Sistem *Automatic Speech Recognition* Menggunakan Metode MFCC dan HMMs Untuk Deteksi Kesalahan Pengucapan Kata Bahasa Inggris

Automatic Speech Recognition System Using MFCC and HMMs Method for Detecting English Language Pronounciation Mistake

Rahmawati Sitti Azizah[1], Dade Nurjanah, Ir., M.T., Phd.[2], Florita Diana Sari, SS., M.Pd.[3]

Fakultas Informatika, Telkom University, Bandung

[1]rahmawatisittiazizah@yahoo.com [2]dadenurjanah@gmail.com [3]floritads@gmail.com

Abstrak - *Automatic Speech Recognition* (ASR) memiliki kemampuan yang dapat membuat komputer mengenali apa yang diucapkan oleh seseorang berdasarkan sinyal suara yang diucapkan oleh seseorang. Dengan kemampuan tersebut sistem ini dapat digunakan untuk mengenali jika seseorang salah dalam mengucapkan sebuah kata. Terutama pada masalah kesalahan pengucapan akibat tertukarnya satu kata dengan kata lain yang mirip.

Metode yang digunakan dalam tugas akhir ini adalah *Mel Frequency Cepstral Coefficient* (MFCC) untuk ekstraksi ciri yang akan mengubah deretan nilai amplitudo menjadi frame-frame yang kemudian akan diolah menggunakan mel-filterbank yang mengadaptasi cara kerja pendengaran manusia sehingga terbentuklah nilai-nilai koefisien yang menjadi fitur ciri. Hasil dari MFCC kemudian diolah menjadi codebook yang nantinya akan menjadi masukkan dalam *Hidden Markov Models* (HMM) untuk menghasilkan sebuah model yang merepresentasikan kata tersebut. Hasil dari ekstraksi ciri dari data tes kemudian dikuantisasi untuk menjadi data yang akan dikenali menggunakan model yang telah didapat.

Pengujian dilakukan dengan menggunakan 10 pasangan kata dengan tingkat kemiripan yang tinggi dan sering tertukar jika dilafalkan secara terpisah. Dari hasil pengujian didapat tingkat akurasi rata-rata setiap pasangan kata sebesar 78,89% pada model HMM 3 state dan 78,33% pada model HMM 5 state.

Kata Kunci : *Automatic Speech Recognition*, MFCC, HMM

1. Pendahuluan

Kemampuan bahasa Inggris seseorang dalam mengucapkan dan memberikan intonasi terhadap kata yang benar secara langsung mempengaruhi kemampuan komunikasi seseorang dalam sebuah percakapan [9]. Penny Ur (1996) menyebutkan kemampuan berbicara merupakan kemampuan yang penting dimiliki seseorang yang mempelajari bahasa Inggris[11]. Merupakan hal yang umum bagi pelajar non Inggris untuk menghadapi kesulitan pada saat proses pembelajaran pronunciation [10].

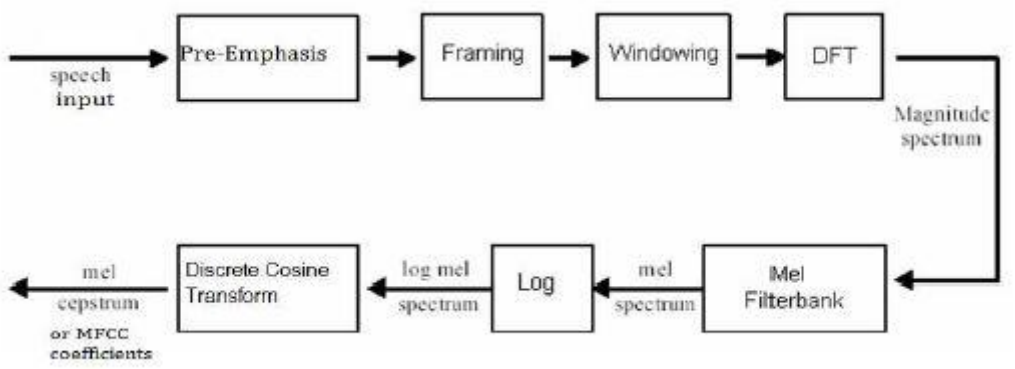
Terdapat enam faktor yang mempengaruhi pronunciation seseorang yaitu bahasa ibu yang digunakan, umur, jumlah phonetic yang mampu dipaparkan, kepribadian, dan motivasi [10]. Jika seseorang tidak mengetahui bagaimana cara pelafalan kata yang benar maka kemungkinan kesalahan pengucapan sangat tinggi. Selain itu sering tertukarnya sebuah kata dengan kata yang lain menjadi masalah yang cukup sering dialami oleh seseorang yang memepelajari bahasa Inggris. Untuk mengatasi hal tersebut diperlukan pemberian informasi dan juga pengenalan terhadap kata-kata tidak diketahui ataupun tertukar, dalam hal ini melalui sebuah sistem komputer yang dapat mengenali jika seseorang melakukan kesalahan pengucapan (pronunciation) dan memberikan contoh pengucapan yang benar kepada pengguna.

