

Implementasi Metode *Principal Component Analysis* dan *Hidden Markov Model* pada Pengenalan Suara

YOULLIA INDRAWATY N, ANDRIANA Z, FADHLIN PRASETIA

Jurusan Teknik Informatika, Fakultas Teknologi Industri
Institut Teknologi Nasional

Email: youllia@itenas.ac.id

ABSTRAK

Informasi sinyal suara yang dikenali memiliki karakteristik yang unik. Dengan adanya keunikan pada sinyal suara dapat diimplementasikan untuk melakukan identifikasi suara pada bidang kesehatan yang difungsikan sebagai pengenalan suara pada tunawicara. Dimana pada pengembangannya aplikasi mampu untuk membantu pembelajaran pada tunawicara. Metode yang dapat melakukan ekstraksi ciri suara salah satunya adalah metode Principal Component Analisis. Metode Principal Component Analysis bekerja dengan melakukan ekstraksi ciri suara dan pengelompokan pada pola suara. Suara yang direkam melalui proses pre-processing audio sebelum dikelompokkan menggunakan Principal Component Analysis untuk mendapatkan nilai koefisien suara. Hasil dari nilai koefisien disimpan sebagai sinyal referensi dan digunakan pada proses pencocokan sinyal suara menggunakan algoritma Hidden Markov Model. Sistem diuji sebanyak 3 sesi pengujian oleh 14 orang penguji. Berdasarkan hasil pengujian diketahui bahwa sistem yang dibangun mencapai tingkat akurasi 85% dari 3 sesi pengujian oleh 14 orang penguji.

Kata Kunci: *Tuna wicara, Ekstraksi ciri, Pencocokan suara*

ABSTRACT

Voice signal information can recognized that have unique characteristic. With that unique on voice signal can be implemented for identifying voice on medical field that can be used for speech recognition on speech disorder. Principal Component Analysis is a method that work for feature extracting and grouping on voice pattern. Recorded voice by pre-processing audio before grouped by Principal Component Analysis for getting the value of voice coefficient. The result of voice coefficient saved as signal reference and used on matching the value of voice coefficient using Hidden Markov Model algorithm. System tested as many as 3 session by 14 examiner. This system has overall accuracy rate 85% from 14 examiner for 3 session.

