Penerapan Metode *Mel Frequency Ceptral Coefficient* dan *Learning Vector Quantization* untuk *Text-Dependent Speaker Identification*

Sukoreno Mukti Widodo [#1], Elisafina Siswanto [#2], Oetomo Sudjana [*3]

[#]*Departemen Teknik Informatika, Institut Teknologi Harapan Bangsa*
Bandung, Indonesia
[1]sukorenomw@gmail.com
[2]elisafina@ithb.ac.id

[*]*Fakultas Teknik Industri, Universitas Parahyangan*
Bandung, Indonesia
[2]oektomo@gmail.com

Abstract*— Voice is one of the natural form for communication. People can recognize other people only by listening their voice. Speaker recognition is a proses to automatically identify speaker based on his/her voice characteristic. Speaker Recognition widely used in many fields, especially in the field of security. Applications made in this study, have input in the form of voice sample audio file that everyone said the same thing or text-dependent speaker recognition. In this study, the speaker recognition system is made to recognize the speaker of an audio file using Mel-Frequency Ceptral Coefficients for extracting voice data to produce features that represent the speaker and Learning Vector Quantization (LVQ) to train the data extraction and matching training data with new data to obtain the identity of the speaker based on the sound. From the experiment result, obtained the highest identification rate is 88.9% using data with a duration about 8 seconds.

***Keywords*— Voice recognition, Learning Vector Quantization, Mel Frequency Ceptral Coefficients, voice features extraction, features selection, Text-Dependent.**

Abstrak*— Suara merupakan salah satu bentuk alami untuk berkomunikasi. Manusia dapat mengenali seseorang hanya dengan mendengarkan orang tersebut berbicara. *Speaker Recognition* adalah proses untuk mengidentifikasi pembicara secara otomatis berdasarkan karakteristik suara. *Speaker Recognition* banyak digunakan pada berbagai bidang, terutama bidang keamanan. Aplikasi yang dibuat pada penelitian ini adalah menerima masukan berupa *file* audio dengan data ucapan. Data ucapan ini bergantung pada teks (*text-dependent*) dengan *output* adalah identitas pembicara yang teridentifikasi. Pada penelitian ini, sistem pengenalan pembicara dibuat untuk dapat mengenali suara pembicara dengan menggunakan metode *Mel-Frequency Cepstral Coefficients* dan metode *Learning Vector Quantization*. Kedua metoda tersebut digunakan untuk melakukan ekstraksi fitur dari data suara, sehingga dihasilkan fitur-fitur yang mewakili pembicara tersebut, untuk melatih data-data hasil ekstraksi dan mencocokan data latih dengan data baru, sehingga didapatkan identitas dari pembicara berdasarkan suara tersebut. Dari hasil pengujian pada sistem ini, didapatkan *identification rate* tertinggi adalah 88,9% dengan menggunakan data dengan durasi sekitar 8 detik.

***Kata Kunci*— Pengenalan suara, *Learning Vector Quantization*, *Mel Frequency Ceptral Coefficients*, ekstraksi fitur suara, seleksi fitur, Text-Dependent.**

I. PENDAHULUAN

Suara merupakan salah satu bentuk alami untuk berkomunikasi. Manusia dapat mengenali seseorang hanya dengan mendengarkan orang tersebut berbicara, sehingga beberapa detik ucapan sudah cukup untuk mengidentifikasi pembicara. *Speaker Recognition* adalah proses untuk mengidentifikasi pembicara secara otomatis berdasarkan karakteristik suara [1].

Speaker Recognition dapat digunakan pada berbagai bidang. Salah satunya adalah bidang keamanan. *Speaker Recognition* dapat membantu proses otentikasi pengguna dengan tingkat keamanan yang lebih baik dibandingkan menggunakan kata sandi yang mudah diketahui oleh orang lain. Contoh lain aplikasi penggunaan *Speaker Recognition* adalah pada bidang kriminalitas, misalnya untuk menemukan pelaku kejahatan berdasarkan *file* suara yang didapatkan di tempat kejadian perkara.