Pada permasalahan tertukarnya sebuah kata dengan kata lain yang mirip dibutuhkan sebuah sistem yang mampu membedakan antara dua buah dengan kemiripan yang tinggi. Metode yang digunakan untuk sistem *Automatic Speech Recognition* (ASR) ini adalah *Mel-Frequency Cepstral Coefficient* (MFCC) sebagai metoda ekstraksi ciri suara dan *Hidden Markov Model* (HMM) sebagai metode untuk pembuatan model. Keduanya dipilih karena MFCC merupakan salah satu metode ekstraksi ciri suara yang cukup baik dari segi akurasi dan pengurangan noise namun memiliki waktu proses yang cukup sedikit dibandingkan ekstraksi ciri yang lain []

2. Ekstraksi Ciri Menggunakan MFCC

Pada ASR ekstraksi ciri menjadi hal yang paling berpengaruh terhadap kehandalan dari sebuah sistem ASR. Karena dari ekstraksi cirilah kita dapat membuat model ciri dari sebuah kata. Karena model ciri menjadi representasi dari sebuah kata yang diucapkan maka tingkat akurasi dari sebuah metode ekstraksi ciri haruslah

baik sehingga ketika diolah pada proses selanjutnya tidak akan membuat tingkat kebenaran pengenalan suara menurun. MFCC merupakan salah satu metode ekstraksi ciri untuk sinyal akustik terbaik [5]. Metode MFCC sendiri menggunakan dua tipe filter yaitu filter linier dibawah 1000 Hz dan filter logaritmik diatas 1000 Hz. Nilai-nilai frekuensi tersebut yang dikenal sebagai frekuensi mel.



Gambar 1 Urutan Komputasi Pada Metode MFCC [7]

Seperti yang terlihat pada gambar 2.3 terdapat 6 tahapan komputasi pada MFCC. Dimulai dengan masukknya inputan data suara yang direpresentasikan kedalam deretan amplitudo-amplitudo hingga membentuk koefisien yang merepresentasikan ciri dari sebuah kata. Setiap tahapan memiliki fungsi sebagai berikut [5]:

a) Pre-Emphasis

Pada tahapan ini akan dilakukan filtering terhadap sinyal suara yang masuk dengan cara mengurangi nilai frekuensi sinyal tersebut sehingga nantinya hanya sinyal berfrekuensi tinggi saja yang dapat melewati filtering. Hal ini dilakukan untuk mengurangi noise dari sebuah suara sehingga hanya data sinyal suara yang sebenarnya saja yang dapat ditangkap oleh sistem.

b) Framing

Pada tahapan ini sampel suara akan dipotong menjadi frame-frame berdurasi lebih pendek (*framing*) sebanyak M yang disimpan kedalam matriks Y berukuran MxW dengan baris y_i menunjukan nomor frame. Pada sampel suara sebenarnya akan ditemui sinyal suara yang tidak stabil sehingga akan susah mencari karakteristik dari sampel suara. Dengan memotong sampel kedalam frame-frame kecil maka sinyal suara yang ada akan lebih stabil sehingga kita mendapatkan karakteristik suara yang lebih stabil. Frame yang dibuat akan overlapping dengan frame-frame lainnya untuk menghindari hilangnya informasi pada saat proses selanjutnya