Keywords: *Speech Disorder, Feature extraction, Voice matching*

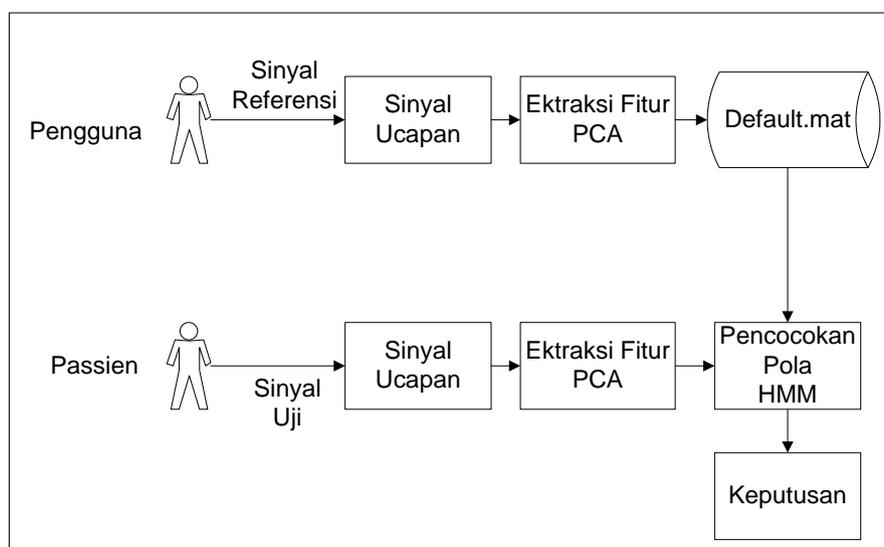
1. PENDAHULUAN

Informasi sinyal suara memiliki karakteristik yang unik. Keunikan sinyal suara dapat diimplementasikan untuk melakukan identifikasi suara pada bidang kesehatan yang difungsikan sebagai pengenalan suara pada tunawicara. Dimana pada pengembangannya aplikasi mampu untuk membantu pembelajaran pada tunawicara. Terdapat kendala yang dihadapi pada pengidentifikasian suara manusia tersebut yaitu, suara dipengaruhi waktu dan suara noise pada lingkungan sekitar. Oleh karena itu, dibutuhkan sebuah algoritma yang dapat mengidentifikasi pola suara salah satunya adalah *Principal Component Analysis*. Metode *Principal Component Analysis* bekerja dengan melakukan ekstraksi ciri suara dan pengelompokan pada pola suara. Suara yang direkam akan melalui proses *pre-processing audio* sebelum dikelompokkan menggunakan *Principal Component Analysis* untuk mendapatkan nilai koefisien suara. Hasil dari nilai koefisien disimpan sebagai sinyal referensi dan digunakan pada proses pencocokan sinyal suara menggunakan algoritma *Hidden Markov Model*. Dalam permasalahan ini pengenalan suara untuk mengetahui ciri dari setiap suara menggunakan metode *Principal Component Analysis* dan untuk pencocokan suara menggunakan algoritma *Hidden Markov Model*.

Metode *Principal Component Analysis* biasa digunakan untuk proses pengolahan citra digital. Pada penelitian ini *PCA* akan diujicobakan sebagai metode yang dapat mengekstraksi ciri suara. Untuk mendapatkan nilai ekstraksi ciri suara *PCA*, dilakukan proses *pre-processing audio*. Tujuannya untuk mengelompokkan suara berdasarkan nilai ekstraksi ciri suara sehingga sistem dapat mengenali suara. Sampel suara atau huruf uji berupa huruf vokal a, i, u, e, dan o.

2. METODOLOGI PENELITIAN

Proses pengenalan ucapan pada sistem ini dapat dijelaskan kedalam sebuah model kerja sistem aplikasi secara umum seperti pada Gambar 1.



Gambar 1. Sistem Kerja Aplikasi Pengenalan Ucapan

Alur kerja dari sistem kerja aplikasi pengenalan suara pada Gambar 1, pengguna melakukan data training dengan merekam sinyal masukan suara dari mikrofon. Dalam melakukan pengujian untuk menghitung sinyal suara menggunakan Persamaan 1.

$$F_s = FS \times dt(\text{detik}) \times \left(\frac{\text{bit}}{8}\right) \times j \quad (1)$$

(<http://dir.unikom.ac.id/s1-final-project/fakultas-teknik-dan-ilmu-komputer/teknik-informatika/2011/jbptunikompp-gdl-annadaraan-25983/7-13.unik-i.pdf/pdf/7-13.unik-i.pdf>)

Dimana:

F_s = Data *sampling* sinyal bit = Jumlah bit resolusi
 FS = Frekuensi *sampling* j = Channel (*mono* = 1)
 dt = Durasi rekaman (detik)

Hasil sinyal suara berupa data *sampling* melalui *proses pre-processing audio* (Gambar2). Data *sampling* dinormalisasi dengan cara dihitung nilai rata-ratanya untuk penyeragaman data menjadi sama. Keluaran dari proses ini berupa nilai *array* yang sudah dinormalisasi menggunakan Persamaan 2, proses ini disebut *DC-Removal*.

$$y[n] = x[n] - x, \quad 0 \leq n \leq N-1 \quad (2)$$

(Sumber: <http://dir.unikom.ac.id>, 2011)

Dimana:

$D[i]$ = hasil *signal* ke -i setelah dilakukan *DC Removal*
 $S[i]$ = *signal* awal ke-i
 N = jumlah *sample*, $n > 0$

