IMPLEMENTASI *SPEAKER RECOGNITION* UNTUK OTENTIKASI MENGGUNAKAN *MODIFIED MFCC – VECTOR QUANTIZATION* ALGORITMA LBG

SPEAKER RECOGNITION IMPLEMENTATION FOR AUTHENTICATION USING MODIFIED MFCC – VECTOR QUANTIZATION LBG ALGORITHM

Reza Aulia Sadewa [1], Tjokorda Agung Budi W [2], Siti Sa'adah [3]
[1.2.3] *Fakultas Informatika, School of Computing, Universitas Telkom*
Jalan Telekomunikasi No.1, Dayeuh Kolot, Bandung 40257
rezaauliasadewa@gmail.com[1], cokagung@telkomuniversity.ac.id[2], tisataz@gmail.com[3]

Abstraksi

Penelitian ini membahas mekanisme otentikasi menggunakan komponen biometrik manusia yang bersifat unik, yaitu sinyal bicara sebagai alternatif dari mekanisme yang lain, seperti kata sandi dan kode PIN. Pertama, fitur atau karakteristik suara direpresentasikan dengan sejumlah koefisien hasil MFCC. Selanjutnya, model suara, yang disebut dengna *codebook* dibentuk menggunakan VQ. Metode ini dimodifikasi dengan mekanisme *thresholding* untuk menolak otentikasi suara aisng atau suara yang *codebook*-nya belum terlatih dan filter Butterworth untuk menangani *noise* pada suara. Pengujian dilakukan dengan menggunakan data sintetik dan biometrik yang masing – masing terdiri dari 10 pembicara. Setengah dari jumlah pembicara dijadikan sebagai pembicara asing. Secara keseluruhan, metode – metode yang dipakai sudah cukup baik sebagai mekanisme otentikasi. Berdasarkan hasil pengujian yang dilakukan, MFCC dan VQ dapat 100 % membedakan suara antar pembicara namun tetap mengotentikasi pembicara asing, yang seharusnya ditolak, sebagai pembicara terlatih yang paling mirip. Dibandingkan dengan suara yang diberi *noise*, suara yang difilter dengan parameter tertentu dapat menghasilkan akurasi yang lebih besar. Metode *thresholding* yang digunakan menghasilkan *true rejection* sekitar 90 % namun menghasilkan *true acceptance* sebesar 70 %. Penerapan toleransi threshold dapat meningkatkan persentase *true acceptance*, namun mekanisme ini harus diteliti lebih lanjut untuk mendapatkan keseimbangan dan optimasi dari hasil *true acceptance* dan *true rejection*.

Kata kunci: *MFCC, VQ,* filter butterworth, *threshold*

Abstract

This research explains about authentication mechanism using human's biometric component, voice. First, the characteristic of the voice is extracted using MFCC then represented by cepstrum coefficients. Those features forms a model by the VQ method. These methods is modified with a proposed thresholding method to reject the unknown voice and a Butterworth Filter to handle the noise. For the experiment, both synthetic and real human voice or biometric data with 10 speaker each is used. Half of the speaker is separated as the unregistered or the untrained voices. Overall, the result shows that the methods is adequate enough to perform a security mechanism. MFCC and VQ combination can truly 100% distinguish the speakers but also authenticate the unregistered speakers, which should be rejected, as the most similar registered speaker. Compared to the noise-added data, the noise-filtered data can increase the acceptance accuracy with a specific filter parameters. The thresholding method produce approximately 90% true rejection but produces only around 70% true acceptance. Hence, the value of the threshold tolerance, which is to increase the authentication accuracy for the registered speaker, needs to be treated more for the next experiment to find the balance between the acceptance and the rejection accuracy.