c) Windowing

Setiap frame diberikan perkalian dengan fungsi *window* untuk meminimalisir diskontinuitas pada awal dan akhir frame yang diakibatkan oleh overlapping pada proses framing. *Hamming Window* digunakan untuk mengintegrasikan semua garis frekuensi terdekat, kelebihan dari metode ini adalah sidelobe yang sedang sehingga memiliki resiko terjadinya kebocoran spektral atau anti aliasing yang kecil namun memiliki noise yang tidak terlalu besar yang tidak akan mempengaruhi akurasi data yang digunakan [7]. Persamaan untuk menentukan *Hamming Window* adalah sebagai berikut :

$$w(k) = 0{,}54 - 0{,}46\cos\left(\frac{2\pi k}{K-1}\right) \quad (1)$$

N = Jumlah sampel, n = indeks window, K=jumlah *frame*

d) Fast Fourier Transform (FFT)

Sinyal suara yang ada masih dalam domain waktu sehingga perlu dilakukan konversi dari domain waktu ke domain frekuensi. Untuk mendapatkan sinyal dalam domain frekuensi dari sebuah sinyal diskrit, salah satu metode yang digunakan adalah *Fast Fourier Transform* (FFT). FFT dilakukan terhadap masing-masing *frame* dari sinyal yang telah di-windowing. FFT menggunakan algoritma *Discrete Fourier Transform* (DCT) versi cepat. Yang dioperasikan pada sinyal diskit yang terdiri dari N sampel.

$$f(n) = \sum_{K=0}^{N-1} y_k e^{-2\pi jkn/N} \quad (2)$$

N = Jumlah sampel, n= indeks sampel, y = sinyal hasil *windowing*

e) Mel-Frequency Wrapping

Mel-Filterbank sebenarnya merupakan sama dengan triangular filterbank biasa, hanya saja range frekuensi linier yang didapat dari FFT dikonversi kedalam skala *Mel-Frequency* untuk mendapatkan batas-batas filterbank berdasarkan skala *Mel-Frequency*. Skala *Mel* dapat diperoleh dengan persamaan :

$$B(f) = 1125 * \ln(1 + \frac{f}{700}) \quad (3)$$

f = frekuensi *linear*(Hz), B (f) = skala *Mel-frequency*

Untuk membuat Mel-Filterbank pertama perlu ditentukan batas atas dan bawah dari filter artinya nilai yang berada diluar batas itu tidak akan masuk kedalam filter kedua batas tersebut dikonversi kedalam skala mel, kemudian kita bagi range kedua batas tersebut sesuai dengan jumlah filter yang ingin dibuat dari sana akan diketahui batas atas dan bawah dari setiap filterbank dalam skala mel. Semua batas tersebut dikonversi kembali kedalam skala frekuensi linier, karena range yang ada tidak dapat merepresentasikan di bin FFT mana sajakah nilai-nilai tersebut dilakukan konversi dari nilai batas frekuensi linier kedalam nilai bin fft terdekat. Setelah itu dibuatlah *filter triangular* berdasarkan batas-batas tersebut. Hasil dari FFT pada tahap sebelumnya kemudian dikalikan dengan Mel-Filterbank.

f) Discrete Cosine Transform

Untuk mendapatkan nilai koefisien dari hasil perkalian mel-filterbank yang masih berada pada domain frekuensi diperlukan pengkonversian kembali kedalam domain waktu karena kita akan mengacu kepada urutan waktu dalam menentukan ciri. Pada langkah ini, hasil log dari perkalian sebelumnya domain waktu menggunakan *Discrete Cosine Transform* (DCT). Hasilnya disebut sebagai *Mel-Frequency Cepstral Coefficient* (MFCC). MFCC bisa didapat dari pendekatan persamaan :

$$C_s(n;m) = \sum_{k=0}^{N-1} \alpha_k \cdot \log(\, fmel_k) \cos(\frac{\pi(2n+1)k}{2N}) \,, \quad (4)$$

