

PENERAPAN PEMISAHAN SUARA BERDASARKAN CIRI SUARA MENGGUNAKAN JARINGAN SARAF TIRUAN

Jayanta¹, Henki Bayu Seta², Ridho Zulfahmi³
Universitas Pembangunan Nasional Veteran Jakarta^{1,2,3},

Jalan RS. Fatmawati, Jakarta Selatan

Sur-el : jayanta@upnvj.ac.id¹, henkiseta@upnvj.ac.id², rido.zet@gmail.com³

Abstract : *One of the biometric data possessed by humans is voice. The advantage of this voice biometric data is that voice cannot be transferred from one person to another, and everyone has different voice specifications. Identification of a person based on voice can be a solution when there are problems that require voice recognition as part of the solution. This research will build a speaker identification system (SIP) where the system will identify the speaker in a conversation between 2-3 speakers. Identification is carried out based on the characteristics of the speaker's voice that is owned by the system. The characteristics of the speaker's voice will be obtained from the extraction of sound characteristics using the results of voice data recording which will be carried out as the initial activity of this research. The research phase consisted of 3 stages, namely, the stage of recording the voice data of 5 speakers, the stage of reducing the noise in the captured voice data, the extraction of voice features with the feature extraction algorithm, and the stage of identifying the speaker based on the available sound characteristics. The feature extraction algorithms used are MFCC and PCA. The advantage of these two algorithms is to reduce feature data so that identification speed is obtained with good accuracy. The results of this study are the availability of a voice corpus that can be provided as research data, sound separation techniques, and speaker identification models in a conversation with 2-3 speakers.*

Keywords: *Biometrik, Jaringan Saraf Tiruan, Sistem Identifikasi Pembicara, PCA, MFCC*

Abstrak : *Salah satu data biometrik yang dimiliki oleh manusia adalah suara. Keunggulan data biometrik suara ini adalah suara tidak dapat dipindahkan dari satu orang ke orang yang lain, dan setiap orang memiliki spesifikasi suara yang berbeda. Identifikasi seseorang berdasarkan pada suara dapat menjadi solusi ketika ada permasalahan yang membutuhkan pengenalan suara sebagai bagian dari solusinya. Penelitian ini akan membangun sistem identifikasi pembicara (SIP) dimana sistem akan melakukan identifikasi pembicara pada suatu percakapan antara 2-3 pembicara. Identifikasi dilakukan berdasarkan pada ciri suara pembicara yang dimiliki oleh sistem. Ciri suara pembicara akan diperoleh dari hasil ekstraksi ciri suara dengan menggunakan hasil perekaman data suara yang akan dilakukan sebagai kegiatan awal penelitian ini. Tahap penelitian yang dilakukan terdiri dari 3 tahap yaitu, tahap perekaman data suara 5 orang pembicara, tahap pengurangan noise pada hasil tangkapan data suara, tahap ekstraksi ciri suara dengan algoritma ekstraksi ciri, dan tahap identifikasi pembicara berdasarkan ciri suara yang tersedia. Algoritma ekstraksi ciri yang digunakan adalah MFCC dan PCA. Keunggulan dari kedua algoritma ini adalah untuk mereduksi data ciri sehingga diperoleh kecepatan identifikasi dengan akurasi yang masih baik. Hasil dari penelitian ini adalah tersedianya corpus suara yang dapat disediakan sebagai data penelitian, teknik pemisahan suara, dan model identifikasi pembicara pada suatu percakapan dengan 2-3 pembicara.*

Kata kunci: *Biometrics, Artificial Neural Networks, Speaker Identification System, PCA, MFCC*

1. PENDAHULUAN

Sistem Identifikasi Pembicara (SIP) telah banyak diteliti dan dikembangkan dengan suara sebagai data masukannya. Suara merupakan data

biometric yang dimiliki oleh manusia dengan spesifikasi berbeda satu dengan yang lain. Sifat suara yang dapat digunakan sebagai pengenalan (identifier) menjadikan suara dapat digunakan dalam sistem identifikasi pembicara. Kecepatan

dan akurasi yang baik merupakan ukuran kemampuan SIP dalam mengidentifikasi pembicara. Data masukan yang digunakan dalam SIP sangat mempengaruhi kinerja SIP, oleh karenanya data masukan yang digunakan dalam SIP menjadi hal penting. Pembentukan data masukan dengan ukuran kecil dan tetap membawa data ciri suara seseorang yang baik harus menjadi perhatian dalam pembentukan SIP. Sistem Identifikasi Pembicara menggunakan klasifikasi pola suara sebagai dasar kerjanya, sehingga menggunakan Jaringan Syaraf Tiruan yang dapat membentuk model rujukan dari proses pembelajaran untuk melakukan pencocokan pola dan klasifikasi data sangat dibutuhkan.

