

Penerapan *Hidden Markov Model* (HMM) pada Pengenalan Penutur

Mukhlisa*¹, Maryati Gultom², Derry Alamsyah³

^{1,2,3}STMIK GI MDP; Jln. Rajawali No. 14, 0711 376400

^{1,2,3}Program Studi Teknik Informatika STMIK GI MDP, Palembang

e-mail: *¹immobulus92@gmail.com, ²gultommaryati@gmail.com, ³derry@mdp.ac.id

Abstrak

Suara manusia bergantung pada posisi atau bentuk rongga yang dimiliki, sehingga karakter suara yang dimiliki setiap orang unik dan menjadi identitasnya. Pengenal penutur (*speaker recognition*) adalah proses mengenali siapa yang bicara berdasarkan informasi yang terdapat di dalam gelombang wicara. Pengenal penutur dapat digunakan sebagai sistem absensi, keamanan dan sebagainya. Sistem pengenal penutur pada penelitian ini dibentuk melalui dua proses utama yaitu pelatihan (*training*) dan pengenalan (*recognition*), dimana Mel Frequency Cepstrum Coefficients (MFCC) digunakan untuk ekstraksi fitur, kemudian model suara dibentuk berdasarkan *Hidden Markov Model* (HMM). Hasil penelitian menunjukkan bahwa pengujian secara real time menggunakan mikrofon tingkat akurasi 30%. Sedangkan pengujian dari file rekaman 100%. Tingkat akurasi sangat bergantung pada kemampuan clustering dan klasifikasi.

Kata kunci—Pengenal Penutur, Mel Frequency Cepstrum Coefficients (MFCC), *Hidden Markov Model* (HMM).

Abstract

Human voice depends on the position or shape of the cavity owned, so the character of the sound that each person is unique and became his identity. Identification of speakers (*speaker recognition*) is the process of identifying who is talking on the information contained in the speech wave. Identification of speakers can be used as attendance systems, security, and so on. *Speaker recognition* system in this study formed through two main processes of training (*training*) and recognition (*recognition*), where Mel Frequency Cepstrum Coefficients (MFCC) are used for feature extraction, then the model is formed based *Hidden Markov sound model* (HMM). The results showed that the test in real time using a microphone accuracy rate of 30%. While testing of the recording file 100%. The level of accuracy depends heavily on the ability of clustering and classification.

Keywords—*Speaker Recognition*, Mel Frequency Cepstrum Coefficients (MFCC) , *Hidden Markov Model* (HMM)

1. PENDAHULUAN

Suara manusia dihasilkan oleh pita suara yang menghasilkan bunyi yang berbeda-beda. Setiap individu manusia memiliki suara yang beraneka ragam tergantung dari posisi atau bentuk rongga seseorang. Hal ini menyebabkan karakter suara yang dimiliki seseorang unik.

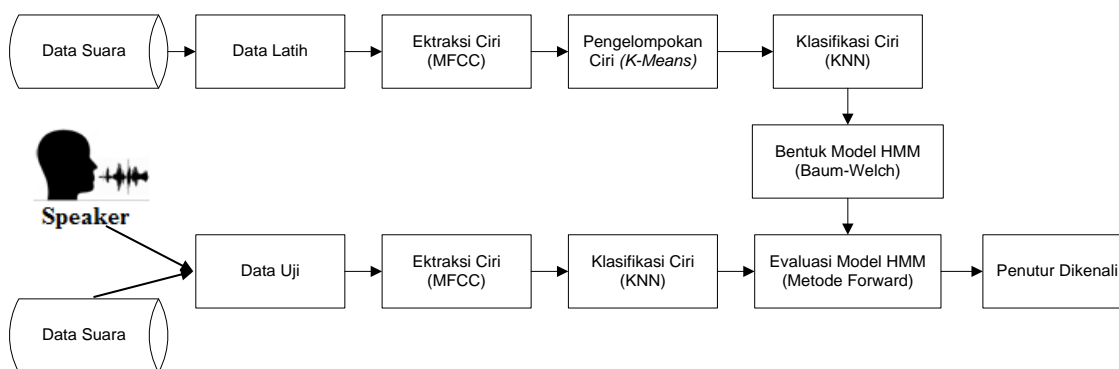
Speaker recognition adalah proses secara otomatis mengenali siapa yang berbicara berdasarkan informasi individu yang terdapat di dalam gelombang suara. Teknik ini

memungkinkan menggunakan suara penutur untuk memverifikasi identitas penutur dan mengontrol layanan seperti menekan nomor telepon dengan suara (*voice dialing*), absensi akademik, kontrol keamanan area rahasia dan akses jarak jauh dengan komputer [1].

