Verifikasi Biometrika Suara Menggunakan Metode MFCC Dan DTW

Darma Putra[1] , Adi Resmawan[2]
[1]Staff pengajar Teknologi Informasi, Fakultas Teknik, Universitas Udayana
[2]Alumni Teknik Elektro, Fakultas Teknik, Universitas Udayana
email : duglaire@yahoo.com[1], adiresmawan@yahoo.com[2]

Abstrak

Teknologi pengenalan suara merupakan salah satu teknologi biometrika yang tidak memerlukan biaya besar serta peralatan khusus. Suara merupakan salah satu dari bagian tubuh manusia yang unik dan dapat dibedakan dengan mudah. Aplikasi yang dibuat dalam penelitian ini adalah sistem verifikasi suara yang dapat memverifikasi/membuktikan identitas yang di klaim oleh seseorang berdasarkan suara yang di-input-kan.

Perangkat lunak ini dirancang menggunakan metode MFCC (Mel Frequency Cepstrum Coefficients) untuk proses ekstraksi ciri dari sinyal wicara dan metode DTW (Dynamic Time Warping) untuk proses pencocokan. Proses MFCC akan mengkonversikan sinyal suara menjadi beberapa vektor yang berguna untuk proses pengenalan. Vector ciri hasil dari proses MFCC selanjutnya akan dibandingkan dengan vector ciri yang tersimpan dalam basis data melalui proses DTW berdasarkan ID yang di klaim oleh pengguna. Bahasa pemrograman yang digunakan dalam merancang perangkat lunak ini adalah Visual C# 2008.

Pengujian dilakukan terhadap 35 orang pengguna yang terdiri dari 27 orang laki-laki dan 8 orang perempuan. Masing-masing orang mengucapkan 5 buah kata yang telah ditentukan sebelumnya, dimana untuk masing-masing kata diucapkan sebanyak 7 kali. Enam buah sampel dijadikan sebagai acuan dan 1 sebagai sampel uji. Hasil pengujian memperlihatkan tingkat akurasi paling rendah adalah 59.664 %, sedangkan tingkat akurasi tertinggi yaitu 93.254 %. Baik buruknya sistem dalam melakukan pengenalan dipengaruhi oleh panjang frame, panjang overlapping, jumlah koefisien fileterbank, dan jumlah koefisien MFCC.

Kata kunci : pengenalan suara, MFCC, DTW, filterbank, verifikasi suara.

Abstract

Voice recognition technology is one of the biometrics technology that does not require great expense and special equipment. Voice is one of human body parts that unique and easily distinguishable. Application made in this research is a voice verification system that can authenticate the identity of the a person based on his/her voice.

The software is designed using MFCC (Mel Frequency Cepstrum Coefficients) for the process of feature extraction from speech signals and method of DTW (Dynamic Time Warping) for the matching process. MFCC process convert the voice signal into a useful vector for the recognition. Vector features result from the process compared with the MFCC feature vector stored in the database through the Dynamic Time Warping process based on ID claims by the user. The programming language used in designing this software is Visual C# 2008.

Test conducted on 35 people consisting of 27 men and 8 women. Each person say 5 predetermined words, where each word is spoken 7 times. Six samples is used as reference and one as a test sample. Test results show the lowest accuracy rate was 59,664%, while the highest level of accuracy was 93,254%. The result of this recognition system is affected by the length of the frame, overlapping length, the number of coefficients fileterbank, and the number of MFCC coefficients.

Key words: speech recognition, MFCC, DTW, filterbank, voice verification.

1. PENDAHULUAN

Perkembangan teknologi terutama dalam bidang komputer saat ini melaju sangat pesat. Hal tersebut dipicu oleh perkembangan ilmu pengetahuan disertai kebutuhan manusia akan teknologi canggih yang dapat mempermudah pekerjaan. Salah satu teknologi dibidang komputer yang banyak diteliti saat ini adalah teknologi biometrika. Teknologi biometrika merupakan suatu teknik pengenalan diri menggunakan bagian tubuh atau perilaku manusia.

Teknologi ini mememuhi dua fungsi penting yaitu identifikasi dan verifikasi. Sistem identifikasi bertujuan untuk memecahkan identitas seseorang. Sedangkan sistem verifikasi bertujuan untuk menolak atau menerima identitas yang diklaim oleh seseorang.

Kebutuhan akan sistem keamanan yang tangguh merupakan salah satu faktor penting kenapa teknologi biometrika terus dikembangkan. Sistem keamanan lama yaitu dengan menggunakan password saat ini sudah banyak kelemahannya. Disamping itu banyak orang hanya menggunakan satu password untuk segala hal, mulai dari e-mail, penggunaan kartu ATM, sampai menjadi keanggotaan mailing list. Kelemahan penggunaan password tersebut dapat diatasi dengan menggunakan teknologi pengenalan suara (Syah, 2009). Teknologi pengenalan suara *(speaker recognition)* merupakan salah satu teknologi biometrika yang tidak memerlukan biaya besar serta peralatan khusus. Pada dasarnya setiap manusia memiliki sesuatu yang unik/khas yang hanya dimiliki oleh dirinya sendiri. Suara merupakan salah satu dari bagian tubuh manusia yang unik dan dapat dibedakan dengan mudah. Disamping itu, sistem biometrika suara memiliki karakteristik seperti, tidak dapat lupa, tidak mudah hilang, dan tidak mudah untuk dipalsukan karena keberadaannya melekat pada diri manusia sehingga keunikannya lebih terjamin (Syah, 2009).

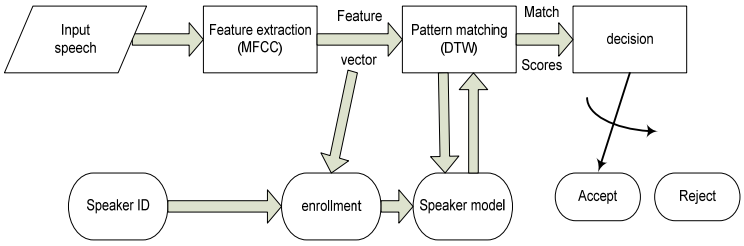
Dari permasalahan diatas, dalam penelitian ini akan dibahas mengenai bagaimana merancang dan membuat suatu perangkat lunak yang dapat melakukan verifikasi terhadap seorang pembicara dengan menggunakan metode MFCC sebagai ekstraksi ciri dan DTW untuk proses pencocokan.