Sistem identifikasi Pembicara dapat digunakan dalam upaya penyidikan dan pengungkapan kasus dengan barang bukti digital audio [1], pembuktian dilakukan dengan membuktikan kesamaan sumber suara pada barang bukti dengan rekaman suara sampling. Hasil penelitian ini melakukan proses kecocokan 20 kata yang diacu oleh FBI dengan hasil analisis menunjukkan bahwa barang bukti digital audio memiliki kesamaan dengan sampling audio yang direkam. Proses identifikasi menggunakan suara sebagai data biometrik dalam proses pengenalan

Manusia berdasarkan ucapan [2], menggunakan metode ekstraksi ciri Mel-Frequency Cepstral Coefficients(MFCC), merupakan metode ekstraksi ciri yang mendekati sistem pendengaran manusia dan mampu untuk mengenali pola ucapan. Penelitian pengenalan suara juga digunakan untuk pengamanan ganda

untuk kata sandi [3], menggunakan jaringan saraf tiruan (JST) untuk mengenali pola kontur intonasi.

Metode yang banyak digunakan pada ekstraksi ciri adalah MFCC dan PCA. Kedua metoda memiliki keunggulan masing-masing, MFCC unggul pada pencapaian tingkat akurasi identifikasi. Penelitian dengan menggunakan MFCC dengan nilai parameter koefisien mel 12, lebar waktu frame 20 ms, overlap 50% dari lebar frame, memberikan nilai akurasi 94,1% [4]. Penelitian dengan menggunakan MFCC pada pengenalan suara (*speaker verification*) melalui jaringan telepon tetap, dengan nilai parameter koefisien mel 20, lebar waktu frame 30 ms, overlap 50% dari lebar waktu frame, memberikan nilai akurasi pengujian sebesar 97,43% [5]. Penelitian identifikasi 8 pembicara (*speaker identification*), menggunakan MFCC dengan nilai parameter koefisien mel 20, lebar waktu frame 40 ms, overlap 50% dari lebar waktu frame, hasil penelitian memberikan nilai akurasi 96% [6].

PCA (*Principal Component Analysis*) adalah teknik yang sering digunakan untuk mereduksi dimensi data dan tetap membawa nilai informasi dengan baik. PCA juga dikenal sebagai metode *Karhunen-Loeve* merupakan salah satu teknik analisis peubah ganda. Hasil transformasi PCA merupakan kombinasi linier dari peubah asli yang tersusun berdasarkan informasi yang dikandungnya atau disebut sebagai vektor eigen atau nilai komponen utama. PCA yang digunakan pada sistem identifikasi pembicara yang mengkombinasi MFCC dan PCA [7] menggunakan metode *text-dependent* mereduksi vector data ciri yang semula

berukuran 1.881 menjadi vektor data ciri berisi 99 nilai komponen utama. Vektor data ciri yang dibentuk berasal dari 36 nilai eigen yang dihasilkan dari proses reduksi dengan persentase rasio sebesar 95%.

Merujuk pada penelitian yang telah dilakukan dan tingkat akurasi yang diperoleh dari penggunaan metode ekstraksi ciri MFCC yang dikombinasi dengan PCA, maka penelitian ini mengusulkan pengembangan sistem identifikasi pembicara yang dilakukan pada situasi percakapan 2-3 orang, dengan menggunakan kombaini MFCC dan PCA dalam proses ekstraksi ciri untuk membentuk basisdata data ciri, dan menggunakan jaringan saraf tiruan dalam proses pengenalannya.

Penelitian ini bertujuan untuk mengidentifikasi pembicara dalam percakapan dengan teknik pencocokan data ciri dari pembicara dengan basisdata data ciri yang sudah dibentuk. Proses pencocokan akan membutuhkan proses pemisahan suara pembicara dalam percakapan terlebih dahulu. Teknik pemisahan suara pembicara akan menjadi luaran dari penelitian ini.

