# IDENTIFIKASI PARAMETER OPTIMAL *GAUSSIAN MIXTURE MODEL* PADA IDENTIFIKASI PEMBICARA DI LINGKUNGAN BERDERAU MENGGUNAKAN RESIDU DETEKSI *ENDPOINT*

***Yanuar Risah Prayogi[1)], Joko Lianto Buliali[2)]***
[1, 2)]Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember,
Jalan Raya ITS Kampus ITS Sukolilo, Surabaya, 60111, Indonesia
e-mail: yanuarrisah@gmail.com[1)], jokolianto@gmail.com[2)]

**ABSTRAK**

*Salah satu permasalahan pada sistem identifikasi pembicara adalah fitur yang dihasilkan kurang tahan terhadap derau. Di lingkungan berderau, kinerja sistem identifikasi pembicara bisa turun secara signifikan. Hal ini disebabkan oleh perbedaan lingkungan ketika pelatihan dan pengujian. Salah satu metode ekstraksi fitur yang digunakan untuk identifikasi pembicara dan sensitif terhadap derau adalah Mel Frequency Cepstral Coefficient (MFCC). Di lingkungan bersih, kinerja yang dihasilkan oleh metode MFCC sangat tinggi, tetapi turun drastis ketika berada di lingkungan berderau.*

*Pada penelitian ini diusulkan memodifikasi metode MFCC menggunakan residu dari algoritma deteksi endpoint. Hasil dari algoritma deteksi endpoint adalah speech dan nonspeech (residu). Nonspeech atau residu ini biasanya tidak dipakai pada proses berikutnya. Pada sinyal suara yang berderau, residu dari algoritma deteksi endpoint sebagian besar diisi oleh derau itu sendiri sehingga bisa dijadikan informasi derau. Residu tersebut diekstrak untuk mendapatkan besaran (magnitude) frekuensi derau. Kemudian magnitude frekuensi derau digunakan untuk menghilangkan derau pada sinyal utama atau speech.*

*Uji coba menggunakan lima tipe derau dengan tujuh tingkat SNR. Tipe derau yang digunakan adalah f16, hfchannel, pink, volvo, dan white. Sedangkan tingkat SNR yang digunakan adalah bersih, 25, 20, 15, 10, 5, dan 0 dB. Hasil uji coba menunjukkan bahwa metode yang diusulkan unggul pada mayoritas pembicara. Selain itu metode yang diusulkan juga unggul pada semua tipe derau dan unggul hampir pada semua tingkat SNR. Metode yang diusulkan menunjukkan rata-rata akurasi sebesar 14.69% lebih tinggi dari metode MFCC, 2.74% dari MFCC+Spectral Subtraction (SS), dan 6.4% dari MFCC+wiener.*

***Kata Kunci**: identifikasi pembicara, lingkungan berderau, Mel Frequency Cepstral Coefficient (MFCC), residu endpoint detection.*

**ABSTRACT**

*One of the problems in the speaker identification system is a feature that generated less resistant to noise. In the noisy environment, the speaker identification system performance can drop significantly. It is caused by environmental differences when training and testing. One feature extraction method used to identify the speaker and sensitive to noise is Mel frequency cepstral coefficient (MFCC). In a clean environment, the performance generated by MFCC method is very high, but dropped dramatically when in the noisy environment.*

*In this study, we propose to modify the MFCC method using endpoint detection residues. Results of endpoint detection algorithm is speech and nonspeech (residue). Nonspeech or residues are usually not used in the next process. At the noisy signal, the residue of endpoint detection algorithm is filled by the noise itself so that it can be used as information noise. The residue is extracted to get the magnitude of the noisy signal. Magnitude of the noisy signal is used to remove noise on the main signal or speech.*

*The experiments using five types of noise with seven levels of SNR. The type of noise that used is f16, hfchannel, pink, volvo, and white. While the level of SNR that used is clean, 25, 20, 15, 10, 5, and 0 dB. Experimental results show that the proposed method superior to the majority of the speakers. In addition the proposed method is also superior to all types of noise and superior in nearly all levels of SNR. The proposed method shows the average accuracy 14.69% higher than MFCC, 2.74% higher than MFCC+Spectral Subtraction (SS), and 6.4% higher than MFCC+wiener.*

***Keywords**: centroid speaker identification, noisy environment, mel frequency cepstral coefficient (MFCC), residue of endpoint detection.*

## I. PENDAHULUAN

IDENTIFIKASI pembicara adalah proses mengenali identitas pembicara menggunakan suara yang diberikan. Pada sistem identifikasi pembicara bertipe *close-set*, identitas pembicara sudah didaftarkan dalam *database*. Ketika pembicara mengklaim sebuah identitas menggunakan suaranya, sistem mencari pembicara didalam *database* yang memiliki kemiripan suara paling tinggi. Identifikasi pembicara ini bisa digunakan sebagai *password* pada sistem keamanan, misalkan pada sebuah perangkat *smartphone*. Untuk membuka dan menggunakan perangkat tersebut diperlukan identifikasi pengguna. Sehingga hanya pengguna tertentu yang mempunyai hak dalam penggunaan perangkat tersebut.

