BAB II
TINJAUAN PUSTAKA

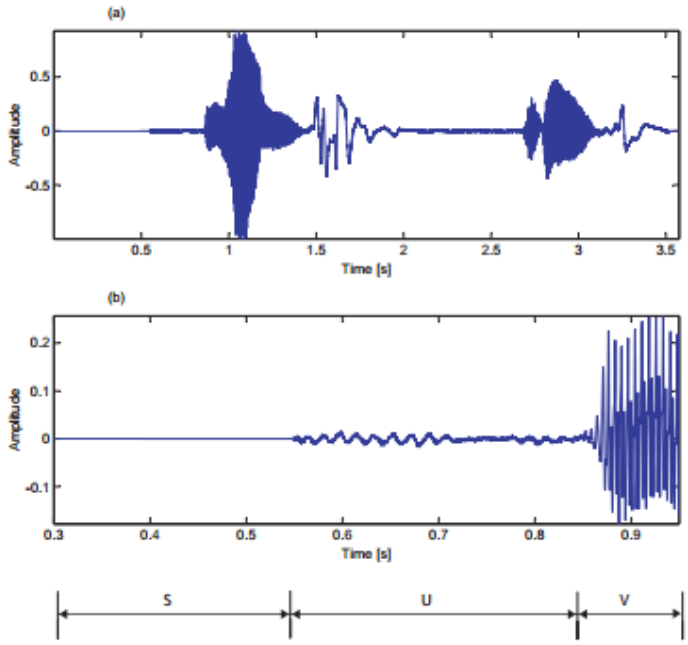
2.1 Suara (Wicara)

Suara khususnya wicara merupakan cara yang natural bahkan paling penting dalam melakukan proses komunikasi. Dalam kehidupan sehari-hari, manusia melakukan berbagai jenis komunikasi dengan sesama manusia, misalnya: body language, berbicara (*speech*) dan lain-lain. Diantara banyak komunikasi yang dilakukan oleh manusia, berbicara (speech) memberikan paling banyak informasi penting dan paling efektif dalam berkomunikasi.

2.1.1 Karakteristik Sinyal Suara

Salah satu parameter yang penting pada sebuah suara adalah frekuensinya (Mandalia & Gareta, 2011). Frekuensi yang dimiliki sebuah suara dapat menjadikan suara tersebut berbeda dengan suara lainnya. Manusia umumnya dapat memproduksi suara dengan frekuensi 70Hz hingga 10kHz. Sedangkan, sistem pendengaran manusia mampu menangkap suara yang dalam rentang 16Hz hingga 20kHz (Mandalia & Gareta, 2011).

Sinyal wicara merupakan sinyal yang bervariasi lambat sebagai fungsi waktu, dalam hal ini ketika diamati pada durasi yang sangat pendek (5 sampai 100 m) karakteristiknya masih stasioner. Tetapi bilamana diamati dalam durasi yang lebih panjang (> 1/5 detik) karakteristik sinyalnya berubah untuk merefleksikan wicara yang keluar dari pembicara. Gambar 1 menunjukkan tiga kondisi dasar sinyal wicara pada manusia.

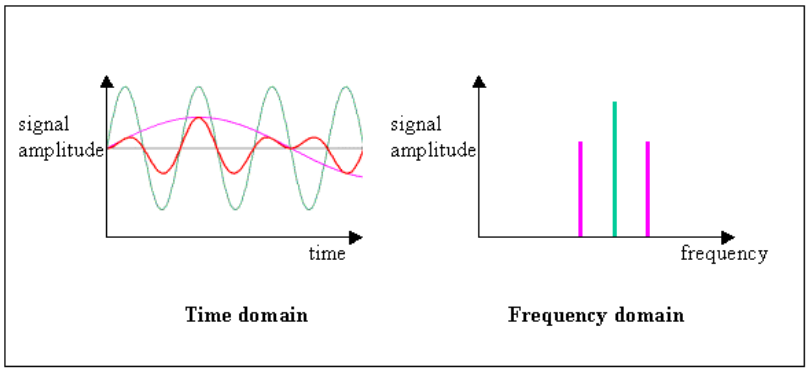


Gambar 2.1 Tiga Representasi Sinyal Suara
Sumber : Speech Recognition using Hidden Markov Model : *performance evaluation in noisy environtment*

Salah satu cara dalam menyajikan sebuah sinyal wicara adalah dengan menampilkannya dalam tiga kondisi dasar, yaitu *silence* (S) atau keadaan tenang dimana sinyal wicara tidak diproduksi. *Unvoice* (U) dimana vocal cord tidak berfibrasi, dan yang ketiga adalah *voiced* (V) dimana *vocal cord* bervibrasi secara periodik sehingga dapat menggerakkan udara ke kerongkongan melalui mekanisme akustik sampai keluar mulut dan menghasilkan sinyal wicara (Nillson, 2002).