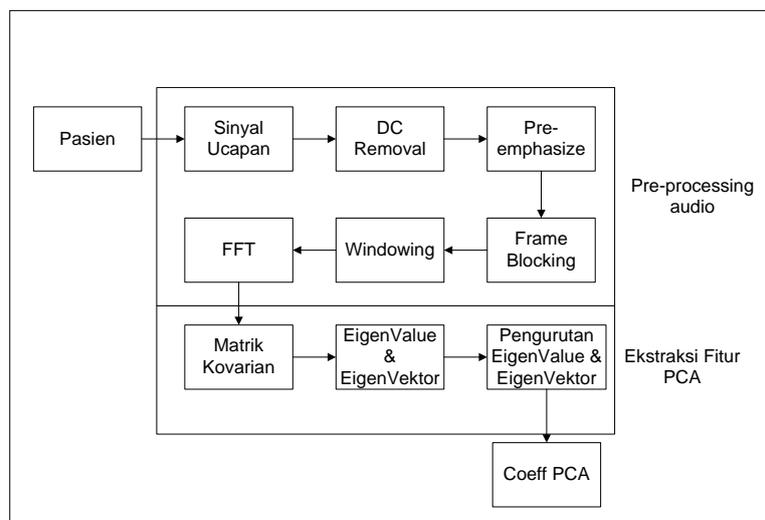
Sinyal suara diperbaiki dari gangguan *noise* (bising) untuk dapat meningkatkan tingkat akurasi dari proses *feature extraction* dengan menggunakan Persamaan 3. Proses perbaikan sinyal disebut *Pre-Emphasize*.

$$y[n] = s[n] - \alpha s[n - 1] \quad (3)$$

(Sumber: <http://dir.unikom.ac.id>, 2011)

Dimana:

$y[n]$ = sinyal hasil *pre-emphasize* $\alpha=0.97$ default nilai *alpha*
 $s[n]$ = sinyal sebelum *pre-emphasize*



Gambar 2. Blok Diagram PCA

Hasil perekaman suara merupakan sinyal analog yang masih bergantung terhadap waktu. Oleh karena itu sinyal tersebut harus dipotong - potong dalam slot - slot waktu tertentu agar bisa dianggap tidak bergantung terhadap waktu. Proses pemotongan sinyal suara bertujuan untuk membagi sinyal suara menjadi beberapa slot dengan panjang tertentu menggunakan Persamaan 4. Proses pemotongan sinyal suara disebut *Frame Blocking*.

$$\text{Jumlah } i = ((I-N)/M)+1 \quad (4)$$

(Sumber: <http://dir.unikom.ac.id>, 2011)

Dimana:

$I = \text{Sample rate}$

$N = \text{Sample point (Sample rate * waktu framing (s))}$

$M = N/2$

Suara yang sudah menjadi slot bagian dianalisis tiap pembatas sinyalnya dengan tujuan untuk mengurangi efek ketidakselarasan pada ujung-ujung slot yang dihasilkan oleh *frame blocking* menggunakan Persamaan 5. Proses mengurangi efek ketidakselarasan disebut *Windowing*.

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

(Sumber: <http://dir.unikom.ac.id>, 2011)

Dimana:

$W(n)$: nilai *windowing*

Sehingga diperoleh sampel sinyal kedalam bentuk frekuensi domain untuk mengurangi kerumitan transformasi yang dihasilkan dari proses *windowing* menggunakan Persamaan 6. Proses ini disebut *FFT*.

$$F(k) = \sum_{n=1}^N f(n) \cdot e^{-j\pi kn T / N} \quad (6)$$

(Sumber: <http://dir.unikom.ac.id>, 2011)

Dimana:

$F(k)$: nilai *FFT point*

Menyusun nilai-nilai varian yang ditemukan pada proses *FFT* berupa parameter matriks. Hasil penyusunan parameter matriks diolah dengan nilai *zero mean* sehingga membentuk nilai matriks kovarian dengan Persamaan 7.