Kelemahan utama pada sistem identifikasi pembicara adalah kurang tahan terhadap lingkungan yang berderau [1]. Di dunia nyata, kinerja sistem identifikasi pembicara bisa turun secara signifikan. Hal ini disebabkan oleh perbedaan lingkungan ketika pelatihan dan pengujian. Ketika pengujian, derau pada sinyal suara ternyata lebih besar atau lebih kecil dibandingkan ketika pelatihan sehingga kinerja dari sistem identifikasi pembicara menurun.

*Mel Frequency Cepstral Coefficient* atau MFCC adalah salah satu metode ekstraksi fitur suara yang terkenal dan sensitif terhadap derau [1]. Pada lingkungan bersih, kinerja yang dihasilkan oleh metode MFCC sangat tinggi, tetapi turun signifikan ketika berada di lingkungan berderau. Oleh karena itu dibutuhkan modifikasi pada metode MFCC.

Untuk mengurangi derau pada MFCC beberapa peneliti menambahkan *Spectral Subtraction* (SS) untuk menghilangkan sinyal derau. Shang-Ming Lee dkk [2] selain menerapkan *Principal Component Analysis* (PCA) pada *Mel-scale filter bank* juga menambahkan *Spectral Subtraction* (SS) untuk mengurangi derau. Peneliti lainnya, Wu Zunjing dan Cao Zhigang [3] selain mengganti fungsi log pada MFCC dengan fungsi pangkat juga menambahkan *Spectral Subtraction* (SS). Dengan penambahan *Spectral Subtraction*, akurasi dua penelitian diatas meningkat dibandingkan dengan tanpa penambahan *Spectral Subtraction*. Kelebihan *Spectral Subtraction* adalah dapat menghilangkan sinyal derau dengan memanfaatkan informasi sinyal derau yang diperoleh dari beberapa *frame* awal. Bagaimanapun akurasi yang dihasilkan tetap turun signifikan seiring turunnya tingkat SNR. Selain itu jumlah *frame* awal yang digunakan harus tepat karena jika jumlah *frame* awal terlalu banyak maka sinyal suara ikut hilang.

Peneliti lainnya yang mencoba menghilangkan derau pada MFCC adalah Paresh M. Chauhan dan Nikita P. Desai [4]. Paresh M. Chauhan dan Nikita P. Desai menambahkan *Wiener filter* setelah proses FFT. *Wiener filter* bekerja di domain frekuensi karena terletak setelah proses FFT. Paresh M. Chauhan dan Nikita P. Desai menguji *Wiener filter* dengan meletakkan di domain waktu (setelah *framing*) dan di domain frekuensi (setelah FFT). Hasil yang diperoleh adalah peletakan *Wiener filter* di domain frekuensi (setelah FFT) lebih efektif dibandingkan di domain waktu (setelah *framing*). Akurasi yang dihasilkan lebih tinggi dibandingkan dengan MFCC. Meskipun demikian, *Wiener filter* menghilangkan derau berdasarkan estimasi sehingga akurasi yang dihasilkan masih turun seiring turunnya tingkat SNR.

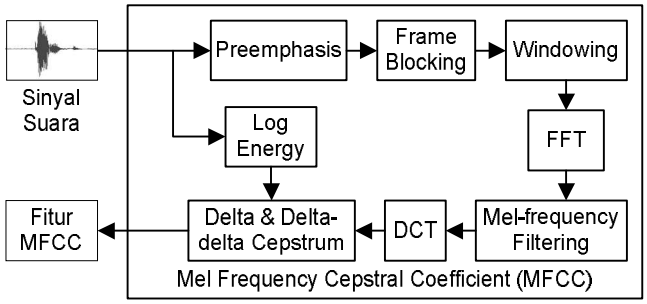
Dari beberapa penelitian sebelumnya, masih ada kebutuhan untuk meningkatkan kinerja metode MFCC baik pada lingkungan bersih maupun berderau. Untuk meningkatkan kinerja MFCC pada sistem identifikasi pembicara, baik pada tingkat SNR rendah maupun tinggi, diperlukan modifikasi yang mampu menghilangkan sinyal derau sesuai dengan tingkat SNR. Dengan demikian diusulkan modifikasi MFCC menggunakan residu dari algoritma deteksi *endpoint*. Motivasi menggunakan residu dari algoritma deteksi *endpoint* adalah karena algoritma deteksi *endpoint* digunakan untuk memisahkan *speech* dengan *nonspeech* [5]. *Nonspeech* ini biasanya dibuang dan tidak akan dipakai pada proses berikutnya sehingga bisa disebut sebagai residu. Didalam sinyal yang berderau, residu dari algoritma deteksi *endpoint* sebagian besar diisi oleh derau itu sendiri [6]. Residu ini dapat dijadikan sebagai informasi sinyal derau seperti yang dilakukan oleh *Spectral Subtraction*. Residu algoritma deteksi *endpoint* diubah ke domain frekuensi sehingga didapatkan *magnitude* frekuensi derau. *Magnitude* tersebut kemudian digunakan untuk menghilangkan derau pada *speech* di domain frekuensi seperti yang dilakukan pada penelitian Paresh M. Chauhan dan Nikita P. Desai [4].

