PERBANDINGAN METODE EKSTRAKSI CIRI *ROUGH SETS-K-MEANS* DAN *K-MEANS* PADA OPTIMASI KASUS PENGENALAN SUARA

Gelar Budidarma Djamal

Program Studi Teknik Informatika
Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia
Jl. Dipati Ukur 114 Bandung
E-mail : gelarbudidarma@gmail.com

ABSTRAK

Speaker verification adalah proses pengenalan suara untuk memverifikasi seorang pembicara. Untuk dapat melakukan *speaker verification*, data suara harus melalui proses ekstraksi ciri suara. Pada penelitian sebelumnya, *k-means* dapat digunakan untuk menambah akurasi pada kasus pengenalan suara. Persentase hasil pengujian adalah sebesar 79.375%. Di samping itu, terdapat penelitian yang membandingkan *K-Means*, PAM, *Rough Sets-K-Means* menggunakan *datasets* kanker leukemia. Hasilnya diketahui bahwa *Rough Sets-K-Means* dan PAM memiliki akurasi yang lebih baik dalam pengelompokan *datasets* kanker dibandingkan dengan *k-means*. Oleh karena itu, dalam penelitian ini akan dilakukan perbandingan *rough sets-k-means* dengan *k-means* untuk mengetahui *algoritma* yang lebih baik digunakan dalam *optimasi* pengenalan suara.

Sebelum data suara diproses, terlebih dahulu data suara diekstraksi menggunakan metode MFCC. Hasil ekstraksi ciri MFCC dikelompokkan menggunakan *rough sets-k-means* dan *k-means*. Hasilnya kemudian disimpan pada *database*. Data hasil ekstraksi ciri diklasifikasi menggunakan LVQ. Pengujian akurasi menggunakan *k-fold cross validation* dengan nilai k=10.

Untuk pengujian akurasi digunakan 180 data latih dan 20 data uji yang berasal dari 5 orang *speaker*. Hasil pengujian akurasi menunjukkan bahwa rough sets-k-means memiliki tingkat persentase lebih baik dibandingkan *k-means* pada kasus pengenalan suara, yaitu sebesar 80%.

Kata kunci : *speaker verification, suara, MFCC, k-means, rough sets-k-means, clustering learning vector quantization, perbandingan algoritma, k-fold cross validation*.

1. PENDAHULUAN

Speaker verification adalah proses verifikasi seorang pembicara, di mana sebelumnya telah diketahui identitas pembicara tersebut berdasarkan data yang telah di inputkan. Untuk dapat melakukan speaker verification, data suara akan melalui proses ekstraksi ciri suara untuk mendapatkan informasi yang terkandung di dalam data suara [1]. Ekstraksi ciri yang digunakan pada penelitian ini adalah metode *Mel Frequency Cepstrum Coefficients* (MFCC), karena MFCC memiliki keunggulan dalam menangkap karakteristik suara dibandingkan dengan ZPC (*Zero Crossings with Peak Amplitudes*) dan LPC (*Linear Predictive Coding*) [1].

Penelitian sebelumnyan yang menerapkan metode MFCC pada kasus verifikasi biometrika suara menyatakan MFCC mempunyai kekurangan pada hasil filter dari MFCC karena pengaruh dari panjang sinyal suara pada nilai-nilai parameter yang digunakan sangat mempengaruhi baik buruknya hasil dari proses MFCC itu sendiri. [2], [3]. Oleh karena itu *clustering* dapat membuat beberapa vektor pusat sebagai wakil dari keseluruhan vektor data yang ada pada hasil parameter MFCC. salah satunya yang telah di uji dengan menggunakan *k-means clustering* pengujian hanya mendapatkan tingkat persentase rata-rata akurasi sebesar 79,375% dengan menggunakan *k-means clustering* [4].

Di samping itu terdapat pula penelitian perbandingan *K-Means*, PAM, *Rough Sets-K-Means*, dengan *datasets cancer leukemia*, *Clustering* digunakan untuk mengklasifikasi *datasets* yang terdiri dari 3859 gen dan satunya dengan 50 gen, di karena kompleksitas pola gen leukemia yang berbeda, ketika diterapkan ketiga algoritma *k-means*, PAM, dan *rough sets-k-means* untuk mengklasifikasi *datasets* gen leukemia, dinyatakan bahwa *rough k-means* dan PAM memiliki akurasi lebih baik dalam mengklasifikasi pola gen *datasets* cancer dibandingkan dengan *k-means* [5]. *Rough K-Means* memungkinkan akan memiliki akurasi yang lebih baik dibandingkan *k-means* pada kasus optimasi pengenalan suara.