Pada penelitian ini, metode yang digunakan untuk ekstraksi fitur suara adalah metode *Mel-Frequency Cepstral Coefficients* (MFCC). Metode ini merupakan metode yang paling terkenal dan paling sering digunakan dalam melakukan ekstraksi fitur suara dalam penelitian *Speaker Recognition* dan *Speech Recognition*. Metode tersebut mampu untuk mengekstraksi karakteristik sinyal suara secara jelas yang relatif berbeda dari setiap sifat saluran suara pembicara dan efektif untuk mengenali suara yang mengandung *noise*. Metode *Learning Vector Quantization* (LVQ) digunakan sebagai pengenalan pola untuk mengidentifikasi pembicara sesuai dengan fitur suara yang telah diekstraksi menggunakan metode MFCC.

II. PENELITIAN TERKAIT

Ada dua jenis *Speaker Recognition*, yaitu [2]: *Text Independent Speaker Identification* (TI-SI) dan *Text Dependent Speaker Identification* (TD-SI). TI-SI adalah jenis proses identifikasi suara pembicara tanpa membatasi

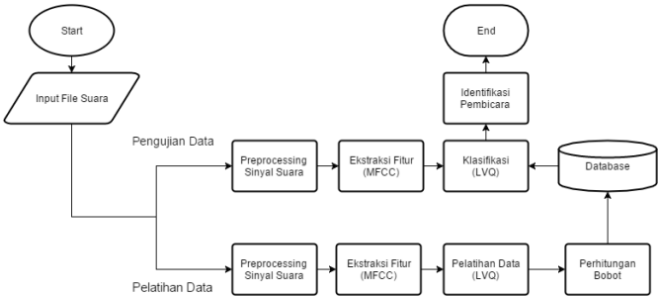
pengenalan dengan kata-kata tertentu. TI-SI membutuhkan pelatihan data yang banyak untuk mendapatkan akurasi yang baik. TD-SI adalah jenis proses identifikasi suara pembicara dengan menggunakan kata-kata yang sama. Tipe ini lebih cocok digunakan untuk teknik identifikasi suara pembicara karena proses identifikasi pembicara bisa didapatkan hanya dengan mengekstraksi ciri suara melalui beberapa kata sebagai sampel.

Metode-metode yang dapat digunakan untuk mengekstraksi fitur suara, diantaranya adalah LPC, LPCC, dan MFCC. Berdasarkan penelitian yang dilakukan oleh Utpal Bhattacharjee pada tahun 2013, metode MFCC yang digunakan untuk pengenalan suara dengan kondisi *noise* sebesar 20 dB memberikan akurasi sebesar 97,03%, sedangkan metode LPCC hanya memberikan akurasi sebesar 73,76% [3]. Untuk metode LPC, tidak direkomendasikan untuk pengenalan pembicara karena lebih cocok untuk komputasi linear, sedangkan suara manusia pada dasarnya adalah nonlinear.

Metode yang digunakan untuk melakukan pengenalan pola suara adalah LVQ. Selain metode LVQ, penelitian *Speaker Recognition* dan *Speech Recognition* biasanya menggunakan metode HMM atau GMM. Metode LVQ lebih cocok digunakan untuk TD-SI karena pengenalan pembicara jenis ini tidak memerlukan banyak pelatihan data, seperti TI-SI, sehingga lebih efisien dalam hal data latih dan waktu pelatihan data untuk penelitian ini [4]. Metode GMM dan HMM lebih banyak digunakan pada model stokastik, di mana lebih cocok digunakan untuk kasus TI-SI [5].

III. Perancangan dan Implementasi

Dalam melakukan pengenalan pembicara, ada beberapa langkah yang dilakukan. Pertama, *user* memasukkan *file* suara yang akan diproses lalu dilakukan praproses pada sinyal suara untuk menghasilkan suara yang jernih. Selanjutnya, dilakukan ekstraksi fitur untuk mendapatkan karakteristik dari suara tersebut yang nantinya akan digunakan untuk pencocokan identifikasi pembicara. Proses pelatihan data dari memasukkan *file* suara sampai dilakukan perhitungan MFCC untuk mendapatkan koefisien *Mel-Frequency Ceptral* dapat dilihat seperti pada Gambar 1.