Metode yang diusulkan akan diimplementasikan pada sistem identifikasi pembicara yang menggunakan Gaussian Mixture Mode (GMM) sebagai metode pelatihan dan pengenalan. Pada metode GMM terdapat parameter jumlah komponen yang menentukan seberapa baik pelatihan dan pengenalan yang dihasilkan. Pada penelitian ini akan diidentifikasi jumlah komponen GMM yang optimal

## II. MEL FREQUENCY CEPSTRAL COEFFICIENT (MFCC)

MFCC adalah metode ekstraksi fitur yang terkenal dan umum digunakan [8]. Proses-proses yang ada pada metode MFCC adalah *preemphasis*, *frame blocking*, *windowing*, *Fast Fourier Transform* (FFT), *Mel-frequency*

*Filtering*, *Discrete Continuous Transform* (DCT), log *energy*, dan *Delta & Delta-delta Cepstrum* [7][8]. Proses-proses yang ada pada metode MFCC ditunjukkan pada Gambar 1.




Gambar 1. Metode MFCC [7][8]

*A. Preemphasis*

Pada sinyal suara yang asli biasanya memiliki energi frekuensi rendah yang cukup banyak. Sedangkan energi frekuensi tinggi sedikit. Perbedaan energi ini terjadi karena sinyal frekuensi rendah disampling dengan frekuensi sampling yang cukup tinggi sehingga menghasilkan nilai numerik yang sama. Tujuan dari *preemphasis* adalah meningkatkan energi dari frekuensi tinggi. Fungsi dari *preemphasis* hampir sama dengan *high-pass filter* yaitu untuk melewatkan komponen frekuensi tinggi dari sinyal suara. Misalkan $s(n)$ adalah sinyal suara maka, *preemphasis* dapat dinyatakan dengan persamaan [9] (1)

$$y(n) = s(n) - \alpha * s(n-1) \,, \tag{1}$$

dimana $y(n)$ adalah sinyal preemphasis dan $\alpha$ adalah konstanta yang bernilai 0.9 sampai 1. Nilai $\alpha$ yang sering digunakan adalah 0.97 [8].

*B. Frame Blocking*

Sinyal suara dibagi menjadi blok-blok kecil yang disebut *frame*. Antara *frame* satu dengan yang lain terjadi tumpang tindih (*overlapping*). Tiap *frame* terdiri dari $N$ sampel dan tumpang tindih sebanyak $M$ sampel. Pada metode MFCC panjang *frame* yang biasa digunakan adalah 30 milidetik atau $N = 480$ sampel untuk frekuensi sampling 16000 Hz. Sedangkan panjang tumpang tindih adalah 10 milidetik atau $M = 160$ sampel untuk frekuensi sampling 16000 Hz. Tujuan *frame blocking* adalah membagi sinyal suara menjadi *frame* kecil dengan sampel yang cukup untuk mendapatkan informasi yang cukup [10]. Jika ukuran *frame* terlalu kecil, informasi yang didapat tidak cukup untuk dipercaya. Sebaliknya jika terlalu besar, informasi didalam *frame* sering berubah-ubah [10].

*C. Windowing*

Proses *windowing* dilakukan untuk tiap *frame*. *Windowing* digunakan untuk meminimalkan ketidaksinambungan pada awal dan akhir *frame* [11]. Dengan menggunakan *window*, sinyal suara pada awal dan akhir *frame* menjadi runcing [10]. Prosesnya adalah sinyal suara dikalikan dengan *window*. Proses perhitungan *windowing* ditunjukkan pada persamaan (2)

$$y(m) = w(m)x(m), \qquad 0 \le m \le N-1 \,, \tag{2}$$

dimana $w(m)$ adalah fungsi *window* dan $x(m)$ adalah sinyal suara. Pada dasarnya banyak fungsi *window* yang bisa digunakan misalkan *blackman window*, *gaussian window*, dan *hamming window*. Pada metode MFCC, *window* yang sering digunakan adalah *hamming window* [10]. Fungsi *hamming window* dinyatakan dengan persamaan [10] (3).

$$w(m) = 0.54 - 0.46 * cos\left(\frac{2\pi m}{N-1}\right), \qquad 0 \le m \le N-1 \,, \tag{3}$$

*D. Fast Fourier Transform (FFT)*

*Fast Fourier Transform* (FFT) merubah sinyal suara dari domain waktu ke domain frekuensi. FFT adalah algoritma versi tercepat untuk menghitung *Discete Fourier Transform* (DFT). DFT dinyatakan dengan persamaan [10] (4)

$$X(k) = \sum_{m=0}^{N-1} x(m) e^{-j2\pi km/N}, \qquad k = 0,1,2,\ldots,N-1, \tag{4}$$

dimana $j$ adalah bilangan imajiner yang bernilai $\sqrt{-1}$ dan $x(m)$ adalah sinyal suara. Frekuensi yang dihasilkan bergantung dari frekuensi sampling. Jika frekuensi sampling 16000 Hz maka frekuensi yang dihasilkan antara 0 sampai 8000 Hz. Hasil dari FFT adalah bilangan kompleks yang terdiri dari bilangan riil ($a$) dan imajiner ($b$). Sama dengan vektor, nilai besaran frekuensi (*magnitude*) didapatkan dengan ($\sqrt{a^2 + b^2}$) akar kuadrat $a$ ditambah $b$.