Pada penelitian ini akan dilakukan pengenalan penutur menggunakan Model Markov Tersembunyi atau biasa disebut *Hidden Markov Model* (HMM). Hal ini dikarenakan metode HMM telah terbukti memiliki tingkat ketepatan yang tinggi untuk identifikasi suara [1] dibanding metode *Vector Quantization* (VQ) [2] dan *fMAPLR* [3].

2. METODE PENELITIAN

Pada penelitian ini pengenalan penutur dilakukan melalui skema yang ditunjukkan pada Gambar 1. Pembagian data dibagi kedalam dua bagian yaitu bagian data latih dan data uji. Data yang digunakan berupa sinyal yang telah diakuisisi dengan isi data berupa penyebutan kata "Satu". Data latih masing-masing penutur direkam sebanyak 10 kali, sedangkan untuk data uji dibagi dua bagian yaitu berdasarkan rekaman sebanyak 10 kali dan *real time*. Selanjutnya dilakukan proses pengelompokan ciri yang didapat dari sinyal suara untuk memprediksi model yang dimiliki oleh masing-masing penutur.



Gambar 1. Skema Pengenalan Penutur

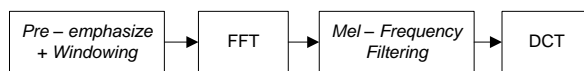
2.1 Mel Frequency Cepstrum Coefficient

Suara manusia dihasilkan dari sebuah kerjasama antara *lungs* (paru-paru), *glottis* (dengan *vocal cords*) dan *articulation tract* (mulut dan rongga hidung). Variasi sinyal suara dipengaruhi oleh tekanan udara yang melewati pita suara dan *vocal tract* yaitu mulut, lidah, gigi, bibir dan langit-langit mulut [4].

Sinyal suara merupakan sinyal yang berubah terhadap waktu secara perlahan atau biasa disebut *quasi-stationary*. Ketika diamati dalam jangka waktu yang sangat singkat (5-100ms) karakteristiknya hampir sama namun, dalam jangka waktu panjang (0.2s atau lebih) karakteristik sinyal berubah dan merefleksikan perbedaan suara yang diucapkan. Oleh karena itu, digunakan spektrum waktu singkat (*short-time spectral analysis*) untuk mengkarakterisasi sinyal suara [5]. Metode yang digunakan untuk proses *ekstraksi* ciri adalah MFCC (*Mel-Frequency Cepstrum Coefficients*).

Langkah-langkah untuk menghitung MFCC ditunjukkan pada Gambar 2, yaitu *Fast Fourier Transform*, memfilter dengan *filterMel* dan *cosine transform* dari vektor *log* energi. MFCC dimulai dihitung dengan mengambil *windowed frame* dari sinyal suara, lalu menggunakan *Fast Fourier Transform* (FFT) untuk memperoleh parameter tertentu dan kemudian diubah menjadi skala *Mel* untuk memperoleh *vektor fitur* yang mewakili amplitudo terkompres secara logaritmik dan informasi frekuensi yang sederhana. Kemudian dihitung dengan mengaplikasikan *discrete cosine transform* kepada *log* dari *Mel-filter bank*. Rumus dari skala *Mel* adalah:

$$mel(f) = 2595 * \log_{10} \left(1 + \frac{f}{700} \right) \tag{1}$$



Gambar 2. Blok Diagram MFCC

2.2 K-Means

Algoritma *K-Means* adalah algoritma pengelompokan (*clustering*) berbasis jarak yang membagi *N* objek pengamatan ke dalam *K* kelompok (*cluster*).