$$C = \begin{bmatrix} c(x_1, x_1) & c(x_1, x_2) \\ c(x_2, x_1) & c(x_2, x_2) \end{bmatrix} \quad (7)$$

(Sumber: Mauridhi Hery Purnomo, Arif Muntasa. 2010)

Dimana:

C : nilai matriks kovarian

c : nilai matriks penyusunan parameter *FFT point*

Hasil penyusunan nilai matriks kovarian dicari cirinya berupa data sampel untuk menentukan nilai *eigenvalue* dan *eigenvector* menggunakan Persamaan 8.

$$Det(\lambda I - C) = 0 \quad (8)$$

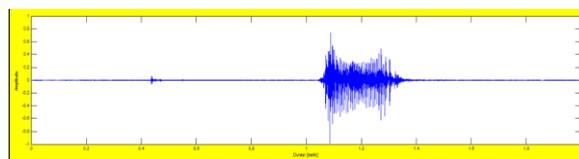
(Sumber: Mauridhi Hery Purnomo, Arif Muntasa. 2010)

Nilai *eigenvalue* dan *eigenvector* diurutkan dengan nilai terbesar merupakan nilai koefisien *PCA*. Hasil nilai fitur ekstraksi ciri berupa nilai koefisien *PCA* disimpan kedalam database dalam bentuk file *.mat*. Untuk pengujian, suara yang sudah direkam dan melalui proses ekstraksi sampai mendapatkan nilai koefisien *PCA* dicocokkan dengan nilai koefisien *PCA* yang sudah disimpan kedalam database. Hasil pencocokan suara menggunakan algoritma *HMM* untuk mencocokkan pola suara pengguna, dengan pola suara referensi dalam menentukan kesamaan jarak antara pola-pola yang berbeda.

3. ANALISIS DAN PEMBAHASAN

Pengolahan sinyal suara tidak dapat langsung menggunakan metode *PCA*. Hal ini dikarenakan, *PCA* sendiri tidak bisa mendapatkan nilai *FFT* point untuk pengolahan suara. Maka, untuk mendapatkan nilai *FFT* point bisa dicari dengan melakukan proses *pre-processing audio*. Dimana pengguna melakukan rekam suara melalui *microphone* untuk mengkonversi sinya suara dari analog menjadi digital. Perekaman suara di dalam aplikasi menggunakan frekuensi sampling standar 16000Hz. Suara dengan format *.wav* ini bisa menggunakan 16 bits/sample dan 1 untuk channel mono. Durasi suara yang direkam apabila lebih pendek lebih mudah untuk diambil perbedaan fiturnya. Dalam analisis ini digunakan contoh durasi rekaman yang diambil adalah 2 detik dengan menggunakan perhitungan Persamaan 1 menghasilkan sinyal ucapan seperti pada Gambar 3.

$$F_s = 16000 \times 2 \times \left(\frac{16}{8}\right) \times 1 = 64000 \text{ byte}$$

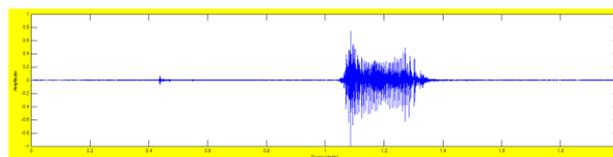


Gambar 3. Hasil Input Sinyal Ucapan

Melakukan proses *DC-Removal* untuk mendapatkan nilai normalisasi dari sampel data suara masukan. Masukan dari proses ini adalah sampel suara dalam bentuk *array* dan outputnya adalah *array* sample suara yang sudah dinormalisasi dengan sampel nilai sampel data 1,2,3 menggunakan Persamaan 2. Dan hasil sinyal dari *DC-Removal* seperti pada Gambar 4.