2. KONSEP DASAR PENGENALAN SUARA

Pengenalan suara dapat dikategorikan menjadi 3 bagian, yaitu : *speech recognition, speaker recognition,* dan *language recognition.* Dalam penelitian ini hanya khusus membahas mengenai *speaker recognition* lebih spesifiknya lagi membahas tentang *speaker verification.*

Speaker recognition adalah suatu proses yang bertujuan mengenali siapa yang sedang berbicara berdasarkan informasi yang terkandung dalam gelombang suara yang di-*input*-kan. *Speaker recognition* dibagi menjadi 2 bagian, yaitu : *speaker verification* dan *speaker identification.*

Speaker verification adalah proses verifikasi seorang pembicara, dimana sebelumnya telah diketahui identitas pembicara tersebut berdasarkan data yang telah diinputkan. *Speaker verification* melakukan perbandingan *one to one* (1:1)*.* dalam arti bahwa fitur-fitur suara dari seorang pembicara dibandingkan secara langsung dengan firur-fitur seorang pembicara tertentu yang ada dalam sistem*.* Bila hasil perbandingan (skor) tersebut lebih kecil atau sama dengan batasan tertentu (*treshold*), maka pembicara tersebut diterima, bila tidak maka akan ditolak (dengan asumsi semakin kecil skor berarti kedua sampel semakin mirip). Gambar dibawah adalah blok diagram dari *speaker verification*.



Gambar 1 Blok Diagram *Speaker Verification* (Darma Putra, 2009)

Speaker identification adalah proses mendapatkan identitas dari seorang pembicara dengan membandingkan fitur-fitur suara yang diinputkan dengan semua fitur-fitur dari setiap pembicara yang ada dalam *database.* Berbeda dengan pada *speaker verification,* proses ini melakukan perbandingan *one to many* (1:N).

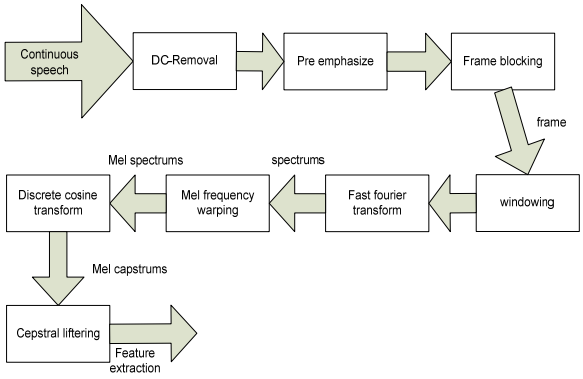
3. FEATURE EKSTRAKSI DENGAN METODE MFCC

MFCC (*Mel Frequency Cepstrum Coefficients*) merupakan salah satu medode yang banyak digunakan dalam bidang *speech technology,* baik *speaker recognition* maupun *speech recognition.* Metode ini digunakan untuk melakukan *feature extraction,* sebuah proses yang

mengkonversikan *signal* suara menjadi beberapa parameter. Beberapa keunggulan dari metode ini adalah (Manunggal, 2005) :

a. Mampu untuk menangkap karakteristik suara yang sangat penting bagi pengenalan suara, atau dengan kata lain dapat menangkap informasi-informasi penting yang terkandung dalam *signal* suara.
b. Menghasilkan data seminimal mungkin, tanpa menghilangkan informasi-informasi penting yang dikandungnya.
c. Mereplikasi organ pendengaran manusia dalam melakukan persepsi terhadap *signal* suara.

MFCC *feature extraction* sebenarnya merupakan adaptasi dari sistem pendengaran manusia, dimana *signal* suara akan difilter secara linear untuk frekuensi rendah (dibawah 1000 Hz) dan secara logaritmik untuk frekuensi tinggi (diatas 1000 Hz). Gambar dibawah ini merupakan block diagram untuk MFCC.



Gambar 2 Blok Diagram Untuk MFCC

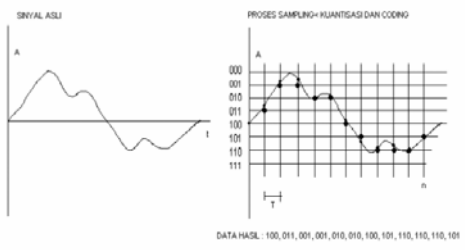
3.1. Konversi Analog menjadi Digital

Signal – signal yang natural pada umumnya seperti *signal* suara merupakan *signal continue* dimana memiliki nilai yang tidak terbatas. Sedangkan pada komputer, semua *signal* yang dapat diproses oleh komputer hanyalah *signal discrete* atau sering dikenal sebagai istilah *digital signal.* Agar *signal* natural dapat diproses oleh komputer, maka harus diubah terlebih dahulu dari data *signal continue* menjadi *discrete.* Hal itu dapat dilakukan melalui 3 proses, diantaranya adalah proses *sampling* data, proses kuantisasi, dan proses pengkodean.

Proses *sampling* adalah suatu proses untuk mengambil data *signal continue* untuk setiap periode tertentu. Dalam melakukan proses *sampling* data, berlaku aturan Nyquist, yaitu bahwa frekuensi *sampling (sampling rate)* minimal harus 2 kali lebih tinggi dari frekuensi maksimum yang akan di *sampling.* Jika *signal sampling* kurang dari 2 kali frekuensi maksimum *signal* yang akan di *sampling,* maka akan timbul efek *aliasing. Aliasing* adalah suatu efek dimana *signal* yang dihasilkan memiliki frekuensi yang berbeda dengan *signal* aslinya.

Proses kuantisasi adalah proses untuk membulatkan nilai data ke dalam bilangan-bilangan tertentu yang telah ditentukan terlebih dahulu. Semakin banyak level yang dipakai maka semakin akurat pula data *signal* yang disimpan tetapi akan menghasilkan ukuran data besar dan proses yang lama.

Proses pengkodean adalah proses pemberian kode untuk tiap-tiap data *signal* yang telah terkuantisasi berdasarkan level yang ditempati.



Gambar 3 Proses Pembentukan *signal* digital.

3.2. *DC-Removal*

Remove DC Components bertujuan untuk menghitung rata-rata dari data sampel suara, dan mengurangkan nilai setiap sampel suara dengan nilai rata-rata tersebut. Tujuannya adalah mendapat normalisasi dari data suara input.