N = jumlah sampel , n = 0,1,2,3…N-1, *fmel* = frekuensi mel

Dimana cep_s adalah hasil akumulasi dari kuadratik magnitude DFT yang dikalikan dengan *Mel-Filter Bank*. Setelah itu didapatlah MFCC. Pada sistem pengenalan suara, biasanya hanya 13 cepstrum koefisien pertama yang digunakan.

g) *Delta feature*

Secara umum metode yang digunakan untuk mendapatkan informasi dari ciri yang dinamis biasa disebut dengan *delta-features*. Turunan waktu dari ciri dapat dihitung dengan rumus :

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \,, \quad (5)$$

Hasil dari perhitungan *delta* akan ditambahkan ke vektor ciri, sehingga menghasilkan vektor ciri yang lebih besar untuk menambah akurasi dari sistem ASR, metode ini akan menghasilkan koefisien delta sebanyak koefisien cepstral yang dihasilkan oleh MFCC

3. *Hidden Markov Model* (HMM)

Hidden Markov Model (HMM) merupakan metode yang menggunakan pendekatan statistik terhadap penyelesaian sebuah permasalah. HMM menggunakan prinsip markov chain untuk menyelesaikan permasalahan yang kita tidak tahu dengan pasti kondisi apa saja yang bisa terjadi. HMM ini dapat memodelkan persoalan – persoalan di dunia nyata yang sifatnya probabilistik.

Terdapat lima elemen dasar yang terdapat pada *hidden markov model*, yaitu [9]:

1. Himpunan *hidden state* : $Q = \{ q_1, q_2, q_3, \ldots, q_n \}$; n =jumlah *hidden state.*
2. Himpunan *observed state* : $V = \{ v_1, v_2, \ldots, v_m \}$; m = jumlah *observed state*.
3. Probabilitas transisi antar *state* : $A = \{a_{ij}\}$, Dimana $a_{ij} = P_t(q_{t+1} = S_j \mid q_t = S_i),\ 1 \leq (i,j) \leq N;$
4. Probabilitas emisi suatu simbol. $B = \{bj(k)\}$, dimana $b_j(k) = P_t(v_k = o_t \mid q_t = S_j)$, $1 \leq k \leq M, 1 \leq j \leq N$.
5. Distribusi Peluang *Initial State* $\pi = \{\pi_i\}$, dimana $\pi_i = P[q_1 = S_i], 1 \leq j \leq N$.

Sebuah HMM dapat direpresentasikan dengan notasi $\lambda = (A, B, \pi)$ dimana A, B dan π berturut – turut menyatakan distribusi peluang transisi antar *state*, distribusi peluang emisi *symbol* observasi, dan distribusi peluang *initial state*.

a) Evaluasi

Evaluasi sendiri merupakan komponen yang dapat menghitung seberapa baik sebuah model dapat menghitung Misalkan terdapat suatu variabel probabilitas *forward* pada waktu ke-*t* dan *state* ke-*i* yang dinotasikan dengan $\alpha_t(i)$ dimana secara matematis :

$$\alpha_t(i) = P(\, o_1 o_2 \ldots o_t,\ q_t = i \mid \lambda) \tag{6}$$

$\alpha_t(i)$= probabilitas *forward*, P = probabilitas observasi, o = observasi

Dalam algoritma ini terdapat 3 tahapan yang dilakukan yaitu inisialisasi, induksi, dan terminasi.

1. Inisialisasi, sebelum dilakukan penghitungan maka perlu ditentukan terlebih dahulu nilai α awal dengan persamaan:

$$\alpha_1(i) = \pi_i b_i(01)\ ,\ 1 \leq i \leq N \tag{7}$$

2. Induksi, setelah diinisialisasi kemudian hitung nilai α untuk setiap kemungkinan state selanjutnya dengan menggunakan persamaan:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\alpha_{ij}\right] b_j(o_{t+1}), 1 \leq t \leq T-1, 1 \leq j \leq N \tag{8}$$