Gambar 1 Urutan proses *Text-Dependent Speaker Identification*

A. Mel Frequency Ceptral Coefficient (MFCC)

MFCC merupakan metode untuk mengekstraksi fitur yang menghitung koefisien *ceptral* berdasarkan variasi frekuensi kritis pada sistem pendengaran manusia. Metode ini dikembangkan oleh Davis and Mermelstein. Tahapan cara kerja MFCC secara umum dapat dilihat pada Gambar 2.

Beberapa keunggulan metode MFCC adalah [3]:

1. Mampu mengekstraksi fitur suara selengkap mungkin dengan data seminimal mungkin.
2. Dapat mereplikasi sistem pendengaran suara manusia, sehingga mendapatkan akurasi tinggi.
3. Meskipun membutuhkan lebih banyak proses komputasi, MFCC menghasilkan performa yang baik dari segi efisiensi dan akurasi.

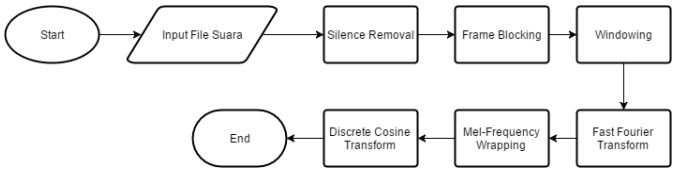
B. Frame Blocking

Frame blocking adalah sebuah proses membagi sinyal suara menjadi beberapa *frame* dengan durasi 20 hingga 30 ms yang berisi *N* sampel. Masing-masing *frame* dipisahkan oleh *M* (M < N), dimana *M* adalah banyaknya pergeseran antar *frame* [6]. *Frame* pertama berisi *N* sampel pertama. *Frame* kedua dimulai *M* sampel setelah permulaan *frame* pertama. *Frame* kedua tersebut *overlap* terhadap *frame* pertama sebanyak *N* - *M* sampel. Hal ini ditunjukkan pada Gambar 3.

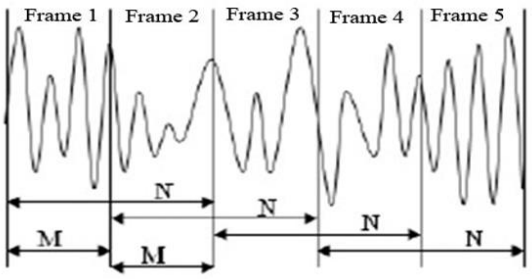
C. Windowing

Proses *windowing* dilakukan untuk meminimalisir diskontinuitas yang terjadi pada sinyal yang disebabkan oleh kebocoran spektral pada saat proses *frame blocking*. Sinyal yang baru memiliki frekuensi yang berbeda dengan sinyal aslinya. Proses *windowing* dilakukan dengan cara mengalikan tiap *frame* dengan jenis *window* yang digunakan. Persamaannya adalah sebagai berikut [6]:

$$y(n) = x(n)w(n), \quad 0 \le n \le N-1 \tag{1}$$



Gambar 2 Tahapan-tahapan MFCC.



Gambar 3 Proses *frame blocking*.

Penelitian suara banyak menggunakan *hamming window* dengan rumus yang sederhana. *Windowing* yang hasilnya baik adalah *hamming window* dengan persamaan berikut [7]:

$$w(n) = 0.54 - 0.46\ cos\left(\frac{2\pi n}{N-1}\right) \quad (2)$$

Di mana,
$y(n)$ = sinyal hasil *windowing* sampel ke- n
$x(n)$ = nilai sampel ke- n sebelum di-*windowing*
$w(n)$ = nilai *window* ke-n
N = jumlah sampel setiap *frame*
n = indeks sampel dalam suatu *frame*