$y[n] = x[n] - \overline{x}$, $0 \le n \le N-1$

Dimana :
y[n] = sampel signal hasil proses DC removal
x[n]= sampel signal asli

$\overline{x}$ *= nilai rata-rata sampel signal asli.*

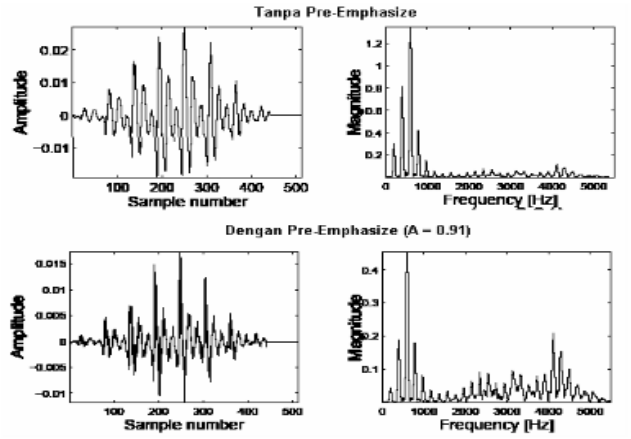
N = panjang signal

3.3. *Pre – emphasize Filetering*

Pre – emphasize Filetering merupakan salah satu jenis *filter* yang sering digunakan sebelum sebuah *signal* diproses lebih lanjut. *Filter* ini mempertahankan frekuensi-frekuensi tinggi pada sebuah spektrum, yang umumnya tereliminasi pada saat proses produksi suara.

Tujuan dari *Pre – emphasize Filetering* ini adalah (Manunggal, 2005) :

a. Mengurangi *noise ratio* pada *signal,* sehingga dapat meningkatkan kualitas *signal.*
b. Menyeimbangkan spektrum dari *voiced sound.* Pada saat memproduksi *voiced sound, glottis* manusia menghasilkan sekitar -12 dB *octave slope.* Namun ketika energy akustik tersebut dikeluarkan melalui bibir, terjadi peningkatan sebesar +6. Sehingga *signal* yang terekam oleh *microphone* adalah sekitar -6 dB *octave slope.* Dampak dari efek ini dapat dilihat pada gambar dibawah ini.



Gambar 4 Contoh dari *pre-emphasize* pada sebuah frame

Pada gambar diatas terlihat bahwa distribusi energi pada setiap frekuensi terlihat lebih seimbang setelah diimplementasikan *pre-emphasize filter.*

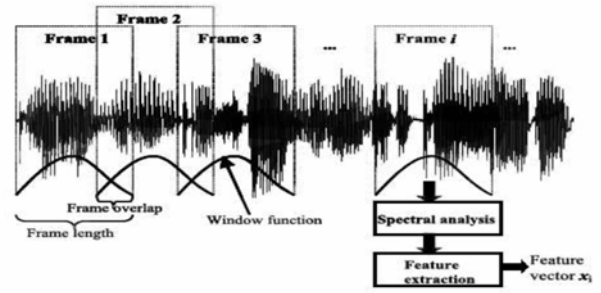
Bentuk yang paling umum digunakan dalam *pre-emphasize filter* adalah sebagai berikut :

$$y[n] = s[n] - \alpha\, s[n-1] \quad , 0.9 \le \alpha \le 1.0$$

Dimana :
y[n] = *signal* hasil *pre-emphasize filter*
s[n] = *signal* sebelum *pre-emphasize filter*

3.4. *Frame Blocking*

Karena *signal* suara terus mangalami perubahan akibat adanya pergeseran artikulasi dari organ produksi vocal, *signal* harus diproses secara *short segments (short frame).* Panjang *frame* yang biasanya digunakan untuk pemrosesan *signal* adalah antara 10-30 milidetik. Panjang *frame* yang digunakan sangat mempengaruhi keberhasilan dalam analisa spektral. Di satu sisi, ukuran dari *frame* harus sepanjang mungkin untuk dapat menunjukkan resolusi frekuensi yang baik. Tetapi di lain sisi, ukuran *frame* juga harus cukup pendek untuk dapat menunjukkan resolusi waktu yang baik.



Gambar 5 *Short Term Spectral Analysis* (Manunggal, 2005)

Proses *frame* ini dilakukan terus sampai seluruh *signal* dapat diproses. Selain itu, proses ini umumnya dilakukan secara *overlapping* untuk setiap *frame*- nya. Panjang daerah *overlap* yang umum digunakan adalah kurang lebih 30% sampai 50% dari panjang *frame. Overlapping* dilakukan untuk menghindari hilangnya ciri atau karakteristik suara pada perbatasan perpotongan setiap *frame.*

3.5. *Windowing*

Proses *framing* dapat menyebabkan terjadinya kebocoran spektral (*spectral leakage*) atau *aliasing. Aliasing* adalah *signal* baru dimana memiliki frekuensi yang berbeda dengan *signal* aslinya. Efek ini dapat terjadi karena rendahnya jumlah *sampling rate,* ataupun karena proses *frame blocking* dimana menyebabkan *signal* menjadi *discontinue.* Untuk mengurangi kemungkinan terjadinya kebocoran spektral, maka hasil dari proses *framing* harus melewati proses *window.*

Sebuah fungsi *window* yang baik harus menyempit pada bagian *main lobe* dan melebar pada bagian *side lobe*-nya.

Berikut ini adalah representasi dari fungsi *window* terhadap *signal* suara yang diinputkan.

$x(n) = x_t(n)w(n)$ n= 0,1,…,N-1

$x(n)$ = nilai sampel *signal* hasil *windowing*

$x_t(n)$ = nilai sampel dari *frame signal* ke i

$w(n)$ = fungsi *window*

N = *frame size*, merupakan kelipatan 2

Ada banyak fungsi *window,* namun yang paling sering digunakan dalam aplikasi *speaker recognition* adalah *hamming window.* Fungsi *window* ini menghasilkan *sidelobe level* yang tidak terlalu tinggi (kurang lebih -43 dB), selain itu *noise* yang dihasilkan pun tidak terlalu besar.

Fungsi Hamming window adalah sebagai berikut :

$$0.54 - 0.46\cos\frac{2\pi n}{M-1}$$

Dimana :
n = 0,1,...,M-1
M = panjang *frame*

3.6. Analisis *Fourier*

Analisis *fourier* adalah sebuah metode yang memungkinkan untuk melakukan analisa terhadap *spectral properties* dari *signal* yang diinputkan. Representasi dari *spectral properties* sering disebut sebagai *spectrogram.*

Dalam *spectrogram* terdapat hubungan yang sangat erat antara waktu dan frekuensi. Hubungan antara frekuensi dan waktu adalah hubungan berbanding terbalik. Bila resolusi waktu yang digunakan tinggi, maka resolusi frekuensi yang dihasilkan akan semakin rendah.