*E. Mel-frequency Filtering*

*Mel-frequency filtering* adalah analisis frekuensi menggunakan *filter bank*. Hasil dari proses ini adalah frekuensi yang dipetakan menggunakan *Mel-filter*, $H(k,m)$. *Mel-frequency filtering* dinyatakan dengan persamaan [12] (5).

$$D(m) = \sum_{k=0}^{N-1} |X(k)| * H(k,m), \qquad m = 1,2,\ldots,M \tag{5}$$

dimana $M$ adalah jumlah *filter bank*, $N$ adalah panjang FFT, $X(k)$ adalah hasil FFT ke-k, dan $H(k,M)$ adalah *Mel-filter*. Nilai $H(k,m)$ dinyatakan dengan persamaan [12] (6).

$$\begin{aligned}
H(k,m) &= 0, && k < f[m-1] \\
H(k,m) &= \frac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])}, && f[m-1] \le k < f[m] \\
H(k,m) &= \frac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m+1] - f[m])}, && f[m] \le k < f[m+1] \\
H(k,m) &= 0, && k \ge f[m+1]
\end{aligned} \tag{6}$$

dimana $f[m]$ respon dari *Mel* frekuensi didapatkan dengan persamaan [12] (7).

$$f[m] = \frac{N}{f_s} Mel^{-1}(Mel(f_l) + m\frac{Mel(f_h) - Mel(f_l)}{M+1}), \tag{7}$$

dimana $f_l$ dan $f_h$ adalah frekuensi batas bawah dan frekuensi batas atas yang ada didalam *filter bank*, $f_s$ adalah frekuensi sampling, $Mel(f)$ adalah *Mel-scale* dan $Mel^{-1}(f)$ adalah *invers Mel-scale*. *Mel-scale* dan *invers Mel-scale* didefinisikan dengan persamaan [12] (8) dan (9).

$$Mel(f) = 2595 * \log_{10}(1 + \frac{f}{700}), \tag{8}$$

$$Mel^{-1}(f) = 700 * (10^{\frac{f}{2595}} - 1), \tag{9}$$

dimana $f$ adalah frekuensi yang akan diskala menggunakan *Mel-scale*. Parameter $f$ dalam satuan Hz.

*F. Discrete Cosine Transform (DCT)*

Setelah didapatkan spektrum Mel, dilakukan operasi DCT. DCT adalah proses yang digunakan untuk mengembalikan dari domain frekuensi ke domain waktu [11]. DCT selain untuk mengembalikan ke domain waktu juga untuk kompresi spektrum [10]. Operasi DCT dapat dinyatakan dengan persamaan [11] (10).

$$C(k) = w(k) \sum_{m=1}^{L} D(m) \cos\left(\frac{\pi(2m-1)(k-1)}{2L}\right), k = 1,2\ldots,N \tag{10}$$

dimana $L$ adalah jumlah *Mel* spektrum *filter*, $D(m)$ adalah mel spektrum, dan $N$ adalah jumlah koefisien *cepstrum*. Sedangkan $w(k)$ dinyatakan dengan persamaan 11. Untuk komponen pertama, $C(0)$, tidak dipakai karena mengandung sedikit informasi [11].

$$w(k) = \begin{cases} \frac{1}{\sqrt{L}} & ,k = 1 \\ \sqrt{\frac{2}{L}} & ,2 \leq k \leq L \end{cases} \quad (11)$$

*G. Delta & Delta-delta Cepstrum*

Pendengaran manusia lebih sensitif terhadap karakteristik dinamik sinyal suara [12]. Turunan pertama (*delta*) dan turunan kedua (*delta-delta*) menggambarkan karakteristik dinamik dari sinyal suara. Nilai *delta cepstrum* didapatkan dari turunan pertama koefisien *cepstrum* yang dinyatakan dengan persamaan (12)

$$d(k) = \frac{\sum_{t=1}^{T} t * (C(k+t) - C(k-t))}{2\sum_{t=1}^{T} t^2}, \qquad 1 \leq k \leq N \quad (12)$$

dimana $C(k)$ adalah koefisien *cepstrum*, $N$ adalah jumlah koefisien *cepstrum*, dan $T$ adalah konstan. Nilai $T$ yang biasa digunakan adalah 2 [12]. Sedangkan *delta-delta cepstrum* didapatkan dari turunan pertama *delta cepstrum*.

*H. Log* Energy

Log *energy* adalah operasi sederhana yang didapatkan dari log penjumlahan kuadrat amplitudo. Log *energy* ($E$) dapat dinyatakan dengan persamaan [8] (13)