Berdasarkan uraian di atas, pada penelitian ini akan diterapkan perbandingan metode ekstraksi ciri pada *Rough Set-K-Means* dengan *K-Means* pada optimasi kasus pengenalan suara. Pada tahap ekstraksi fitur menggunakan MFCC sebagai ekstraksi ciri suara yang berguna bagi proses

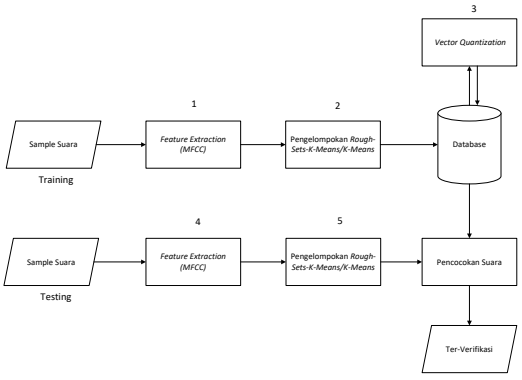
pengenalan suara. *Rough Set-K-Means* dengan *K-Means* digunakan untuk mengoptimasi hasil pada filter MFCC. *Learning Vector Quantization* digunakan sebagai klasifikasi untuk mendapatkan *model pattern*. dan membandingkan hasil optimasi akurasi *K-Means* dengan *Rough Sets-K-Means.*

2. MODEL ANALISIS DAN DESAIN IMPLEMENTASI

Untuk melihat perbandingan dari penggunan metode *k-m*eans dan *rough sets-k-means* pada kasus pengenalan suara, maka berikut ini adalah analisis proses dari sistem pengenalan suara menggunakan fitur ekstarksi Metode *Mel Frequency Cepstrum Coefficients* (MFCC), yang meliputi proses latih (training mode) dan proses pengujian (testing mode).

Training Mode adalah bagian yang bertugas mempelajari dan memodelkan suara pembicara. Pembelajaran terhadap suara pembicara dilakukan dengan membuat model dari suara pembicara kemudian menyimpannya ke dalam database.

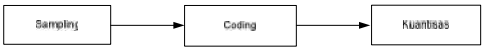
Testing Mode merupakan bagian yang bertugas mengenali suara pembicara. Pengenalan suara dilakukan dengan melakukan pencocokan data uji dengan data model pattern pada database. Pada pencocokan, perbandingan yang dilakukan adalah perbandingan 1:1 dimana model suara dari A langsung dibandingkan dengan hasil pembelajaran pemodelan suara A yang ada dalam database. Berikut alur proses pengenalan suara yang dapat dilihat pada gambar 1.



Gambar 1. Alur Proses Pengenalan Suara

2.1 Pre Processing

Sinyal suara yang akan diproses bersifat analog sehingga jika akan dilakukan pengolahan secara digital, sinyal suara tersebut harus dikonversi menjadi sinyal digital, berupa urutan angka dengan tingkat presisi tertentu yang dinamakan *analog to digital conversion* dengan menggunakan *analog-todigital converter* (ADC). Konsep Kerja ADC terdiri dari tiga proses :



Gambar 2. Konsep Kerja Analog Digital Conventer

Keterangan konsep kerja ADC :

1. *Sampling* adalah konversi sinyal kontinu dalam domain waktu menjadi sinyal diskrit, melalui proses *sampling* sinyal pada selang waktu tertentu. Sehingga jika $x_0(t)$ adalah sinyal input, maka outputnya adalah $x_0(nT)$, dengan T adalah interval *sampling*.
2. Kuantisasi adalah proses untuk membulatkan nilai data kedalam bilangan bilangan tertentu yang telah ditentukan terlebih dahulu.
3. *Coding*, pada proses ini, tiap nilai diskrit yang telah didapat, direpresentasikan dengan angka binary n-bit.