2.1.2 Representasi Sinyal Suara

Sinyal suara/wicara dan karakteristiknya dapat direpresentasikan ke dalam dua domain yang berbeda, yaitu domain waktu dan domain frekuensi. Sebuah sinyal suara dapat diubah ke dalam domain waktu atau frekuensi untuk merubah perspektif dalam menyelesaikan masalah pengolahan sinyal.



Gambar 2.2 Sinyal Domain Waktu dan Domain Frekuensi
Sumber : http://www.ni.com/tutorial/13042/en/

A. Domain Waktu

Sinyal dalam domain waktu merepresentasikan besanya amplitudo yang ada pada satuan waktu saat *sampling*. Domain waktu ini merupakan bentuk umum sinyal yang sering dilihat yang biasa disebut bentuk gelombang (*waveform*)

B. Domain Frekuensi

Representasi ini sering digunakan dalam berbagai proses pengolahan sinyal digita dibanding domain waktu karena memiliki banyak informasi penting di dalamnya. Domain frekuensi ini merepresentasikan besarnya amplitudo terhadap frekuensi-frekuensi yang terdapat dalam sebuah sinyal.

Domain frekuensi pada sebuah sinyal didapatkan dengan menggunakan transformasi *Fourier* dari domain frekuensi

2.1.3 Energi Sinyal Suara

Untuk pengkuran nilai energi pada sinyal wicara kita harus melibatkan fungsi window. Hal ini karena dalam pengukuran energi sinyal wicara kita harus menyusunnya dalam frame-frame tertentu. Ini merupakan standar dalam teknologi speech processing, sebab secara umum dalam pengolahan sinyal wicara kita terlibat dengan sinyal dengan durasi yang terlalu panjang bila dihitung dalam total waktu pengukuran. Fenomena ini juga dikenal sebagai *short term speech signal energy*.

Untuk menghitung energi sinyal wicara kita gunakan formulasi dasar seperti berikut:

$E = \sum_{t=0}^{T}(V(t)w(t))^2$ ....(2.1)

Dimana $w(t)$ = merupakan fungsi window seperti hamming, hanning, bartlett, dan boxcar dan $V(t)$ =sinyal suara;

2.1.4 Filter pada Sinyal Suara

Filtering merupakan salah teknik yang digunakan dalam pengolahan sinyal digital yang berguna untuk memilah frekuensi-frekuensi suara sesuai dengan rentang yang ada. Frekuensi sinyal suara yang terdapat di luar batas yang ditentukan akan diabaikan/dihapus (Lacanette, 1991).

A. Filter Low-pass

Low Pass Filter (LPF) adalah filter yang hanya melewatkan sinyal dengan frekuensi yang lebih rendah dari frekuensi *cut-off* (f_c) dan akan melemahkan sinyal dengan frekuensi yang lebih tinggi dari frekuensi *cut-off* (f_c). Pada filter LPF yang ideal sinyal dengan frekuensi diatas frekuensi *cut-off* (f_c) tidak akan dilewatkan sama sekali.

B. Filter High-pass

Filter high-pass adalah suatu rangkaian yang akan melewatkan suatu sinyal yang berada diatas frekuensi *cut-off* (ω_c) sampai frekuensi *cut-off* (ω_c) filter tersebut dan akan menahan sinyal yang berfrekuensi dibawah frekuensi *cut-off* (ω_c) filter tersebut.

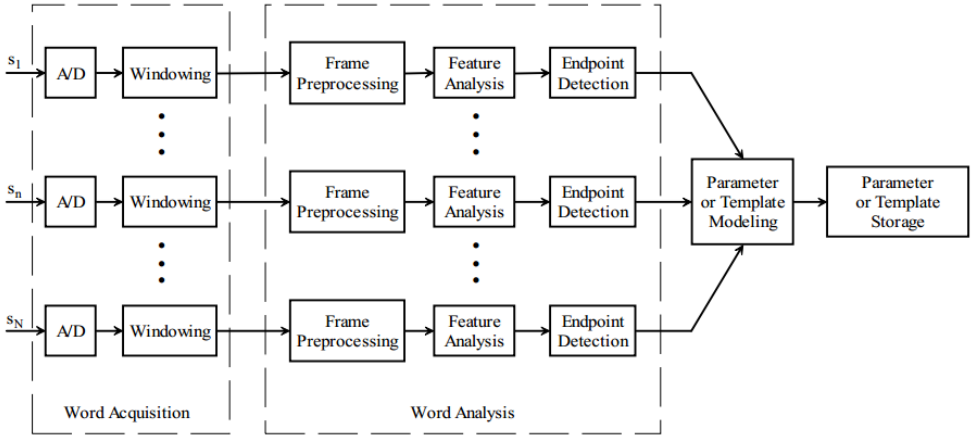
2.2 Automatic Speech Recognition (ASR)

Sistem pengenalan suara ini sudah mulai dikembangkan sekitar 30 tahun yang lalu dan hingga kini masih menjadi objek penelitian yang terus dikembangkan. Mengingat manfaat ASR yang dapat digunakan dalam berbagai hal penting seperti pendukung keamanan, personal asisten, dan bidang-bidang yang lainnya. salah satu implementasi dari ASR yang sering kita lihat dan gunakan adalah aplikasi *Text to Speech* yang terdapat dalam perangkat *smartphone* saat ini.