3. Terminasi

Setelah diketahui semua nilai α maka kita dapat menghitung $P(O|\lambda)$ menggunakan rumus :

$$(O|\lambda) \ = \sum_{i=1}^{N} \alpha_t(i) \tag{9}$$

$\alpha_t(i)$ = probabilitas *forward* ke-i,

b) Decoding

Algoritma *viterbi* merupakan metode yang sering digunakan untuk solusi masalah *decoding* yaitu untuk mencari deretan *state* yang terbaik. Algoritma ini hampir mirip dengan algoritma *forward* hanya saja disini hasil yang diperoleh tidak dijumlahkan namun di cari nilai terbesarnya. Untuk mencari deretan *state* terbaik $Q = \{q_1, q_2, \ldots, q_t\}$ dari barisan observasi $O = \{o_1, o_2, \ldots, o_t\}$, kita perlu mencari kuantitas dari [9] :

$$\delta_t(i) = \max_{q_1 q_2 \ldots q_{t-1}} P[q_1 q_2 \ldots q_{t=i}, o_1\, o_2 \ldots o_t \mid \lambda] \tag{10}$$

$\delta_t(i)$ = probabilitas yang terbaik, q = *state*

c) Learning

Hal berikutnya yang menjadi perhatian adalah masalah penentuan parameter-parameter (A, B, π) yang dapat menghasilkan model yang paling optimal berdasarkan kriteria optimal tertentu. Metode yang biasa digunakan untuk memecahkan masalah ketiga ini adalah algoritma *Baum-Welch* atau sering disebut sebagai algoritma *forward – backward*. Algoritma ini adalah metode *iterative* yang berfungsi untuk mencari nilai – nilai maksimum lokal dari fungsi probabilitas $P(O|\lambda)$. Proses *training* ini berlangsung terus sampai kondisi minimum local terpenuhi. Model hasil *training* harus lebih baik daripada model sebelumya.

Formula *Baum-Welch re*-estimasi untuk mean dan kovarian pada masing masing *state* HMM adalah:

$$\hat{\mu} = \frac{\sum_{t=1}^{T} L_j(t) ot}{\sum_{t=1}^{T} L_j(t)} \qquad (2\text{-}12)$$

$$\hat{\Sigma_j} = \frac{\sum_{t=1}^{T} L_j(t)(o_t - \mu_j)(o_t - \mu_j)'}{\sum_{t=1}^{T} L_j(t)} \qquad (2\text{-}13)$$

L_j = probabiliti *state* j, $\hat{\mu}$ = *mean*, $\hat{\Sigma_j}$=nilai kovarian

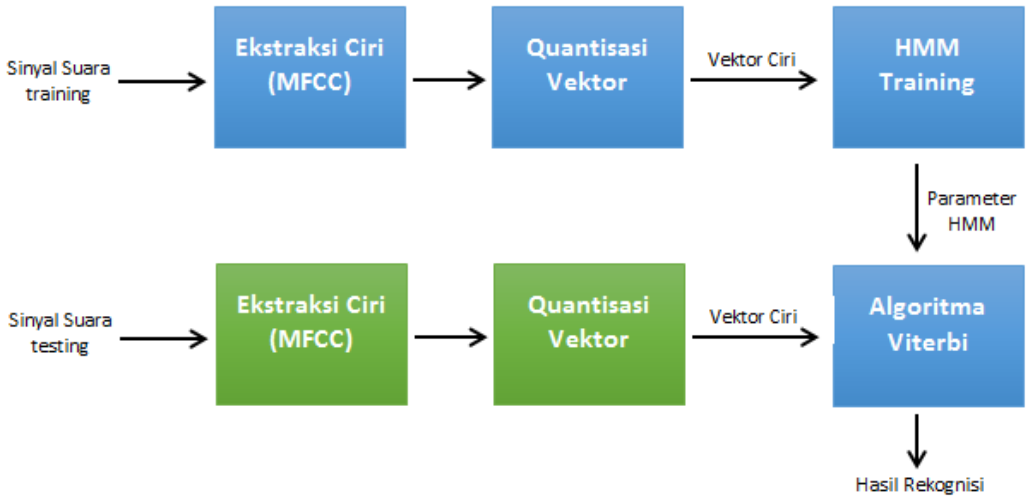
Parameter vektor akan diestimasi dengan menggunakan algoritma *forward-backward* hingga diperoleh nilai probabilitas P(O|M) terbesar berdasarkan observasi pada masing masing *state*.