$$x = \frac{1+2+3}{3} = 2$$

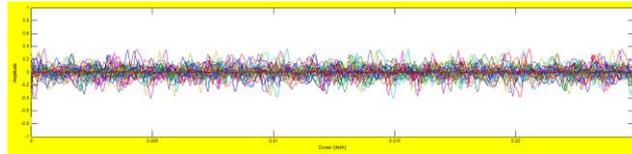
$$n = 1-2 = -1$$



Gambar 4. Hasil proses DC-Removal

Setelah sinyal suara dinormalisasi, sinyal suara diperbaiki dari gangguan *noise* (bising) untuk dapat meningkatkan tingkat akurasi dari proses *feature extraction* menggunakan Persamaan 3. Proses ini disebut *Pre-emphasize* dan hasilnya seperti pada Gambar 5.

$$y = (0 - (-1 * 0, 97)) = 0.97$$



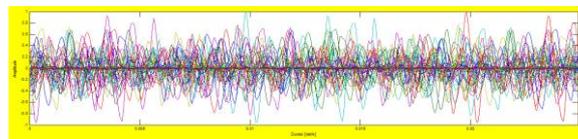
Gambar 5. Hasil proses *Pre-emphasize*

Selanjutnya melakukan proses *Frame Blocking* bertujuan untuk membagi sampel sinyal menjadi beberapa frame slot dengan panjang tertentu dihitung menggunakan Persamaan 4. Dan hasil sinyal dari *Frame Blocking* seperti pada Gambar 6.

$$N = 32000 * 0,02 = 640 \text{ Sample Point}$$

$$M = 640/2 = 320.$$

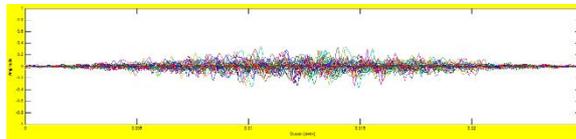
$$\text{Sehingga Jumlah Frame : } ((I-N) / M) + 1 = ((32000-640) / 320) + 1 = 99$$



Gambar 6. Hasil proses *Frame Blocking*

Melakukan proses *Windowing* untuk mengurangi efek aliasing pada ujung-ujung frame yang dihasilkan oleh proses frame blocking menggunakan Persamaan 5. Hasil sinyal dari *Windowing* seperti pada Gambar 7.

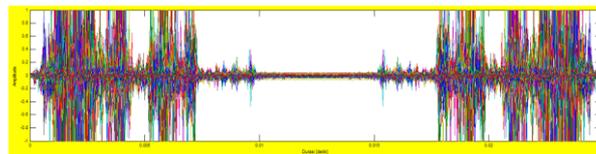
$$W_{1,0} = 0.54 - 0.46 \cos \frac{2 \times 3.14 \times 0}{640 - 1} = 0.08$$



Gambar 7. Hasil proses *Windowing*

Melakukan proses *FFT (Fast Fourier Transform)* merubah *signal digital* dari *time domain* ke *frequency domain* menggunakan Persamaan 6, sehingga diperoleh sampel signal dalam *frequensi domain*. Dan hasil dari proses *FFT* seperti pada Gambar 8.

$$F_{1,0} = \frac{1}{3} \left[-0.16 \left(\cos \left(\frac{2\phi i * 0 * 0}{3} \right) \right) \right] - j \sin \left(\cos \frac{2\phi i * 0 * 0}{3} \right) + \left[0.07 \left(\cos \left(\frac{2\phi i * 0 * 1}{3} \right) \right) \right] - j \sin \left(\cos \frac{2\phi i * 0 * 1}{3} \right) + \left[0.16 \left(\cos \left(\frac{2\phi i * 0 * 2}{3} \right) \right) \right] - j \sin \left(\cos \frac{2\phi i * 0 * 2}{3} \right) + = 0.02 + 0j = 0.02$$



