LAMPIRAN

```
//Kode Matlab untuk ekstrasi Suara MFCC


function c = mfccAlgorithm(s, fs)
frameSize = 256;                    % frame size
frameDistance = 100;                    % inter frame distance
len = length(s);
numberOfFrames = 1 + floor((len - frameSize)/double(frameDistance));
frameVectors = zeros(frameSize, numberOfFrames); % vector of frame
vectors

for i=1:numberOfFrames
    index = 100*(i-1) + 1;
    for j=1:frameSize
        frameVectors(j,i) = s(index);
        index = index + 1;
    end
end

hammingWindow = hamming(frameSize);           % hamming window
afterWinMat = diag(hammingWindow)*frameVectors;
freqDomMat = fft(afterWinMat);  % FFT into freq domain

filterBankMat = melFilter(20, frameSize, fs);             % matrix
for a mel-spaced filterbank
nby2 = 1 + floor(frameSize/2);
melSpectrum = filterBankMat*abs(freqDomMat(1:nby2,:)).^2; % mel
spectrum
c = dct(log(melSpectrum));                          % mel-frequency
cepstrum coefficients
c(1,:) = [];                                % exclude 0'th order
cepstral coefficient
```

```
//Kode Matlab untuk Perhitungan Vector Quantitation

function codebk = VectorQuantizationCodeBook(d, k)
e = 0.0001;                                % splitting parameter
codebk = mean(d, 2);                        % code book
distortion = int32(inf);
numOfCentroids = int32(log2(k));           % number of code
words/centroids

for i=1:numOfCentroids
    codebk = [codebk*(1+e), codebk*(1-e)];  % the splitting
    while(1==1)
        dis = distanceCalculated(d, codebk);          % distance of
each point to every code word
        [m,ind] = min(dis, [], 2);         % ind maps points in 'd' to
closest centroid
        t = 0;
        lim = 2^i;
        for j=1:lim
            codebk(:, j) = mean(d(:, ind==j), 2);    % updating
centroids to better mean values
            x = distanceCalculated(d(:, ind==j), codebk(:, j));  % x is
a cluster i.e vector of neighbouring ...
            len = length(x);                      % ... points of a
centroid
            for q = 1:len
                t = t + x(q);
            end
        end
        if (((distortion - t)/t) < e)      % distortion condition
breaks the loop
            break;
        else
            distortion = t;
        end
    end
end
```

```
%penghitingan Jarak

function d = distanceCalculated(x, y)

[m1, n1, k1] = size(x);
[m2, n2, k2] = size(y);

d = zeros(n1, n2);

if (n1 < n2)
    copies = zeros(1,n2);
    for n = 1:n1
        d(n,:) = sum((x(:, n+copies) - y) .^2, 1);
    end
else
    copies = zeros(1,n1);
    for p = 1:n2
        d(:,p) = sum((x - y(:, p+copies)) .^2, 1)';
    end
end

d = d.^0.5;
```

// Kode Matlab untuk input data training

```
clc;
close all;
clear all;
Fs=8000;
Nseconds = 1;
samp=4;
k=20;

for i= 1:1:samp
    fprintf('Ucapkan kata KIRI setelah menekan tombol enter');
    input('');

    x= wavrecord(Nseconds*Fs,Fs,'double');
    wavplay(x,Fs)
    v = mfccAlgorithm(x, Fs);
    code{i} = VectorQuantizationCodeBook(v, k);
    figure(1);
    plot(VectorQuantizationCodeBook(v, k));
    title('Hasil Ekstrasi Suara Kata KIRI');
end
for i= (samp+1):1:2*samp
    fprintf('Ucapkan kata KANAN setelah menekan tombol enter');
    input('');
    x= wavrecord(Nseconds*Fs,Fs,'double');
    wavplay(x,Fs)
    v = mfccAlgorithm(x, Fs);
    code{i} = VectorQuantizationCodeBook(v, k);
    figure(1);
    plot(VectorQuantizationCodeBook(v, k));
    title('Hasil Ekstrasi Suara Kata KANAN');
end
for i= (2*samp+1):1:3*samp
    fprintf('Ucapkan kata MAJU setelah menekan tombol enter');
    input('');
    x= wavrecord(Nseconds*Fs,Fs,'double');
    wavplay(x,Fs)
    v = mfccAlgorithm(x, Fs);
    code{i} = VectorQuantizationCodeBook(v, k);
    figure(1);
    plot(VectorQuantizationCodeBook(v, k));
    title('Hasil Ekstrasi Suara Kata MAJU');
end
for i= (3*samp+1):1:4*samp
    fprintf('Ucapkan kata MUNDUR setelah menekan tombol enter');
    input('');
    x= wavrecord(Nseconds*Fs,Fs,'double');
    wavplay(x,Fs)
    v = mfccAlgorithm(x, Fs);
    code{i} = VectorQuantizationCodeBook(v, k);
    figure(1);
    plot(VectorQuantizationCodeBook(v, k));
    title('Hasil Ekstrasi Suara Kata MUNDUR');
end
```

```
for i= (4*samp+1):1:5*samp
    fprintf('Ucapkan kata STOP setelah menekan tombol enter');
    input('');
    x= wavrecord(Nseconds*Fs,Fs,'double');
    wavplay(x,Fs)
    v = mfccAlgorithm(x, Fs);
    code{i} = VectorQuantizationCodeBook(v, k);
    figure(1);
    plot(VectorQuantizationCodeBook(v, k));
    title('Hasil Ekstrasi Suara Kata STOP');
end

save('code.mat','code')
```

```
// Kode Matlab untuk Pengujian Speech Recognition

clc;
close all;
Fs=8000;
Nseconds = 1;
k = 20;


code=load('code.mat');
code=code.code;

while(1)
fprintf('Ucapkan kata setelah menekan tombol enter');
input('');
x= wavrecord(Nseconds*Fs,Fs,'double');
v = mfccAlgorithm(x, Fs);
distmin = inf;
k1 = 0;
for l = 1:length(code)      % each trained codebook, compute
distortion
       d = distanceCalculated(v, code{l});
       dist = sum(min(d,[],2)) / size(d,1);
       if dist < distmin
           distmin = dist;
           k1 = l;
       end
end

if(k1<=4)
    disp('KIRI')
    fwrite(s2,'L');

;

elseif(k1>=5)&&(k1<=8)
    disp('KANAN')
     fwrite(s2,'R');
;

elseif(k1>=9)&&(k1<=12)
    disp('MAJU')
    fwrite(s2,'F');
;

elseif(k1>=12)&&(k1<=16)
    disp('MUNDUR')
    fwrite(s2,'B');
;

elseif(k1>=16)&&(k1<=20)
    disp('STOP')
   fwrite(s2,'S');
;
   else
    disp('PERINTAH TIDAK DIKENALI') ;
   end
end
```