SIMULASI DAN ANALISIS SPEAKER RECOGNITION MENGGUNAKAN METODE *MEL FREQUENCY CEPSTRUM COEFFICIENT* (MFCC) DAN *GAUSSIAN MIXTURE MODEL* (GMM)

SPEAKER RECOGNITION SIMULATION AND ANALYSIS USING MEL FREQUENCY CEPSTRUM COEFFICIENT (MFCC) DAN GAUSSIAN MIXTURE MODEL (GMM) METHOD

Doanda Khabi Putra [1], Iwan Iwut Triasmoro [2], Ratri Dwi Atmaja [3]
[1.2.3] *Fakultas Teknik Elektro, School of Electrical Engineering, Universitas Telkom*
Jalan Telekomunikasi No.1, Dayeuh Kolot, Bandung 40257
doanda.khabi.p@gmail.com[1], iwaniwut@telkomuniversity.ac.id[2], ratridwiatmaja@telkomuniversity.ac.id[3]

Abstraksi

Tugas akhir ini membahas mengenai pengenalan pembicara (*speaker recognition*), yaitu mekanisme pengenalan identitas subjek berdasarkan ciri suaranya. Pertama, sinyal suara subjek yang diuji diekstraksi cirinya menggunakan metode MFCC (*Mel Frequency Cepstrum Coefficient*). Tahapan di dalam MFCC termasuk diantaranya adalah *pre-emphasis*, *framing*, *windowing*, FFT (*Fast Fourier Transform*), *mel scaling* dan DCT (*Discrete Cosine Transform*), yang mana keluaran MFCC adalah *feature vector* yang dinamakan *cepstrum*. Selanjutnya, *cepstrum* dari masing-masing subjek akan dimodelkan menggunakan metode GMM (Gaussian Mixture Model). Tahapan di dalam GMM termasuk diantaranya adalah *Expectation-step* dan *Maximization-step*, yang mana keluaran GMM adalah distribusi *Gaussian* dengan parameter *mean* (μ) dan *variance* (σ^2) yang unik untuk masing-masing subjek. Proses klasifikasi dilakukan dengan membandingkan parameter distribusi *Gaussian* antara data latih dan data uji.

Pada penelitian internasional sebelumnya oleh kelompok mahasiswa di Preston University dan Jinnah Women University, Pakistan, dengan judul "*Speaker Identification Using GMM with MFCC*" diperoleh akurasi sebesar 87.5% dengan metode ekstraksi ciri MFCC, metode *clustering* K-Means, metode *modelling* GMM dan diklasifikasikan menggunakan *log probability*. Pada tugas akhir ini, kita akan melewatkan tahap *clustering* dan tahap klasifikasi dilakukan dengan melakukan perbandingan pada distribusi Gaussian memanfaatkan parameter *mean* (μ) dan *variance* (σ^2), dimana merupakan cara paling cepat dan mudah. Pada tugas akhir ini, diusahakan akurasi yang didapat mampu mendekati penelitian yang sudah ada mengingat tahap klasifikasi yang digunakan bisa dikatakan cara 'kasar' dalam penggunaan *Gaussian Mixture Model* (GMM) sehingga tidak bisa diekspektasikan lebih baik, meskipun banyak faktor lain yang bisa mempengaruhi akurasi simulasi.
Kata kunci: *Speaker recognition*, *Mel Frequency Cepstrum Coefficient* (MFCC), *Gaussian Mixture Model* (GMM), *Expectation Maximization* (EM)

Abstract

This essay discusses about speaker recognition, a system that recognize subject identity by their voice. First, subject's voice features are extracted using MFCC (Mel-Frequency Cepstrum Coefficients) method. Steps in the MFCC are pre-emphasis, framing, windowing, FFT (Fast Fourier Transform), mel scaling and DCT (Discrete Cosine Transform), which produce feature vector called cepstrums. These cepstrums are then modelled using GMM (Gaussian Mixture Model). Steps in th GMM are Expectation-step and Maximization-step, which produce gaussian distribution along with its parameters, mean (μ) and variance (σ^2) which are different for every subjects. Classification step is done by comparing between training data parameters and testing data parameters. If the comparation gets high score, it means two datas are match, vice versa.