Gambar 8. Hasil proses *FFT*

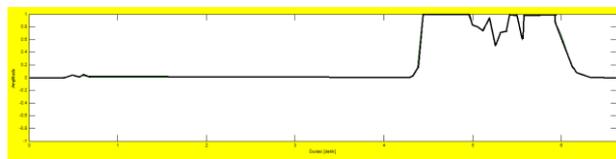
Maka dengan menggunakan persamaan matrik kovarian, melihat hasil dari *FFT* sebelumnya menggunakan 3 parameter dengan jumlah data sebanyak 3 data. Pembentukan matriks kovarian berdasarkan hasil perhitungan dari Persamaan 7 disusun sebagai berikut.

$$C = \begin{bmatrix} 0,0040735 & -0,0005045 & 0,0040735 \\ -0,0005045 & 0,0006335 & -0,0005045 \\ 0,0040735 & -0,0005045 & 0,0040735 \end{bmatrix}$$

Hasil penyusunan nilai matriks kovarian dicari cirinya berupa data sampel untuk menentukan nilai *eigenvalue* dan *eigenvector* menggunakan Persamaan 8. Setelah mendapatkan nilai *eigenvector* dan *eigenvalue* dilakukan pengurutan sehingga dapat digunakan sebagai nilai koefisien *PCA*. Dan mendapatkan nilai dari ekstraksi ciri *PCA* seperti pada Gambar 9.

$$\text{eigVektor} = \begin{bmatrix} -0,0663 & 0,7071 & -0,7040 \\ -0,9956 & 0 & 0,0937 \\ 0,0040735 & -0,7071 & -0,7040 \end{bmatrix}$$

$$\text{eigValue} = \begin{bmatrix} 0,0006 \\ 0 \\ 0,00824 \end{bmatrix}$$



Gambar 9. Hasil proses PCA

Fungsi yang diuji pada aplikasi adalah fungsi merekam suara, pencocokan suara dan fungsi metode *PCA*. Pengujian pada fungsi merekam suara dilakukan agar diketahui apakah aplikasi dapat merekam suara, melakukan proses ekstraksi *PCA* dan menyimpan suara kedalam database. Pengujian pada fungsi pencocokan suara dilakukan agar diketahui apakah aplikasi dapat melakukan pencocokan suara dengan membandingkan nilai dari koefisien *PCA*. Selanjutnya dapat diketahui kesimpulan dari hasil pengujian seluruh fungsi yang diterapkan pada Aplikasi Pencocokan Suara.

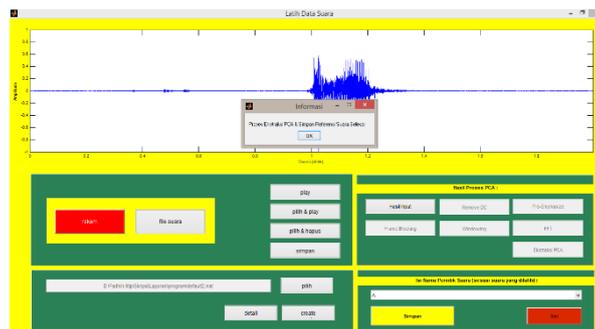
3.1 Fungsi Merekam Suara

Pengujian fungsi merekam suara dilakukan untuk mengetahui apakah suara yang direkam dapat melakukan proses ekstraksi *PCA* sebelum simpan kedalam *database* yang telah ditentukan nilai dari sinyal suara tersebut. Kriteria tujuan aktor merekam suara dan melakukan proses ekstraksi ciri *PCA* sebelum dimasukkan kedalam database. Setelah fungsi merekam suara tersebut diimplementasikan ke dalam sistem, fungsi merekam suara diuji dengan tujuan untuk melihat berhasil atau tidaknya penerapan fungsi perekaman suara pada sistem. Pengujian dilakukan dengan uji *blackbox* dengan informasi aktor melakukan pengujian, kondisi awal sebelum melakukan pengujian, kondisi terakhir yang diharapkan, skenario pengujian, kondisi akhir yang didapatkan, hasil pengujian, dan kesimpulan yang disimpulkan setelah dilakukannya pengujian. Uji *blackbox* pengujian fungsi perekaman suara diketahui pada Tabel 1.