Keywords: *MFCC, VQ, butterworth filter, threshold*

1. Pendahuluan

Suara manusia bersifat unik [9]. Dalam penerapan *speaker recognition* sebagai mekanisme otentikasi pada sebuah perangkat, masalah yang terjadi adalah bagaimana sebuah sistem dengan *speaker recognition* benar – benar dapat membedakan suara baik pengguna yang satu dengan yang lain maupun menolak otentikasi pengguna yang asing atau pengguna yang sebelumnya tidak terdaftar. Selain itu, sistem tersebut harus memiliki mekanisme tersendiri untuk menangani jenis intervensi terhadap suara, seperti *noise* dan rentang *silence*, sesuatu yang tidak bisa dihindari jika sistem sudah diterapkan pada kehidupan sehari – hari.

Dalam membedakan suara yang satu dengan yang lain, sinyal suara harus diubah terlebih dahulu ke model matematis sehingga karakteristik atau fiturnya dapat terukur dengan jelas dan karakeristik antara sinyal - sinyal suara dapat dibandingkan kemiripannya.

Tujuan dari penelitian ini adalah membangun sebuah sistem otentikasi pada sebuah perangkat yang dapat membedakan suara manusia yang satu dengan yang lain dan bisa digunakan pada keadaaan baik yang bersifat *controlled – enviroment* maupun yang tidak. Selain itu, sistem tersebut dapat mengetahui suara orang – orang yang tidak dikenal atau yang sebelumnya belum dilatih.

2. Dasar Teori

2.1 *Speaker Recognition*

Secara garis besar, tahap dari *speaker recognition* adalah tahap pelatihan dan tahap pengujian. Tahap pelatihan adalah tahap di mana karakteristik atau fitur suara sejumlah orang tertentu disimpan dalam sistem sebagai model. Pada tahap pengujian, suara yang tidak diketahui atau belum terlatih yang diterima oleh sistem, dibandingkan dengan model yang sebelumnya sudah disimpan. Selanjutnya sistem akan menyatakan kecocokannya dengan salah satu suara orang yang tersimpan dalam bentuk model berdasarkan tingkat kemiripan fiturnya. Dari segi ruang sampel, speaker recognition dibagi menjadi 2 yaitu, identifikasi dan verifikasi. Identifikasi adalah membandingkan atau mencari kemiripan suara yang masuk dengan sekumpulan model suara pembicara, sedangkan verifikasi hanya terdapat satu model pembicara yang disimpan di dalam sistem, sehingga hasil otentikasinya bersifat biner.

2.2 Filter Butterworth

Konsepnya, filter ini dapat menghaluskan dengan meratakan sampai ke titik di mana sinyal suara hampir tidak memiliki riak atau rata pada batas frekuensi tertentu [7]. Jenis filter Butterworth yang biasa digunakan untuk menghilangkan *noise* adalah *low – pass* atau mengaplikasikan filter di atas frekuensi yang diinginkan [8]. Penerapan filter ini dilakukan untuk setiap nilai amplitudo pada sinyal dalam domain waktu [1].

Hal ini dirumuskan pada persamaan (1)

$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] - b_1 y[n-1] - b_2 y[n-2] \quad (1)$$

Di mana $y[n]$ adalah hasil nilai amplitudo ke – n setelah dilakukan pengaplikasian filter pada nilai amplitudo sebelum proses filter, $x[n]$, dengan yang ber - *order* 2. Koefisien dan didapatkan dari nilai riiil dan nilai imajiner dari filter Butterworth, yang dirumuskan pada persamaan (2)

$$\frac{1 - r^2}{\quad} \quad (2)$$

$$= \frac{}{(1-r)^2 + r^2} \text{ dan } = \frac{2r}{(1-r)^2 + r^2}$$

Di mana dan dirumuskan pada persamaan (3)