Previous research done by student group at Preston University and Jinnah Women University, Pakistan, with title "Speaker Identification Using GMM with MFCC" gets accuration score 87,5% using feature extraction method MFCC, clustering method K-Means, modelling method GMM and classification by log probability. In this essay, we will pass the clustering step and classify by doing comparation between gaussian distribution using parameter mean (μ) and variance (σ^2), which is the fastest and easiest way. In this essay, we do the best thing to get as close as possible with previous research accuration score knowing that the classification step can be called the 'rough' one in GMM usage so we are not expecting high, even though there are so many factors that can influence the simulation accuracy.
Keywords: *MFCC, VQ, butterworth filter, threshold*

1. Pendahuluan

Biometrik adalah teknologi yang berfungsi mengenali subjek berdasarkan ciri biologisnya. Aspek biologis yang dimanfaatkan dalam biometrik antara lain wajah, sidik jari, iris mata, DNA dan suara. Keunggulan biometrik dalam hal otentifikasi dibandingkan dengan password karakter adalah kemampuannya membedakan satu individu dengan individu lain secara akurat, sulit diduplikasi, dan tidak mudah hilang. Saat ini perkembangan perangkat digital dan Internet semakin pesat, apalagi didukung dengan adanya gagasan mengenai *Internet of Things* (IoT) yang mana akan menyambungkan seluruh perangkat elektronik ke jaringan internet tentu masalah keamanan akan menjadi bahasan yang penting dan biometrik adalah salah satu solusi terbaik untuk keamanan perangkat digital. Melihat prospek biometrik yang cukup baik baik di masa depan, menurut penulis cukup baik untuk mengangkat topik ini dan yang akan dibahas pada tugas akhir ini adalah biometrik berbasis suara yaitu pengenalan pembicara (*speaker recognition*).

2. Dasar Teori

2.1. Speaker Recognition

Speaker recognition adalah usaha mengidentifikasi seseorang melalui karakteristik suaranya [7]. Secara alami otak kita juga mampu melakukan *speaker recognition* dimana kita mampu mengenali suara ayah, suara ibu, suara adik-kakak, maupun suara teman-teman kita tanpa perlu melihat wajah mereka. *Speaker Recognition* juga dapat diimplementasikan ke dalam bentuk perangkat lunak agar dapat mengidentifikasi suara secara digital.

Speaker Recognition pada umumnya terdiri dari tiga tahap, yaitu ekstraksi ciri, *modelling*, dan klasifikasi [1]. Pada ekstraksi ciri, tiap data suara diambil fitur cirinya dan dikeluarkan dalam bentuk vektor. Pada *modelling*, tiap vektor ciri dari hasil ekstraksi ciri dimodelkan dengan parameter tertentu sehingga memudahkan untuk klasifikasi. Pada tahap klasifikasi, hasil *modelling* ciri dari data uji dicari kesesuaiannya dengan model ciri dari data latih sehingga bisa diketahui pemilik suara tersebut.

2.2. Perbedaan Suara Manusia

Suara setiap orang bisa berbeda-beda disebabkan banyak faktor diantaranya adalah perbedaan bentuk dan ukuran pita suara, perbedaan bentuk dan ukuran laring, ukuran tubuh, dan bagaimana suara itu sendiri biasa diartikulasikan oleh seseorang.

Ciri utama suara seseorang dipengaruhi tiga hal, yaitu *pitch*, *volume* dan *timbre* [6]. *Pitch* adalah nada, yang artinya suara dengan frekuensi tertentu. *Volume* adalah tingkat kekerasan seseorang bicara (*amplitudo*). *Timbre* adalah warna nada yang menjadi ciri khas tiap orang. Semua orang bisa mengucapkan *do-re-mi-fa-sol-la-si-do* dengan nada yang sama, namun kita tetap bisa membedakan satu orang dengan orang yang lain karena *timbre* yang berbeda. *Timbre* dipengaruhi oleh pita suara.

2.3. Mel Frequency Cepstrum Coefficient (MFCC)

MFCC adalah suatu metode ekstraksi ciri yang keluarannya berupa *feature vector* bernama *cepstrum*. Tahap dalam MFCC adalah sebagai berikut [2] :

a. Pre-Emphasis

Pre-emphasis berfungsi untuk menstabilkan nilai magnitude dari sinyal suara. Rumus dari *pre-emphasis* adalah sebagai berikut:

$$s_n' = s_n - a s_{n-1}$$

Dimana s_n adalah nilai sampel ke-n dan a adalah konstanta *pre-emphasis*.

b. Framing

Framing berfungsi membagi sinyal suara menjadi beberapa *frame* dengan panjang sampel tertentu. Panjang tiap *frame* umumnya sangat singkat sekitar 20 sampai 40 ms. Keuntungan dari *framing* adalah memudahkan analisis dan mengurangi alokasi memori.