2.2 Pre Processing

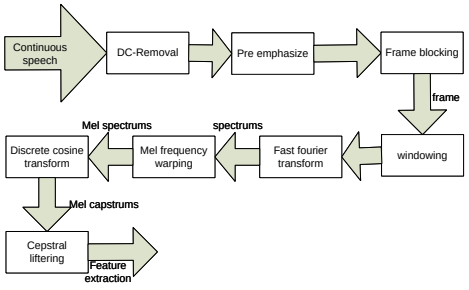
Dalam analisis ini digunakan contoh dari *dataset voice* terdiri dari 5 *speaker*, yang diambil dalam waktu 1 detik dan kata yang di ucapkan adalah kata "A" dalam intonasi yang berbeda.

$$X = FS\ x\ dt(detik)\ x\ \left(\frac{bit}{8}\right) x\ j \quad (1)$$

X = Data sampling
FS = Frekuensi sampling
dt = durasi rekaman (detik)
bit = jumlah bit resolusi
j = 1 untuk *mono*, dan 2 untuk *stereo*

2.3 MFCC

Metode MFCC terdiri dari 8 proses yaitu *dc removal, pre-emphasize, frame blocking, windowing, fast fourier transfrom (*FFT*), filterbank, discrete cosine transform* (DCT) dan *cepstral liftering* proses dapat dilihat pada gambar 3.



Gambar 3. Proses metode Mel Frequency Cepstrum Coefficients (MFCC)

2.3.1 DC Removal

DC Removal Components bertujuan untuk menghitung rata-rata dari data sampel suara, dan mengurangkan nilai setiap sampel suara dengan nilai rata-rata tersebut. Tujuannya adalah mendapat normalisasi dari data suara input [2].

$$D[n] = s[n] - \bar{x} \ , 0 \leq n \leq N-1 \ (2)$$

Dimana :
D[n] = sampel signal hasil proses DC removal
s[n] = sampel signal asli
$\bar{x}$ = nilai rata-rata sampel signal asli.
N = panjang *signal*

2.3.2 Pre Emphasize

Pre emphasize merupakan salah satu jenis *filter* yang sering digunakan sebelum sebuah *signal* diproses lebih lanjut. *Filter* ini mempertahankan frekuensi-frekuensi tinggi pada sebuah spektrum, yang umumnya tereliminasi pada saat proses produksi suara [2].

Dimana $0.9 \leq \alpha \leq 1.0$, dan $\alpha \in R$. Formula diatas dapat dijadikan sebagai *first order differentiator*, sebagai berikut:

$$y[n] = s[n] - \alpha\, s[n-1] \ (4)$$

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

Pada umumnya nilai α yang paling sering digunakan adalah antara 0.9 sampai 1.0.

2.3.3 Frame Blocking

Hasil perekaman suara merupakan sinyal analog yang berada dalam domain waktu yang bersifat variant time, yaitu suatu fungsi yang bergantung waktu. Oleh karena itu sinyal tersebut harus dipotong-potong dalam slot-slot waktu tertentu agar dapat dianggap invariant. Sinyal suara dipotong sepanjang 20 milidetik .Setiap potongan tersebut disebut *frame*. Untuk menghitung jumlah Frame digunakan rumus

$$\textit{Jumlah frame} = ((I-N)/M)+1) \ (5)$$

Dimana:
I = Sample rate
N = Sample point (Sample rate * waktu framing(s))
M = N/2

2.3.4 Windowing

Proses *framing* dapat menyebabkan terjadinya kebocoran spektral (*spectral leakage*) atau *aliasing*. *Aliasing* adalah *signal* baru dimana memiliki frekuensi yang berbeda dengan *signal* aslinya. Efek ini dapat terjadi karena rendahnya jumlah *sampling rate*, ataupun karena proses *frame blocking* dimana menyebabkan *signal* menjadi *discontinue.* Untuk mengurangi kemungkinan terjadinya kebocoran spektral, maka hasil dari proses *framing* harus melewati proses *window* [2].

Fungsi pada proses windowing ini menggunakan fungsi *Hamming Window* sebagai berikut :

$$w(n) = 0.54 - 0.46\cos\frac{2\pi n}{M-1} \ (6)$$

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

2.3.5 Fast Fourier Transfrom

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

$$S[k] = \sum_{n=0}^{N-1} s[n]\, e^{-\frac{j2\pi nk}{N}}, 0 \leq k \leq N-1 \ (7)$$

Dimana:
N = jumlah sampel yang akan diproses ($N \in$ N)
S(n) = nilai sampel *signal*
K = variable frekuensi discrete