Langkah-langkah algoritma *K-Means* secara umum yaitu [6]:

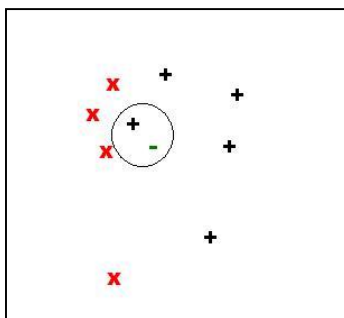
1. Tentukan jumlah kelompok
2. Alokasikan data ke dalam kelompok secara acak
3. Hitung pusat kelompok (*centroid*/rata-rata) dari data yang ada di masing-masing kelompok
4. Alokasikan masing-masing data ke *centroid*/rata-rata terdekat
5. Kembali ke langkah 3, apabila masih ada data yang berpindah kelompok atau apabila perubahan nilai *centroid* di atas nilai ambang yang ditentukan atau apabila perubahan nilai pada fungsi objektif yang digunakan masih di atas nilai ambang yang ditentukan.

Pada penelitian ini penulis menggunakan pengukuran pada jarak ruang (*distance space*) *Euclidean* karena ingin memberikan jarak terpendek antara dua titik *linier*.

$$D(x_2, x_1) = \|x_2 - x_1\| = \sqrt{\sum_{j=1}^p |x_{2j} - x_{1j}|^2} \tag{2}$$

2.3 K-Nearest Neighbour

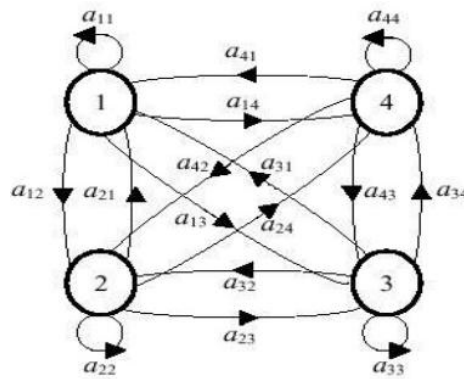
Algoritma *K-Nearest Neighbor (K-NN)* merupakan algoritma yang melakukan klasifikasi berdasarkan kedekatan lokasi (jarak) suatu data dengan data yang lain. Pada algoritma *K-NN*, data berdimensi *q*, jarak dari data tersebut ke data yang lain dapat dihitung. Nilai jarak inilah yang digunakan sebagai nilai kedekatan /kemiripan antara data uji dengan data latih. Nilai *K* pada *K-NN* berarti *K*-data terdekat dari data uji [6]. Gambar 3 memberikan contoh algoritma *K-NN*.



Gambar 3. Algoritma KNN

2.4 Hidden Markov Model

Hidden Markov Model (HMM) merupakan suatu model stokastik yang menggambarkan dua keterkaitan antara peubah yaitu peubah yang tak teramati (*hidden state*) dari waktu ke waktu, serta peubah yang teramati (*observable state*). Pada penelitian ini, jenis HMM yang digunakan adalah model *ergodic*, seperti yang ditunjukkan pada Gambar 4.



Gambar 4. HMM Ergodic Model

Pembentukan HMM, pertama tentukan hidden state-nya $S = \{S_1, S_2, \dots, S_N\}$, dimana N adalah jumlah hidden state. Pada penelitian ini jumlah hidden state-nya adalah 6. Karena menurut error minimum berada pada state yang berjumlah 6 [7]. Kemudian alokasikan probabilitas transisi antar state (*matriks transisi*),

$$A = \{a_{ij}\}, \text{ untuk } 1 \leq i, j \leq N, \quad (3)$$

$$a_{ij} = P[q_{t+1} = S_j / q_t = S_i], \quad (4)$$

yaitu peluang berada di state j pada waktu $t + 1$ jika pada waktu t berada di state i . Input untuk membangun HMM yaitu sinyal suara yang telah ditransformasi menjadi barisan vektor ciri, $O = \{O_1, O_2, \dots, O_M\}$, dimana M adalah panjang observasi, ke peluang kemunculannya pada suatu barisan hidden state tertentu. Pada penelitian ini panjang observasi diambil dari jumlah kluster yaitu 16 yang didapat dari kaidah Thumb, $k \approx \sqrt{\frac{n}{2}}$, dimana n adalah hasil dari ekstraksi suara penutur.