2. METODOLOGI PENELITIAN

Tahap penelitian yang dilakukan terdiri dari 4 tahap yaitu, tahap perekaman data suara 5 orang pembicara, tahap pengurangan noise pada hasil tangkapan data suara, tahap ekstraksi ciri suara dengan algoritma ekstraksi ciri, dan tahap identifikasi pembicara berdasarkan ciri suara yang tersedia. Algoritma ekstraksi ciri yang digunakan adalah MFCC dan PCA. Keunggulan

dari kedua algoritma ini adalah untuk mereduksi data ciri sehingga diperoleh kecepatan identifikasi dengan akurasi yang masih baik, sebagaimana terlihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Tahap pengumpulan data dilakukan dengan cara merekam suara dari 5 orang pembicara dengan mengucapkan kata yang berbeda sebanyak 10 kali pengambilan. Perekaman data ini akan menggunakan alat perekam suara dengan *noise*, dan perekaman suara dengan reduksi *noise*. Data yang sudah direkam akan dikurangi *noisenya* untuk dilakukan ekstraksi ciri. Reduksi *noise* hanya akan dilakukan pada hasil perekaman data dengan situasi perekaman suara pada lingkungan biasa, sedangkan perekaman data pada situasi dengan *noise* yang rendah tidak dilakukan reduksi *noise*.

Tahap ekstraksi ciri akan dilakukan pada data yang masih memiliki *noise*, data dengan *noise* yang rendah, dan data yang telah melalui reduksi *noise*. Ciri yang diperoleh akan digunakan pada proses klasifikasi untuk mengetahui data yang memberikan nilai akurasi yang tinggi. Tingkat akurasi ini digunakan untuk menjamin bahwa pengenalan suara akan benar. Data yang memberikan nilai akurasi yang tinggi pada saat proses pelatihan, akan digunakan sebagai data acuan untuk digunakan dalam proses identifikasi.

Tahap terakhir yang dilakukan pada penelitian ini adalah melakukan pemisahan suara yang terdapat dapat percakapan 2-3 orang dengan mengacu pada data acuan suara yang telah dimiliki. Tahap terakhir ini membutuhkan suatu model pemisahan suara oleh karenanya pada tahap akhir ini dilakukan pembuatan model untuk melakukan pemisahan suara pembicara pada percakapan.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Pengumpulan suara dilakukan dari 6 sukarelawan laki-laki dewasa dengan mengucapkan kalimat yang sudah ditentukan “Kebebasan adalah sesuatu yang harus Anda perjuangkan”, waktu pengucapan 2,9 detik, rekam suara dilakukan menggunakan program sederhana yang dikembangkan menggunakan Bahasa pemrograman *Python* seperti terlihat dibawah ini, dan *Audacity 2.4.2* yang merupakan perangkat lunak bebas pakai, format *.wav* digunakan sebagai kualitas suara dengan jumlah kanal 2 (stereo), lebar pita 16 bit, dan tingkat pengambilan sampel sebesar (fs) 44100 putaran per detik (Hz/s), sehingga jumlah data sampel per orang untuk sekali rekam adalah 127.890 (mono), jumlah ini didapat dari hasil perkalian antara waktu rekam dengan besar tingkat pengambilan suara yang digunakan.

Sintak Python untuk rekam suara:

```
import os
import sounddevice as sd
from scipy.io.wavfile import write
os.system('cls')
fs = 44100 # Sample rate
seconds = 3 # Duration of recording
val = input('Masukan nama file : ')
```

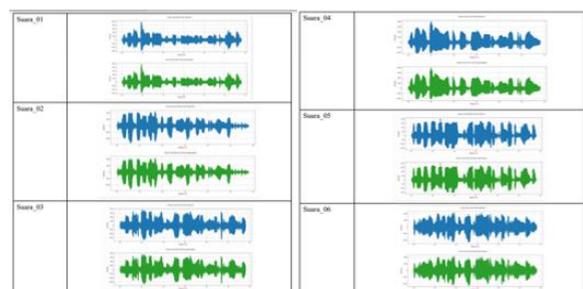
```
val = val + '.wav'
print(val)
print('Mulai Rekam Suara .....')
myrecording = sd.rec(int(seconds *
fs), samplerate=fs, channels=2)
sd.wait() # Wait until recording is
finished
## Save as WAV file
write("C:/DATA_JAYANTA/Croping/SUARA
/Rekaman/"+val, fs, myrecording)
print('Rekam Suara selesai
.....')
```

Rekam suara dilakukan 2 tahap, tahap pertama 3 relawan melakukan satu kali rekam suara bersama, dilakukan menggunakan satu mikrofon dan hasil rekam suara tersebut diberi label “Suara_Gb.wav”. Tahap kedua setiap relawan melakukan 3(tiga) kali perekaman, dan untuk setiap hasil rekaman diberi label, “Suara_02, Suara_02a, Suara_02b” untuk relawan pertama, “Suara_01, Suara_01a, Suara_01b” untuk relawan kedua, dan seterusnya.