c. Windowing

Windowing berfungsi meredam noise yang muncul di kedua ujung *frame*. Teknik windowing yang umum adalah *Hamming Window*, dimana rumusnya sebagai berikut:

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

Dimana N adalah jumlah sampel tiap frame dan n adalah bilangan bulat dari 0 hingga $N-1$.

d. Fast Fourier Transform (FFT)

FFT adalah mengubah sinyal dari domain waktu ke domain frekuensi. Adapun rumus FFT secara matematis adalah sebagai berikut:

$$F_k = \sum_{j=0}^{\frac{N}{2}-1} f_{2j}\, e^{-\frac{2\pi i k(2j)}{N}} + \sum_{j=0}^{\frac{N}{2}-1} f_{2j+1}\, e^{-\frac{2\pi i k(2j+1)}{N}}$$

e. Mel Frequency Filter Bank

Pada tahap ini, sinyal suara pada domain frekuensi diubah menjadi domain frekuensi mel dimana rumusnya adalah sebagai berikut:

$$Mel(f) = 2595(\ln\left(1 + \frac{f}{700}\right))$$

Dimana $Mel(f)$ adalah nilai frekuensi mel dari f. Hasil akhir dari tahap ini adalah didapatkannya sejumlah *mel filter bank*. Nilai *mel* filter bank menunjukkan seberapa besar energi pada rentang frekuensi yang ada pada masing-masing filter *mel*.

f. Transformasi non Linear

Transformasi non-linear berfungsi untuk mengambil nilai logaritma natural dari setiap mel filter bank dimana rumusnya adalah sebagai

berikut:

$$f'_k = \ln(f_k)$$

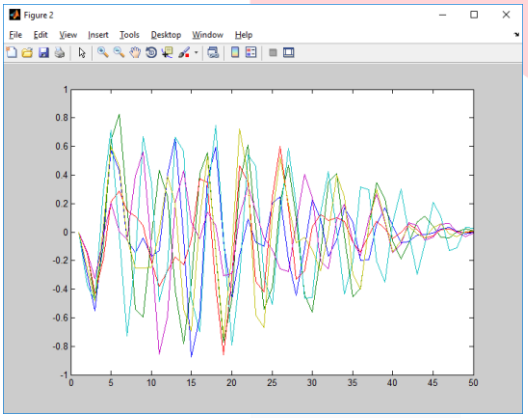
Dimana f_k adalah *mel frequency filter bank* dan k adalah jumlah *mel frequency filter bank* pada tiap *frame*.

g. Discrete Cosine Transform (DCT)

DCT berfungsi untuk mengembalikan sinyal suara pada domain frekuensi ke domain waktu sehingga didapatkan koefisien *cepstrum*. Adapun rumusnya adalah sebagai berikut:

$$c_n = \sum_{k=1}^{K} (f'_k) \cos\left[n(k-0{,}5)\frac{\pi}{K}\right]$$

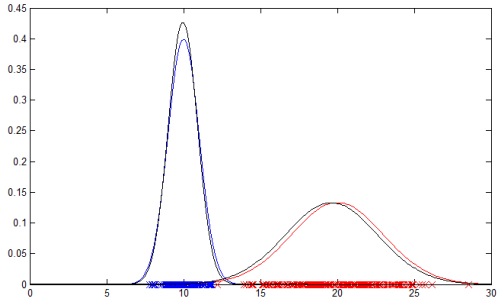
Dimana K adalah jumlah mel frequency filter bank, f'_k berasal dari hasil transformasi non linear, n adalah bilangan bulat dari 1 hingga N (jumlah total sampel) sehingga didapatkan N buah koefisien cepstrum.



Gambar 2.1 Contoh hasil MFCC

2.4. Gaussian Mixture Model (GMM)

GMM adalah algoritma yang berfungsi memodelkan sejumlah data menjadi sebuah distribusi *gaussian* dengan parameter *mean* (μ) dan *variance* (σ^2) tertentu [8]. *Mean* (μ) adalah titik pusat dari distribusi gaussian sedangkan *variance* (σ^2) adalah ukuran persebaran nilai pada set data. Kelebihan yang dimiliki GMM adalah mampu memodelkan lebih dari satu *gaussian* untuk sebuah set data [8].