Probabilitas kemunculan peubah yang terobservasi pada suatu state (*matriks emisi*) dinotasikan dengan

$$B = \{b_j(k)\}, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M, \quad (5)$$

$$b_j(k) = P[O_k \text{ terhadap } t / q_t = S_j], \quad 0 \leq b_j(k) \text{ dan } \sum_{k=1}^M b_j(k) = 1 \quad (6)$$

Matriks distribusi probabilitas pada tahap awal berada di state i dinotasikan dengan

$$\pi = \{\pi_i\}, \text{ untuk } 1 \leq i \leq N \quad (7)$$

$$\pi_i = P[q_1 = S_i], \quad 0 \leq \pi_i \text{ dan } \sum_{i=1}^N \pi_i = 1. \quad (8)$$

Dengan menggunakan notasi-notasi tersebut, maka suatu HMM dapat dilambangkan dengan

$$\lambda = (A, B, \pi). \quad (9)$$

2.5 Evaluasi Observasi: Algoritma Forward-Backward

Variabel algoritma *forward* $\alpha_i = P(O_1, O_2, \dots, O_T, q_t = S_i | \lambda)$. Berikut ini langkah-langkah dalam algoritma *Forward* [8]:

1. Inisialisasi

$$a_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N \tag{10}$$

2. Induksi

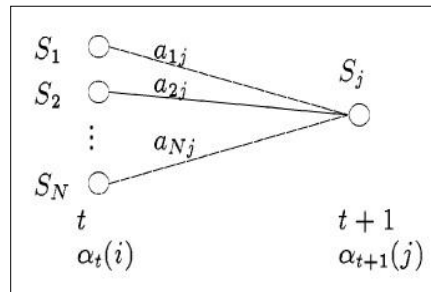
$$\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i) a_{ij}] b_j(o_{t+1}), \tag{11}$$

dengan $1 \leq t \leq T-1$ dan $1 \leq j \leq N$

3. Terminasi

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i), \quad 1 \leq i \leq N \tag{12}$$

Ilustrasi algoritma *forward* dapat dilihat pada Gambar 5.



Gambar 5. Ilustrasi Alur Algoritma *Forward*

Variabel algoritma *Backward*: $\beta_t = P(O_1, O_2, \dots, O_T, q_t = S_i | \lambda)$. Berikut ini langkah-langkah dalam algoritma *Backward* [8]:

1. Inisialisasi

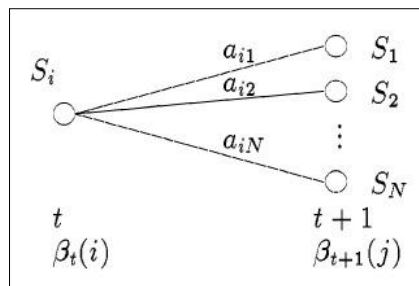
$$\beta_T(i) = 1, \quad 1 \leq i \leq N \tag{13}$$

2. Induksi

$$\beta_t(i) = [\sum_{j=1}^N \alpha_{i,j} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j)], \tag{14}$$

dengan $t = T-1, T-2, \dots, 1$ dan $1 \leq i, j \leq N$

Ilustrasi untuk algoritma *backward* dapat dilihat pada Gambar 6.



Gambar 6. Ilustrasi Alur Algoritma *Backward*

2.6 Model Latih Algoritma Baum Welch

Algoritma *BaumWelch* melibatkan algoritma *forward* dan algoritma *backward*. Untuk menggambarkan prosedur *update* parameter HMM, diperlukan variabel $\xi_t(i, j)$ yang merupakan peluang gabungan *state* i dan *state* j terhadap peluang observasi pada model yang diberikan dan $\gamma_t(i)$ *state* pada waktu dan merepresentasikan peluang berada di *state* i pada waktu t . Secara matematis nilai $\xi_t(i, j)$ dan $\gamma_t(i)$ dapat diformulasikan dengan persamaan berikut [8]:

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) = \frac{\alpha_t[i] a_{i,j} b_j[o_{t+1}] \beta_{t+1}[j]}{P(O|\lambda)} \quad (15)$$

Variabel *state*:

$$\gamma_t(i) = P(q_t = S_i | O, \lambda) = \sum_{j=1}^n \xi_t(i, j) \quad (16)$$

Dengan menggunakan persamaan (15) dan (16) maka persamaan untuk meng-*update* parameter-parameter (A , B , π) pada HMM dapat dirumuskan seperti pada persamaan (17,18,19).

