hanning=hanning(window_samp+1);

for nfr = 1:length(st_locs)
    if (end_locs(nfr) - st_locs(nfr) < loc_diff)
        sig_vec(1:end_locs(nfr) - st_locs(nfr)+1,nfr) = ...
            sig(st_locs(nfr):end_locs(nfr));
        sig_hann(1:end_locs(nfr) - st_locs(nfr)+1,nfr) = ...
            sig_vec(1:end_locs(nfr) - st_locs(nfr)+1,nfr).*hanning(end_locs(nfr) ...
            - st_locs(nfr)+1);
    else
        sig_vec(:,nfr) = sig(st_locs(nfr):end_locs(nfr));
        sig_hann(:,nfr) = sig_vec(:,nfr).*std_hanning;
    end;
end;

% Proses perhitungan FFT dari masing-masing data window / frame
for nfr = 1:length(st_locs)
    fft_vec(nfr,:) = fft(sig_hann(:,nfr)',fft_points );

```

```

temp_fft=fft_vec(nfr,:);
temp_phase=phase(temp_fft);
phase_vec(nfr,:) = temp_phase;
end;

% Menggambar sinyal input yang telah di-window menjadi beberapa frame
figure(1);
for frames = 1:first_50ms
    subplot(5,1,frames);
    stem(abs(fft_vec(1:fft_points/2,frames)));
    title(['Frame ke-',int2str(frames)]);
end;

% Menghitung Magnitude

% Menjumlahkan sinyal suara 50 milidetik pertama (belum ada ucapan)
sum_of_first50ms = sum(abs(fft_vec(1:first_50ms,1:floor(fft_points/2)+1)),1);
mag_avg = (1/first_50ms)*(sum_of_first50ms); % Magnitude spektrum noise rata-rata
(ada bias)

% Rata-rata tiga half-overlapped window dan menghilangkan bias
for noise = first_50ms:length(st_locs)
    st_frm = max(1,noise - 1);
    end_frm = min(length(st_locs), noise + 1);
    temp_spec_sub = max(0,sum(abs(fft_vec(st_frm:end_frm,1:floor(fft_points/2)+1)),1)/3
- mag_avg );
    spec_sub(noise,1:floor(fft_points/2)+1)=temp_spec_sub; % Semula bukan comment

    % Baris di bawah semula adalah comment saja ....
%
spec_sub(noise,floor(fft_points/2)+2:fft_points)=temp_spec_sub(floor(fft_points/2):-1:2);

speech(noise,:) = spec_sub(noise,:); % .*exp(j*phase_vec(noise,1:fft_points));

```

```

    % Semula baris di bawah tidak dikali dengan %.*exp(j*phase_vec(noise,1:fft_points))
    % speech(noise,:) = spec_sub(noise,:).*exp(j*phase_vec(noise,1:fft_points));
end;

```

```

% Pengurangan Noise Residual (PNR)

```

```

for noise = 1:length(st_locs)
    if speech(noise,:) >= mag_avg
        % Jika frame yang dianalisis lebi besar atau sama dengan mag_avg,
        % maka sinyal speech sesudah PNR sama dengan sinyal ucapan
        speech_arnr(noise,:) = speech(noise,:);
    else
        st_frm = max(1,noise - 1); % Ini jika noise residual ditekan
        end_frm = min(length(st_locs), noise + 1);
        speech_arnr(noise,:) = min(speech(st_frm:end_frm,:),[],1);
    end
    % Frame sekarang digantikan oleh nilai minimum yang dipilih dari analisis frames
    tetangga
end;

```

```

% Atenuasi sinyal tambahan jika terjadi Nonspeech Activity Speech terdeteksi
for frames = 1:length(st_locs) % Loop ini untuk menghitung threshold untuk setiap frame
    % Baris di bawah semula tanpa eps
    spch_detect(frames) = 20*log10(sum(speech_arnr(frames,.)+eps./mag_avg)/(2*pi));
    % spch_detect(frames) = 20*log10(sum(speech_arnr(frames,)./mag_avg)/(2*pi));
end;

```

```

% Atenuasi sinyal sepanjang Non-Speech Activity

```

```

for act = 1:length(st_locs)
    % Jika threshold kurang dari -12dB, maka frame diklasifikasi sebagai no speech
    activity.
    if spch_detect(act) >= -12
        attn_speech(act,1:floor(fft_points/2)+1) = speech_arnr(act,:);
        % Jika threshold besar dari -12dB, maka sinyal speech ada dan tidak ada atenuasi
    else

```