Gambar 2.2 Contoh hasil modelling GMM

Langkah awal yang harus dilakukan adalah memodelkan set data yang ada menjadi sebuah fungsi probabilistik :

$$p(x|\theta) = \sum_z p(x,z|\theta)$$

Dimana x adalah data, θ adalah parameter mean (μ) dan variance (σ^2), dan z adalah keanggotaan data terhadap gaussian tertentu. Setelah kita mendapatkan model probabilistik maka langkah berikutnya adalah sebagai berikut :

a. Lower Bound

Lower bound adalah fungsi yang memiliki arah yang sama dengan $p(x|\theta)$ namun nilainya kurang dari atau setara dengan $p(x|\theta)$ [4]. Fungsi *lower bound* didapatkan dengan memanfaatkan *Jensen's inequality* terhadap $\log p(x|\theta)$. Dimana *Jensen's inequality* menyatakan :

$$\log \sum_n \lambda_n x_n \geq \sum_n \lambda_n \log x_n$$

Dengan menerapkan persamaan di atas terhadap fungsi probabilistik data maka didapat :

$$\log p(x|\theta) \geq E_{z\sim q}[\log p(x,z|\theta)] + E_{z\sim q}[\log q(z|\theta)]$$

Bagian *lower bound* inilah yang akan kita maksimalkan bukan $p(x|\theta)$, karena *lower bound* memiliki keterkaitan dengan *hidden variable* z [4]. Sebelum melakukan maksimalisasi harus dicari terlebih dahulu titik inisiasi dimana kita akan memulai maksimalisasi, yaitu titik dimana *lower bound* dan $p(x|\theta)$ memiliki nilai yang sama :

Misalkan $q(z) = p(z|x,\theta)$ maka :

$$\sum_z q(z) \log\left[\frac{1}{q(z)} p(x,z|\theta)\right] = \log p(x|\theta)$$

Dengan asumsi di atas maka kita sudah mendapatkan titik inisiasi dimana kita akan memulai maksimalisasi parameter.

b. Expectation Maximization

Algoritma EM adalah sebuah prosedur iteratif untuk menghitung estimasi *Maximum Likelihood* (ML) yang muncul pada *hidden* data [4]. Pada estimasi ML, kita ingin menghitung model parameter dengan kemungkinan paling besar bagi data yang terobservasi.

Setiap iterasi pada algoritma EM mengandung dua buah proses : *E-step* dan *M-step* [4]. Pada *E-step*, *hidden* data diestimasikan menggunakan data terobservasi dan estimasi model parameter saat ini. Pada *M-step*, fungsi *likelihood* (*lower bound*) dimaksimalisasi dengan asumsi bahwa *hidden* data diketahui, yaitu menggunakan *hidden* data yang diestimasi pada *E-step*.

Dengan memperhatikan persamaan *lower bound* kita mendapatkan:

$$\log p(x|\theta) \geq E_{z\sim p(z|x,\theta)}[\log p(x,z|\theta)]$$

$$+E_{z\sim p(z|x,\theta)}[\log p(z|x,\theta)]$$

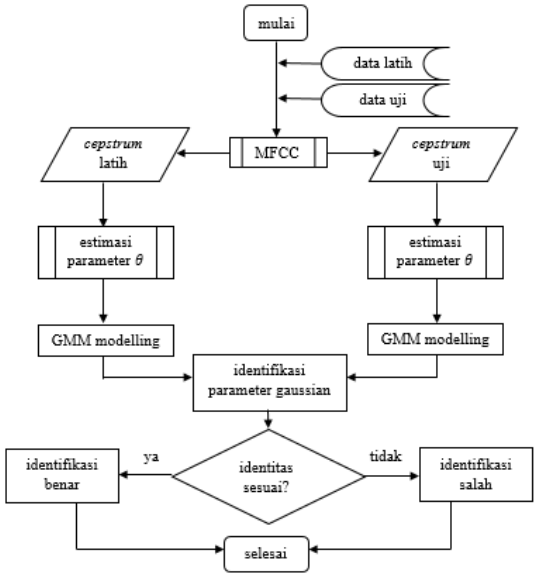
Pada proses maksimalisasi bagian *entropy* bisa diabaikan karena bersifat independen dari parameter θ [5], sehingga bisa kita asumsikan :

$$Q(\theta,\theta^{old}) = E_{z\sim p(z|x,\theta)}[\log p(x,z|\theta)]$$
$$= \sum_z p(z|x,\theta^{old})[\log p(x,z|\theta)]$$

Menggunakan persamaan di atas kita akan melakukan algoritma EM untuk mencari parameter yang cocok :