D. Fast Fourier Transform

Fast Fourier Transform (FFT) adalah tahap untuk mengubah setiap *frame* yang terdiri dari *N* sampel dari domain waktu ke dalam domain frekuensi. Langkah pertama untuk menginterpretasikan FFT adalah dengan menghitung nilai frekuensi dari setiap sampel tengah dari FFT. Jika sampel waktu yang diterima FFT dalam bentuk riil, maka hanya keluaran $X(m)$ dari $m = 2$ sampai $m = N/2$ yang independen. Dalam kasus ini, hanya perlu menghitung nilai frekuensi FFT untuk *m* selama $0 \leq m \leq N/2$. Jika sampel waktu yang diterima berbentuk kompleks, maka semua nilai frekuensi FFT untuk *m* dihitung selama $0 \leq m \leq N\text{-}1$.

Amplituda FFT dihitung menggunakan rumus sebagai berikut [7]:

$$X(m) = X_{real}(m) + jX_{imag}(m) \quad (3)$$

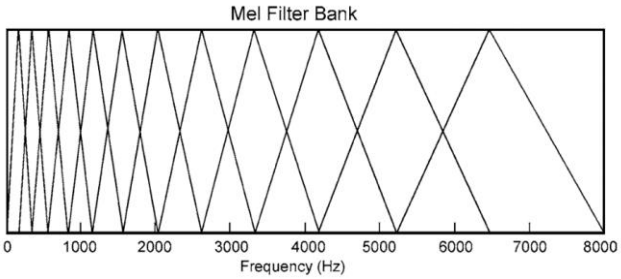
Magnituda FFT dihitung menggunakan persamaan berikut [7]:

$$X_{mag}(m) = \sqrt{X_{real}(m)^2 + X_{imag}(m)^2} \quad (4)$$

E. Mel-Frequency Wrapping

Selanjutnya adalah tahap membentuk *filterbank*. *Mel-Frequency Wrapping* menggunakan *filterbank* untuk menyaring sinyal suara yang telah diubah ke dalam bentuk domain frekuensi. *Filterbank* adalah sistem yang membagi *input* sinyal ke dalam kumpulan analisis sinyal yang sesuai dengan wilayah yang berbeda sesuai spektrumnya [7]. Biasanya, daerah spektrum yang diberikan oleh analisis sinyal kolektif menjangkau seluruh suara yang terdengar oleh pendengaran manusia, sekitar 20 Hz sampai 20 kHz.

Filterbank yang digunakan dalam metode MFCC, khususnya untuk proses *Mel-Frequency Wrapping*, adalah *Mel-Filterbank*. *Mel-filterbank* yang terdiri dari rangkaian *Triangular Window* yang saling *overlap* akan menyaring sinyal sebanyak *N* sampel, seperti yang dapat dilihat pada Gambar 4.



Gambar 4 Mel Filterbank

Nilai *Mel* dapat dihitung dengan persamaan berikut:

$$mel(f) = 1125\ \ln(1 + \frac{f}{700}) \quad (5)$$

Nilai frekuensi dari nilai *Mel* dapat dihitung dengan persamaan berikut:

$$mel^{-1}(f) = 700\ (\exp\left(\frac{mel(f)}{1125}\right) - 1) \quad (6)$$

Di mana,
Mel(f)= nilai *Mel* (konversi dari nilai frekuensi).
$Mel^{-1}(f)$ =nilai frekuensi (konversi dari nilai *Mel*).
f = frekuensi.