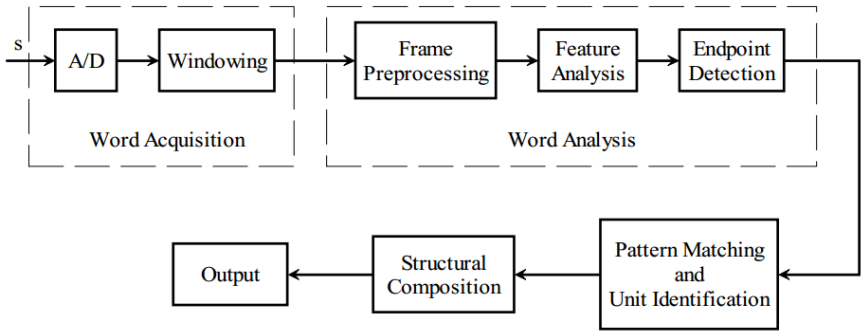
2.2.1 Karakteristik Sistem Pengenalan Suara

Secara umum sistem pengenalan suara memiliki 2 tahap utama yaitu tahap pelatihan(*training*) dan tahap pengenalan(*testing*) (Srichai, 1998). Pada kedua tahap ini dilakukan ekstraksi fitur sinyal suara untuk mendapatkan informasi penting yang ada dalam sinyal tersebut.

Pada tahap pelatihan, fitur vektor dari kata yang sama digunakan untuk membangun sebuah model atau template. Kemudian model beserta parameter tersebut digunakan pada saat pengenalan sebuah kata atau kalimat untuk mendapatkan keputusan kata yang dikenali.



(a)



(b)

Gambar 2.3 Diagram Sistem Pengenalan Suara. Tahap training(a), tahap pengenalan(b)

2.2.2 Klasifikasi Sistem Pengenalan Suara

Berdasarkan review terhadap sejumlah penelitian terkait ASR, Prakash et al. (2013) menyatakan bahwa sistem pengenalan wicara dapat diklasifikasikan menjadi beberapa kelompok, diantaranya berdasarkan jenis kata pada wicara, berdasarkan pembicara, dan besar kosa kata yang digunakan dalam pengenalan tersebut. Pengenalan suara menjadi lebih kompleks karena adanya variasi sinyal suara.

A. Jenis Wicara

Wicara sendiri merupakan kata atau rangkaian kata yang memiliki sebuah arti. Oleh karena itu, wicara dapat terdiri atas sebuah kata, beberapa kata, kalimat bahkan beberapa kalimat sekaligus. Macam-macam wicara antara lain :

a. Kata terisolasi

Pengenalan wicara jenis ini membutuhkan jeda yang cukup panjang antara wicara satu dengan wicara lainnya. Dalam hal ini, sistem tidak dapat mengenali lebih dari 1 kata sekaligus karena membutuhkan pengucapan yang jelas dan jeda waktu antar kata yang cukup panjang.

b. Kata - kata terhubung

Pada sistem kata – kata yang terhubung ini sejenis dengan jenis sebelumnya hanya saja memungkinkan bagi wicara lain dideteksi secara bersamaan dengan jeda waktu yang minimum.

c. Wicara Kontinyu

Wicara yang diucapkan hampir mendekati wicara secara natural. Jenis ini sangat sulit dibuat karena harus menggunakan metode-metode tertetu untuk menentukan batas-batas dalam sebuah wicara. Dengan bertambahnya kosa kata yang digunakan, tingkat kesulitan untuk membedakan rangkaian kata yang tepat juga semakin bertambah.

d. Wicara spontan

Tidak ada pelatihan yang dilakukan dalam jenis ini. Wicara yang dikenali merupakan wicara manusia yang natural sebagaimana manusia berkomunikasi dengan manusia lainnya. Pada saat berbicara secara spontan, mungkin saja terdapat kesalahan dalam pengucapan atau pengucapan yang bahkan tidak termasuk sebagai kata.