Setelah rekam suara selesai dilakukan, langkah berikutnya membuat suara pembanding diberi label “audio_tx”, dengan proses berikut:

```
audio_00 = Suara_01
audio_tx = Suara_Gb
```

Gambar sinyal hasil rekam suara relawan, dan hasil proses untuk mendapatkan suara pembanding (“suara_tx”) dapat dilihat pada gambar berikut:



Gambar 2. Sinyal hasil rekam suara relawan

3.1 Reduksi Noise

Untuk mengurangi gangguan tingkat kebisingan suara (*noise*) terhadap hasil rekam sinyal suara, maka dilakukan proses reduksi *noise* menggunakan algoritma penapisan sinyal suara rendah (*Low-pass filter*), didalam penerapan *Low-pass filter* digunakan beberapa parameter yang dilibatkan, diantaranya adalah algoritma penapisan *Butterworth*, nilai frekuensi Nyquist. Dalam penelitian ini digunakan besaran nilai parameter untuk kebutuhan proses *Low-pass filter* yang dapat dilihat pada tabel 1.

Tabel 1. Nilai Parameter untuk Kebutuhan Proses Low-pass Filter

Periode Sampel (t)	: 2,9 detik
Tingkat Pengambilan Sampel (fs)	: 277088.47173 (rad/s) atau 44100 Hz
Frekuensi Batas yang Diinginkan dari filter	: 12.5664 (rad's) atau 2 Hz
Frekuensi Nyquist	: 0,5 * fs
Jumlah data Sampel	: Int (t*fs) = 127890 sampel

Implementasi Fungsi Low Pass Filter dalam bahasa pemrograman Python, adalah sebagai berikut:

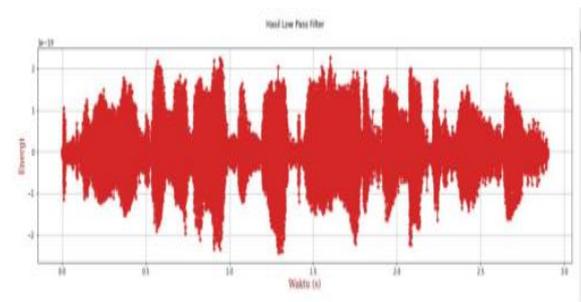
```
def butter_lowpass(cutOff, fs, order):
    nyq_frq = 0.5 * fs
    normalCutoff = cutOff / nyq_frq
    b, a = butter(order, normalCutoff, btype='low', analog = False)
    return b, a
def butter_lowpass_filter(data, cutOff, fs, order):
    b, a = butter_lowpass(cutOff, fs, order=order)
    y = lfilter(b, a, data)
    return y
cutOff = 12.5664 #cutoff frequency in rad/s = 2 Hz
```

```
fs = 277088.47173 #sampling
frequency in rad/s 277088.47173 = 44100 Hz
order = 6 # order of filter
```

Proses eksekusi fungsi low pass filter, dilakukan sebagai berikut:

```
Dt_low_00 =
(butter_lowpass_filter(audio_00, cutOff, fs, order))
Dt_low_tx =
(butter_lowpass_filter(audio_Gb, cutOff, fs, order))
```

Low pass filter dilakukan terhadap data suara relawan dan suara pembanding, contoh hasil filterisasi untuk Suara_06 dapat dilihat pada gambar 3.



Gambar 3. Hasil filterisasi untuk Suara_06

3.3. Ekstraksi Ciri

Setelah dilakukan penapisan terhadap sinyal suara, langkah selanjutnya dilakukan proses ekstraksi ciri untuk mendapatkan nilai dominan dari sinyal suara. *Mel Frequency Cepstral Coefficients* (MFCC) digunakan sebagai algoritme untuk ekstrasi ciri suara. Dalam penelitian ini digunakan besaran nilai parameter untuk kebutuhan proses *Mel Frequency Cepstral Coefficients* untuk mendapatkan nilai ciri suara dapat dilihat apa tabel 2.

Tabel 2. Nilai Parameter untuk Kebutuhan Proses Mel Frequency Cepstral Coefficients

Sinyal suara	:	2,9 detik
Tingkat Pengambilan Sampel (fs)	:	44100 Hz
Panjang Jendela analisis dalam hitungan detik (<i>winlen</i>)	:	0,06 ms (6 detik)
Langkah di antara jendela berturut-turut dalam hitungan detik (<i>winstep</i>)	:	0,02 ms (2 detik)
Jumlah cepstrum yang akan dikembalikan (<i>numcep</i>)	:	20
Jumlah filter di filterbank (<i>nflit</i>)	:	22
Ukuran FFT (<i>nfft</i>)	:	4096
Tepi pita terendah dari filter mel (<i>lowfreq</i>)	:	0
Tepi pita tertinggi dari filter mel (<i>highfreq</i>)	:	0,5 * fs
Apply preemphasis filter with preemph as coefficient (<i>preemph</i>)	:	0,97