Estimasi dilakukan terhadap parameter vektor pada inisial HMM dengan menggunakan metode forward/backward hingga diperoleh parameter vektor yang konvergen (tidak dapat diestimasi lagi). Kriteria update adalah nilai probabiltias observasi terhadap model P(O|M) lebih tinggi dari nilai iterasi sebelumnya.

Pada tahap awalnya HMM melakukan pembelajaran untuk memodelkan beberapa contoh kata. Hasil dari pembelajaran adalah model yang telah dilakukan estimasi(M). Kemudian HMM digunakan untuk mengenali kata/observasi (O) berdasarkan hasil pembelajaran tersebut. P(O|M) adalah kemungkinan rangkaian observasi O terhadap model M.

4. Perancangan Sistem

Sistem yang dibuat merupakan sistem yang dapat membedakan pasangan kata bahasa Inggris yang memiliki tingkat kemiripan yang tinggi dan sering tertukar dalam pengucapannya. Sistem ini memiliki dua bagian besar yaitu bagian *training* untuk membuat model yang akan digunakan dalam pengenalan kata dan *testing* untuk melakukan skenarion pengujian, untuk lebih jelasnya lihat gambar.



Gambar 2 Blok Proses Sistem

Pada proses *training* sinyal suara yang disimpan dalam file berekstensi .wav dibuat fitur cirinya dengan menggunakan MFCC kemudian fitur ciri dikuantisasi sehingga ukurannya menjadi sama, kemudian fitur ciri tersebut digunakan sebagai input untuk membuat model HMM, yang hasil parameternya digunakan sebagai inputan pada proses algoritma viterbi untuk pengenalan

Sedangkan pada proses *testing*, ciri yang sudah didapatkan dari proses *testing* yang telah dikuantisasi sehingga ukuran tiap fitur ciri menjadi sama, dibandingkan dengan model HMM yang telah dibuat pada proses *training* menggunakan algoritma viterbi. Jika terdapat kata yang cocok maka system akan mengeluarkan kata tersebut sebagai hasil dari proses pengenalan

5. Skenario Pengujian dan Hasil

Kata yang dipilih dalam tugas akhir ini merupakan pasangan kata yang sering tertukar dalam pelafalannya oleh mahasiswa telkom. Data ini berasal dari observasi yang dilakukan oleh Ibu Florita Diana Sari selaku dosen bahasa Inggris Universitas Telkom. Berikut 10 pasangan kata yang akan digunakan dalam tugas akhir ini :

Tabel 1 Daftar Pasangan Kata

No	Pasangan Kata
1	*Seventeen, Seventy*
2	*Six, Sick*
3	*Feature, Future*
4	*Produce, Product*
5	*Three, Tree*
6	*Many, Money*
7	*Axe, Ask*
8	*Reply, Replay*
9	*Curtain, Certain*
10	*Access, Assess*

Dataset terdiri dari 10 pasangan kata yang merupakan hasil *generate* dari aplikasi *text to speech* berbasis web yang tersedia di *website* http://www.naturalreaders.com. *File wav* yang didapatkan merupakan *file* yang bersih dari *noise* suara, sehingga akan menjadikan proses ekstraksi lebih baik. Terdapat sebelas penutur dengan dengan struktur 5 penutur pria dan 6 pembicara wanita yang semuanya merupakan suara natural manusia dengan aksen America (bukan robot). Maka terdapat sebelas *file* suara dalam satu kata.

a) Pengujian Pengaruh Parameter panjang Codebook

Data suara menggunakan data sintetis telah dibagi menjadi dua bagian, 16 sebagai data training dan 6 sebagai data testing pada setiap pasangan kata. Pemilihan panjang codebook sebenarnya tidak ada aturan ukuran tertentu, namun dalam tugas akhir ini ukuran codebook yang digunakan adalah 64,128, dan 256 dengan nilai default state hmm= 5. Berikut hasil pengujian sistem dengan variasi panjang codebook :