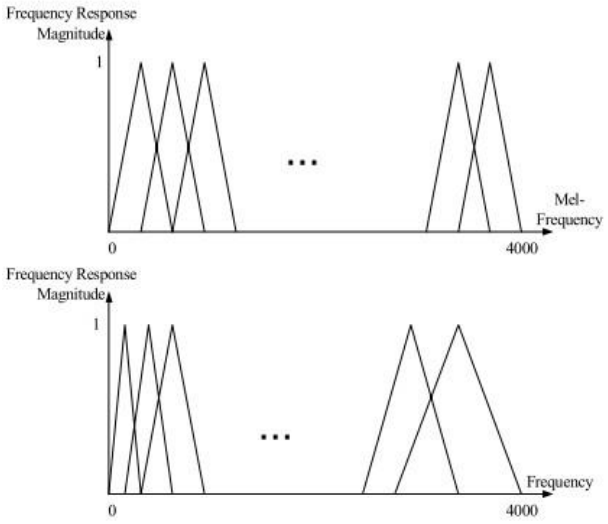
$$= \tan\frac{}{2}\cos\frac{2r+1}{2r}\pi \text{ dan } \quad (3)$$

$$= \tan\frac{}{2}\sin\frac{2r+1}{2r}\pi$$

Di mana ◆?◆= 0, 1, 2, . . , 2◆?◆− 1 ; ◆?◆ adalah *order* , ◆??◆ adalah frekuensi. Order akan mempengaruhi seberapa besar sensitivitas filter terhadap frekuensi *cut-off*.

2.3 *Mel Freqiency Cepstral Coefficient (MFCC)*

MFCC adalah representasi fitur sinyal suara dalam bentuk angka. Angka ini didapatkan dari pengaplikasian sejumlah filter berbentuk segitiga pada sinyal suara yang dipetakan pada domain frekuensi *mel*. Domain ini mengikuti persepsi pendengaran manusia, yaitu skala linear pada frekuensi di bawah 1000 Hz dan logaritmik untuk frekuensi yang lebih besar [4]. Perbedaan domain frekuensi biasa dan frekuensi *mel* beserta filternya terlihat pada gambar 1. di bawah ini.



Gambar .. Filter pada domain frekuensi mel dan frekuensi [3]

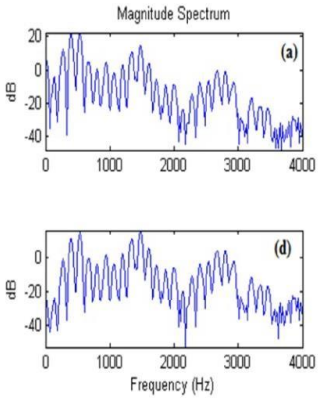
Sinyal suara pada domain waktu terdiri dari banyak sampel atau nilai amplitudo (*db*) dengan durasi (*ms*) tertentu tertentu. Nilai – nilai ini merupakan dasar dari pemrosesan sinyal suara. Jumlah sampel pada tiap satuan waktu dipengaruhi kualitas *sampling rate* saat suara tersebut direkam. Terdapat beberapa tahap MFCC, yaitu :

1. *Pre – emphasis*

Tahap ini digunakan untuk meratakan atau menstabilkan nilai – nilai *magnitude*, yaitu bentuk skalar dari *amplitude* [5] dengan konstanta *pre – emphasis* tertentu. *Pre – emphasis* dirumuskan pada persamaan (4)

$$y_n = x_n - \alpha x_{n-1} \quad (4)$$

Di mana x_n adalah nilai sampel ke – *n* dan α adalah konstanta *pre – emphasis*. Dampak dari *pre – emphasis* diilustrasikan pada gambar 2.2



Dari gambar 2. terlihat bahwa dengan dilakukannya *pre – emphasis* sinyal suara lebih bersifat "lurus".

2. *Framing*

Sinyal suara dipecah menjadi sejumlah *frame* dengan panjang tertentu. Panjang setiap *frame* harus cukup singkat, yaitu sekitar 20 sampai 40 ms, karena sinyal suara bersifat stabil pada rentang waktu yang singkat [6]. Selain itu, *frame* yang satu harus saling mendahuli atau *overlapping* dengan *frame* sebelunya untuk menjamin kestabilan sinyal [3].

Untuk tahap selanjutnya, proses dilakukan untuk setiap *frame*.