Implementasi Fungsi Ekstraksi Ciri MFCC dalam bahasa pemrograman *Python* adalah sebagai berikut:

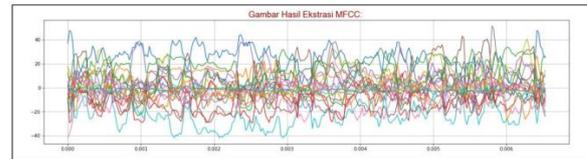
```
def Fung_MFCC(Dt_Sinyal):
    features_mfcc =
mfcc(Dt_Sinyal, sampling, winlen=0.060
, winstep=0.02, numcep=13,
nflit=22, nfft=4096, lowfreq=0,
highfreq = (sampling/2),
preemph=0.97, ceplifter=22,
appendEnergy=False, winfunc =
np.hamming)
return features_mfcc
```

Proses eksekusi fungsi Ekstraksi Ciri MFCC, dilakukan sebagai berikut:

```
Vek_mfcc_00 = (Fung_MFCC(Dt_low_00))
Vek_mfcc_tx = (Fung_MFCC(Dt_low_tx))
```

Ekstraksi Ciri dilakukan terhadap data suara relawan dan suara pembanding, ukurandata

hasil ekstraksi MFCC adalah $288 \times 20 = 5.760$, salah satu hasil ekstraksi teknik MFCC untuk Suara_01 dapat dilihat pada gambar 4.



Gambar 4. Hasil Ekstraksi Teknik MFCC

Dalam penelitian ini memanfaatkan teknik ekstraksi ciri PCA untuk mereduksi ukuran data yang dihasilkan Teknik MFCC. Implementasi PCA terhadap hasil MFCC dalam Bahasa program Python, sebelum dilakukan proses PCA terlebih dahulu dilakukan pembobotan (penskalaan) terhadap hasil ekstraksi ciri MFCC, dengan instruksi Python sebagai berikut:

```
sc = StandardScaler()
X = sc.fit_transform(Vek_mfcc_00)
```

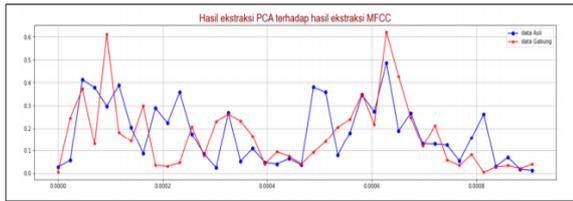
Kemudian hasil pen-skalaan yang berbentuk matriks diubah menjadi vector dengan memanfaatkan fungsi *numpy* dalam *Python* yaitu “*reshape(-1)*”, implementasinya fungsi tersebut adalah sebagai berikut:

```
X = X.reshape(-1)
```

Selanjutnya hasil *reshape()* dijadikan masukkan pada proses PCA, implementasi fungsi PCA adalah sebagai berikut:

```
pca = PCA(n_components=2)
fit = pca.fit(X)
data_01 = fit.components_
```

Implementasi fungsi PCA menghasilkan ukuran data sebesar 40 data terhadap hasil MFCC, salah satu hasil fungsi PCA terhadap MFCC dapat dilihat pada gambar 5.



Gambar 5. Hasil Ekstraksi PCA terhadap hasil ekstraksi MFCC

Setelah dilakukan ekstraksi ciri dengan Teknik MFCC, juga ekstraksi ciri dari kombinasi secara serial terhadap metode MFCC dengan PCA, maka ukuran data dapat tereduksi dengan signifikan, ketika terhadap hasil ekstraksi ciri MFCC, dan hasil ekstraksi ciri MFCC dengan PCA dilakukan operasi fungsi cross correlation dan convolve, maka banyaknya jumlah data meningkat 49.998% untuk data ciri MFCC, sedangkan untuk data ciri MFCC dengan PCA juga mengalami peningkatan jumlah data sebesar 49.367%, hasil reduksi data tersebut dapat dilihat pada tabel 3.