2.3.6 Filterbank

Filterbank menggunakan representasi konvolusi dalam melakukan filter terhadap sinyal. Konvolusi dapat dilakukan dengan melakukan multiplikasi antara spektrum sinyal dengan koeffisien *filterbank*. Berikut ini adalah rumus yang digunakan dalam perhitungan *filterbank*s. Berikut perhitungan untuk *filterbank* :

$$Y[i] = \sum_{j=1}^{N} S[j]H_i[j] \ (8)$$

N= jumlah *magnitude spectrum* ($N \in$ N)
S[j] = magnitude spectrum pada frekuensi j
$H_i[j]$ = koefisien *filterbank* pada frekuensi j

2.3.7 Discrete Cosine Transfrom

Proses ini merupakan langkah akhir dari *feature extraction*. Hasil dari DCT ini adalah fitur-fitur yang dibutuhkan oleh penulis untuk melakukan proses

analisa terhadap pengenalan suara tersebut. Menggunakan rumus:

$$C_n = \sum_{k=0}^{k} (logS_k)\cos\left[n\left(k-\frac{1}{2}\right)\frac{\pi}{k}\right] \quad (9)$$

Dimana
k = koefisien 10

2.3.8 Cepstral Liftering

Hasil dari proses utama MFCC *feature extraction* memiliki beberapa kelemahan. *Low order* dari *cepstral coefficients* sangat sensitif terhadap *spectral slope,* sedangkan bagian *high order*nya sangat sensitif terhadap *noise.* Oleh karena itu, *cepstral liftering* menjadi salah satu standar teknik yang diterapkan untuk meminimalisasi sensitifitas tersebut. *Cepstral liftering* dapat dilakukan dengan mengimplementasikan fungsi *window* terhadap *cepsral features* [4].

$$w[n] = \left\{1 + \frac{L}{2}\sin\left(\frac{n\pi}{L}\right)\right\} \quad (10)$$

Dimana:
L = jumlah *cepstral coefficients*
n = *index* dari *cepstral coefficients*

2.4 K-Means Clustering

Clustering merupakan faktor yang paling fundamental dalam *pattern recognition*. Masalah utama dari *clustering* adalah mendapatkan beberapa nilai vektor pusat yang dapat mewakili keseluruhan vektor dari hasil *feature extraction*. K-means *clustering* adalah salah satu metode yang digunakan untuk mempartisi vektor hasil *feature extraction* ke dalam k vektor pusat [6].

K-Means Clustering adalah proses memetakan vektor-vektor yang berada pada lingkup wilayah yang luas besar menjadi sejumlah tertentu (k) vektor. Wilayah yang terwakili oleh vektor pusat hasil dari proses kuantisasi disebut sebagai *cluster*. Sebuah vektor pusat hasil dari proses kuantisasi dikenal sebagai *codewords*. Sedangkan kumpulan dari vektor pusat dikenal *sebagai codebooks* [6]

2.5 Rough Sets-K-Means Clustering

Rough Sets-K-Means algoritma adalah penggunaan teori *rough set* dan algoritma *k-means* untuk menangani tidak pastian yang terlibat pada analisis cluster [5].

Langkah pada *Rough Sets-K-Means* sebagai berikut:

1. Pilih *cluster* awal n objek ke dalam *cluster* k.
2. Tetapkan setiap objek pada batas *lower* (L(x)) atau batas *upper* (U(x))
   pada *cluster* atau masing-masing cluster sebagai:

   Pada setiap objek v, biarkan d(v,xi) menjadi jarak antara sendirinya dan *cluster* pada *centroid* xi, perbedaan antara d(v,xi) - d(v,xj), 1≤ i, j ≤ k digunakan untuk menentukan anggota dari v, sebagai berikut:
   a. Apabila d(v,xi) - d(v,xj) ≤ awal, maka v∈U(xi) & v∈U(xj). Selanjutnya, v tidak akan menjadi bagian dari setiap batas *lower*.
   b. Jika tidak, v∈L (x), sehingga d (v, x) adalah minimum untuk 1≤ i ≤ k. Selain itu, v∈U (xi).

$$C_{i=}\begin{cases} W_{lower}\times\frac{\sum_{a\in L(c)}A_j}{|L(c)|} + W_{upper}\times\frac{\sum_{a\in U(c)-L(c)}A_j}{|U(c)-L(c)|} \\ \qquad if\ |U(c)-L(c)\neq\phi| \\ W_{lower}\times\frac{\sum_{a\in L(x)}A_j}{|L(C)|} \quad otherwise \end{cases} \quad (11)$$