3. *Windowing*

Setelah proses *framing*, terjadi diskontinuitas berupa *noise* pada frekuensi besar di kedua ujung nilai tiap *frame* [3]. Hal ini dapat diperhalus dengan penerapan fungsi *Hamming window* (5) di bawah ini

$$= 0{,}54 - 0{,}46 \cos\left(2\pi \frac{n}{N-1}\right) \quad (5)$$

Di mana N adalah jumlah sampel tiap *frame* sedangkan $n = 0, 1, 2, \ldots, (N-1)$

4. *Fast Fourier Transform* (FFT)

FFT adalah sebuah algoritma yang digunakan untuk melakukan *Discrete Fourier Transform* (DFT) [2]. DFT dirumuskan pada persamaan (6)

$$F_n \equiv \sum_{n=0}^{N-1} x_n e^{\frac{2\pi i n(n)}{N}} \quad (6)$$

Gambar 2. Dampak pre-emphasis [5]

Di mana X_k koefisien hasil dari DFT ke k, x_n adalah sampel sinyal ke n dalam domain waktu, N adalah jumlah sampel, i adalah komponen bilangan kompleks. Bilangan eksponen e beserta pangkatnya membentuk suatu bilangan kompleks, yang terdiri dari komponen riil dan imajiner. Komponen riil merepresentasikan *magnitude*, sedangkan komponen imajiner merepresentasikan fase.

Secara keseluruhan, proses ini mengubah sinyal suara pada domain waktu menjadi domain frekuensi pada rentang yang telah

ditentukan.Rentang frekuensi maksimum didapatkan dari setengah nilai *sampling rate*.

5. *Mel frequency filter bank*

Pada tahap ini, sinyual suara pada domain frekuensi dikonversi menjadi domain frekuensi *mel* yang dirumuskan pada persamaan (7)

$$\text{frekuensi}_{mel} = 2595 \ln\left(1 + \frac{f}{700}\right) \quad (7)$$

Di mana frekuensi$_{mel}$ adalah nilai frekuensi *mel* dari frekuensi f.

Hasil akhir dari tahap ini adalah didapatkannya sejumlah *mel filter bank*. Nilai – nilai pada *mel filter bank* menunjukkan seberapa besar energi pada rentang frekuensi yang ada pada masing - masing filter *mel*.

6. Transformasi non linear

Tahap ini adalah mengambil nilai logaritma natural dari setiap *mel frequency filter bank*, dengan persamaan (8)

$$f_n = \ln f_n \quad (8)$$

Di mana $f$adalah *mel frequency filter bank*, $f = 1,2,..,K$; sedangkan K adalah jumlah *mel frequency filter bank* pada setiap *frame*.

7. *Discrete cosine transform* (DCT)

Proses ini mengembalikan sinyal domain frekuensi kembali menjadi domain waktu sehingga didapatkannya koefisien *cepstrum*. Persamaannya terdapat pada (9)

$$c_i = \sum_{n=1}^{K} (f_n) \cos\left[i\left(n - 0{,}5\right)\frac{\pi}{K}\right] \quad (9)$$

Di mana K adalah jumlah *mel frequency filter bank*, f_n berasal dari persamaan (8), $i = 1,2,..,I$ sehingga didapat I buah koefisien *cepstrum*. Koefisien pertama diabaikan karena nilai ini tidak representatif [10].

2.4 *Vector Quantization (VQ)*

Konsep dasar dari VQ adalah pengurangan jumlah vektor pada ruang dimensi tertentu yang

Kumpulan *centroid* dari masing – masing *cluster* dinamakan *codebook*. Proses pengelompokkan vektor terhadap *centroid* pada suatu *cluster* yang terdekat dilakukan dengan algoritma LBG [Linde, Buzo, dan Gray, 1980].