$$E = log \sum_{n=1}^{N} s(n)^2 \quad (13)$$

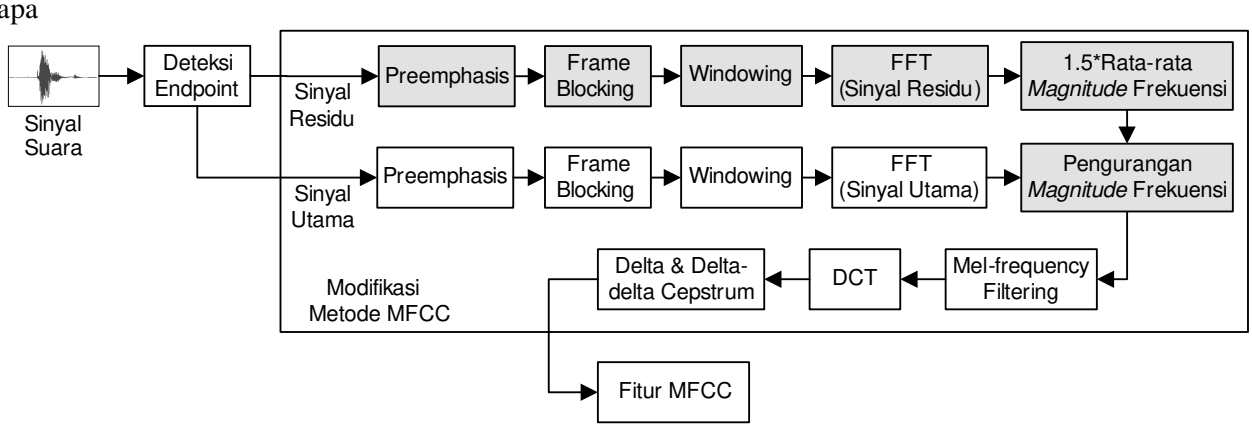
dimana $s(n)$ adalah sinyal suara didalam *frame* dan $N$ adalah panjang *frame*. Sinyal suara yang digunakan pada log *energy* adalah sinyal suara yang belum dilakukan operasi *preemphasis*. Kemudian dilakukan operasi *frame blocking* untuk membagi sinyal suara menjadi *frame-frame* kecil.

## III. METODE YANG DIUSULKAN

Sistem identifikasi pembicara secara umum dibagi menjadi empat bagian yaitu praproses, ekstraksi fitur, pelatihan, dan pengenalan. Bagian yang mengalami modifikasi adalah bagian ekstraksi fitur yaitu MFCC yang ditunjukkan pada Gambar 2. Bagian modifikasi metode MFCC terdapat penambahan blok proses yang ditandai dengan warna abu-abu. Bagian praproses terdiri dari proses deteksi *endpoint*. Sedangkan bagian pelatihan dan pengenalan menggunakan *Gaussian Mixture Model* (GMM).

Modifikasi metode MFCC menggunakan residu deteksi *endpoint* didasarkan pada *Spectral Subtraction* (SS) [13] dan penelitian Paresh dan Nikita [4]. Metode *Spectral Subtraction* menghilangkan derau berdasarkan informasi yang didapatkan dari beberapa *frame* awal. Sedangkan Paresh dan Nikita menghilangkan derau menggunakan Wiener *filter* yang diletakkan di domain frekuensi (setelah FFT) [4].

Penambahan beberapa blok proses dimaksudkan untuk mengurangi derau di domain frekuensi. Blok proses *preemphasis*, *frame blocking*, *windowing*, dan FFT (sinyal residu) digunakan untuk mengekstrak frekuensi derau. Cara kerjanya sama dengan metode *Spectral Subtraction* yaitu mendapatkan informasi frekuensi derau dari beberapa




Gambar 2. Modifikasi Metode *Mel Frequecy Cepstral Coefficient* (MFCC) Menggunakan Residu Deteksi *Endpoint*

frame awal. Bedanya, informasi frekuensi derau didapatkan dari residu deteksi *endpoint*. Kemudian dihitung rata-rata magnitude frekuensi untuk semua *frame*. Hasilnya digunakan untuk mengurangi magnitude frekuensi sinyal utama. Pengurangan ini terjadi di domain frekuensi seperti penelitian Paresh dan Nikita yang menghilangkan derau menggunakan Wienar *filter* di domain frekuensi [4].

### A. *Praproses (Deteksi* Endpoint*)*

Praposes adalah proses yang dikerjakan pada tahap awal sebelum ekstraksi fitur. Praproses sinyal suara digunakan untuk meningkatkan kualitas sinyal suara. Praproses yang digunakan pada penelitian ini adalah deteksi *endpoint*. Deteksi *endpoint* adalah algoritma yang memisahkan antara *speech* dan *nonspeech*. *Speech* berisi sinyal utama sedangkan *nonspeech* adalah daerah sunyi/hening. Daerah *nonspeech* ini bisa disebut juga sebagai residu dari algoritma deteksi *endpoint*.

Deteksi *endpoint* yang digunakan pada penelitian ini berdasarkan *short-time energy*. *Short-time energy* menggambarkan kuantitas sinyal suara yang dinyatakan dengan kuadrat dari amplitudo [5]. Pada *short-time energy* terdapat variabel *alpha*, jumlah sampel dalam satu *frame* atau *N*, jumlah sampel yang tumpang tindih atau *M*, fungsi window atau *W* dan *threshold*.

### B. *Preemphasis, Frame Blocking, Windowing, dan FFT*

Sinyal suara setelah melewati deteksi *endpoint* akan terbagi menjadi dua bagian yaitu sinyal utama (*speech*) dan sinyal residu (*nonspeech*). Sinyal residu kemudian diekstrak untuk mendapatkan *magnitude* frekuensi menggunakan beberapa proses yaitu *preemphasis*, *frame blocking*, *windowing*, dan FFT. *Magnitude* frekuensi sinyal residu digunakan untuk mengurangi *magnitude* frekuensi sinyal utama. Blok proses pada bagian modifikasi metode MFCC ditunjukkan pada Gambar 2 yang diberi warna abu-abu.