B. Pembicara

Setiap orang memiliki suara yang unik yang ditentukan baik berdasarkan kondisi fisik atau mentalnya. Secara umum, pada sistem pengenalan suara, pembicara dikelompokkan menjadi 2 jenis yaitu :

a. Terikat

Sistem dengan pembicara yang terikan dirancang untuk pembicara tertentu saja sehingga akurasinya lebih baik disbanding pembicara lain yang tidak termasuk dalam kelompok tersebut. Biasanya sistem jenis ini mudah untuk dikembangkan dan lebih murah. Tetapi kurang mampu beradaptasi pada pembicara lainnya.

b. Tidak Terikat

Sistem ini dirancang untuk mengenali segala jenis pembicara sehingga lebih sulit dikembangkan, mahal, dan menghasilkan tingkat akurasi yang lebih rendah dari jenis terikat. Namun jenis ini lebih fleksibel

C. Tipe Kosa Kata

Ukuran dari kosa kata pada sebuah sistem pengenalan wicara berpengaruh pada kompleksitas sistem, kebutuhan pemrosesan, dan akurasi yang didapatkan. Beberapa aplikasi membutuhkan hanya beberapa kata saja, sedangkan yang lainnya membutuhkan ukuran kosa kata yang besar. Oleh karena itu, ukuran kosa kata dibagi menjadi :

a. Kecil (maksimal 10 kata)

b. Sedang (puluhan hingga ratusan kata)

c. Besar (ribuan kata)
d. Sangat Besar (puluhan ribu kata)
e. Diluar Kosa Kata (memetakan kata pada kosakata menjadi kata yang belum diketahui)

2.3 Mel Frequency Cepstrum Coefficient(MFCC)

Ekstraksi fitur pada ASR (Automatic Speech Recognition) merupakan proses perhitungan urutan dari fitur vektor yang mampu merepresentasikan sinyal wicara yang ada secara optimal (Dave, 2013). Fitur yang biasa digunakan adalah *cepstral coefficient*. MFCC merupakan metode ekstraksi fitur yang menghitung koefisien cepstral yang didasarkan pada variasi dari frekuensi kritis pada telinga manusia. Filter dipetakan secara linear pada frekuensi rendah (< 1 kHz) dan logaritmik pada frekuensi tinggi (> 1kHz) untuk mendapatkan karakteristik suara yang penting (Vibha, 2009). Beberapa keunggulan dari metode ini adalah (Manunggal, 2005) :

1. Mampu menangkap karakteristik suara yang sangat penting bagi pengenalan suara
2. Menghasilkan data seminimal mungkin, tanpa menghilangkan informasi-informasi penting yang terkandung di dalamnya
3. Mereplikasi organ pendengaran manusia dalam melakukan persepsi terhadap sinyal suara

Adapun tahapan-tahapan dalam MFCC adalah sebagai berikut.

1) Pre-Emphasize Filtering

Proses *filtering* ini berfungsi untuk mempertahankan frekuensi-frekuensi tinggi pada sebuah *spectrum*, yang umumnya tereliminasi pada saat proses produksi suara (Putra & Resmawan, 2009).

Bentuk paling umum yang digunakan dalam *pre-emphasize filtering* adalah

$$y[n] = s[n] - \alpha s[n-1], 0.9 \leq \alpha \leq 1.0 \ldots\ldots\ldots\ldots \quad (2.2)$$

Dimana :

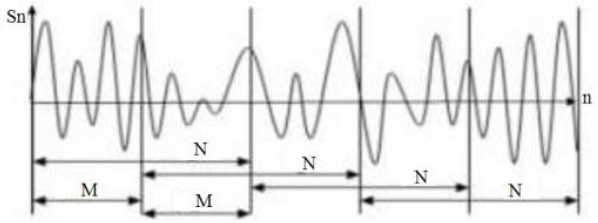
$y[n]$ = sinyal hasil *pre-emphasize filtering*

$s[n]$ = sinyal sebelum *pre-emphasize filtering*

2) Frame Blocking

Tahap ini sinyal suara analog dibagi menjadi beberapa *frame* yang terdiri dari N sampel, masing-masing frame dipisahkan oleh M, dengan M adalah banyaknya pergeseran antar *frame* (M<N). *Frame* pertama berisi sampel N pertama. *Frame* kedua dimulai M sampel setelah permulaan *frame* pertama, sehingga *frame* kedua ini *overlap* terhadap *frame* pertama sebanyak N-M sampel.

Selanjutnya, *frame* ketiga akan dimulai M sampel setelah *frame* kedua. Proses ini berlanjut sampai seluruh suara tercakup dalam *frame*. Hasil dari proses ini adalah matriks dengan N baris dan beberapa kolom sinyal X[N]. Proses ini ditunjukkan pada dibawah, Sn adalah nilai sampel yang dihasilkan dan n adalah urutan sampel yang akan diproses.



Gambar 2.4 Proses *FrameBlocking*
Sumber : Aria (2013)

3) Windowing

Proses *framing* dapat menyebabkan terjadinya kebocoran spektral yaitu sinyal yang baru memiliki frekuensi yang berbeda dengan sinyal aslinya. Efek ini dapat terjadi karena rendahnya jumlah *sampling rate* ataupun karena proses *frame blocking* dimana menyebabkan sinyal menjadi tidak kontinyu. Untuk mengurangi kemungkinan terjadinya kebocoran spektral ini maka hasil dari proses *framing* harus melewati proses *windowing*.