Pengurangan jumlah vektor ini akan mempercepat proses saat verifikasi, di mana vektor – vektor data uji tidak akan dibandingkan dengan semua vektor data latih yang jumlahnya sangat banyak, melainkan hanya dengan *centroid*-nya saja yang jumlahnya lebih sedikit [4]. Parameter yang mewakili kemiripan sejumlah vektor data uji dengan suatu *codebook* adalah semakin kecilnya tingkat distorsi, yang dirumuskan pada persamaan (10)

$$D_T(X,C) = \sum_{i=1}^{T} \min d(x_i, c_j) ; 1 \le j \le K \quad (10)$$

Di mana D_T adalah nilai distorsi data uji X yang terdiri dari T buah vektor terhadap *codebook* C yang terdiri dari K buah *centroid*, d adalah jarak *Euclidean*, yang terdapat pada persamaan (11)

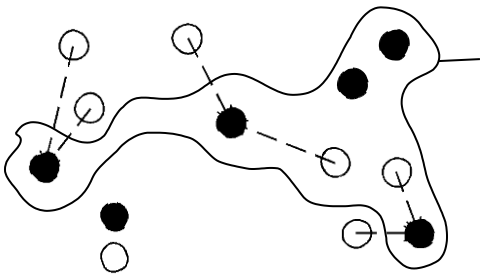
$$d(x,c) = d(x,c) \quad (11)$$

$$= \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \cdots + (x_n - c_n)^2}$$

Di mana $x$dan $c$adalah vektor yang masing – masing memiliki koordinat $x_{1..n}$dan $c_{1..n}$

Dengan kata lain, distorsi suatu *codebook* terhadap data uji adalah jumlah semua jarak *Euclidean* antara vektor – vektor data uji terhadap *centroid* yang terdekat pada suatu *codebook*. Konsep ini diperjelas melalui gambar 2.4

codebook direpresentasikan dengan *centroid* pada setiap *cluster*. Koordinat dari *centroid* didapatkan dengan metode *k – means*, yaitu pengambilan rata – rata koordinat untuk semua vektor yang berada pada *cluster*.



: *centroid*

: vektor data uji

Gambar 3. Distorsi data uji terhadap codebook

Di mana distorsi adalah jumlah dari seluruh jarak yang direpresentasikan dengan garis putus – putus.

Tahap dari algoritma VQ-LBG [4] adalah :

1. Inisiasi satu buah *centroid* dengan koordinat didapatkan dari rata – rata dari semua vektor
2. Naikkan jumlah centroid menjadi dua kali lipat dengan aturan (2-13)

$$\begin{aligned} c_n^+ &= c_n(1+\varepsilon) \\ c_n^- &= c_n(1-\varepsilon) \end{aligned} \quad (12)$$

Di mana c_n adalah centroid yang berjumlah n pada iterasi tertentu dan ε adalah konstanta *split factor* yang bernilai 0,01.

3. Vektor – vektor dikelompokkan berdasarkan jarak *Euclidean* terdekat terhadap centroid yang bersangkutan
4. Ulangi tahap 3 dan 4 sampai memenuhi aturan (13)

$$\frac{\sum_{m=1}^{M} D_m' - \sum_{m=1}^{M} D_m}{\sum_{m=1}^{M} D_m'} > 0{,}1 \quad (13)$$

Di mana D_m' adalah "distorsi lokal" *cluster* setelah tahap ke – 4 , D_m adalah distorsi *cluster* sebelumtahap ke – 4 , dan M adalah jumlah cluster.

5. Kembali ke tahap 2 dan proses selanjutnya diulangi sampai centroid mencapai jumlah yang ditentukan.

Karena konsep dari LBG adalah melipat gandakan jumlah *centroid* menjadi dua kali lipat setiap iterasi, maka jumlah *centroid* yang ditentukan harus bilangan pangkat 2.