- Pilih setting awal untuk θ^{old}
- Iterasi hingga konvergen
- *E-Step* : Gunakan x dan θ^{old} untuk menghitung $p(z|x,\theta^{old})$
- *M-Step* : $\theta^{new} = \text{argmax}_{\theta} Q(\theta,\theta^{old})$
- Asumsikan θ^{new} sebagai θ^{old} dan kembali ke *E-step*

3. Perancangan Sistem



Gambar 3.1 Diagram alir keseluruhan

Langkah-langkah pada diagram alir di atas :

- Data latih (keseluruhan data) dan data uji (salah satu data yang akan diuji) yang berupa sinyal suara diekstraksi cirinya melalui MFCC.
- Hasil keluaran dari MFCC berupa *cepstrum*, masing-masing *cepstrum* latih (ekstraksi ciri data latih) dan *cepstrum* uji (ekstraksi ciri data uji).
- *Cepstrum* latih diestimasi parameter (θ) nya. Estimasi parameter untuk *cepstrum* latih dilakukan satu-persatu tiap data karena kita tidak melalui tahap *clustering*, dimana dengan tahap *clustering*, estimasi bisa langsung dilakukan sekaligus data keseluruhan.
- *Cepstrum* uji diestimasi parameter (θ) nya.
- Menggunakan parameter *cepstrum* latih, dimodelkanlah distribusi *gaussian* untuk data latih, begitu pula untuk data uji.
- Distribusi *gaussian* data latih dan data uji diperbandingkan parameternya. Apabila parameter data uji sesuai atau mendekati dengan salah satu parameter data latih (misal : parameter subjek A), maka identifikasi data uji akan diklasifikasikan sebagai subjek A.
- Validasi dilakukan secara manual dengan memperbandingkan hasil identifikasi sistem dengan identitas aktual.

4. Hasil Implementasi

Subjek	Kata				
	India	Deli	Hari	Senin	Selasa
Adi Arief Wicaksono					
Ali Akbar Himawan					
Bagus Teguh Nugroho					
Muhammad Bayu Saputro					
Doanda Khabi Putra					
Ihsan Prawono					
Presida Kremina Yuanisa					
Ricky Luckerman					
Ristianisa Hasana					
M. Tama Yodha Akbar					

Subjek	Kata				
	Rabu	Kamis	Jumat	Sabtu	Minggu
Adi Arief Wicaksono					
Ali Akbar Himawan					
Bagus Teguh Nugroho					
Muhammad Bayu Saputro					
Doanda Khabi Putra					
Ihsan Prawono					
Presida Kremina Yuanisa					
Ricky Luckerman					
Ristianisa Hasana					
M. Tama Yodha Akbar					

Gambar 4.1 Tabel hasil simulasi

Pada percobaan simulasi awal diperoleh akurasi sebesar 67%. Simulasi kemudian dilanjutkan dengan mengubah beberapa variabel seperti jenis *window*, lebar *frame*, jumlah *cepstrum* dan opsi *liftering*.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan hasil simulasi *speaker recognition* dan optimasi akurasi dengan beberapa variabel didapat kesimpulan sebagai berikut :

1) Metode MFCC dan metode GMM bisa dikombinasikan untuk membentuk sistem *speaker recognition*.
2) Tahap *clustering* bisa dihilangkan namun berakibat pada menurunnya akurasi identifikasi.
3) Menghilangkan tahap *clustering* mengakibatkan sistem tidak mampu memanfaatkan kelebihan GMM yang mampu mengestimasi parameter pada banyak data sekaligus. Sistem *speaker recognition* tanpa *clustering* terpaksa harus melakukan estimasi parameter satu-persatu tiap data pembicara.