Tabel 2 Hasil pengujian menggunakan variasi panjang *codebook*

Pasangan kata	Codebook =64		Codebook =128		Codebook =256	
	FRR	CRR	FRR	CRR	FRR	CRR
Seventeen, Seventy	0%	100%	17%	83%	0%	100%
Six, Sick	33%	67%	33%	67%	33%	67%
Feature, Future	50%	50%	50%	50%	67%	33%
Produce, Product	0%	100%	0%	100%	17%	83%
Three, Tree	17%	83%	17%	83%	0%	100%
Many, Money	17%	83%	17%	83%	50%	50%
Axe, Ask	0%	100%	33%	67%	33%	67%
Reply, Replay	0%	100%	0%	100%	0%	100%
Curtain, Curtain	17%	83%	17%	83%	33%	67%
Access, Assess	50%	50%	33%	67%	17%	83%
Rata-Rata	18,33%	81,67%	21,67%	78,33%	25%	75%

Dari tabel 2 jika dilihat dari nilai rata-rata akurasi dari pasangan kata berbanding terbalik dengan peningkatan panjang codebook. Penurunan ini terjadi secara konstan dengan nilai penurunan ±3%. Hal ini sangat mungkin terjadi karena jumlah frame dari dataset yang ada berada dibawah 100 sehingga jika dibuat codebook lebih besar dari jumlah framenya maka kemungkinan besar akan ada nilai-nilai yang tidak diperlukan yang akhirnya menurunkan tingkat akurasi.

Sedangkan jika kita mengamati berdasarkan akurasi setiap pasangan maka akan terlihat reaksi yang berbeda tiap pasangan kata, untuk pasangan kata (seventeen, seventy), (produce, product), (three, tree), (reply, replay) cenderung tidak terlalu terpengaruh oleh perubahan codebook dan memiliki tingkat akurasi diatas 70% begitu pula dengan pasangan kata (six, sick), (feature, future) yang tingkat akurasinya cenderung lemah dikisaran 33-67%. Sedangkan untuk kasus (access, assess) perubahan nilai codebook mampu meningkatkan nilai akurasi dari pasangan kata tersebut, pasangan (many, money), (axe, ask), (curtain, certain) menunjukan pada batas tertentu semakin besar nilai codebook malah menurunkan nilai akurasi.

b) Pengujian Pengaruh Jumlah State HMM

Data suara menggunakan data sintetis telah dibagi menjadi dua bagian, 16 sebagai data training dan 6 sebagai data testing pada setiap pasangan kata. Pemilihan jumlah state sebenarnya tidak ada aturan, dalam

tugas akhir ini jumlah state yang akan digunakan adalah 3, dengan nilai default codebook = 64. Berikut hasil pengujian sistem dengan variasi jumlah state HMM :

Tabel 3 Hasil pengujian menggunakan variasi jumlah state HMM

Pasangan kata	Jumlah state =3		Jumlah state =5		Jumlah state =9	
	FRR	CRR	FRR	**CRR**	**FRR**	**CRR**
Seventeen, Seventy	0%	100%	17%	83%	0%	100%
Six, Sick	33%	67%	33%	67%	33%	67%
Feature, Future	50%	50%	50%	50%	67%	33%
Produce, Product	0%	100%	0%	100%	17%	83%
Three, Tree	17%	83%	17%	83%	0%	100%
Many, Money	50%	50%	17%	83%	50%	50%
Axe, Ask	50%	50%	33%	67%	33%	67%
Reply, Replay	0%	100%	0%	100%	0%	100%
Curtain, Curtain	0%	100%	17%	83%	33%	67%
Access, Assess	17%	83%	33%	67%	17%	83%
Rata-Rata	21,67%	78,33%	21,67%	78,33%	25%	75%

Dari tabel 3 jika dilihat dari nilai rata-rata akurasi dari pasangan kata tidak terlalu terpengaruh oleh jumlah state. Tingkat akurasi tertinggi diperoleh pada saat jumlah state=5. Hal ini terjadi karena bervariasinya jumlah suku kata pada semua pasangan kata. Sehingga menyebabkan jumlah state yang digunakan haruslah tidak terlalu kecil agar kata yang memiliki suku kata yang panjang dapat terwakili dan juga tidak terlalu besar agar kata-kata yang pendek tidak kelebihan informasi yang tidak dibutuhkan.