3.6.1. *Discrete Fourier Transform (DFT)*

DFT merupakan perluasan dari transformasi *fourier* yang berlaku untuk *signal-signal* diskrit dengan panjang yang terhingga. Semua *signal* periodik terbentuk dari gabungan *signal-signal* sinusoidal yang menjadi satu yang dapat dirumuskan sebagai berikut :

$$S[k] = \sum_{n=0}^{N-1} s[n] e^{-j2\pi nk/N}, 0 \leq k \leq N-1$$

N = jumlah sampel yang akan diproses ($N \in$ N)

S(n) = nilai sampel *signal*

K = variable frekuensi discrete, dimana akan bernilai ($k = {}^{N}/_{2}, k \in$ N)

Dengan rumus diatas, suatu *signal* suara dalam domain waktu dapat kita cari frekuensi pembentuknya. Hal inilah tujuan penggunaan analisa *fourier* pada data suara, yaitu untuk merubah data dari domain waktu menjadi data spektrum di domain frekuensi. Untuk pemrosesan *signal* suara, hal ini sangatlah menguntungkan karena data pada domain frekuensi dapat diproses dengan lebih mudah dibandingkan data pada domain waktu, karena pada domain frekuensi, keras lemahnya suara tidak seberapa berpengaruh.

Untuk mendapatkan spektrum dari sebuah *signal* dengan DFT diperlukan N buah sampel data berurutan pada domain waktu, yaitu x[m] sampai x[m+N-1]. Data tersebut dimasukkan dalam fungsi DFT maka akan menghasilkan N buah data. Namun karena hasil dari DFT adalah simetris, maka hanya N/2 data yang diambil sebagai spektrum.

3.6.2. *Fast Fourier Transform* (FFT)

Perhitungan DFT secara langsung dalam komputerisasi dapat menyebabkan proses perhitungan yang sangat lama. Hal itu disebabkan karena dengan DFT, dibutuhkan N^2 perkalian bilangan kompleks. Karena itu dibutuhkan cara lain untuk menghitung DFT dengan cepat. Hal itu dapat dilakukan dengan menggunakan algoritma *fast fourier transform* (FFT) dimana FFT menghilangkan proses perhitungan yang kembar dalam DFT.

3.7. *Mel Frequency Wrapping*

Mel Frequency Wrapping umumnya dilakukan dengan menggunakan *Filterbank. Filterbank* adalah salah satu bentuk dari *filter* yang dilakukan dengan tujuan untuk mengetahui ukuran energi dari *frequency band* tertentu dalam *signal* suara. *Filterbank* dapat diterapkan baik pada domain waktu maupun pada domain frekuensi, tetapi untuk keperluan MFCC, *filterbank* harus diterapkan dalam domain frekuensi.

Filterbank menggunakan representasi konvolusi dalam melakukan *filter* terhadap *signal.* Konvolusi dapat dilakukan dengan melakukan multiplikasi antara spektrum *signal* dengan koefisien *filterbank.* Berikut ini adalah rumus yang digunakan dalam perhitungan *filterbanks.*

$$Y[i] = \sum_{j=1}^{N} S[j] H_i[j]$$

N = jumlah *magnitude spectrum* ($N \in$ N)

S[j] = *magnitude spectrum* pada frekuensi j

$H_i[j]$ = koefisien *filterbank* pada frekuensi j ($1 \leq i \leq M$)

M = jumlah *channel* dalam *filterbank*

Persepsi manusia terhadap frekuensi dari *signal* suara tidak mengikuti *linear scale.* Frekuensi yang sebenarnya (dalam Hz) dalam sebuah *signal* akan diukur manusia secara subyektif dengan menggunakan *mel scale. Mel frequency scale* adalah *linear* frekuensi *scale* pada frekuensi dibawah 1000 Hz, dan merupakan *logarithmic scale* pada frekuensi diatas 1000 Hz.

3.8. *Discrete Cosine Transform (DCT)*

DCT merupakan langkah terakhir dari proses utama MFCC *feature extraction.* Konsep dasar dari DCT adalah mendekorelasikan *mel spectrum* sehingga menghasilkan representasi yang baik dari property spektral local. Pada dasarnya konsep dari DCT sama dengan *inverse fourier transform.* Namun hasil dari DCT mendekati PCA (*principle component analysis*). PCA adalah metode static klasik yang digunakan secara luas dalam analisa data dan kompresi. Hal inilah yang menyebabkan seringkali DCT menggantikan *inverse fourier transform* dalam proses MFCC *feature extraction.*

Berikut adalah formula yang digunakan untuk menghitung DCT.

$$C_n = \sum_{k=1}^{K} (\log S_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right]; \quad n = 1,2,\ldots,K$$

S_k = keluaran dari proses *filterbank* pada *index* k
K = jumlah koefisien yang diharapkan

Koefisien ke nol dari DCT pada umumya akan dihilangkan, walaupun sebenarnya mengindikasikan energi dari frame *signal* tersebut. Hal ini dilakukan karena, berdasarkan penelitian-penelitian yang pernah dilakukan, koefisien ke nol ini tidak *reliable* terhadap *speaker recognition.*

3.9. *Cepstral Liftering*

Hasil dari proses utama MFCC *feature extraction* memiliki beberapa kelemahan. *Low order* dari *cepstral coefficients* sangat sensitif terhadap *spectral slope,* sedangkan bagian *high order*nya sangat sensitif terhadap *noise.* Oleh karena itu, *cepstral liftering* menjadi salah satu standar teknik yang diterapkan untuk meminimalisasi sensitifitas tersebut.

Cepstral liftering dapat dilakukan dengan mengimplementasikan fungsi *window* terhadap *cepstral features.*

$$W[n] = \begin{cases} 1 + \frac{L}{2}\sin\left(\frac{n\pi}{L}\right) & n = 1,2,\ldots,L \\ 0 \end{cases}$$

L = jumlah *cepstral coefficients*
N = *index* dari *cepstral coefficients*

Cepstral liftering menghaluskan spektrum hasil dari *main processor* sehingga dapat digunakan lebih baik untuk *pattern matching.*

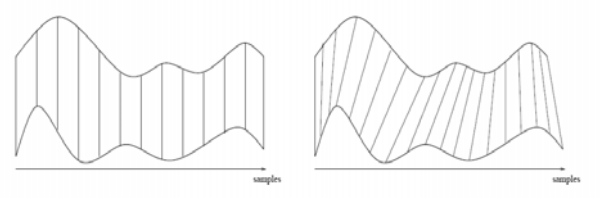
4. PENCOCOKAN DENGAN METODE DTW (*DYNAMIC TIME WARPING*)

Satu masalah yang cukup rumit dalam *speech recognition* (pengenalan wicara) adalah proses perekaman yang terjadi seringkali berbeda durasinya, biarpun kata atau kalimat yang diucapkan sama. Bahkan untuk satu suku kata yang sama atau vocal yang sama seringkali proses perekaman terjadi dalam durasi yang berbeda. Sebagai akibatnya proses matching antara sinyal uji dengan sinyal referensi (*template*) seringkali tidak menghasilkan nilai yang optimal.