### C. *Rata-rata* Magnitude *Frekuensi*

Rata-rata *magnitude* frekuensi adalah rata-rata *magnitude* frekuensi sinyal residu pada semua *frame*. Dari proses *frame blocking* dihasilkan beberapa *frame* yang mana setiap *frame* berisi nilai *magnitude* setiap frekuensi. Nilai rata-rata yang dimaksud adalah rata-rata setiap frekuensi dari semua *frame* yang sudah dihasilkan. Misalkan dari proses *frame blocking* dihasilkan tiga *frame* yaitu *F1*, *F2*, dan *F3*. Kemudian dilakukan operasi FFT untuk merubah dari domain waktu ke domain frekuensi sehingga dihasilkan *magnitude* frekuensi. Didalam *F1* dihasilkan vektor *magnitude* frekuensi dengan panjang *D*. Rata-rata *magnitude* frekuensi didapatkan dengan menjumlahkan vektor *F1*, *F2*, dan *F3* dibagi jumlah *frame* dalam contoh ini adalah tiga. Rata-rata digunakan untuk mendapatkan estimasi *magnitude* fekuensi derau.

*Magnitude* sinyal yang dirata-rata adalah sinyal residu hasil dari proses deteksi *endpoint*. Karena sinyal residu letaknya diawal atau diakhir dari sinyal suara, maka *magnitude* yang didapatkan tidak penuh seperti *magnitude* sinyal suara yang berada ditengah. Ketika diawal, sinyal suara cenderung naik sedangkan diakhir sinyal suara cenderung turun sehingga mengurangi *magnitude* yang dihasilkan. Oleh karena itu, pada proses rata-rata *magnitude* frekuensi terdapat nilai pengalih. Nilai pengalih yang digunakan adalah 1.5. Nilai pengalih didapatkan dari uji coba pada sampel data dan didapatkan nilai terbaik sebesar 1.5.

### D. *Pengurangan* Magnitude *Frekuensi*

Setelah didapatkan rata-rata *magnitude* frekuensi derau, selanjutnya *magnitude* frekuensi dari sinyal utama dikurangi *magnitude* frekurensi dari sinyal residu. Hasil dari proses ini adalah *magnitude* frekuensi yang bebas derau. Proses ini dilakukan untuk semua *frame* pada sinyal utama.

## IV. UJI COBA DAN EVALUASI

*Dataset* yang digunakan pada penelitian ini ada dua yaitu *dataset* pembicara dan *dataset* derau. *Dataset* pembicara diperoleh dari database CHAINS corpus dengan alamat web http://chains.ucd.ie/corpus.php. Sedangkan *dataset* derau diperoleh dari database *Signal Processing Information Base* (SPIB) dengan alamat web http://spib.linse.ufsc.br. Database pembicara yang terdiri dari 36 pembicara dengan kondisi berbicara sendiri (*solo speech*). Teks yang digunakan pada penelitian ini adalah kalimat tunggal nomor tiga dengan teks "*play in the street up ahead*". Tipe derau yang digunakan pada penelitian ini adalah derau tipe *white noise*, *pink noise*, *cockpit noise 3* (F16), *vehicle interior noise* (Volvo 340), dan *HF channel noise*. Frekuensi sampling pada *dataset* derau adalah 16000 Hz.

Lama waktu yang dihasilkan manusia ketika berbicara satu kalimat untuk kali kedua atau lebih tidak pernah sama meskipun untuk kalimat yang sama. Untuk menirukan kondisi seperti itu perlu dilakukan duplikasi dengan modifikasi. Duplikasi dengan modifikasi pada penelitian ini adalah memperbanyak file teks dengan tempo yang berbeda-beda. Dari satu file teks kemudian diduplikasi dengan modifikasi sebanyak sepuluh file dengan tempo

−50%, −40%, −30%, −20%, −10%, +10%, +20%, +30%, +40%, dan +50%. Nilai tempo positif menunjukkan sinyal suara menjadi lebih pendek dan nilai negatif menunjukkan sinyal suara menjadi lebih panjang. Aplikasi yang digunakan untuk duplikasi dengan modifikasi file suara adalah Audacity. Dalam satu pembicara jumlah file menjadi sebelas. Karena jumlah pembicara ada 36 pembicara sehingga total dalam satu dataset sebanyak 396 file.

Dataset pembicara dan dataset derau diolah untuk pembuatan dataset sinyal suara bersih dan dataset sinyal suara berderau. Pembuatan dataset sinyal suara berderau dengan cara menambahkan sinyal derau ke sinyal suara yang bersih. Dalam satu dataset sinyal suara yang bersih hanya ditambahi satu tipe derau dengan satu tingkat SNR. Untuk membuat dataset sinyal suara berderau dengan tipe derau dan tingkat SNR yang lain perlu dilakukan duplikasi dataset sinyal suara yang bersih kemudian ditambahi sinyal derau dengan tipe derau dan tingkat SNR yang berbeda. Tingkat SNR yang digunakan adalah bersih, 25, 20, 15, 10, 5, dan 0 dB. Karena frekuensi *sampling dataset* pembicara 44100 Hz dan *dataset* derau 16000 Hz maka perlu dilakukan penyamaan frekuensi *sampling* menjadi 16000 Hz. Kemudian amplitudo dari setiap *dataset* yang terbentuk dinormalisasi sehingga amplitudo sinyal bernilai antara -1 sampai 1.