Nilai *Mel* didapat diubah ke dalam Hertz (Hz) dengan menggunakan persamaan 6. Namun, untuk membuat filterbank, nilai frekuensi *f* harus dikonversikan ke dalam nilai sampel FFT terdekat dengan menggunakan persamaan berikut [8]:

$$f[m] = \left(\frac{N}{F_s}\right) mel^{-1}\left(mel + m\frac{mel(f_h) - mel(f_l)}{M+1}\right) \quad (7)$$

Di mana,
$f[m]$ = nilai *sample* FFT
N = numlah FFT tiap *frame*
F_s = frekuensi *sampling*
f_h = frekuensi batas atas
f_l = frekuensi batas bawah
M = jumlah filter

Proses filter sinyal dilakukan untuk mendapatkan *log energy* pada tiap filter dengan menggunakan persamaan [8]:

$$S[m] = \ln\left[\sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k]\right], 1 \leq m \leq M \quad (8)$$

Di mana,
$S[m]$ = nilai *log energy*
$X_a[k]$ = nilai magnituda (Hz)
H_m[k] = nilai *filterbank*
N = jumlah nilai FFT tiap *frame*
m = indeks filter
k = indeks *input* sampel FFT (k = 0, 1, ..., N - 1)

F. Ceptrum

Pada tahap terakhir ini, nilai Mel akan dikonversikan kembali ke dalam domain waktu, yang hasilnya disebut *Mel Frequency Ceptral Coefficient.* Konversi ini dilakukan dengan menggunakan Discrete Cosine Transform (DCT). C_0 adalah nilai rata-rata dalam dB yang dapat digunakan untuk estimasi energi yang berasal dari filterbank. Koefisien DCT adalah nilai amplitudo dari spektrum yang dihasilkan. Perhitungan DCT dapat dilihat pada persamaan berikut [8]:

$$c_i = \sum_{m=0}^{M-1} S[m] \cos\left(\frac{\pi n\,(m-0.5)}{M}\right) \qquad (9)$$

G. Learning Vector Quantization

Learning Vector Quantization (LVQ) adalah satu metode untuk melakukan klasifikasi pola, dimana setiap keluarannya merepresentasikan sebuah kelas atau kategori tertentu. LVQ merupakan salah satu algoritma dari *supervised neural network* [9].

Pada Gambar 5 dapat dilihat bahwa LVQ terdiri atas lapisan *input* (*x*), lapisan kompetitif, dan lapisan *output* (*y*). Lapisan *input* adalah masukan data dan lapisan kompetitif adalah pada saat terjadinya kompetisi pada *input* untuk masuk dalam suatu kelas berdasarkan kedekatan jaraknya. Dalam lapisan kompetitif, proses pembelajaran dilakukan secara terawasi. *Input* akan bersaing untuk dapat masuk ke dalam suatu kelas.

Algoritma dalam melakukan LVQ adalah sebagai berikut [9]:

1. Inisialisasi vektor referensi; inisialisasi *learning rate* (α), dimana 0 < α < 1.
2. Selama kondisi berhenti adalah *false*, maka lakukan langkah 2 s.d. 4.
3. Untuk setiap *input* pelatihan vektor *x*, lakukan langkah 3 s.d. 4.
4. Cari *J* hingga $\|x - w_j\|$ adalah nilai minimal.
5. Perbarui *w_j* , dimana kondisinya adalah:
   - Jika *T* = *C_j* maka:

   $$w_j(baru) = w_j(lama) + \alpha[x - w_j(lama)] \qquad (10)$$

   - Jika $T \neq C\_j$ maka:

   $$w_j(baru) = w_j(lama) - \alpha[x - w_j(lama)] \qquad (11)$$

6. Kurangi *learning rate*.
7. Uji kondisi berhenti sebagai kondisi yang mungkin menetapkan sebuah jumlah yang tetap dari iterasi.

Untuk mendapatkan hasil kelas dari *input* yang dimaksud, maka dilakukan perhitungan kembali data *input* dengan jarak terpendek dari bobot akhir. Jika hasil perhitungan jarak bobot didapatkan bobot ke-*x* merupakan jarak terkecil diantara bobot-bobot lainnya, maka vektor *input* pertama tersebut dapat dimasukan ke dalam kelas bobot ke-*x*. Langkah tersebut diulangi sampai seluruh data *input* selesai diproses, sehingga didapatkan banyak kelas sejumlah data *input*. Untuk mendapatkan hasil akhir data *input* tersebut masuk ke dalam kelas, maka dilakukan proses *voting*. Hal ini dapat terlihat pada Gambar 6, yaitu memilih hasil kelas yang terbanyak sebagai kelas akhir dari data *input* tersebut.