Tabel 1. Pengujian Fungsi Merekam Suara

Nama Fungsi	Merekam suara
Tujuan	Untuk merekam suara dan melakukan proses ekstraksi ciri PCA sebelum dimasukkan kedalam database.
Aktor	Admin atau <i>user</i> .
Kondisi Awal	Halaman utama
Kondisi yang Diharapkan	Sistem dapat merekam dan melakukan proses ekstraksi ciri PCA sebelum dimasukkan kedalam database.
Skenario	1) Aktor menekan menu latih data suara 2) Aktor menekan tombol rekam dan menyimpan huruf sesuai dengan apa yang sudah diucapkan. 3) Aktor menentukan sinyal ucapan yang baru direkam.
Kondisi Akhir yang Didapatkan	Aktor masuk ke menu latih data suara dan melakukan proses perekaman kemudian aktor menyimpan sinyal suara yang baru direkam
Hasil	Berdasarkan kondisi ahir yang diharapkan, sinyal suara yang direkam dan disimpan ke database sesuai dengan kriteria tujuan perekaman.
Kesimpulan	Sinyal suara yang direkam sesuai dengan kriteria tujuan perekaman.

Hasil pengujian yang dilakukan Aktor terhadap fungsi merekam suara dengan mengikuti skenario diatas yang dinyatakan pada Tabel 1. Aktor dalam memulai pengujian merekam suara dapat di terima sistem untuk melakukan pelatihan suara saat melakukan pengujian identifikasi suara seperti ditunjukkan pada Gambar 10.



Gambar 10. Pengujian Fungsi Rekam Suara

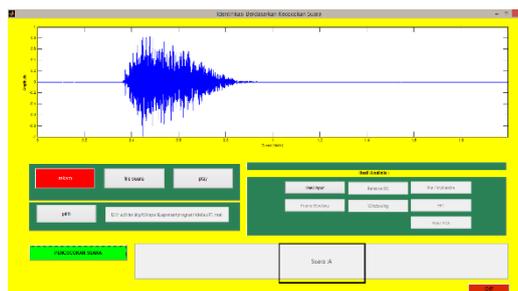
3.2 Fungsi Pencocokan Suara

Pengujian fungsi pencocokan suara dilakukan untuk mengetahui apakah suara uji berhasil dicocokkan dengan suara referensi yang disimpan dalam database. Kriteria tujuan aktor merekam suara dan melakukan proses ekstraksi ciri PCA sebelum melakukan proses pencocokan suara. Setelah fungsi pencocokan suara tersebut diimplementasikan kedalam sistem, fungsi pencocokan suara diuji dengan tujuan melihat berhasil atau tidaknya penerapan fungsi pencocokan suara pada sistem. Pengujian dilakukan dengan uji blackbox dengan informasi aktor melakukan pengujian, kondisi awal sebelum melakukan pengujian, kondisi terahir yang diharapkan, skenario pengujian, kondisi ahir yang didapatkan, hasil pengujian, dan kesimpulan yang disimpulkan setelah dilakukannya pengujian. Uji blackbox pengujian fungsi perekaman suara diketahui pada Tabel 2.

Tabel 2 Pengujian Fungsi Pencocokan Suara

Nama Fungsi	Pencocokan suara
Tujuan	Untuk melatih suara uji dan melakukan perbandingan dengan suara referensi yang sudah dimasukkan kedalam database.
Aktor	Admin atau <i>user</i> .
Kondisi Awal	Halaman utama
Kondisi yang Diharapkan	Sistem dapat melatih dan melakukan proses pencocokan suara dengan algoritma HMM
Skenario	4) Aktor menekan menu identifikasi suara 5) Aktor menekan tombol rekam dan setelah proses perekaman aktor menekan tombol pencocokan suara
Kondisi Akhir yang Didapatkan	Aktor masuk kedalam menu identifikasi suara dan melakukan proses perekaman dan pencocokan suara.
Hasil	Berdasarkan kondisi ahir yang diharapkan, sinyal suara yang direkam dapat dicocokkan dengan database.
Kesimpulan	Sinyal suara yang diuji dan dilakukan perbandingan dengan suara referensi sesuai dengan kriteria tujuan pencocokan suara.