Ketika pelatihan menggunakan *dataset* sinyal suara yang bersih. Sedangkan ketika pengenalan menggunakan *dataset* sinyal yang berderau. Tidak semua file yang ada pada pembicara digunakan untuk pelatihan dan pengenalan. Dari sebelas file pada setiap pembicara hanya sepuluh yang digunakan ketika pelatihan. Sedangkan ketika pengenalan hanya menggunakan satu file pada setiap pembicara. Jika file nomor satu dari setiap pembicara pada *dataset* sinyal suara yang berderau digunakan untuk pengenalan maka file nomor dua sampai sebelas dari setiap pembicara pada *dataset* sinyal suara yang bersih digunakan untuk pelatihan. Kemudian diulangi sebanyak lima kali untuk file yang sama dan diambil nilai akurasi maksimum. Uji coba diulangi lagi untuk file nomor dua dari setiap pembicara pada *dataset* sinyal suara yang berderau sebagai data pengenalan dan file nomor satu, tiga, sampai sebelas dari setiap pembicara pada *dataset* sinyal suara yang bersih sebagai data pelatihan. Begitu seterusnya sampai file nomor sebelas sebagai data pengenalan dan file nomor satu sampai sepuluh sebagai data pelatihan. Kemudian diambil nilai rata-rata untuk setiap pembicara.

Uji coba diulangi sebanyak lima kali untuk file yang sama dikarenakan metode *Gaussian Mixture Model* (GMM) yang digunakan untuk tidak bisa menemukan global optimum. Yang ditemukan oleh GMM adalah lokal optimum. Hal ini terjadi karena ada proses random di GMM yaitu ketika pemilihan *M* vektor fitur sebagai komponen GMM. Variabel *M* adalah jumlah komponen GMM yang digunakan ketika uji coba.

Pada uji coba dilakukan percobaan menggunakan metode yang diusulkan, MFCC, MFCC+SS, dan MFCC+wiener dengan parameter jumlah komponen GMM sebesar 8, 16, 32, dan 64. Nilai parameter yang digunakan oleh keempat metode tersebut sama seperti yang ditunjukkan pada Tabel I dan II. Dari hasil percobaan tersebut kemudian dilakukan uji t untuk mengetahui pada jumlah komponen GMM berapa metode yang diusulkan menghasilkan akurasi lebih tinggi dari metode pembanding.

Terdapat tiga uji t yaitu uji t akurasi metode ekstraksi fitur terhadap pembicara yang berbeda-beda, uji t akurasi metode ekstraksi fitur terhadap tipe derau yang berbeda-beda, dan uji t akurasi metode ekstraksi fitur terhadap tingkat SNR yang berbeda-beda. Hasil uji t dengan jumlah komponen GMM sebesar 8, 16, 32, dan 64 ditunjukkan pada Tabel III, Tabel IV, dan Tabel V. Pada Tabel III, nilai t hitung ketika komponen GMM sebesar 8 kurang dari nilai t kritis sehingga kemungkinan yang bisa dipilih adalah komponen GMM 16, 32, dan 64. Pada Tabel IV, nilai t hitung ketika komponen GMM sebesar 8 dan 16 kurang dari nilai t kritis sehingga kemungkinan yang bisa dipilih adalah komponen GMM 32 dan 64. Sedangkan pada Tabel V nilai t hitung dari semua komponen GMM kurang dari nilai t kritis sehingga tidak ada pilihan lain kecuali memilih nilai t hitung yang paling besar yaitu ketika komponen GMM sebesar 32. Dengan demikian komponen GMM yang bisa dipilih dari kemungkinan yang ada adalah komponen GMM sebesar 32.

TABEL I
NILAI PARAMETER YANG DIGUNAKAN PADA UJI COBA

| Parameter | Nilai |
|---|---|
| Alpha preemphasis | 0.97 |
| Frame length | 480 |
| Frame shift | 160 |
| Panjang FFT | 1024 |
| Jumlah Mel filter bank (*M*) | 25 |
| Frekuensi batas bawah Mel filter bank | 300 |
| Frekuensi batas atas Mel filter bank | 4000 |
| Jumlah koefisien DCT (*N*) | 13 |
| Jumlah komponen GMM | 32 |

TABEL II
NILAI PARAMETER PADA ALGORITMA *ENDPOINT DETECTION*

| Parameter | Nilai |
|---|---|
| Alpha | 0.97 |
| Frame length | 480 |
| Frame shift | 160 |
| Threshold | rata-rata log energi |

TABEL III
NILAI UJI T AKURASI METODE EKSTRAKSI FITUR TERHADAP PEMBICARA YANG BERBEDA-BEDA (T KRITIS = 1.69)

| Komponen GMM | Metode yang diusulkan dibandingkan MFCC | Metode yang diusulkan dibandingkan MFCC+SS | Metode yang diusulkan dibandingkan MFCC+wiener |
|---|---|---|---|
| 8 | 7.989 | 0.018 | 4.526 |
| 16 | 10.856 | 2.496 | 3.994 |
| 32 | 12.382 | 5.089 | 3.201 |
| 64 | 11.095 | 4.080 | 2.150 |