Sebuah teknik yang cukup popular di awal perkembangan teknologi pengolahan sinyal wicara adalah dengan memanfaatkan sebuah teknik *dynamic-programming* yang juga lebih dikenal sebagai *Dynamic Time Warping* (DTW). Teknik ini ditujukan untuk mengakomodasi perbedaan waktu antara proses perekaman saat pengujian dengan yang tersedia pada *template* sinyal referensi. Prinsip dasarnya adalah dengan memberikan sebuah rentang '*steps*' dalam ruang (dalam hal ini sebuah frame-frame waktu dalam sample, frame-frame waktu dalam template) dan digunakan untuk mempertemukan lintasan yang menunjukkan *local match* terbesar (kemiripan) antara time frame yang lurus. Total `*similarity cost*' yang diperoleh dengan algorithm ini merupakan sebuah indikasi seberapa bagus *sample* dan *template* ini memiliki kesamaan, yang selanjutnya akan dipilih *best-matching template*.

DTW (*DynamicTime Warping*) adalah metode untuk menghitung jarak antara dua data *time series.* Keunggulan DTW dari metode jarak yang lainnya adalah mampu menghitung jarak dari dua vektor data dengan panjang berbeda.

Jarak DTW diantara dua vektor dihitung dari jalur pembengkokkan optimal (*optimal warping path*) dari kedua vektor tersebut. Ilustrasi pencocokan dengan metode DTW ditunjukkan pada gambar dibawah ini.



(a) (b)

Gambar 6 Pencocokan *sequence* (a) alignment asli dari 2 *sequence* (b) alignment dengan DTW (Darma Putra, 2009).

Dari beberapa teknik yang digunakan untuk menghitung DTW, salah satu yang paling handal adalah dengan metode pemrograman dinamis. Jarak DTW dapat dihitung dengan rumus:

$$D(U,V) = \gamma(m,n)$$

$$\gamma(m,n) = d_{base}(u_i, v_j) + \min\begin{cases}\gamma(i-1, j)\\ \gamma(i-1, j-1)\\ \gamma(i, j-1)\end{cases}$$

5. HASIL PENGUJIAN

Pengujian terhadap aplikasi yg telah dibuat dilakukan dengan mencari rasio kesalahan pencocokan yang menyatakan probabilitas terjadinya kesalahan pencocokan pada sistem. Terdapat 2 jenis rasio kesalahan pencocokan, yaitu: rasio kesalahan kecocokan (*false match rate*) dan rasio kesalahan ketidakcocokan (*false non match rate*).

1. Rasio Kesalahan Kecocokan
   False match rate (FMR) menyatakan probabilitas sampel dari pengguna cocok dengan acuan yang diambil secara acak milik pengguna yang berbeda. *False match rate* disebut juga *false positive*. Rasio kesalahan kecocokan dihitung dengan rumus:
   $$FMR = \frac{jumlah\ kesalahan\ cocok}{jumlah\ seluruh\ pencocokan}\ x\ 100$$
2. Rasio Kesalahan Ketidakcocokan
   False non match rate (FNMR) menyatakan probabilitas sampel dari pengguna tidak cocok dengan acuan lain yang diberikan pengguna yang sama. *False non match rate* disebut juga *False negative*. Rasio kesalahan ketidakcocokan dihitung dengan rumus:
   $$FNMR = \frac{jumlah\ kesalahan\ tidak\ cocok}{jumlah\ seluruh\ pencocokan}\ x\ 100$$
3. Nilai Ambang (*Thresold Value*)
   Nilai ambang, yang sering dilambangkan dengan T, memegang peranan penting dalam memutuskan terjadinya kesalahan dalam pencocokan. Nilai FMR/FNMR tergantung pada besarnya nilai ambang yang digunakan. Nilai T akan dibandingkan dengan skor hasil dan bila memenuhi kondisi Skor ≤ T, maka pengguna dinyatakan sah, bila tidak, maka pengguna dinyatakan tidak sah (dengan asumsi semakin kecil skor, kedua data yang dibandingkan semakin mirip).

Pengujian pada penelitian ini dilakukan dengan jumlah pengguna 35 orang yang terbagi menjadi 210 sampel acuan dan 35 sampel uji sehingga total pencocokan yang dilakukan adalah 7350. Enam sampel dari masing-masing pengguna akan dijadikan sebagai sampel acuan atau reference dan satu sampel untuk pengujian.

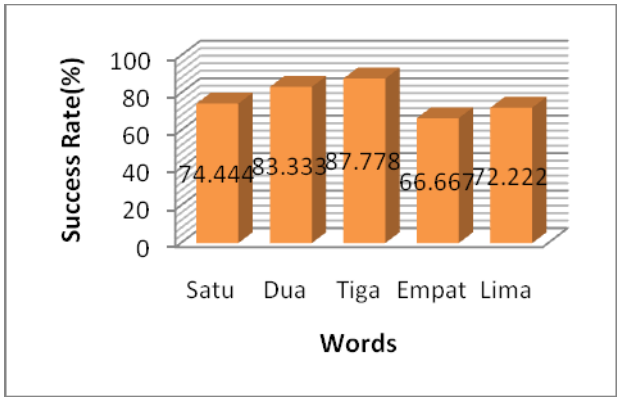
Ada beberapa pengujian yang dilakukan dalam penelitian ini, diantaranya adalah :

- Pengujian terhadap suku kata yang diucapkan(satu, dua, tiga, empat dan lima)

- Pengujian terhadap jumlah sampel acuan yang digunakan (1, 3 dan 6 sampel acuan)
- Pengujian terhadap jumlah pengguna (10,20 dan 35 pengguna)
- Pengujian terhadap jumlah koefisien MFCC yang digunakan (11, 15, 19 dan 23 koefisien MFCC)
- Pengujian terhadap panjang frame (N) dan panjang pergeseran frame (M) yang digunakan (N=20, M=10 dan N=30, M=15)

5.1. Analisa Hasil Pengujian Terhadap Suku Kata yang Diucapkan

Hasil pengujian sistem verifikasi suara terhadap suku kata yang diucapkan dapat ditampilkan dalam bentuk grafik perbandingan akurasi berikut ini :