Tabel 3. Hasil Ekstraksi Data

Ukuran data asli	Tanpa konvolusi cross correlation		Setelah konvolusi menggunakan hasil cross correlation	
	MFCC	MFCC + PCA	MFCC	MFCC + PCA
127.890 (mono)	5760	40	11519	79

3.4. Pemodelan Sistem Identifikasi Percakapan

Perhitungan jarak euclidean dan nilai similarity digunakan untuk mengetahui tingkat kemiripan antara sinyal suara perorangan (relawan) dengan sinyal suara gabung (“Suara_Gb”), jika nilai jarak Euclidean sama dengan nol (0), dan nilai similarity sama dengan 1, memberi arti bahwa kedua sinyal tersebut identik sama. *Cosine similarity* merupakan

rumus fungsi similarity yang digunakan untuk mendapatkan nilai kemiripan bila diterapkan dalam instruksi bahasapemrogram python adalah sebagai berikut:

```
def cos_similar(x, y):
    return np.dot(x, y) /
        (np.sqrt(np.dot(x, x)) *
         np.sqrt(np.dot(y, y)))
```

Untuk mengeksekusi fungsi tersebut adalah sebagai berikut:

```
cos_dst_mfcc =
cos_similar(Rshp_mfcc_tx, Rshp_mfcc_00)
```

Sedangkan untuk mengeksekusi jarak Euclidean adalah sebagai berikut:

```
d_Tx_mfcc =
(distance.euclidean(Rshp_mfcc_tx,
Rshp_mfcc_00))
```

Dalam mengetahui tingkat kemiripan sinyal juga digunakan *algoritme cross correlation*, eksekusi proses perhitungan *cross Correlation* adalah sebagai berikut:

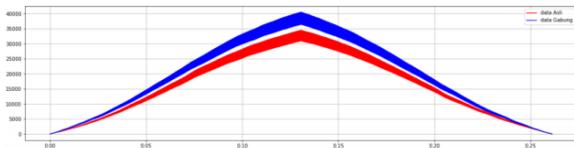
```
Dt_Corr =
(signal.correlate(Rshp_mfcc_00,
Rshp_mfcc_tx, mode='same') /
len(Rshp_mfcc_tx))
```

Kemudian nilai dari cross correlation dikonvolusikan dengan nilai ciri suara relawan dan suara gabung, hal ini dilakukan juga untuk mengetahui nilai kemiripan dari kedua sinyal tersebut, proses eksekusinya adalah sebagai berikut:

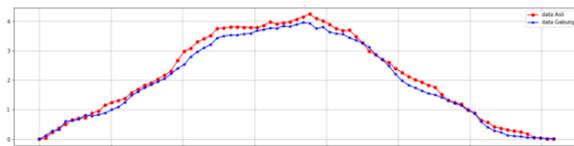
```
Rshp_mfcc_tx =
np.convolve(Rshp_mfcc_tx *
Dt_Corr)
Rshp_mfcc_00 = np.convolve
(Rshp_mfcc_00 * Dt_Corr)
```

Satu contoh hasil konvolusi untuk hasil ekstraksi ciri MFCC, dan ekstraksi ciri PCA

terhadap Hasil ekstraksi ciri MFCC dapat dilihat pada gambar 6.



Gambar 6. Hasil Ekstraksi MFCC dengan Cross Correlation



Gambar 7. Grafik Hasil Ekstraksi PCA terhadap hasil ekstraksi MFCC

Kemudian langkah selanjutnya melakukan proses perhitungan jarak Euclid dan similarity terhadap hasil – hasil ekstraksi ciri dengan memanfaatkan perhitungan *cross correlation*, eksekusi proses tersebut adalah sebagai berikut:

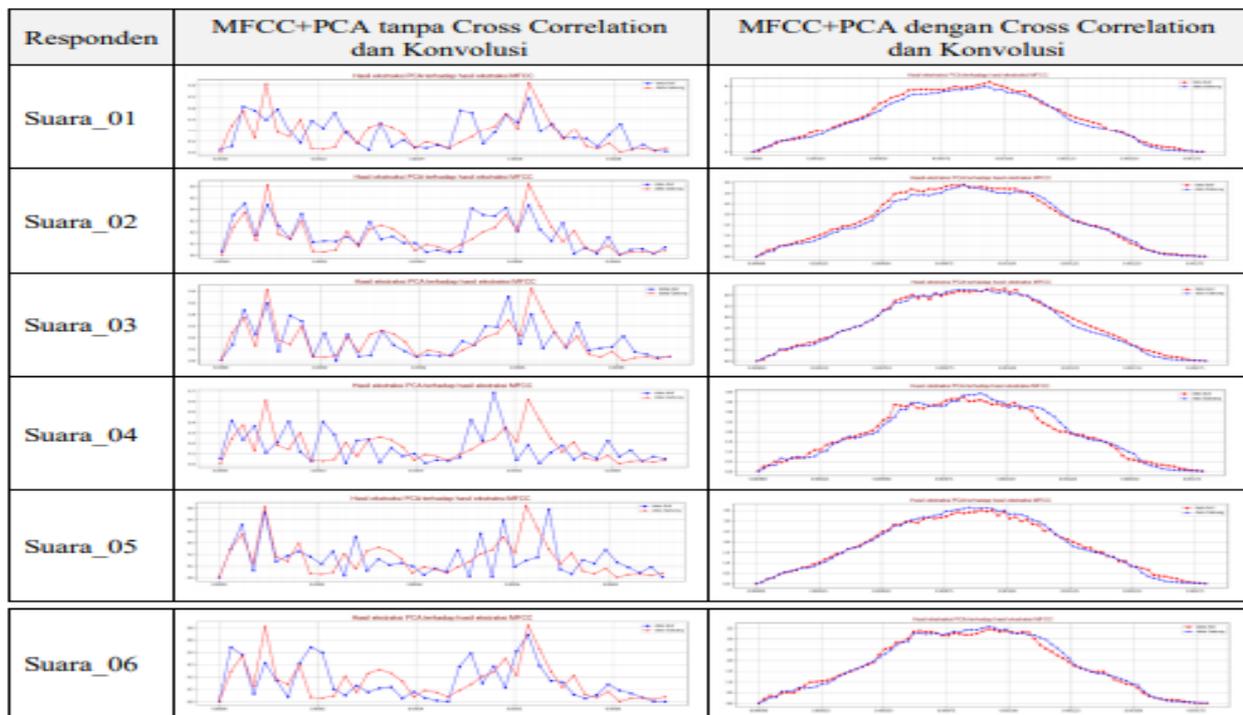
```
d_Corr_mfcc =
(distance.euclidean(Rshp_mfcc_tx,
Rshp_mfcc_00))
cos_dst_mfcc =
cos_similar(Rshp_mfcc_tx,Rshp_mfcc_0
0)
```

Hasil dari proses di atas baik dengan memanfaatkan *cross correlation* maupun tidak, dapat dilihat pada tabel 4. Dari tabel tersebut diketahui bahwa penerapan teknik *cross correlation* dan konvolusi dapat meningkatkan kemiripan semua sinyal yang ada, meskipun jarak rata-rata memberikan nilai yang buruk untuk hasil ekstraksi ciri MFCC, hal tersebut dapat terkoreksi dengan baik ketika teknik *cross correlation* dan konvolusi digunakan. Dari tabel diatas juga memperlihatkan bahwa semua suara dapat kita gunakan untuk mendekati suara pembanding (“suara_tx”), hanya suara_06 memberikan nilai kemiripan dan nilai rata-rata jarak Euclid yang lebih baik tanpa menggunakan *cross correlation* dan convolusi. Berikut adalah gambar salah satu grafik untuk hasil perhitungan jarak Euclid dan nilai *similarity* setelah dilakukan perkalian dengan hasil *cross correlation* pada hasil ekstraksi ciri MFCC dan hasil ekstraksi PCA terhadap hasil ekstraksi MFCC.

Tabel 4. Hasil Penerapan Teknik Cross Correlation dan Konvolusi

	Tanpa cross correlation					
	Suara 01	Suara 02	Suara 03	Suara 04	Suara 05	Suara 06
Nilai Jarak Euclid						
MFCC	785.2942 49949660 6	834.646 0281742 195	864.291 2122566 097	814.389 5109614 318	871.12 569745 1292	886.082 7206209 361
MFCC + PCA	0.941469 59589781 76	0.62915 1809979 1178	0.70078 9554716 6781	1.25412 6389164 9836	0.9591 954962 362991	0.91317 2491920 3381
Similarity						
MFCC	0.752450 54244340 8	0.73346 6806721 8356	0.71134 3749150 9614	0.75471 5302470 8405	0.7296 879197 277742	0.66167 8982474 2138
MFCC + PCA	0.778297 18155198 11	0.90103 5048899 4064	0.87723 1796072 7133	0.60686 6535334 7603	0.7699 147054 985093	0.79141 3443814 386
	Dengan signal correlation					
	Suara 01 Adiat	Suara 02 Dhimas	Suara 03 Ikmal	Suara 04 Tymmi	Suara 05 Rahmat	Suara 06 Gusti
Nilai Jarak Euclid						
MFCC	32.07159 43198337 4	9.39389 7135791 553	18.4030 3855731 6567	9.86910 6224844 659	4.5337 833724 41597	65.2000 4667960 426
MFCC + PCA	0.021164 88649781 799	0.01735 8886906 56719	0.01343 1104617 574068	0.02614 7708244 835553	0.0144 084384 445543 46	0.01484 2054282 43288
Nilai Similarity						
MFCC	0.999942 34583044 7	0.99964 3120169 939	0.99995 2651443 0203	0.99923 2222446 7245	0.9998 079797 52763	0.99962 1212807 5045

Gambar 8. Hasil Perhitungan Jarak Euclid dan Nilai *Similarity* setelah dilakukan Perkalian dengan Hasil *Cross Correlation* pada Hasil Ekstraksi Ciri MFCC dan Hasil Ekstraksi PCA terhadap Hasil Ekstraksi MFCC.