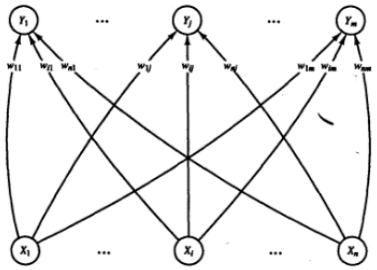
IV. PENGUJIAN

Pada penelitian ini, pengujian dilakukan dengan menggunakan jumlah data latih yang beragam. Data uji yang digunakan adalah sebanyak 3 *file* audio untuk setiap 1 pembicara. Data uji diambil dari data yang bukan merupakan data latih. Setiap orang akan diambil 1 rekaman suara untuk setiap kata. Jumlah data uji setiap pembicara adalah 3 suara, sehingga total data uji adalah 30 buah suara. Pada pengujian ini dilakukan 3 kali pelatihan data dengan parameter yang sama untuk mendapatkan 3 bobot akhir yang berbeda (disebut *T*1, *T*2, dan *T*3) karena setiap pelatihan bobot awal diambil secara acak. *Identification rate* akhir yang didapat adalah hasil *identification rate* rata-rata dari 3 kali pengujian pada parameter yang sama. *Identification rate* merupakan persentase keberhasilan sistem menentukan pembicara dari suatu *file* audio dengan tepat.

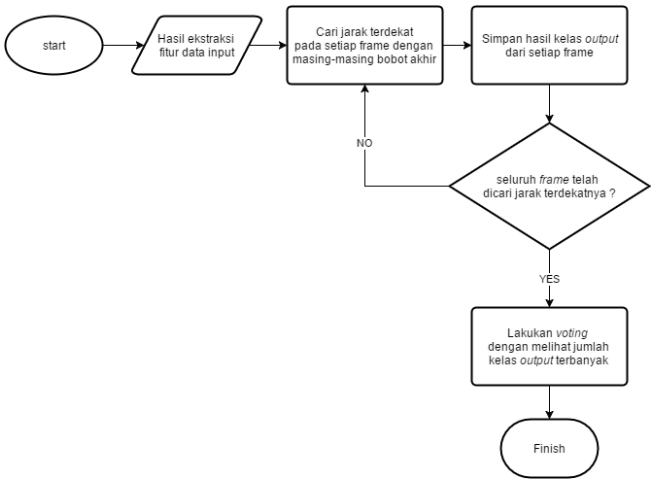
Dalam penelitian ini dilakukan 3 skenario pengujian yakni:

1) Pengujian sama kata

Pada pengujian ini, pembicara hanya mengucapkan satu kata yang sama, yaitu urutan nomor telepon ‘0819-1051-3704’ untuk data latih dan uji. Data latih yang digunakan untuk setiap orang adalah 10 buah *file* suara dan 9 orang pembicara, sehingga total data latih untuk pengujian ini adalah sebanyak 90 buah. Untuk data uji, masing-masing pembicara hanya mengucapkan 1 kali, sehingga data uji yang didapat adalah sebanyak 9 buah. Hasil pengujian dapat dilihat pada Tabel I.



Gambar 5 Arsitektur LVQ



Gambar 6 *Flowchart* proses *voting output* kelas akhir

2) Pengujian dengan durasi data 2, 4 dan 8 detik

Pada pengujian ini, pembicara hanya mengucapkan 3 kata yaitu kata "*unlock*". Urutan angka berupa nomor telepon dan urutan angka 1-10 untuk data latih dan uji. Masing - masing durasi kata yang diucapkan adalah 2 detik, 4 detik dan 8 detik. Data latih yang digunakan untuk setiap orang adalah 10 *file* suara untuk setiap pembicara dari 9 orang pembicara. Total data latih untuk pengujian ini adalah sebanyak 90 buah untuk masing-masing kata. Untuk data uji, masing-masing pembicara hanya mengucapkan 1 kali, sehingga data uji yang didapat adalah sebanyak 10 buah untuk masing-masing kata. Hasil pengujian ini dapat dilihat pada Tabel II.