2.5 Penaganan Data yang tidak dikenal

Dengan metode VQ, data uji yang memiliki karakteristik di luar data latih akan tetap terotentikasi sebagai data yang memiliki tingkat distorsi yang terkecil dengan *codebook* tertentu. Olehkarena itu, *threshold* berupa nilai distorsi maksimum dan minimum masing – masing *codebook* harus dilatih dengan menggunakan data – data di luar data yang digunakan untuk membentuk *codebook* namun masih pembicara yang sama. Pada saat verifikasi penanganan ini dilakukan dengan aturan (14)

$$(D_{min} \le D_n \le D_{max} \rightarrow \text{identitas pembicara valid} \quad (14)$$

Dengan hanya menggunakan nilai distorsi minimum dan maksimum, terdapat kelemahan, yaitu nilainya tetap statis. Olehkarena itu, skenario pendekatan *threshold* yang kedua adalah menambah toleransi *threshold* tersebut dengan penambahan jarak atau *range threshold* dengan selisih nilai mimimum dan maksimumnya. Alternatif toleransi *threshold* yang kedua adalah nilai selisih yang digunakan dibagi konstanta tertentu.

2.6 Akurasi sistem

Akurasi sistem dihitung dengan persamaan (15)

$$A = \frac{B}{T} 100\% \quad (15)$$

Di mana A adalah akurasi sistem, B adalah jumlah "benar", sedangkan T adalah jumlah data yang diujikan. Dalam proses pengujian, hasil dinyatakan "benar" jika memenuhi kondisi :

1. *True acceptance*, keadaan di mana suara yang masuk terotentkasi sesuai dengan pembicara yang sebenarnya
2. *True rejection*, keadaan di mana suara asing atau suara yang *codebook* nya belum terlatih dapat ditolak oleh sistem

3. Rancangan Sistem

Sistem secara keseluruhan dijelaskan pada gambar 4.

Gambar 4. Rancangan Sistem

Di mana D_{min} adalah distorsi minimum, D_{max} adalah distorsi maksimum, sedangkan D_n adalah distorsi data uji dengan *codebook* yang bersangkutan.

Baik data latih maupun data uji, keduanya masuk ke tahap ekstraksi dan filter. Perbedaannya adalah data latih akan dibentuk terlebih dahulu menjadi *codebook* dan dilakukan penentuan nilai *threshold*-

nya masing – masing sebelum dibandingkan dengan data uji.

4. Hasil Pengujian

Pengujian menggunakan 2 jenis data suara, yaitu data sintetik dan biometrik yang masing –masing terdiri dari 10 pembicara. Setengah dari jumlah pembicara dijadikan sebagai pembicara asing. Data sintetik adalah data yang dibentuk dengan menggunakan aplikasi text – to – speech pada www.acapela-group.com dan diperbanyak dengan mengubah *pitch*, *formant*, dan sebagian *volume* menggunakan aplikasi REAPER buatan Cockos untuk setiap data. Data biometrik adalah suara asli manusia yang direkam menggunakan mikrofon.

Rangkuman dari beberapa skenario pengujian antara lain :

1. Untuk membedakan suara, MFCC – VQ dapat mengotentikasi seluruh suara terlatih. Namun, suara asing tetap terotentikasi dengan *codebook* yang paling mirip, sehingga terjadi *false positive*.
2. Mekanisme *threshold* yang digunakan dapat menolak otentikasi suara asing sekitar 95 % namun hanya menghasilkan sekitar 74 % *true acceptance* bagi suara yang terlatih. Hal ini dapat ditangani dengan menerapkan toleransi *threshold* dengan konstantanya sebesar 4. Hal ini menyebabkan kenaikan akurasi bagi suara terlatih menjadi sekitar 86 %, namun sedikit menurunkan akurasi *true rejection* sekitar 3 %.
3. Pengubahan parameter – parameter seperti jumlah *centroid* pada *codebook*, filter mel, dan koefisien cepstrum daopat memengaruhi hasil sistem namun tidak cukup signifikan.
4. Pengujian 1 pembicara dengan penambahan *noise* maupun rentang *silence* menebabkan setiap data sama sekali tidak ada yang terotentikasi
5. Selisih SNR paling kecil didapatkan saat data uji dan data latih, baik yang memiliki *noise* maupun tidak, keduanya tetap difilter
6. Hasil pengujian filter terbaik untuk 5 pembicara data sintetik yang diberi *noise* adalah menggunakan *order* = 10 dan *cut-off* = 1000 Hz, sehingga menghasilkan sekitar 73 % *true acceptance*.
7. Penerapan filter pada data biometrik mengurangi akurasi, karena baik data latih maupun data ujinya terdapat *noise* yang levelnya dapat berbeda – beda.