Sedangkan jika kita mengamati berdasarkan akurasi setiap pasangan maka terlihat lebih dari setengah dari pasangan kata tidak terpengaruh dengan perubaha jumlah state. Namun, pada beberapa pasangan kata seperti (axe, ask), (access, assess) perubahan state memberikan pengaruh yang cukup besar dan memiliki reaksi yang bertolak belakang, jika pada pasangan (axe, ask) akurasi tertinggi diperoleh pada saat jumlah state-5 maka pada (access, assess) justru akurasi terkecil yang didapat, sedangkan pada saat state-3 (access, assess) mendapatkan nilai tertinggi dan (axe, ask) mendapat nilai terburuk

6. Kesimpulan dan Saran

Berdasarkan hasil analisis terhadap pengujian yang dilakukan pada sistem, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem *Automatic Speech Recognition* berhasil diimplementasi dengan menggunakan algoritma MFCC sebagai ekstraksi ciri dengan digabungkan dengan *Hidden Markov Model* (HMM) sebagai algoritma pencocokan.
2. Sistem menghasilkan nilai akurasi rata-rata sebesar 78,9% dengan akurasi terbaik diperoleh sebesar 81,67% pada saat jumlah state HMM 5 dan panjang *codebook* 64.
3. Panjang codebook, jumlah *state* HMM memberikan pengaruh terhadap akurasi system secara keseluruhan.
4. Pada pengujian terhadap pasangan kata diketahui bahwa sistem ASR dengan menggunakan metode MFCC dan HMM mampu membedakan pasangan kata yang memiliki tingkat kemiripan dan kecenderungan untuk tertukar pelafalannya cukup tinggi.

Pengembangan lebih lanjut yang dapat dilakukan terhadap tugas akhir ini adalah sebagai berikut :

1. Pengaruh karakteristik kata terhadap parameter yang dibutuhkan dapat diteliti lebih jauh sehingga akan berguna untuk mengembangkan sistem yang dapat mengenali kata lebih banyak lagi.
2. Penggunaan threshold untuk menghindari ketika kata yang diucapkan sama sekali berbeda dari model hanya dengan acuan pasangan kata ataupun beberapa kata saja.

Daftar Pustaka

[1] F. Zhang, "A Study of Pronunciation Problems of English Learners in China," *Asian Social Science Journal*, vol. 5, no. 6.

[2] P. Ur, A Course in Language Teaching, Cambridge: Cambridge University Press, 1996.

[3] S. Sharma, A. Shukla dan P. Mishra, "Speech and Language Recognition using MFCC and DELTA-MFCC," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 12, no. 9, 2014.

[4] E. Riyanto dan S. , "Perbandingan Metode Ekstrasi Ciri Suara Mffcc, Zcpa, Dan Lpc," *Teknik Informatika STMIK HIMSYA*, 2014.

[5] J. Kenworthy, Teaching English Pronunciation, London: Longman, 1987.

[6] G. Kelly, How to Teach Pronunciation, Oxfordshire: Bluestone Press, 2000.

[7] M. Gales dan S. Young, "The Application of Hidden Markov Models in Speech Recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, 2007.

[8] H. Combrinck dan E. Botha, "On The Mel-Scaled Cepstrum," *Department of Electrical and Electronic Engineering, University of Pretoria*, 1996.

[9] R. S. Chavan dan G. S. Sable, "An Overview of Speech Recognition Using HMM," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 6, 2013.

[10] R. S. Chavan dan G. S. Sable, "An Implementation of Text Dependent Speaker Independent Isolated Word Speech Recognition Using HMM," *International Journal Of Engineering Sciences & Research Technology*, vol. 2, no. 9, 2013.

[11] P. Blunsom, *Hidden Markov Model*, Lecture Notes, 2004.

[12] M. Anusuya dan S. Katti, "Speech Recognition by Machine: A Review," *International Journal of Computer Science and Information Security*, vol. 6, no. 3, 2009.

[13] R. dan E. Haryanto, "Improving Students' Pronunciation through Communicative Drilling Technique at Senior High School (SMA) 07 South Bengkulu, Indonesia," *International Journal of Humanities and Social Science*, vol. 2, no. 21, 2012.