3) Pengujian dengan 4 data uji

Pada pengujian ini, dilakukan dengan menggunakan 4 data uji yang berbeda dari setiap pembicara. Total pembicara adalah 9 orang, sehingga total data uji untuk pengujian ini adalah 36 data. Perbedaan dengan pengujian pada skenario B adalah penambahan jumlah data uji. Hasil pengujian dapat dilihat pada Tabel III.

TABEL I

HASIL PENGUJIAN SAMA KATA

No	*alpha*	decy	epoh	T1	T2	T3	AVG
1	0.1	0.1	500	11.11%	11.11%	11.11%	11.11%
2	0.05	0.100	500	11.11%	22.22%	11.11%	14.81%
3	0.03	0.100	500	11.11%	11.11%	11.11%	11.11%
4	0.05	0.100	700	55.56%	55.56%	55.56%	55.56%
5	0.05	0.095	700	44.44%	44.44%	44.44%	44.44%
6	0.05	0.085	700	44.44%	44.44%	44.44%	44.44%
7	0.05	0.100	1000	88.89%	88.89%	88.89%	88.89%
8	0.03	0.100	1000	88.89%	88.89%	88.89%	88.89%
9	0.03	0.095	1000	88.89%	88.89%	88.89%	88.89%
10	0.05	0.100	3000	88.89%	88.89%	88.89%	88.89%
11	0.05	0.100	5000	88.89%	88.89%	88.89%	88.89%

TABEL II

HASIL PENGUJIAN SAMA KATA

Durasi	T1	T2	T3	AVG
2s	80.00%	60.00%	60.00%	66.67%
4s	55.56%	66.67%	66.67%	62.96%
8s	88.89%	88.89%	88.89%	88.89%

TABEL III

HASIL PENGUJIAN 4 DATA UJI

Durasi	T1	T2	T3	AVG
2s	73,33%	73,33%	73,33%	73,33%
4s	80,85%	80,85%	80,85%	80,85%
8s	88,89%	88,89%	88,89%	88,89%

V. KESIMPULAN

Untuk menjawab tujuan penelitian ini, maka kesimpulan yang dapat diambil adalah nilai *max epoch* sangat mempengaruhi *identification rate*, sedangkan nilai *alpha* dan *alpha decay* hanya sedikit mempengaruhi *identification rate*. Semakin kecil nilai *alpha* dan nilai *alpha decay*, maka *identification rate* yang didapatkan akan semakin baik. *Identification rate* tertinggi sebesar 88,89% adalah pada pengujian ke-7 dengan parameter *alpha* 0,05, *alpha decay* 0,1, *max epoch* 1000. Hal ini disebabkan karena semakin kecil nilai alpha, maka ketelitian pembelajaran akan semakin baik. Jika nilai *alpha* terlalu kecil, maka hasil yang didapatkan tidak stabil. Begitu pula dengan nilai *max epoch.* Semakin besar nilai *max epoch*, maka pembelajaran jaringan LVQ akan semakin baik karena banyak dilakukan iterasi. Jika nilai *max epoch* terlalu besar, maka akan memakan waktu lebih lama dalam melakukan komputasi dengan hasil yang tidak signifikan.

Frame size sebesar 512 menghasilkan akurasi 10% lebih baik dibandingkan dengan *frame size* 256. Akurasi tertinggi untuk *frame size* 512 adalah 88,89%. Akurasi tertinggi untuk *frame size* 256 adalah 77,78%. Hal ini disebabkan karena panjang *frame* ideal adalah sebesar 20-30 ms dengan rekaman suara menggunakan frekuensi *sampling* sebesar 16 KHz. Hasil paling optimal untuk ukuran sampel dalam *frame* adalah sebesar 512.