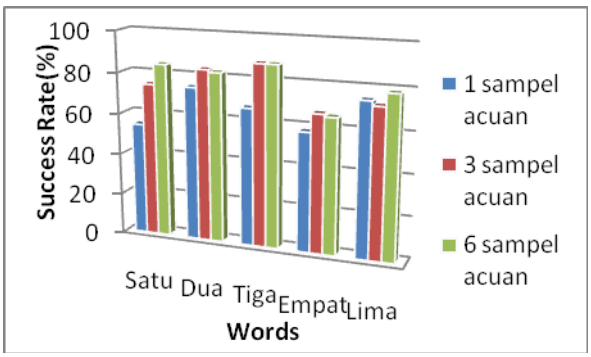
Gambar 7 Grafik perbandingan akurasi sistem berdasarkan suku kata yang diucapkan

Hasil pengujian menggunakan 10 orang pengguna, 30 sampel acuan, 10 sampel uji dan 15 koefisien MFCC didapatkan nilai akurasi tertinggi sebesar 87.778 % pada pengucapan kata ‘tiga’. Nilai akurasi paling rendah adalah 66.667 % pada kata ‘empat’. Rata-rata akurasi yang diperoleh adalah 76.888 %.

Tingkat keberhasilan sistem dalam melakukan verifikasi terhadap pengguna dapat dikatakan merata yaitu dari 66.667 % sampai 87.778 % dengan kata lain tidak terdapat hasil yang terlalu rendah. Dalam pengujian ini sistem dapat dikatakan berhasil dalam melakukan verifikasi terhadap pengguna.

5.2. Analisa Hasil Pengujian Terhadap Jumlah Sampel Acuan

Hasil pengujian sistem verifikasi suara terhadap jumlah sampel acuan yang digunakan dapat ditampilkan dalam bentuk grafik perbandingan akurasi berikut ini :

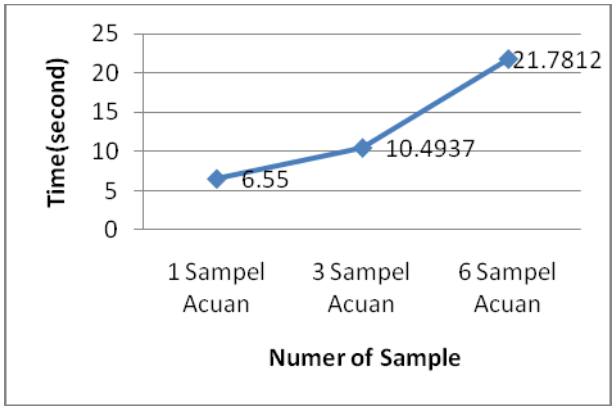


Gambar 8 Grafik perbandingan akurasi sistem berdasarkan jumlah sampel acuan

Rata-rata akurasi sistem saat menggunakan 1 sampel acuan adalah 65.555 %, 76.888 % saat menggunakan 3 buah sampel acuan dan 78.444 % saat ditambahkan 3 sampel acuan lagi. Rata-rata akurasinya meningkat seiring dengan penambahan sampel acuan yang dilakukan. Namun ada juga yang mengalami sedikit penurunan seperti terlihat pada grafik diatas yaitu pada kata ‘dua’, ‘empat’, dan ‘lima’. Hal tersebut hanya terjadi pada beberapa pengguna saja karena perekaman dilakukan pada lingkungan yang dipengaruhi oleh *noise*.

Melalui grafik diatas dapat ditarik suatu kesimpulan yaitu : semakin banyak suara yang ditrainingkan oleh pengguna maka semakin meningkat pula kemampuan sistem dalam melakukan pengenalan terhadap pengguna. Namun semakin banyak sampel yang ditrainingkan (sampel acuan) maka semakin lama juga waktu yang diperlukan untuk melakukan pengenalan.

Berikut ini adalah Grafik Pengaruh Jumlah Sampel Acuan Terhadap Waktu :

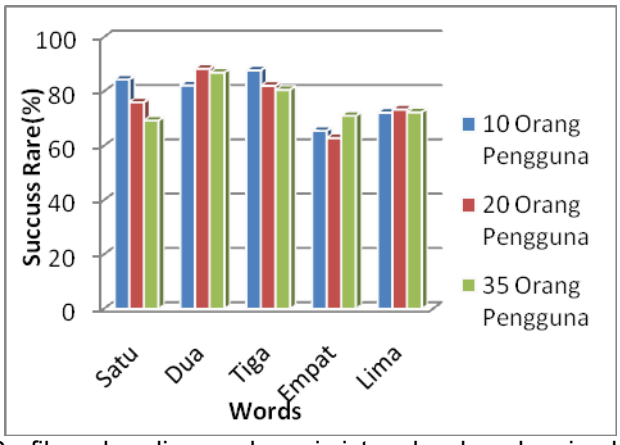


Gambar 9 Grafik pengaruh jumlah sampel acuan terhadap waktu

Melalui grafik diatas dapat ditarik satu kesimpulan yaitu semakin banyak sampel acuan yang dipakai maka semakin meningkat waktu yang diperlukan untuk pemrosesan.

5.3. Analisa Hasil Pengujian Terhadap Jumlah Pengguna

Hasil pengujian sistem verifikasi suara terhadap jumlah pengguna yang terdapat dalam basis data dapat ditampilkan dalam bentuk grafik perbandingan akurasi berikut ini :

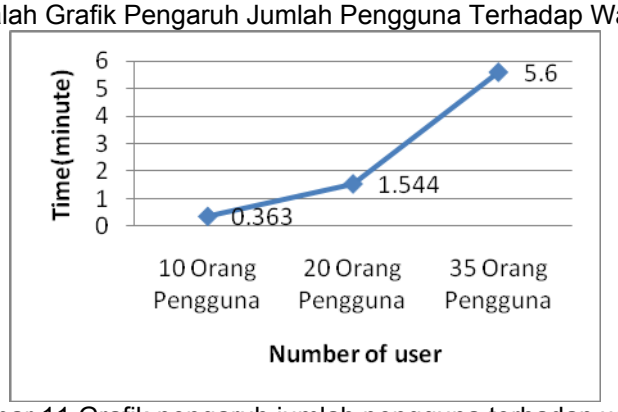


Gambar 10 Grafik perbandingan akurasi sistem berdasarkan jumlah pengguna

Rata-rata akurasi yang diperoleh ketika digunakan 10 orang pengguna sebesar 78.444 %. Setelah ditambahkan 10 pengguna rata-rata akurasi menjadi 76.579 %. Akurasinya berkurang sebesar 1.865 %, kemudia ditambahkan 15 pengguna lagi sehingga totalnya menjadi 35 orang pengguna, rata-rata akurasi yang diperoleh adalah sebesar 76.067 %.