Konsep *windowing* adalah meruncingkan sinyal ke angka nol pada permulaan dan akhir setiap *frame*. Proses ini dilakukan dengan mengalikan antar *frame* dengan jenis *window* yang digunakan. Proses *windowing* ini dapat dituliskan dalam persamaan berikut :

$$y(n) = x(n)w(n), 0 \leq n \leq N-1 \quad \text{..... (2.3)}$$

Dimana

$y(n)$ = sinyal hasil *windowing* sampel ke- n

$x(n)$ = nilai sampel ke- n

$w(n)$ = nilai *window* ke- n

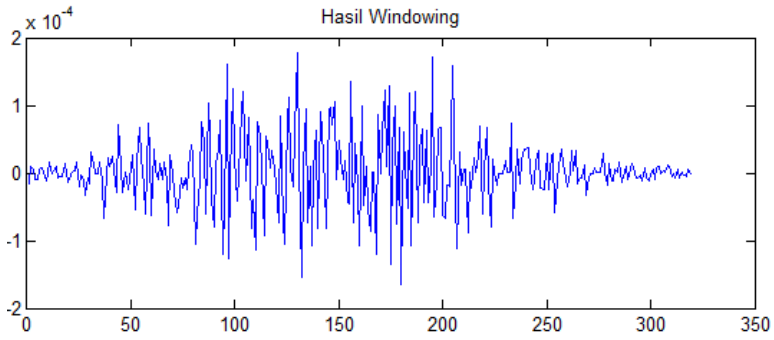
N = jumlah sampel dalam *frame*

Penelitian suara banyak menggunakan *Window Hamming* karena kesederhanaan formulanya dan nilai kerja *window*. Dengan pertimbangan tersebut, maka penggunaan *Window Hamming* cukup beralasan. Persamaan *Window Hamming* adalah :

$$w(n) = 0.54 - 0.46\ cos\frac{2\pi n}{N-1} \text{... ..... (2.4)}$$

Dimana

n = 0,1,…,*N*-1



Gambar 2.5 Contoh hasil *Windowing* sinyal suara

4) Fast Fourier Transform (FFT)

Tahapan selanjutnya ialah mengubah setiap *frame* yang terdiri dari N sampel dari domain waktu ke dalam domain frekuensi. Output dari proses ini disebut dengan nama spektrum atau periodogram. Sinyal dalam domain frekuensi dapat diproses dengan lebih mudah dibandingkan data pada domain waktu, karena pada domain frekuensi, amplitudo suara tidak terlalu berpengaruh. *Fast Fourier Transform (FFT)* adalah algoritma yang mengimplementasikan *Discrete Fouries Transform (DFT)* yang dioperasikan pada sebuah sinyal waktu diskrit yang terdiri dari sampel menggunakan persamaan berikut.

$$Real\ X[k] = \sum_{i=0}^{N-1} x[i].\cos(\frac{2\pi ki}{N}) \quad \text{..... (2.5)}$$

$$Imajiner\ X[k] = -\sum_{i=0}^{N-1} x[i].\sin(\frac{2\pi ki}{N}) \quad \text{..... (2.6)}$$

Dimana

N = jumlah data

k = 0,1,2, … , $\frac{N}{2}$

$x(i)$ = data pada titik ke- i

Proses selanjutnya adalah menghitung nilai *magnitude* dari FFT. Persamaan yang digunakan adalah persamaan berikut :

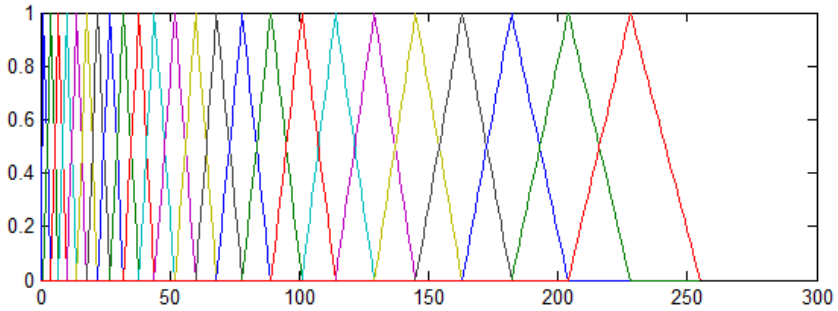
$$|X[k]| = \sqrt{(Real\ X[k])^2 + (Imajiner\ X[k])^2} \quad \text{..... (2.7)}$$

5) Mel-Frequency Wrapping

Persepsi sistem pendengaran manusia terhadap frekuensi sinyal suara tidak hanya bersifat linear. Penerimaan sinyal suara untuk frekuensi rendah (<1k Hz) bersifat linear, dan untuk frekuensi tinggi (>1k Hz) bersifat logaritmik. Jadi, untuk setiap nada dengan frekuensi sesungguhnya , sebuah pola diukur dalam sebuah skala yang disebut “mel” (berasal dari Melody). Skala ini didefinisikan sebagai :

$$F_{mel} = \begin{cases} 2595 \times log_{10}\left(1 + \frac{F_{Hz}}{700}\right), & F_{Hz} > 1000 \\ F_{Hz}, & F_{Hz} < 1000 \end{cases} \quad \text{..... (2.8)}$$

Sebuah pendekatan untuk simulasi spektrum dalam skala mel adalah dengan menggunakan *filter bank* dalam skala mel seperti yang ditunjukkan pada gambar di bawah ini dimana setiap *frame* yang diperoleh dari tahapan sebelumnya difilter menggunakan M *filter* segitiga sama tinggi dengan tinggi satu.