1. Probabilitas *state* transisi:

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (17)$$

dengan $1 \leq i \leq N$, $1 \leq j \leq M$

2. Simbol probabilitas emisi

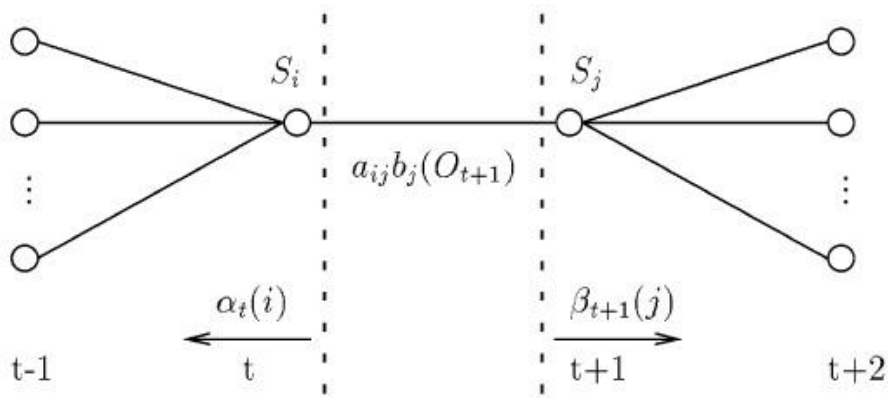
$$\bar{b}_i(k) = \frac{\sum_{t=1}^T O_t = v_k \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)} \quad (18)$$

dengan $1 \leq i \leq N$, $1 \leq j \leq M$

3. Probabilitas *state* awal

$$\bar{\pi}_i = \gamma_1(i), 1 \leq i \leq N \quad (19)$$

Ilustrasi mengenai algoritma *Baum-Welch* dapat dilihat pada Gambar 7.



Gambar 7. Ilustrasi Perhitungan pada Algoritma Baum Welch

3. HASIL DAN PEMBAHASAN

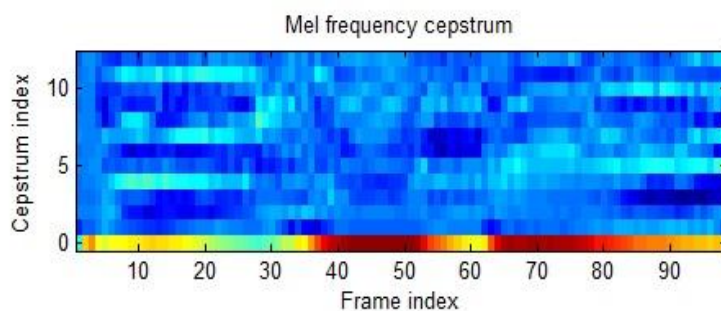
Aplikasi pengenalan penutur dibagi kedalam empat bagian yaitu tampilan utama, latihan, uji dari data *real time* dan uji dari data rekaman ditunjukkan oleh Gambar 8.



Gambar 8. Aplikasi Pengenal Penutur. (a) Tampilan Utama (b) Tampilan Latih (c) Tampilan Uji *real time* (d) Uji rekaman

3.1 Ekstraksi Ciri

Pada proses ekstraksi ciri dari masing-masing penutur akan ada hasil gelombang *spektrogram* dari masing-masing penutur.

Gambar 9. *Spektrogram* Penutur

Gambar 9 memperlihatkan warna berbeda-beda dari setiap penutur. Daerah gelap menunjukkan puncak (*formants*) dalam *spektrum*. Hal ini dikarenakan *frekuensi* dan *amplitudo* terhadap waktu masing-masing penutur yang berbeda.