Keterangan:
C(n) = *index* pada centroid
U(c) = *centroid Upper*
L(c) = *centroid Lower*
W(lower) = batas bawah
W(Upper) = batas atas

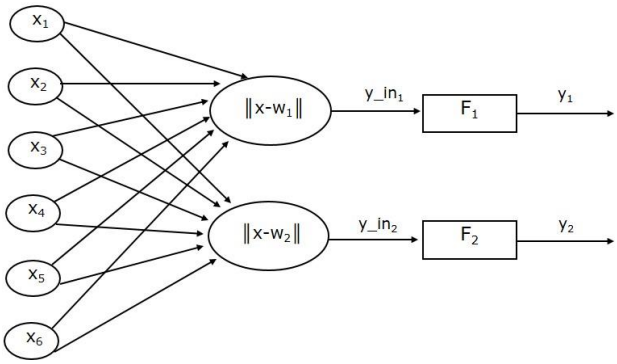
3. Pada setiap cluster xi hitung ulang pusat menurut persamaan berikut kombinasi yang tertimbang dari data point pada
   Jika tidak dimana 1≤ j ≤ k. *Parameter Wlower* dan *Wupper* menyerupai dengan kepentingan berbeda pada batas *lower* dan batas *upper*

Jika kriteria konvergen terpenuhi, yaitu pusat *cluster* yang sama dengan yang di iterasi sebelumnya, kemudian berhenti, kembali ke langkah 2.

2.6 Learning Vector Quantization

Learning Vector Quantization (LVQ) adalah suatu metode untuk melakukan pembelajaran atau pelatihan pada lapisan kompetitif yang terawasi. LVQ belajar mengklasifikasikan vektor masukan ke kelas target yang ditentukan oleh pengguna [7].

Jaringan LVQ terdiri dari dua lapisan tersembunyi yaitu lapisan kompetitif dan lapisan linear. Lapisan kompetitif disebut juga *Self Organizing Map* (SOM). Disebut lapisan kompetitif karena neuron-neuron berkompetisi dengan algoritma kompetisi yang akan menghasilkan neuron pemenang (*winning neuron*). Pada jaringan LVQ, bias pada lapisan kompetitif dihilangkan. Lapis linear mengalih ragamkan kelas-kelas pada lapisan kompetitif ke klasifikasi target yang ditentukan oleh pengguna [7]. Arsitektur LVQ dapat dilihat pada Gambar 4 berikut



Gambar 4. Arsitektur LVQ

2.7 Euclidean Distance

Euclidean distance adalah sebuah metode yang digunakan untuk mengukur jarak (*distance*). *Euclidean distance* sebenarnya merupakan generalisasi dari teorema phytagoras. *Euclidean distance* diformulasikan pada persamaan berikut:

$$d(x,y) = \sqrt{\sum_{i=1}^{r}(x_i - y_i)^2} \quad (12)$$

d = jarak antara x dan y
r = jumlah fitur dalam vektor
x_i, y_i = nilai fitur ke i dari x dan y

Jarak minimal yang mungkin antara dua vektor data adalah 0. Selain itu, jarak dari *x* ke *y* akan sama dengan *y* ke *x*, $d(x,y) = d(y,x)$ [6].

3. PENGUJIAN

Berikut adalah pengujian pada *clustering k-means* dan *rough sets-k-means* yang dilakukan menggunakan *k-fold cross validation* dengan nilai k=10. Nilai 10 menunjukan jumlah dataset (D), dimana dataset (D) dibagi menjadi 10 bagian yaitu D1, ...D10 dan iterasi (pengujian) dilakukan sebanyak sepuluh kali.

Pada pengujian ini digunakan nilai parameter yang optimal yang telah dilakukan pada pengujian parameter yaitu *learning rate* α=0,001, penurunan *learning rate* dec α=0,001 dan maksimal epoh =100 digunakan 180 data latih dan 20 data uji.

3.1 Pengujian K-Means

Setelah dilakukan pengujian dengan menggunakan *10-fold cross validation* pada *cluster k-means* yang pengujiannya dilakukan sebanyak 10 kali, maka didapatkan rekap hasil pengujian sebagai berikut pada tabel 1.