Gambar 2.6 *Mel Filter Bank* dengan 24 buah *filter*

Dalam mel-frequency *wrapping*, sinyal hasil FFT dikelompokkan ke dalam berkas *filter triangular* ini. Proses pengelompokan tersebut adalah setiap nilai FFT dikalikan terhadap *filter* yang bersesuaian dan hasilnya dijumlahkan. Proses *wrapping* terhadap sinyal dalam domain frekuensi dilakukan menggunakan persamaan berikut.

$$X_i = log_{10}(\sum_{k=0}^{N-1} X(k).H_i(k)) \quad \text{..... (2.9)}$$

Dimana

X_i = nilai *frequency wrapping* pada *filter* $i = 1, 2, \dots, n$(jumlah *filter*)

X_n = nilai *magnitude* frekuensi pada k frekuensi

$X_i(k)$ = nilai tinggi *filter* i segitiga dan k frekuensi, dengan $k = 0,1, \dots, N-1$ (jumlah *magnitude* frekuensi)

6) Cepstrum

Cepstrum biasa digunakan untuk mendapatkan informasi dari suatu sinyal suara yang diucapkan oleh manusia. Pada tahap terakhir pada MFCC ini, spektrum log mel akan dikonversi menjadi domain waktu menggunakan *Discrete Cosine Transform (DCT)* menggunakan persamaan berikut.

$$c_j = \sum_{i=1}^{M} X_i \,.\cos(\frac{j(i-1)}{2}.\frac{\pi}{M}) \text{ ..... (2.10)}$$

Dimana

C_i = nilai koefisien $C\ ke\ j$

j = 1,2,… jumlah koefisien yang diharapkan

X_i = nilai X hasil *mel-frequeny wrapping* pada frekuensi $i = 1,2,\dots,n$ (jumlah *wrapping*)

M= jumlah filter

Hasil dari proses ini dinamakan Mel-Frequency Cepstrum Coefficients (MFCC)

2.4 Hidden Markov Model (HMM)

2.4.1 Markov Model

Markov Model biasa disebut sebagai *Markov Chain* atau Rantai Markov. Model ini ditemukan oleh Andrey Markov yang berdasar kepada teori probabilitas yang dapat digunakan untuk memodelkan sebuah rangkaian kejadian berdasarkan atas waktu. Pada Markov Model, probabilitas pada sebuah state hanya bergantung pada nilai probabilitas pada state sebelumnya. Dimana sifat ini biasa dikenal dengan karakteristik Markov (Wiggers & RothKrantz, 2003).

Model ini merupakan bagian dari *finite state* atau *finite automaton*. *Finite automaton* sendiri adalah kumpulan state yang transisi antar state-nya dilakukan berdasarkan masukan observasi.

Pada rantai markov, setiap transisi antar state berisi probabilitas yang mengindikasikan kemungkinan jalur tersebut akan diambil. Jumlah probabilitas semua transisi yang keluar dari sebuah simpul sama dengan satu (Aria, 2013).

2.4.2 Hidden Markov Model

HMM merupakan model stokastik dimana suatu sistem yang dimodelkan diasumsikan sebagai markov proses dengan kondisi yang tidak terobservasi. Suatu HMM dapat dianggap sebagai jaringan Bayesian dinamis yang sederhana(*simplest dynamic Bayesian network*) (Prasetyo, 2010).

Hidden Markov Model (HMM) adalah sebuah sistem yang diasumsikan sebuah proses Markov dengan parameter yang tak diketahui, dan tantangannya adalah menentukan parameter-parameter tersembunyi (hidden) dari parameter yang dapat diamati (Lestary, 2010). Setiap kondisi memiliki distribusi kemungkinan disetiap output yang berbeda. Oleh karena itu urutan langkah yang dibuat oleh HMM memberikan suatu informasi tentang urutan state. Sifat tersembunyi(hidden) berarti bahwa walaupun parameter model diketahui, model tersebut tetap tersembunyi. Secara umum (Adami, 2010), HMM terdiri atas elemen-elemen berikut :