Hasil ekstraksi suara dari masing-masing penutur akan menghasilkan nilai ciri yang berbeda. Hasil ekstraksi fitur didapat dari nilai rata-rata *magnitude* dimana nilai tersebut merupakan hasil dari nilai analisis *frame duration* 25 ms, nilai analisis *frame shift* 10 ms, nilai *pre-emphasis coefficient* 0,97, nilai *frequency range* yang bernilai dari 300 sampai 3700, nilai *filterbank* 20, nilai *cepstral coefficient* 13, nilai *cepstral sine filter* parameter 22. Hasil ekstraksi fitur dari masing-masing penutur dapat dilihat pada Tabel 1. Hasil ekstraksi merupakan bilangan sembarang bilangan riil yang berdistribusi berdasarkan model suara seseorang.

Tabel 1 Nilai Ciri Masing-Masing Penutur

No Ciri	Penutur 1	Penutur 2	Penutur 3	Penutur 4
1	29,22610	16,04124	26,7916	27,63623
2	-5,57114	-11,2210	-8,7203	-5,64174
3	-0,12888	-0,09575	0,0319	1,36485
4	-5,03435	-2,99690	-6,2693	-2,16470
5	-0,18315	-3,61515	1,8317	3,38628
6	2,34311	-3,59859	9,9454	0,29152
7	-4,39527	-1,88607	9,3361	-0,18776
8	-0,28316	-2,48388	1,4107	-6,85013
9	4,08185	-6,03756	2,5771	-1,16713
10	4,37304	0,35020	-0,5023	8,83512
11	-0,80525	1,45071	1,1137	5,03163
12	-0,38473	-0,30243	5,6076	7,77480
13	1,19224	-1,22829	-1,1413	-2,16825

3.2 Klasifikasi Ciri

Untuk melakukan klasifikasi ciri perlu dilakukan pengelompokan. Hal ini dikarenakan ciri yang didapat hanya berupa data mentah. Pada penelitian ini penulis menggunakan algoritma *K-Means*. Hasil *centroid* yang diperoleh dari jumlah pengelompokan sebanyak 16 kelompok berdasarkan kaidah *Thumb* $k \approx \sqrt{\frac{n}{2}}$, dimana n adalah nilai hasil MFCC. Hasil *centroid* dapat dilihat pada Tabel 2.

Tabel 2 Pengelompokan Ciri

No Pengelompokan	Centroid
1	57,3000997070308
2	-4,07408052610624
3	-5,57250557390312
4	3,04384381688059
5	-9,78095907038631
6	0,466027979736970
7	0,837583575180137
8	-0,426714379405893
9	-1,17634593977207
10	5,13794154819773
11	2,01781248157463
12	0,0571543921558319
13	-7,47826263895440
14	1,34483886304448
15	-2,16822428329575
16	-3,04853318250440

Setelah dilakukan pengelompokan dari masing-masing ciri penutur maka pengelompokan tersebut harus diklasifikasi. Pada penelitian ini penulis menggunakan algoritma *K-Nearest Neighbor* (K-NN) sebagai algoritma klasifikasi. Nilai ciri atau hasil ekstraksi dari masing-masing penutur diklasifikasi berdasarkan jarak terdekat dari nilai *centroid* yang diperoleh dari pengelompokan sebelumnya sehingga setiap nilai ciri tersebut dapat diklasifikasi. Hasil nilai klasifikasi dari nilai ciri penutur dapat dilihat pada Tabel 3.

Tabel 2 merupakan titik pusat untuk mengelompokan data kedalam kelas-kelas tertentu, dalam hal ini kelas observasi. Sebagai contoh, Tabel 3 menyajikan pengelompokan berdasarkan jarak nilai ciri dengan nilai *centroid* pada Tabel 2.

Tabel 3 Klasifikasi Ciri Penutur 1

No Ciri	Nilai Ciri	Klasifikasi
1	27,95505	9
2	-7,17575	6
3	-1,22671	16
4	-3,88804	3
5	-2,92968	10
6	-1,41628	16
7	1,64985	13
8	2,80069	12
9	1,65864	13
10	0,60345	15
11	1,22852	4
12	0,46984	15
13	-0,05473	14

3.3 Pemodelan

Proses pemodelan yang digunakan oleh penulis untuk penelitian ini adalah *Hidden Markov Model* yang akan menghasilkan *state transisi* dan *state emisi*. Nilai-nilai *state transisi* dapat dilihat pada Tabel 5 dan *state emisi* dapat dilihat pada Tabel 6.

