LAMPIRAN A

LISTING PROGRAM

```
############################

% Tampil sinyal suara


[y fs]=wavread('      ');

wavplay(y,fs);


% Menggambar sinyal suara

figure(1);

panjang_y=length(y);

t_y=1/fs:1/fs:panjang_y/fs;

% subplot(211);plot(211);plot(t_y,y);title('Gambar Sinyal');ylabel('Amplituda
(volt)');xlabel('Waktu (detik)');

% subplot(212);plot(y);

plot(211);plot(t_y,y);title('Gambar Sinyal');ylabel('Amplituda
(volt)');xlabel('Waktu (detik)');


% Stem sinyal

figure(2);

Y_potong_gbr=(y(10000:10100));

panjang_Y=length(Y_potong_gbr);

t_Y_potong_gbr=1/fs:1/fs:Y_potong_gbr/fs;

subplot(311);plot(Y_potong_gbr);title('Potong Gambar');ylabel('Amplituda
(volt)');xlabel('Sampel');

subplot(312);stem(Y_potong_gbr);title('Stem Gambar');ylabel('Amplituda
(volt)');xlabel('Sampel');
```

```
% Windowing

w=hamming(panjang_Y);

wh=Y_potong_gbr.*w;

subplot(313);plot(wh);title('sinyal di-Hamming');ylabel('Amplituda
(volt)');xlabel('Sampel');


% FFT

Y_FFT=abs(fft(Y_potong_gbr,512));

Y_dB=20*log(Y_FFT)

figure(3);

plot(Y_dB);title('FFT sinyal');ylabel('Magnituda (dB)');xlabel('Frekuensi
(Hertz)');

pause;

close all;


##################################################################
Main.m
######
function main(mysterySignal, speakerDatabase)
% Ini Adalah main function yang merekam sinyal suara misteri
% ... dari seseorang yang berusaha "meng-akses sistem".
% ... speaker Database terdiri dari contoh suara dari orang yang "diterima".

% ini men-set threshhold untuk envelope detection
envThreshhold = .2;

error = '';

% penggal 300 sample (untuk menghilangkan microphone "turn-on" spike)...
% ... dan zero-kan signal
mysterySignal = zerovect(mysterySignal);

% Normalize mysterySignal
mysterySignal = mysterySignal * ( 1 / sqrt(mysterySignal' * mysterySignal) );
```

```
% jalankan envelope detector untuk mencari awal dan akhir untuk setiap
pengucapan
[vowelStart, vowelEnd] = newenvdetect(mysterySignal, envThreshhold);

numberOfVowels = length(vowelStart);

% Untuk setiap ucapan pada sinyal misteri
for k = 1 : numberOfVowels

   % Isolate kth vowel of mysterySignal
   mysVowelSig = mysterySignal( vowelStart(k) : vowelEnd(k) );

   % Create vocal model of kth vowel
   [freqkHz, mysVowelModelMag] = vocalModel(mysVowelSig);
   figure (k)
   subplot(511)
   semilogy(freqkHz, mysVowelModelMag);
   title('mysterySig')

   % Figure out which vowel (a, e, i, o...) the kth vowel is
   % Find formant frequencies of kth vowel
   [formantFreq, magForm] = formantfind(freqkHz, mysVowelModelMag);
   formantFreq

   % Database of known first and second formant values for vowel sounds
   % Note: we are using only select vowels appearing in the groups'...
   % ... names to emulate a security database
   % format: vowel = [minFormant1 maxFormant1 minFormant2 maxFormant2]

    a = [ .670 .810 1.500 1.900 ]; % seperti cat
   ah = [ .675 .830 .900 1.500]; % seperti dog
   ay = [ .455 .570 1.875 2.500 ]; % seperti pay
   ee = [ .200 .350 2.000 2.650 ]; % seperti meet
   eh = [ .510 .675 1.620 1.960 ]; % seperti bet
   ih = [ .350 .500 1.975 2.145 ]; % seperti fish
   oh = [ .400 .600 .850 1.500 ]; % seperti boat

   % membandingkan hasil kalkulasi dari suara misteri dengan nilai di database
   if ( formantFreq(1) > a(1) && formantFreq(1) < a(2)...
      && formantFreq(2) > a(3) && formantFreq(2) < a(4) )

      mysVowel(k) = 1; % 'a'

   elseif ( formantFreq(1) > ah(1) && formantFreq(1) < ah(2)...
      && formantFreq(2) > ah(3) && formantFreq(2) < ah(4) )

      mysVowel(k) = 2; % 'ah'
```

```
    elseif ( formantFreq(1) > ay(1) && formantFreq(1) < ay(2)...
        && formantFreq(2) > ay(3) && formantFreq(2) < ay(4) )

        mysVowel(k) = 3; % 'ay'

    elseif ( formantFreq(1) > ee(1) && formantFreq(1) < ee(2)...
        && formantFreq(2) > ee(3) && formantFreq(2) < ee(4) )

        mysVowel(k) = 4; % 'ee'

    elseif ( formantFreq(1) > eh(1) && formantFreq(1) < eh(2)...
        && formantFreq(2) > eh(3) && formantFreq(2) < eh(4) )

        mysVowel(k) = 5; % 'eh'

    elseif ( formantFreq(1) > ih(1) && formantFreq(1) < ih(2)...
        && formantFreq(2) > ih(3) && formantFreq(2) < ih(4) )

        mysVowel(k) = 6; % 'ih'

    elseif ( formantFreq(1) > oh(1) && formantFreq(1) < oh(2)...