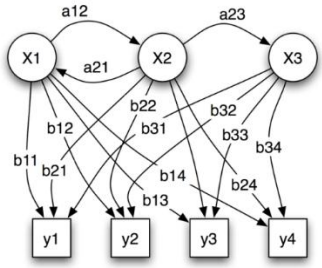
1. Himpunan nilai output observasi $O = \{o_1, o_2, \dots, o_M\}$, dimana M adalah jumlah simbol observasi.
2. Himpunan state $\Omega = \{1,2, \dots, N\}$. Dimana N menyatakan jumlah state yang terdapat pada HMM.
3. Himpunan probabilitas transisi antar state. Diasumsikan bahwa state berikutnya tergantung pada state pada saat ini. Asumsi ini menyebabkan proses perhitungan menjadi lebih mudah dan efisien untuk dilakukan. Probabilitas transisi dapat dinyatakan dengan sebuah matriks A= $\{a_{ij}\}$, dimana a_{ij} adalah probabilitas transaksi dari state i ke state j. Sebagai contoh :

   $$a_{ij} = P(s_t = j | s_{t-1} = i), \qquad 1 \leq i, j \leq N \quad \text{..... (2.11)}$$

   dimana s_t merupakan state pada waktu ke- t.
4. Himpunan probabilitas output $B = \{b_i(k)\}$ pada setiap state. Yang juga disebut probabilitas emisi, $b_i(k)$ adalah probabilitas dari simbol output o_k pada state i yang didefinisikan sebagai

   $$b_i(k) = P(v_t = o_k | s_t = i) \quad \text{.....(2.12)}$$

   dimana v_t adalah simbol observasi pada waktu ke- t.
5. Himpunan state awal $\pi = \{\pi_i\}$, dimana π_i adalah probabilitas state i menjadi state awal pada urutan state HMM.



Gambar 2.7 Parameter Probabilistik pada *Hidden Markov Model*
Sumber:http://www.google.com/imgres?imgurl=http://en.academic.ru/pictures/enwiki/72/HiddenMarkovModel.png

Dimana :

x = kondisi

y = observasi yang mungkin

a = kemungkinan keadaan transisi

b = kemungkinan output

2.4.3 Penyelesaian masalah dengan HMM

Dalam penggunaannya terdapat 3 permasalahan dasar pada HMM untuk dapat melakukan pengenalan terhadap suara (Uchat, 2009). Pertama , masalah evaluasi. Dimana diberikan sebuah λ dari HMM dan barisan observasi $O = O_1,O_2,...,O_t$ dimana terdapat probabilitas obsevasi yang dihasilkan oleh model $p\{O \mid \lambda\}$.

Kedua, masalah *decoding* diberikan sebuah model λ dan barisan observasi $O = O_1,O_2,...,O_t$ dimana kemiripan maksimal barisan state di model yang menghasilkan observasi.

Ketiga adalah masalah pembelajaran dimana diberikan model λ dan barisan pengamatan $O = O_1,O_2,...,O_t$ dimana kita harus menyesuaikan paramter $\lambda = (A,B,\pi)$ untuk memaksimalkan $p\{O \mid \lambda\}$.

2.4.1 Evaluation (evaluasi)

Diberikan barisan obervasi O = O_1,O_2,...,O_t, sebuah model λ=(A,B,π) and p{O | λ) . Ini dapat dihitung menggunakan probabilitas sederhana, namum proses perhitungan ini memiliki kompleksitas N^T. Hal tersebut akan menghasilkan perhitungan dengan nilai yang sangat besar. Sehingga digunakan metode lain untuk yang menggunakan variabel tambahan yaitu αt (i), yang dinamakan variabel maju *(forward)*.

A. Prosedur Forward

Inisialisasi untuk $1 \leq i \leq N$

$$\propto_1 (i) = \pi_i f_i(O_1) \quad \text{..... (2.13)}$$

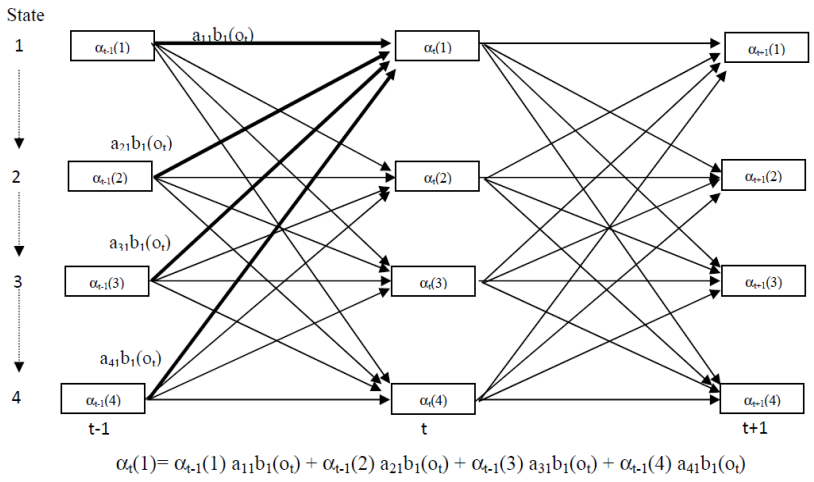
Rekursi maju untuk $t = 1,2,\dots,T-1\ ;\ 1 \leq j \leq N$

$$\propto_{t+1} (j) = \left[\sum_{i=1}^{N} \propto_t (i) a_{ij}\right] f_j(O_{t+1}) \quad \text{..... (2.14)}$$