Tabel 4 Probabilitas *State* Transisi

HiddenStates	1	2	3	4	5	6
1	0,192536	1,9 7E-01	2,0 3E-01	0,127705	1,3 8E-01	0,14153
2	8,5 8E-02	0,227146588	0,085996	0,242296	0,146088	2,13E-01
3	4,5 1E-02	0,331594236	2,3 1E-01	2,7 4E-01	7,7 9E-02	4,03E-02
4	0,11361	1,7 1E-01	1,8 5E-01	2,09E-01	1,3 5E-01	0,186845
5	0,06857	2,6 3E-01	2,2 7E-02	0,243388	0,400499	1,61E-03
6	2,2 2E-01	0,178479317	0,125715	2,11E-01	0,134159	1,29E-01

Tabel 5 Probabilitas *State* Emisi

State	Hidden States					
Observasi	1	2	3	4	5	6
1	2,37627546991454e-88	4,90150634287460e-89	4,34813794582834e-51	1,73806482651127e-53	1,03909188882053e-113	0,350683342804920
2	7,65572596307492e-42	8,18325002599665e-157	1,77042171570092e-92	0,296306635849983	1,31224176214866e-48	3,93474080628769e-245
3	2,20150754668137e-59	2,74654607345107e-100	0,235743302472375	1,43861548589867e-126	2,88022394393422e-06	3,09538990406524e-46
4	6,05747132983383e-124	2,08946531270740e-67	3,75248115589803e-36	2,63050451955544e-12	0,372473863110867	1,40322709003652e-105
5	0,354186604541197	9,64039416476544e-23	4,57723846250166e-20	2,08600458002870e-72	1,52753319616937e-89	5,93820307063932e-109
6	7,07020577001503e-27	0,401419356364558	0,249606329605100	4,18849186720468e-12	3,02153781612886e-09	4,29978230051850e-128
7	4,42867346749692e-78	3,64216234000102e-71	4,34053467218211e-55	2,97496328753777e-66	1,11204409081490e-20	0,245478339963444
8	4,23434743333178e-06	0,318464332419233	0,466133064960843	5,52482537504039e-29	1,55998294532824e-33	1,47827271578553e-19
9	0,308779219287989	9,19041291446880e-32	0,0485173029616822	0,066657125343362	2,40858713177679e-38	7,15261111720347e-84
10	0,0780708707847486	1,18262862033468e-16	2,62510721597766e-69	0	1,61871514020060e-45	5,49810578116724e-132
11	1,07374237331175e-19	0,144545382695049	1,82061458567632e-28	0,182703298302050	0,0954173443975096	0,117442266882932
12	8,95785021652770e-61	7,02047384522813e-22	1,42020972857650e-31	3,20940241545834e-12	0,532105518730302	7,76614884518748e-75
13	0,164812829833121	0,0779093194043792	8,45174994418415e-21	0,196535562493086	3,90515840405227e-07	3,9552442515048e-143
14	0	0	5,95137405319813e-271	0	3,44469609560831e-18	0,0350683324804920
15	0	0	0	0	0	0
16	0,0941462412055117	0,0576616091167809	9,64883493813968e-28	0,197576427420187	1,12371330611138e-28	0,251327716068211

Tabel 4 dan 5 merupakan salah satu model HMM yang dibentuk melalui hasil *training*. Dimana probabilitas transisi dimodelkan berdasarkan Tabel 4 dan emisi pada Tabel 5.

3.4 Evaluasi

Hasil pengujian yang dilakukan secara langsung (*real time*) melalui mikrofon rata-rata tingkat akurasi 30%. Pengujian yang kedua dilakukan dengan cara mengambil *file* rekaman. Hasil pengujian yang dilakukan melalui *file* rekaman rata-rata tingkat akurasi 100%. Hasil pengujian suara yang diambil adalah nilai *log likelihood* yang tertinggi dari empat nilai *log* yang ada. Hasil pengujian secara *real time* dapat dilihat pada Tabel 6 dan hasil pengujian *from file* rekaman dapat dilihat pada Tabel 7.