5. Penutup

5.1 Kesimpulan

Secara keseluruhan, combinasi dari MFCC dan VQ sudah cukup baik dalam membedakan suara antar pembicara. Akurasi sistem dapat berkurang jika data terdapat gangguan – gangguan seperti *noise* dan rentang *silence*. Selain itu, fitur MFCC terikat pada dua komponen, yaitu pembicara dan kata yang diucapkan. Penggunaan kkata yang berbeda dari model yang sudah dilatih akan mengurangi akurasi sitem.

Noise dapat ditangani dengan Filter Butterworth, namun penentuan parameter yang tidak tepat dapat merusak fitur – fitur hasil ekstraksi MFCC sehingga terjadi penurunan akurasi sistem.

Mekanisme *threshold* yang digunakan cukup efektif dalam menolak otentikasi suara asing namun kurang efektif dalam mengotentikasikan suara yang terlatih. Penerapan toleransi *threshold* dapat meningkatkan akurasi suara yang terlatih tetapi tetap dapat mengurangi hasil akurasi *true rejection*.

5.2 Saran

Untuk penelitian selanjutnya, terutama dengan metode yang sama, sebaiknya :

1. Jumlah pembicara diperbanyak
2. Dilakukan pengujian terhadap masing – masing kombinasi parameter
3. Ditambahkan mekanisme yang dapat menetukan parameter filter yang tepat secara otomatis sesuai dengan keadaan *noise* dari masing – masing sinyal suara
4. Karena dasar teori dalam perancangan mekanisme *thresholding* maupun penentuan konstanta toleransi *threshold* harus dilakukan penelitian lebih lanjut maupun pengujian dengan skenario yang lebih banyak untuk mendapatkan persentase *true acceptance* dan *true rejection* yang seimbang dan optimal.

Daftar Pustaka

[1] Alem, N., & Perry, M. (1995). *Design of Digital Low-pass Filters for Time-Domain Recursive Filtering of Impact Acceleration Signals.* Alabama.

[2] Fast Fourier Transform. (1992). In *NUMERICAL RECIPES IN FORTRAN 77: THE ART OF SCIENTIFIC COMPUTING* (pp. 490-502). Cambridge University Press.

[3] HAN, W., CHAN, C.-F., CHOY, C.-S., & PUN, K.-P. (2006). An Efficient MFCC Extraction Method in Speech Recognition. *IEEE*, 145-148.

[4] Kamale, H. E., & Kawitkar, R. S. (2008). Vector Quantization Approach for Speaker Recognition. *International Journal of Computer Technology and Electronics Engineering* , 110-114.

[5] Loweimi, E., Ahadi, S. M., Drugman, T., & Loveymi, S. (n.d.). On the importance of Pre-emphasis and Window Shape in Phase-based Speech Recognition.

[6] Mayrhofer, R., & Kaiser, T. (n.d.). Towards usable authentication on mobile phones: An evaluation of speaker and face recognition on off-the-shelf handsets.

[7] Parashar, A., & Ghosh, P. K. (n.d.). Speech Enhancement and Denoising Using Digital Filters. *International Journal of Engineering and Science Technology*, 1094-1103.

[8] Robertson, D. G., & Dowling, J. J. (2003). Design and responses of Butterworth and critically damped digital filters. *Journal of Electromyography*, 569-573.

[9] Trilok, N. P., Cha, S.-H., & Tappert, C. C. (2004). Establishing the Uniqueness of the Human Voice for Security Applications. *Proceedings of Student/Faculty Research Day, CSIS, Pace University*, (pp. 8.1-8.6).

[10] Zheng, F., Zhang, G., & Song, Z. (2001). Comparisson of different implementation of MFCC. *J. Computer Science & Technology*, 582-589.