Setelah diketahui tingkat kemiripan dan jarak *Euclidean* nya, selanjutnya dilakukan pemisahan suara, teknik pemisahan suara yang dilakukan dalam penelitian ini adalah melakukan penguatan tingkat kejelasan bunyi satu suara dari suara-suara lain dalam suara yang dijadikan pembandingan. Implementasi proses tersebut adalah sebagai berikut:

```
audio_Gbx = audio_Tx - audio_00 -
audio_00 - audio_00
wavio.write("C:/DATA_JAYANTA/Croping/suara_xpr.wav", audio_Gbx,
sampling, sampwidth=3)
Path_Data_Suara_Gb =
"C:/DATA_JAYANTA/Croping/suara_xpr.wav"

wave_obj =
sa.WaveObject.from_wave_file(Path_Data_Suara_Gb)
play_obj = wave_obj.play()
play_obj.wait_done()
```

Dari proses di atas kita akan memperkuat suara "audio_00", dan melemahkan atau memburamkan suara lain yang ada pada sumber suara pembandingan "audio_Gbx", ketika "data_00" dihitung nilai kemiripan dan jarak *Euclidean* nya terhadap data pembandingan "audio_Gbx" didapat nilai sebagai berikut 0.9983490473836057 untuk nilai kemiripan sinyal, 0.018403744328999 723 untuk nilai jarak. Ketika dihitung kemiripan dan jarak antara "audio_Gbx" dengan "audio_Tx" tidak menghasilkan nilai 1.00 dan 0.00, tetapi menghasilkan nilai 0.9998297520126006 (0.9998972050902559) untuk nilai kemiripan dan 0.004282893411371005 (0.003811924221705252) untuk nilai jarak, hal ini menginformasikan bahwa ada pengaruh operasi pengurangan "audio_00" terhadap "audio_Tx".

4. KESIMPULAN

Penerapan kombinasi Teknik ekstraksi ciri MFCC dan PCA dengan proses *Cross Correlation* dan konvolusi terhadap data asli dapat mereduksi ukuran atau jumlah data sebesar 99%, tanpa mengurangi hasil perhitungan tingkat similarity dan jarak Euclidean nya, jika dibandingkan nilai hasil perhitungan *similarity* dan jarak Euclidean yang dilakukan terhadap hasil ekstraksi ciri MFCC dan PCA tanpa proses *cross correlation* dan konvolusi. Penelitian perlu dilanjutkan dengan data memanfaatkan kombinasi data relawan untuk mengetahui tingkat kemirip dan jaraknya euclidean apakah lebih baik, jika dibandingkan dengan hasil yang telah dilakukan dalam penelitian ini.

DAFTAR PUSTAKA

- [1] Rababah, H., & Al-Taani, A. T, 2017, *An automated scoring approach for Arabic short answers essay questions*, In 2017 8th International Conference on Information Technology (ICIT) (pp. 697-702), IEEE.
- [2] Adhitia, R., & Purwarianti, A, 2009, Penilaian Esai Jawaban Bahasa Indonesia Menggunakan Metode SVM-LSA dengan Fitur Generik, *Jurnal Sistem Informasi*, 5(1), 33-41.
- [3] Roy, S., Narahari, Y., & Deshmukh, O. D, 2015, *A perspective on computer assisted assessment techniques for short free-text answers*, In International Computer Assisted Assessment Conference (pp. 96-109), Springer, Cham.
- [4] Yannakoudakis, H., Briscoe, T., & Medlock, B, 2011, *A new dataset and method for automatically grading ESOL texts*, In Proceedings of the 49th annual meeting of the association for computational linguistics: human language

technologies (pp. 180-189).

- [5] Suresh, A., & Jha, M, 2018, *Automated essay grading using natural language processing and support vector machine*, *International Journal of Computing and Technology*, 5(2), 18-21.
- [6] F. Rahutomo et al., 2018, *Open Problems in Indonesian Automatic Essay Scoring System*, *Int. J. Eng. Technol.*, vol. 7, no. 4.44, p. 156.
- [7] Aji, R. B., Baizal, Z. A., & Firdaus, Y, 2011, *Automatic essay grading system menggunakan metode Latent Semantic Analysis*, In Seminar Nasional Aplikasi Teknologi Informasi (SNATI).