Tabel 6 Hasil Pengujian Secara *Real Time*

No	Hasil Pengujian			
	Penutur 1	Penutur 2	Penutur 3	Penutur 4
1	×	×	×	×
2	✓	×	×	×
3	×	×	✓	✓
4	×	×	✓	×
5	×	×	✓	✓
6	×	×	×	×
7	✓	✓	×	×
8	✓	×	×	×
9	×	×	✓	✓
10	×	×	×	✓
Tingkat Akurasi	30%	10%	40%	40%

Tabel 7 Hasil Pengujian *from File* Rekaman

No	Hasil Pengujian			
	Penutur 1	Penutur 2	Penutur 3	Penutur 4
1	✓	✓	✓	✓
2	✓	✓	✓	✓
3	✓	✓	✓	✓
4	✓	✓	✓	✓
5	✓	✓	✓	✓
6	✓	✓	✓	✓
7	✓	✓	✓	✓
8	✓	✓	✓	✓
9	✓	✓	✓	✓
10	✓	✓	✓	✓
Tingkat Akurasi	100%	100%	100%	100%

4. KESIMPULAN

Dari hasil penelitian dan pembahasan yang dilakukan, maka dapat disimpulkan bahwa:

1. Metode *Hidden Markov Model* (HMM) dapat digunakan untuk pengenalan penutur.
2. Tingkat akurasi pengujian secara *real time* sebesar 30% dan pengujian dari *file* rekaman sebesar 100%.
3. Hasil yang diperoleh untuk pengujian dari *file* rekaman lebih tinggi dibanding pengujian secara *real time* karena algoritma K-NN kurang baik dalam mengklasifikasi ciri yang dihasilkan dari MFCC.

5. SARAN

Adapun saran-saran yang ingin disampaikan sebagai berikut :

1. Pengembangan lebih lanjut guna mendapatkan hasil yang lebih akurat, metode HMM dapat dikombinasikan dengan metode klasifikasi yang lain.
2. Dalam pengembangan lebih lanjut menggunakan metode *Hidden Markov Model* (HMM) yang digunakan untuk pengenalan suara dapat digunakan dalam aplikasi untuk keamanan dan absensi.

DAFTAR PUSTAKA

- [1] AbdallahS, Osman I and Mustafa M. Text-Independent Speaker Identification Using Hidden Markov Model. 2012. *World of Computer Science and Information Technology Jurnal*, Vol. 2, No. 6.
- [2] Wang, Jun, dkk. *Sequential Model Adaptation for Speaker Verification*. 2008. INTERSPEECH.
- [3] Vyawahare, S.S. Speaker Recognition. 2013. *International Journal of Engineering Research & Technology*, Vol. 2.
- [4] Holmes, J. And Holmes, W. *Speech Synthesis and Recognition*. 2001. Taylor and Francis, London.

- [5] Nilsson, M and Ejnaronson , *Speech Recognition using Hidden Markov Model*, Thesis Blekinge Institute of Technology, Sewen, 2002.
 - [6] Prasetyo, Eko. *Data Mining Konsep dan Aplikasi Menggunakan MATLAB*. 2012. Andi Offset. ISBN: 978-979-29-3282-9.
 - [7] Rabiner, L.R. *A Tutorial in Hidden Markov Models and Selected Applications in Speech Recognition*. *Proceedings of IEEE 1989*, Vol. 77, No. 2.
 - [8] Li, Xiaolin, dkk. *Training Hidden Markov Model With Multiple Observation A Combinatorial Method*. *Proceedings of IEEE 2000*, Vol. 22, No. 4.
 - [9] Irawan, F.A. *Buku Pintar Pemrograman MATLAB*. MediaKom.2012. ISBN: 978-979-877-273-3.
 - [10] Simarmata, Janner. *Rekayasa Perangkat Lunak*. Andi Offset. 2010. ISBN: 978-979-29-1347-7.
-