Rata-rata akurasi yang diperoleh relatif sama ketika jumlah pengguna ditambahkan, terdapat sedikit penurunan akurasi pada beberapa kata yang diujikan, namum ada juga yang meningkat seperti terlihat pada grafik diatas. Hal tersebut wajar karena semakin banyak pengguna maka semakin banyak juga pencocokan yang dilakukan oleh sistem. Sehingga semakin banyak pula kemungkinan kesalahan sistem dalam melakukan pengenalan. Disamping itu kualitas dari sampel suara yang diujikan juga tidak sama (pengaruh *noise* dari lingkungan) karena proses perekaman tidak dilakukan pada satu tempat yang sama.

Berikut ini adalah Grafik Pengaruh Jumlah Pengguna Terhadap Waktu :

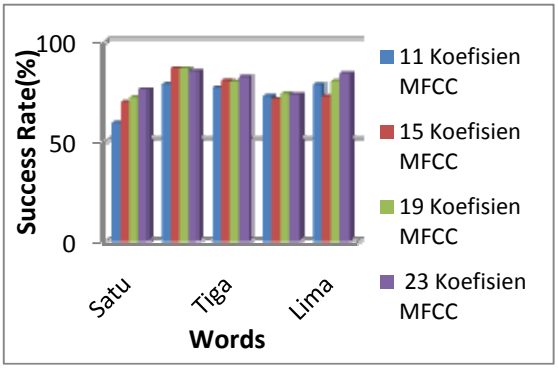


Gambar 11 Grafik pengaruh jumlah pengguna terhadap waktu

Melalui grafik diatas dapat dilihat bahwa semakin banyak pengguna yang terdaftar dalam basis data, maka semakin lama waktu proses yang diperlukan. Hal tersebut terjadi karena saat pengujian, sistem melakukan perbandingan 1 : N, dimana setiap sampel uji dibandingkan dengan seluruh sampel acuan yang ada dalam basis data. Namun dalam penggunaannya, sistem verifikasi ini melakukan perbandingan 1 : 1, dimana sistem hanya akan melakukan perbandingan terhadap ID yang diklaim oleh user saja. Sehingga penambahan jumlah pengguna tidak akan berpengaruh terhadap waktu pemrosesan yang diperlukan oleh sistem.

5.4. Analisa Hasil Pengujian Terhadap Jumlah Koefisien MFCC

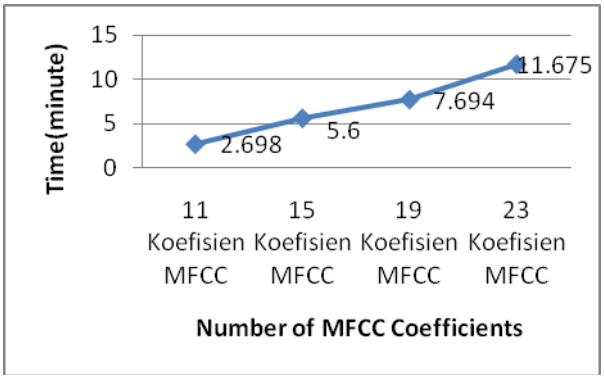
Hasil pengujian sistem verifikasi suara terhadap jumlah koefisien MFCC yang digunakan dapat ditampilkan dalam bentuk grafik perbandingan akurasi berikut ini :



Gambar 12 Grafik perbandingan akurasi sistem berdasarkan jumlah koefisien MFCC

Rata-rata akurasi yang diperoleh dengan pengujian menggunakan 11, 15, 19, dan 23 koefisien MFCC secara berurutan adalah sebesar : 73.260 %, 76.067 %, 78.6052, 80.3864 %. Dari hasil tersebut dapat disimpulkan yaitu semakin besar jumlah koefisien MFCC yang digunakan maka semakin baik kemampuan sistem dalam melakukan pengenalan terhadap pengguna begitu juga sebaliknya, semakin kecil jumlah koefisien MFCC yang digunakan maka semakin kecil tingkat akurasi sistem dalam melakukan pengenalan. Namun, semakin banyak jumlah koefisien MFCC yang digunakan, maka waktu yang diperlukan dalam proses pengenalan juga semakin lama, begitu juga sebaliknya.

Berikut ini adalah Grafik Pengaruh Jumlah Koefisien MFCC Terhadap Waktu :

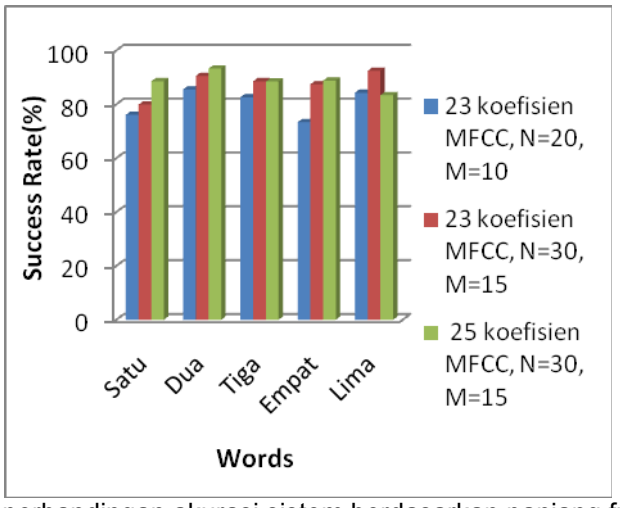


Gambar 13 Grafik pengaruh jumlah kofisien MFCC terhadap waktu

Sumbu y pada grafik diatas menyatakan waktu (menit) dan sumbu x menyatakan jumlah koefisien MFCC yang digunakan. Berdasarkan grafik tersebut dapat disimpulkan bahwa semakin besar jumlah koefisien MFCC yang digunakan, maka semakin besar juga waktu yang diperlukan untuk melakukan pemrosesan. Peningkatan jumlah koefisien MFCC menyebabkan semakin banyak pula perhitungan dan *looping* yang dilakukan oleh sistem sehingga meningkatkan waktu pemrosesan.