        && formantFreq(2) > oh(3) && formantFreq(2) < oh(4) )

        mysVowel(k) = 7; % 'oh'
    else
        errmsg = sprintf('no vowel match in vowel number %d', k);
        disp(errmsg)
        continue

    end

    % Output untuk ditampilkan ucapan mana yang ditemukan
    disp(sprintf(' Found %d', mysVowel(k)));

    numberOfPeople = length(speakerDatabase);

    % untuk setiap orang dalam database atau jumlah pengucap kata yang dikenali
    for l = 1 : numberOfPeople

        % Give each person a score for the current vowel and put it...
        % ... into a matrix with a row for each person and a column...
        % ... for each vowel
        matchMatrix(l, k) = speakerDatabase(l).vowel(mysVowel(k)).model'*
mysVowelModelMag;
        figure(k)
        subplot(5,1,(l+1))
```

```
      semilogy(freqkHz , speakerDatabase(l).vowel(mysVowel(k)).model)
      title(sprintf('speaker number %d',l))
   end

end

% Add up the elements of each row in matchMatrix to get a total "score"...
% ... for each person
for m = 1 : size(matchMatrix, 1)
   scoreVector(m) = sum( matchMatrix(m, :));
end

% Cari nilai tertinggi
[highestScore, highestScoreIndex] = max(scoreVector);

matchMatrix
scoreVector

% Set threshhold score yang harus diberikan untuk memberikan hasil yang positif
thresholdMultiplier = length(nonzeros(matchMatrix(1,:)));
threshholdScore = .85 * thresholdMultiplier;


if (highestScore > threshholdScore)
   disp(['Hello ' speakerDatabase(highestScoreIndex).name]);
   soundsc(speakerDatabase(highestScoreIndex).welcome,8000);

else
   disp('No Match');
end

#################################################################
databaseCreationScript.m
#####################
% databaseCreationScript
% We ran this after recording everybodies' vowel sounds to get the...
% ... models we needed for the speaker recognition.  Each vowel sound...
% ... is a field in a structure.

% dhamodel = Damen Hattori's "a" model sound
knownSpeaker(1).name = 'Damen Hattori';
knownSpeaker(1).welcome = HelloDamen;
knownSpeaker(1).vowel(1).model   = dhamodel; % vowel 1 = a
knownSpeaker(1).vowel(2).model   = dhahmodel; % vowel 2 = ah
knownSpeaker(1).vowel(3).model   = dhaymodel; % vowel 3 = ay
knownSpeaker(1).vowel(4).model   = dheemodel; % vowel 4 = ee
knownSpeaker(1).vowel(5).model   = dhehmodel; % vowel 5 = eh
```

```
knownSpeaker(1).vowel(6).model   = dhihmodel; % vowel 6 = ih
knownSpeaker(1).vowel(7).model   = dhohmodel; % vowel 7 = oh
knownSpeaker(1).vowel(8).model   = dhohmodel; % vowel 8 = ue

knownSpeaker(2).name = 'Josh Long';
knownSpeaker(2).welcome = HelloJosh;
knownSpeaker(2).vowel(1).model   = jlamodel;
knownSpeaker(2).vowel(2).model  = jlahmodel;
knownSpeaker(2).vowel(3).model   = jlaymodel;
knownSpeaker(2).vowel(4).model   = jleemodel;
knownSpeaker(2).vowel(5).model   = jlehmodel;
knownSpeaker(2).vowel(6).model   = jlihmodel;
knownSpeaker(2).vowel(7).model   = jlohmodel;
knownSpeaker(2).vowel(8).model   = jluemodel;

knownSpeaker(3).name = 'Matt McDonell';
knownSpeaker(3).welcome = HelloMatt;
knownSpeaker(3).vowel(1).model   = mmamodel;
knownSpeaker(3).vowel(2).model  = mmahmodel;
knownSpeaker(3).vowel(3).model   = mmaymodel;
knownSpeaker(3).vowel(4).model   = mmeemodel;
knownSpeaker(3).vowel(5).model   = mmehmodel;
knownSpeaker(3).vowel(6).model   = mmihmodel;
knownSpeaker(3).vowel(7).model   = mmohmodel;
knownSpeaker(3).vowel(8).model   = mmuemodel;

knownSpeaker(4).name = 'Chris Pasich';
knownSpeaker(4).welcome = HelloChris;
knownSpeaker(4).vowel(1).model   = cpamodel;
knownSpeaker(4).vowel(2).model  = cpahmodel;
knownSpeaker(4).vowel(3).model   = cpaymodel;
knownSpeaker(4).vowel(4).model   = cpeemodel;
knownSpeaker(4).vowel(5).model   = cpehmodel;
knownSpeaker(4).vowel(6).model   = cpihmodel;
knownSpeaker(4).vowel(7).model   = cpohmodel;
knownSpeaker(4).vowel(8).model   = cpuemodel;

% Give the models names in the structure
for k = 1:4;
knownSpeaker(k).vowel(1).name = 'a' ;
knownSpeaker(k).vowel(2).name = 'ah' ;
knownSpeaker(k).vowel(3).name = 'ay' ;
knownSpeaker(k).vowel(4).name = 'ee' ;
knownSpeaker(k).vowel(5).name = 'eh' ;
knownSpeaker(k).vowel(6).name = 'ih' ;
knownSpeaker(k).vowel(7).name = 'oh' ;
knownSpeaker(k).vowel(8).name = 'ue' ;
```

```
end

% Create final database to use
speakerDatabase = knownSpeaker(1 : 4);
```