Ket: Nilai t hitung kurang dari nilai t kritis

TABEL IV
NILAI UJI T AKURASI METODE EKSTRAKSI FITUR TERHADAP TIPE DERAU YANG BERBEDA-BEDA (T KRITIS = 2.13)

| Komponen GMM | Metode yang diusulkan dibandingkan MFCC | Metode yang diusulkan dibandingkan MFCC+SS | Metode yang diusulkan dibandingkan MFCC+wiener |
|---|---|---|---|
| 8 | 2.855 | 0.010 | 10.775 |
| 16 | 3.415 | 1.327 | 10.855 |
| 32 | 3.643 | 2.334 | 6.040 |
| 64 | 3.778 | 2.810 | 3.909 |

Ket: Nilai t hitung kurang dari nilai t kritis

TABEL V
NILAI UJI T AKURASI METODE EKSTRAKSI FITUR TERHADAP TINGKAT SNR YANG BERBEDA-BEDA (T KRITIS = 1.94)

| Komponen GMM | Metode yang diusulkan dibandingkan MFCC | Metode yang diusulkan dibandingkan MFCC+SS | Metode yang diusulkan dibandingkan MFCC+wiener |
|---|---|---|---|
| 8 | 1.884 | 0.032 | 6.214 |
| 16 | 1.916 | 1.357 | 5.357 |
| 32 | 2.004 | 1.401 | 6.432 |
| 64 | 1.997 | 1.365 | 5.955 |

Ket: Nilai t hitung kurang dari nilai t kritis

## V. KESIMPULAN

Dari hasil uji coba dapat diambil kesimpulan bahwa jumlah komponen GMM yang optimal untuk sistem identifikasi pembicara menggunakan residu *deteksi* endpoint sebanyak 32 komponen. Selain itu sistem identifikasi pembicara menggunakan residu dari deteksi *endpoint* menghasilkan akurasi lebih tinggi dari metode MFCC, MFCC+SS, dan MFCC+wiener yang dibuktikan dengan nilai uji t yang lebih tinggi dari nilai t kritis.

## VI. DAFTAR PUSTAKA

[1] Y. Zhang dan W.H. Abdulla, (2007), Robust Speaker Identification in Noisy Environment using Cross Diagonal GTF-ICA Feature, *6th International Conference on Information, Communication & Signal Processing*, hal. 1-4.

[2] S. M. Lee, S. H. Fang, J. W. Hung, dan L. S. Lee, (2002), Improved MFCC Feature Extraction by PCA-Optimized Filter Bank for Speech Recognition, *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'01)*, hal. 49-52.

[3] Wu Zunjing dan Cao Zhigang (2005), Improved MFCC-Based Feature for Robust Speaker Identification, *Tsinghua Science and Technology*, 10(2), hal. 158-161.

[4] P.M. Chauhan dan N.P. Desai, (2014), Mel Frequency Cepstral Coefficients (MFCC) Based Speaker Identification in Noisy Environment Using Wiener Filter, *2014 International Conference on Green Computing Communication and Electrical Engineering*, hal. 1-5.

[5] S. Yong dan C. Leimin, (2011), Performance Comparison of New Endpoint Detection Method in Noise Environments, *2011 International Conference on Electric Information and Control Engineering (ICEICE)*, hal. 1523-1527.

[6] Bing-Fei Wu dan Kun-Ching Wang, (2005), Robust Endpoint Detection Algorithm Based on the Adaptive Band-Partitioning Spectral Entropy in Adverse Environments, *IEEE Transactions on Speech and Audio Processing*, 13(5), hal. 762-775.

[7] S. Young, (1996), A Review of Large-Vocabulary Continuous-Speech Recognition, *IEEE Signal Processing Magazine*, 13(5).

[8] W. Han, C.F. Chan, C.S. Choy, dan K.P. Pun, (2006), An Efficient MFCC Extraction Method in Speech Recognition, *2006 IEEE International Symposium on Circuits and Systems*, doi: 10.1109/ISCAS.2006.1692543.

[9] R. Vergin dan D. O'Shaughnessy, (1995), Pre-Emphasis and Speech Recognition, *Canadian Conference on Electrical and Computer Engineering*, 2, hal. 1062-1065.

[10] S. Gupta, J. Jaafar, W.F. Ahmad, dan A. Bansal, (2013), Feature Extraction Using MFCC, Signal & Image Processing: An International Journal

(SIPIJ), 4(4), hal. 101-108.

[11] C.G.K Leon, (2009), Robust Computer Voice Recognition Using Improved MFCC Algorithm, *2009 International Conference on New Trends in Information and Service Science*, hal. 835-840.

[12] W. Junqin dan Y. Junjun, (2011), An Improved Arithmetic of MFCC in Speech Recognition System, *2011 International Conference on Electronics, Communication and Control (ICECC)*, hal. 719-722.

[13] J.S. Lim dan A.V. Oppenheim, (1979), Enhancement and Bandwidth Compression of Noisy Speech, *Proceedings of The IEEE*, 67(12), hal. 1586-1604