```

c = sqrt(10)/100;
% Jika threshold kurang dari -12dB, maka frame tersebut diatenuasi dengan faktor
% c, dengan  $20 \cdot \log(c) = -30$  dB
attn_speech(act,1:floor(fft_points/2)+1) =
c*(abs(fft_vec(act,1:floor(fft_points/2)+1)));
end
attn_speech(act,floor(fft_points/2)+2:fft_points)=attn_speech(act,floor(fft_points/2):-
1:2);
attn_speech(act,:)=attn_speech(act,:).*exp(j*phase_vec(act,:));
end;

%attn_speech=speech;

% Proses sintesis : menghitung Inverse Fourier Transform (IFT) dari setiap window
for frame = 1:length(st_locs)
attn_speech(frame,1:257)=speech_arnr(frame,1:257);
attn_speech(frame,258:512)=attn_speech(frame,256:-1:2);
attn_speech(frame,:)=attn_speech(frame,:).*exp(j*phase_vec(frame,:));
temp_isig = ifft(attn_speech(frame,:),fft_points);
isig(frame,1>window_samp) = temp_isig(1>window_samp);
end;

% Window data dibuat overlap (ditambah) untuk merekonstruksi sinyal keluaran speech
recon_sig=[];
recon_sig(1>window_samp/2) = isig(1,1>window_samp/2);
for frm = 1:length(st_locs) - 1
temp_frm = isig(frm>window_samp/2 + 1>window_samp) + isig(frm
+1,1>window_samp/2);
recon_sig(length(recon_sig) + 1: length(recon_sig) + window_samp/2) =temp_frm;
end;

recon_sig(length(recon_sig) + 1: length(recon_sig) + window_samp/2) =...
isig(length(st_locs),window_samp/2 + 1>window_samp);
recon_sig=real(recon_sig);

```

```

% Menghitung SNR
p_jg_min=min(length(asli),length(recon_sig));
asli=asli(1:p_jg_min,:);
sig=sig(1:p_jg_min,:);
recon_sig=(recon_sig(:,1:p_jg_min))';
daya_asli=sum(asli.^2);
% daya_delta_asli_noise=sum((abs(asli)-abs(sig)).^2);
% daya_delta_asli_enhance=sum((abs(asli)-abs(recon_sig)).^2);
daya_delta_asli_noise=sum((asli-sig).^2);
daya_delta_asli_enhance=sum((asli-recon_sig).^2);
SNR_awal=10*log10(daya_asli./daya_delta_asli_noise);
SNR_akhir=10*log10(daya_asli./daya_delta_asli_enhance);
delta_SNR = SNR_akhir - SNR_awal;
disp(['Nilai SNR sebelum proses enhance = ', num2str(SNR_awal), ' dB']);
disp(' ');
disp(['Nilai SNR sesudah proses enhance = ', num2str(SNR_akhir), ' dB']);
disp(' ');
disp(['Peningkatan nilai SNR yang diperoleh = ', num2str(delta_SNR), ' dB']);
disp(' ');

% Menggambar sinyal sebelum dan sesudah direduksi noise-nya

figure(2);
p_asli=length(asli);
p_sig=length(sig);
p_recon_sig=length(real(recon_sig));
t_asli=1/fs:1/fs:p_asli/fs;
t_sig=1/fs:1/fs:p_sig/fs;
t_recon_sig=1/fs:1/fs:p_recon_sig/fs;
subplot(311);plot(t_asli,asli);ylabel('Amplituda (volt)');xlabel('Waktu
(detik)');title('Sinyal asli');grid on; zoom on;

```

```

% figure(3);
subplot(312);plot(t_sig,sig);ylabel('Amplituda (volt)');xlabel('Waktu (detik)');title('Sinyal
terkorupsi noise');grid on; zoom on;
% figure(4);
subplot(313);plot(t_recon_sig,real(recon_sig));ylabel('Amplituda (volt)');xlabel('Waktu
(detik)');title('Sinyal sudah di-enhance');grid on; zoom on;

figure(5);
clf;plot(mag_avg);title('Magnitude noise rata-rata');grid on; zoom on;

figure(6);clf;subplot(111);plot(abs(fft_vec(1>window_samp,300)));title('FFT masing-
masing frame');
grid on;zoom on;
% subplot(212);plot(mag_avg);grid on; zoom on;
% subplot(313);plot(real(recon_sig(1>window_samp,300)));grid on; zoom on;

% soundsc(asli,fs); pause;
soundsc(sig,fs); pause;
soundsc(0.99.*recon_sig,fs);pause;
close all;
% soundsc(recon_sig,22000);

```