Akurasi identifikasi pembicara dengan data latih berdurasi panjang lebih baik daripada akurasi identifikasi pembicara pada data latih yang berdurasi pendek. *Identification rate* tertinggi adalah sebesar 88,89% untuk identifikasi pembicara dengan durasi data 8 detik, sedangkan *identification rate* tertinggi untuk identifikasi pembicara dengan data berdurasi 4 detik adalah 62,66%. Hal ini disebabkan karena MFCC mendekati sistem pendengaran manusia, sehingga semakin panjang durasi data *input* hasil yang didapatkan akan semakin baik. Pada penelitian berikutnya, proses ekstraksi fitur bisa menggunakan metode lain, seperti *Gammatone Frequency Cepstral Coefficients* (GFCC). Metode GFCC ini dapat lebih mengenali warna suara pembicara dari ucapannya dan menghasilkan akurasi pengenalan sebesar 98,5% [10].

DAFTAR REFERENSI

[1] Kshamamayee Dash, Debananda Padhi, Bhoomika Panda, and Sanghamitra Mohanty, "Speaker Identification Using Mel Frequency Cepstral Coefficient And Bpnn," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 4, April 2012.

[2] Zhizheng Wu, Anthony Larcher, and Kong Aik Lee, "Vulnerability evaluation of speaker verification under voice conversion spoofing: the effect of text constraints.," in *INTERSPEECH*, 2013, pp. 950-954.

[3] Utpal Bhattacharjee, "A Comparative Study Of LPCC And MFCC Features For The Recognition Of Assamese Phonemes," *International Journal of Engineering Research and Technology*, vol. 2, no. 1, January 2013.

[4] Penghua LI, Shunxing Zhang, Huizong Feng, and Yuanyuan Li, "Speaker Identification Using Spectrogram And *Learning Vector Quantization*," *Journal of Computational Information Systems*, vol. 11, no. 9, 2015.

[5] Geeta Nijhawan and M.K Soni, "Speaker Recognition Using Mfcc And Vector Quantisation," *International Journal on Recent Trends in Engineering and Technology*, vol. 11, no. 1, Juli 2014.

[6] Angga Setiawan, Achmad Hidayanto, and R. Rizal Isnanto, "Aplikasi Pengenalan Ucapan dengan Ekstraksi Mel-Frequency Cepstrum Coefficients Melalui Jaringan Syaraf Tiruan *Learning Vector Quantization* untuk Mengoperasikan Kursor Komputer," *TRANSMISI*, vol. 13, no. 2, pp. 82-86, 2011.

[7] Richard G Lyons, *Understanding Digital Signal Processing 3rd Edition*. Boston: Prentice Hall, 2011.

[8] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon, *Spoken Language Processing: A Guide To Theory, Algorithm And System Development*. New Jersey: Prentice Hall, 2001.

[9] Laurene Fausett, *Fundamental of Neural Networks: Architectures, Algorithms, and Applications*.: Prentice Hall, 1994.

[10] Shaveta Sharma and Parminder Singh, "Speech Emotion Recognition using GFCC and BPNN.," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 18, no. 6, pp. 321-322, 2014.

Sukoreno Mukti Widodo, mahasiswa jurusan Teknik Informatika di Institut Teknologi Harapan Bangsa yang lulus pada tahun 2016.

Elisafina Siswanto, lahir di Bandung pada tahun 1989, menerima gelar Sarjana Teknik dari Institut Teknologi Harapan Bangsa pada tahun 2011 jurusan Teknik Informatika, dan menyelesaikan pendidikan Magister Informatika di Institut Teknologi Bandung pada tahun 2014 Saat ini aktif sebagai pengajar di Departement Teknik Informatika, Institut Teknologi Harapan Bangsa di Bandung. Minat penelitian adalah pada bidang Pembelajaran Mesin dan Pemrosesan Bahasa Alami.

Oetomo Sudjana, adalah lulusan Teknik Elektro dari Universitas Udayana, Bali pada tahun 2010 dan menerima gelar Magister Teknik Elektro dari Institut Teknologi Bandung pada tahun 2014. Saat ini aktif sebagai pengajar di Teknik Industri, Universitas Parahyangan.