5.5. Analisa Hasil Pengujian Terhadap Panjang Frame (N) dan Panjang pergeseran frame(M)

Hasil pengujian sistem verifikasi suara terhadap panjang frame dan panjang pergeserannya dapat ditampilkan dalam bentuk grafik perbandingan akurasi berikut ini :

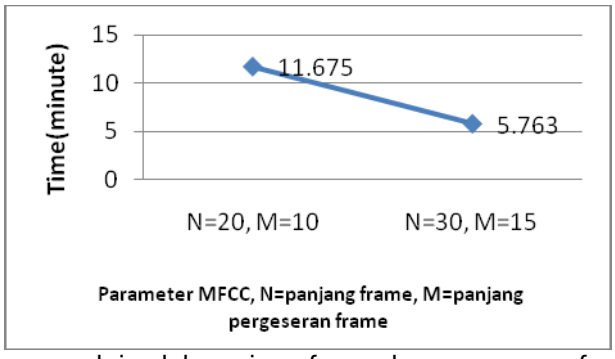


Gambar 14 Grafik perbandingan akurasi sistem berdasarkan panjang frame dan panjang pergeseran frame

Hasil pengujian dengan N=20, M=10 dan 23 koefisien MFCC didapatkan rata-rata akurasi sebesar 80.3864 %, setelah panjang frame dan pergeserannya dirubah menjadi N=30, M=15 dan koefisien MFCC tetap 23 didapatkan rata-rata akurasi sebesar 87.7946 %, meningkat sebesar 7.4082 %. Untuk pengujian terakhir penulis mencoba menambah jumlah koefisien MFCC menjadi 25 koefisien sedangkan parameter yang lain tetap sama dan didapatkan rata-rata akurasi sebesar 88.508 %.

Berdasarkan hasil pengujian tersebut diketahui bahwa dengan menggunakan N=30 ms dan M=15 ms kinerja sistem verifikasi lebih baik dibandingkan saat menggunakan N=20 ms dan M=10 ms dimana keduanya menggunakan frekuensi sampling sebesar 12800 Hz.

Berikut ini adalah Grafik Pengaruh Panjang *Frame* dan Panjang Pergeseran *Frame* Terhadap Waktu :



Gambar 15 Grafik pengaruh jumlah panjang *frame* dan pergeseran *frame* terhadap waktu

Grafik diatas menunjukkan bahwa semakin besar panjang *frame* yang digunakan, maka semakin kecil waktu pemrosesan yang diperlukan oleh sistem. Hal tersebut disebabkan oleh semakin sedikitnya proses perhitungan dan *looping* yang dilakukan oleh sistem.

6. PENUTUP

6.1. Kesimpulan

Berdasarkan uraian pembahasan dan analisa hasil dapat disimpulkan beberapa hal sebagai berikut:

1. Metode *Mel Frequency Cepstrums Coefficients* adalah metode yang baik untuk ekstraksi fitur dalam pengenalan suara.
2. Semakin banyak *training* yang dilakukan oleh setiap pengguna, semakin baik pula kemampuan sistem dalam melakukan pengenalan.
3. Metode *Dynamic Time Warping* dapat digunakan untuk membandingkan dua buah fitur suara hasil dari proses MFCC.
4. Nilai-nilai parameter MFCC yang digunakan sangat mempengaruhi baik buruknya hasil dari proses MFCC itu sendiri, sehingga berpengaruh terhadap tingkat kesuksesan saat pencocokan.
5. Hal-hal yang dapat mempengaruhi baik buruknya kinerja sistem verifikasi suara yang dibuat adalah panjang frame(N), panjang pergeseran frame(M), jumlah koefisien filterbank dan jumlah koefisien MFCC.
6. Pada penelitian ini, hasil terbaik yang diberikan oleh sistem adalah pada saat digunakan nilai-nilai parameter MFCC sebagai berikut : N=30 ms, M=15 ms, 33 koefisien filterbank dan 25 koefisien MFCC. Pengujian dilakukan terhadap kata satu, dua, tiga, empat, lima dengan 36 orang pengguna, 6 buah sampel acuan dan 1 buah sampel uji untuk masing-masing kata, diperoleh rata-rata akurasi sebesar 88.508 %.
7. Sisterm verifikasi suara memperlihatkan hasil yang buruk saat nilai-nilai parameter MFCC yang digunakan adalah N=20 ms, M=10 ms, 23 koefisien filterbank, 11 koefisien MFCC dilakukan terhadap 35 orang pengguna, 210 sampel acuan, 35 sampel uji terhadap kata satu, dua, tiga, empat dan lima didapatkan rata-rata akurasi sebesar 73.260 %.

7. DAFTAR PUSTAKA

[1] Campbell, J. 1997. *Speaker Recognition : A Tutorial.*___. IEEE.
[2] Darma Putra. 2009. Sistem Biometrika. Konsep Dasar, Teknik Analisis Citra, dan Tahapan Membangun Aplikasi Sistem Biometrika. Yogyakarta : Andi.
[3] Goananta Wangsa, Anak Agung Gede. 2008. Tugas Akhir: Sistem Identifikasi Telapak Tangan Dengan Menggunakan Metode Alihragam Fourier. Bukit Jimbaran: Universitas Udayana.
[4] Hartanto, B. 2008. Memahami Visual C#.Net Secara Mudah. Yogyakarta : Andi.
[5] Kartikasari,YE. 2006. Pembuatan Software Pembuka Program Aplikasi Komputer Berbasis Pengenalan Suara. Surabaya. Politeknik Elektronika Negeri Surabaya.
[6] Manunggal, HS. 2005. Perancangan dan Pembuatan Perangkat Lunak Pengenalan Suara Pembicara dengan Menggunakan Analisa MFCC *Feature Extraction*. Surabaya : Universitas Kristen Petra.

[7] Morton, Jeff. 2009.http://www.codeproject.com/KB/audiovideo/SoundCatcher/ SoundCatcher_Source.zip. Akses tanggal : 20 April 2009.
[8] Shannon, BJ., Paliwal,KK. 2003. A Comparative Study of Filter Bank Spacing for Speech Recognition. ___. Microelectronic Engineering Research Conference.
[9] Sitanggang, D., Sumardi., Hidayatno, A. 2002. Pengenalan Vokal Bahasa Indonesia Dengan Jaringan Syaraf Tiruan Melalui Transformasi Fourier. Semarang. Jurusan Teknik Elektro Undip.
[10] Syah, DPA. 2009. Sistem Biometriks Absensi Karyawan Dalam Menunjang Efektifitas Kinerja Perusahaan. http://donupermana.wordpress.com/ makalah/sistem-biometrik-absensi/. Akses tanggal : 23 Pebruari 2010.
[11] Xafopoulos, A. 2001. Speaker Verification(an overview). Greece. Aristotle University Of Thessaloniki.
[12] ___. 2009. About EER. http://www.bioid.com/sdk /docs/About_EER.htm. Akses tanggal : 15 Febuari 2009.
[13] ___. 2009. http://msdn.microsoft.com/en- us/library/aa446573(loband).aspx#wavei nout_topic_004/. Akses tanggal : 05 Desember 2009.