Tabel 1. Hasil Pengujian pada *K-Fold Cross Validation K-Means*

Pengujian	True	False	Akurasi (%)
1	14	6	70
2	16	4	80
3	15	5	75
4	14	6	70
5	16	4	80
6	14	6	70
7	15	5	75
8	16	4	80
9	15	5	75
10	15	5	75
Rata-rata akurasi (%)			75%

3.1 Pengujian Rough Sets-K-Means

Setelah dilakukan pengujian dengan menggunakan *10-fold cross validation* yang pengujiannya dilakukan sebanyak 10 kali, maka didapatkan rekap hasil pengujian sebagai berikut pada tabel 2.

Tabel 2. Hasil Pengujian *K-Fold Cross Validation Rough Sets-K-Means*

Pengujian	True	False	Akurasi (%)
1	17	3	85
2	17	3	85
3	18	2	90
4	18	2	90
5	15	5	75
6	14	6	70
7	15	5	75
8	17	3	85
9	15	5	75
10	14	6	70
Rata-rata akurasi (%)			80%

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan pada penelitian ini, maka diperoleh kesimpulan bahwa perbandingan antara rough sets-k-means dan k¬-means pada ektrasksi ciri suara MFFC dapat disimpulkan dengan menggunakan nilai k (k=6) dan nilai parameter pada pembelajaran learning vector quantization menggunakan parameter learning rate (α=0,001), penurunan learning rate (dec α=0,1) dan maksimum epoh=100 digunakan karena pengujian akurasi terbaik pada

penggunaan datasets yang digunakan, tetapi menggunakan parameter berbeda dapat memungkinkan dapat mengubah hasil akurasi, sehingga dapat disimpulkan Bahwa rough sets-k-means memiliki akurasi lebih baik pada kasus pengenalan suara dengan persentase 80%, dan k-means hanya memiliki persentase 75%. Maka dapat disimpulkan bahwa rough sets-k-means lebih cocok digunakan untuk kasus pengenalan suara.

4.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, penelitian ini masih dapat dikaji lebih lanjut. Adapun saran untuk kajian lebih lanjut sebagai berikut:

1. Ekstraksi Ciri yang digunakan adalah MFCC, sehingga memungkinkan dengan menggunakan ekstraksi ciri suara yang berbeda seperti LPC atau ZPC akan mempengaruhi hasil yang lebih baik pada akurasi kasus pengenalan suara.
2. *Noise* dapat mempengaruhi pada tingkat keberhasilan pengenalan suara, diharapkan ada metode yang lebih baik yang dapat mengatasi masalah *noise*.
3. Pada proses pembelajaran perlu dilakukan dengan menggunakan metode klasifikasi lain, seperti menggunakan *Backpropagation Hiden Markov Model, Leans Blue Grey Algorithm* yang dapat digunakan sebagai pembanding tingkat akurasi pada pengenalan suara.

DAFTAR PUSTAKA

[1] E. S. Riyanto, Perbandingan Ekstraksi Ciri Suara MFCC, ZCPA dan LPC, STMIK HIMSYA.

[2] Darma Putra Adi Resmawan, "Verifikasi Biometrika Suara Menggunakan Metode MFCC dan DTW," *LONTAR KOMPUTER,* vol. Vol. 2 NO.1, pp. 8-21, 2011.

[3] R. D. Inung Wijayanto, "Jenis Tipe Jangkauan Suara Pada Pria dan Wantia Menggunakan Metoda Mel-Frequency Cepstral Coefficient dan Jaringan Syaraf Tiruan Backpropagation," *Fakultas Elektro dan Komunikasi Insitut Teknologi Telkom Bandung.*

[4] A. D. Andriana, "Perangkat Lunak Untuk Membuka Aplikasi Pada Komputer Dengan Perintah Suara Menggunakan Metode Mel Frequency Cepstrum Coefficients," *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA),* vol. 2, pp. 21-26, 2013.

[5] Parvesh Kumar, Siri Krishan Wasan, "Comparative Study of K-Means, Pam and Rough K-Means," *2009 International Symposium on Computing, Communication, and Control (ISCCC 2009),* pp. 136-140, 2011.

[6] E. Prasetyo, Data Mining Mengolah Data Menjadi Informasi Menggunakan Matlab, Yogyakarta: Andi, 2014.

[7] K. S. M. Ghassemian, Speaker Identification Features, Models and Robustness, Denmark: Kongens Lyngby, 2009.

[8] N. S. V. M. Manisha Modak, "K-means, Covering Based Rough Set using," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 4, no. 12, pp. 451-460, 2014.