4) Jenis *window* dengan akurasi terbaik adalah *Chebyshev* atau *Blackman* dengan perolehan akurasi 74%, namun *Blackman* memiliki waktu komputasi lebih singkat yaitu 175 ms dibandingkan dengan *Chebyshev* yang memiliki waktu komputasi sebesar 188 ms. Sehingga jenis *window* terbaik jatuh pada *Blackman*.
5) Lebar *frame* tidak terlalu mempengaruhi akurasi karena nilai yang dihasilkan fluktuatif dan tidak berubah signifikan, namun waktu komputasi akan semakin baik apabila *frame* semakin lebar. Sehingga perolehan lebar *frame* terbaik ada pada angka 1024 dengan waktu komputasi 115 ms.
6) Jumlah *cepstrum* dengan akurasi terbaik ada pada angka 50 dengan perolehan nilai 67%. Meskipun waktu komputasi bukan yang terbaik, yaitu 155 ms namun tidak terlalu jauh dibandingkan jumlah *cepstrum* 5 dengan waktu komputasi 128 ms dimana hanya memperoleh akurasi 63%. Sehingga jumlah *cepstrum* terbaik yang bisa digunakan adalah pada angka 50.
7) Penggunaan *liftering* atau tanpa *liftering* hampir tidak mempengaruhi akurasi pada simulasi. Dengan menggunakan *liftering* diperoleh akurasi 67% sedangkan tanpa *liftering* diperoleh akurasi 68%. Nilai yang berdekatan ini diperkirakan hanya bersifat fluktuatif sehingga sejauh ini bisa disimpulkan bahwa opsi *liftering* tidak mempengaruhi akurasi meskipun secara teori tidak berkata demikian. Berdasarkan waktu komputasi, penggunaan *liftering* memiliki waktu komputasi lebih besar yaitu 155 ms sedangkan dengan menghilangkan *liftering* didapat waktu komputasi 146 ms. Sehingga opsi yang paling baik sejauh ini adalah dengan tidak menggunakan *liftering* karena memiliki waktu komputasi paling singkat.

5.2. Saran

Perancangan sistem *speaker recognition* ini masih terbilang sederhana dan masih bisa dikembangkan lebih jauh lagi. Berikut adalah beberapa saran yang dapat digunakan untuk membuat sistem *speaker recognition* lebih baik lagi :

1) Menguji coba berbagai metode ekstraksi ciri dan modelling yang lain seperti contohnya HMM, SVM, DWT atau ANN untuk memperoleh kemungkinan hasil akurasi dan waktu komputasi yang lebih baik.
2) Menggunakan metode *clustering* seperti K-Means atau KNN untuk mengurangi distorsi antar data pembicara.
3) Menambah data pembicara agar semakin mempertegas perbedaan akurasi antar variabel optimasi terutama opsi *liftering* karena secara teori *liftering* mampu membawa dampak signifikan terhadap akurasi sistem *speaker recognition*.

6. Daftar Pustaka

1) Mahboob, Tahira. Khanum, Memoona. Khiyal, Malik Sikandar Hayat. Bibi, Ruqia. 2015. *Speaker Identification Using GMM with MFCC*. Pakistan : Jinnah Women University dan Preston University.
2) Aulia Sadewa, Reza. 2015. *Speaker Recognition Implementation for Authentication using Modified MFCC-Vector Quantization LBG Algorithm*. Bandung : Universitas Telkom.
3) Soleymani. 2012. *EM & Gaussian Mixture Models (GMM)*. Iran : Sharif University of Technology.
4) Borman, Sean. 2004. *The Expectation Maximization Algorithm*. Salt Lake City : The University of Utah.
5) Anonim. 2009. *Expectation Maximization*. Salt Lake City : The University of Utah.
6) Jaleel, Asad. 2015. *Why Does Everyone Have a Different Voice*. https://www.quora.com/Why-does-everyone-have-a-different-voice (diakses 24 April 2017)
7) Peacocke, Richard D. dan Graf, Daryl H. 1990. *An Introduction to Speech and Speaker Recognition*. Ontario : Bell-Northern Research.
8) Snowbell. 2016. *What is 'Mixture' in A Gaussian Mixture Model*. https://stats.stackexchange.com/questions/236295/what-is-mixture-in-a-gaussian-mixture-model (diakses 24 April 2017)
9) Anonim. *Decorrelated and Liftered Filter-Bank Energies for Speech Recognition. Brisbane* : Griffith University.
10) Oppenheim, Alan V. Dan Schafer, Ronald W. 2004. *From Frequency to Quenfrency : A History of the Cepstrum*. IEEE Signal Processing Magazine.
11) Gultom, Maryati. Mukhlisa. Alamsyah, Derry. 2015. *Rancang Bangun Aplikasi Pengenal Penutur Menggunakan Metode Hidden Markov Model (HMM)*. Palembang : STMIK GI MDP.
12) Tandyo, Anny. Martono. Widyatmoko, Adi. *Speaker Identification Menggunakan Transformasi Wavelet Diskrit dan Jaringan Saraf Tiruan Back-Propagation*. Jakarta Barat : Universitas Bima Nusantara.

13) Putra, Darma. Resmawan, Adi. 2011. *Verifikasi Biometrika Suara Menggunakan Metode MFCC dan DTW*. Bali : Universitas Udayana.