Hasil pengujian yang dilakukan Aktor terhadap fungsi pencocokan suara dengan mengikuti skenario diatas dinyatakan pada Tabel 2. Aktor dalam memulai pengujian mencocokkan suara dapat di terima sistem untuk melakukan eksekusi pencocokan suara saat melakukan identifikasi suara seperti ditunjukkan pada Gambar 11.



Gambar 11. Pengujian Jarak Kecocokan Suara

Pengujian dilakukan kepada 14 penguji yang sebelumnya suara dari masing-masing penguji telah di rekam dan disimpan sebagai suara referensi. Pengujian dilakukan sebanyak 3 sesi. Dengan satu sesi terdiri dari 10 kali pengulangan huruf untuk setiap 5 huruf yang sudah ditentukan sehingga mendapatkan 50 sampel untuk setiap sesi yang dilakukan. Hasil dari pengujian diketahui bahwa sistem yang dibangun mencapai tingkat akurasi 85% dari 3 sesi pengujian seperti pada Tabel 3.

Tabel 3 Pengujian Aplikasi

Penguji	Jumlah Pengujian			Jumlah Match			Persentasi Match (%)		
	Sesi 1	Sesi 2	Sesi 3	Sesi 1	Sesi 2	Sesi 3	Sesi 1	Sesi 2	Sesi 3
Penguji 1	50	50	50	44	45	46	88	90	92
Penguji 2	50	50	50	41	45	43	82	90	86

4. KESIMPULAN

Dari hasil implementasi dan pengujian fungsi yang dilakukan, dapat disimpulkan bahwa penggunaan Metode Principal Component Analysis dan Algoritma Hidden Markov Model dalam aplikasi pencocokan suara mampu mengenal suara dengan tingkat akurasi mencapai 85% dengan 3 sesi pengujian oleh 14 orang penguji. Dengan digunakannya metode Principal Component Analysis dan algoritma Hidden Markov Model dalam aplikasi pencocokan suara, aplikasi mampu mengelompokkan suara berdasarkan nilai ekstraksi ciri suara.

DAFTAR PUSTAKA

- [1] Sinung Tegar. 2008. "Aplikasi Pengenalan Ucapan Sebagai Pengaktif Peralatan Elektronik". Universitas Diponegoro
- [2] AS Wahid Faizin. 2014. " Speech Recognition / Pengenalan Suara". Universitas Islam Negeri (UIN) Sunan Kalijaga Yogyakarta
- [3] Inung Wijayanto. 2012. " Jenis Tipe Jangkauan Suara Pada Pria dan Wanita Menggunakan Metode Mel-Frequency Cepstral Coefficient dan Jaringan Syaraf Tiruan Backpropagation". Insitut Teknologi Telkom Bandung
- [4] Lawrence Rabiner, dkk. 1993. "Fundamental of Speech Recognition". Prentice Hall PTR
- [5] Dadang Ardiansyah. 2011. " Identifikasi Wajah Pada Sistem Keamanan Brangkas Menggunakan Principal Component Analysis". Politeknik Elektronika Negeri Surabaya
- [6] Muhammad Eko B P . 2010. "Teori Dasar Hidden Markov Model". Institut Teknologi Bandung
- [7] Mauridhi Hery Purnomo, Arif Muntasa. 2010. "Konsep Pengolahan Citra Digital dan Ekstraksi Fitur". Graha Ilmu
- [8] Muhammad Eko B P . 2010. "Teori Dasar Hidden Markov Model". Institut Teknologi Bandung