Perhitungan probabilitas

$$p(O|\lambda) = \sum_{i=1}^{N} \propto_t (i) \quad \text{..... (2.15)}$$

Diagram trellis dapat digunakan untuk menvisualisasikan perhitungan pobabilitas dari HMM. Pada gambar dibawah menunjukan HMM untuk 4 state. Setiap kolom pada trelis menujukan kemungkinan state pada waktu ke t. Setiap state dalam satu kolom terhubung pada setiap state pada kolom yang berdekatan dengan peluang transisi diberikan pada elemen aij dari matriks transisi A



Gambar 2.8 *Diagram Treilis* Untuk Perhitungan Prosedur Maju
Sumber : Aria (2013)

B. Prosedur Backward

Inisialisasi untuk $1 \leq i \leq N$

$$\beta_T(i) = 1 \quad \text{..... (2.16)}$$

Rekursi mundur untuk $t = T-1, T-2, \ldots, 1\ ;\ 1 \leq i \leq N$

$$\beta_t(i) = \sum_{i=1}^{N} \propto_{ij} f_j(O_{t+1})\beta_{t+1}(j) \quad \text{..... (2.17)}$$

Perhitungan probabilitas

$$p(O|\lambda) = \sum_{i=1}^{N} \pi_i f_i(O_1)\beta_1(i) \quad \text{..... (2.18)}$$

2.4.2 Decoding

Dalam kasus ini akan dicari barisan state yang memiliki kemiripan maksimal untuk barisan observasi O = O = $O_1,O_2,\ldots,O_t$ dan model $\lambda = (A,B,\pi)$. Salah satu pendekatan untuk menemukan "most likely state" q_t saat t = t dan menghubungkan seluruh q_t.

Pada metode ini yang dikenal sebagai Algoritma Viterbi. Untuk membantu proses perhitungan, ditambahkan sebuah variabel bantu. Algoritma viterbi

merupakan algoritma induktif dimana mempertahankan setiap barisan state terbaik untuk tiap state N sebagai perantara state untuk barisan pengamatan O = $O_1,O_2,...,O_t$.

A. Inisialisasi

$$\delta_1(i) = \pi_i b_i(O_1) \quad \text{..... (2.19)}$$

$$\Psi_1(i) = 0 \quad \text{..... (2.20)}$$

Untuk $1 \leq i \leq N$

a. Untuk $t = 2, \dots, T$

$$\delta_t(i) = \max_i\left[\delta_{t-1}(i)a_{ij}\right] b_j(t) \quad \text{..... (2.21)}$$

$$\Psi_t(j) = \arg\max_i\left[\delta_{t-1}(i)a_{ij}\right] \quad \text{..... (2.22)}$$

Untuk $1 \leq j \leq N$

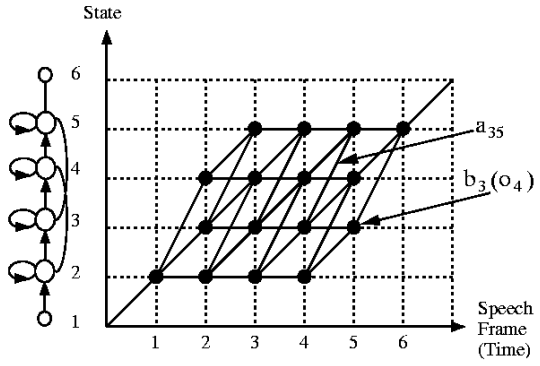
b. Terminasi

$$\Delta^* = max_i[\delta_T(i)] \quad \text{..... (2.23)}$$

$$x_T^* = \arg max_i\,[\delta_T(i)] \quad \text{..... (2.24)}$$

c. Telusur balik untuk $t = T-1, T-2, \dots, 1$

$$x_T^* = \Psi_{t+1}(x_T^*)\ dan\ X^* = \{x_1^*, x_2^*, \dots x_T^*\} \quad \text{..... (2.25)}$$



Gambar 2.9 Proses Rekursif Untuk Menentukan Jalur Terpendek Menggunakan Algoritma Viterbi

Sumber: http://www.google.com/imgres?imgurl=http://izanami.tl.fukuoka-u.ac.jp/SLPL/HMM/HTKBook/img96.gif

Dapat dilihat pada gambar, melalui treliss dan rute yang dibantuk oleh algoritma Viterbi untuk menemukan barisan yang telah dihasilkan. Kemudian diambil rute yang memiliki jarak terkecil.

2.4.3 Learning (pembelajaran)

Umumnya, masalah pembelajaran adalah bagaimana menyesuaikan parameter HMM yang ada. Diberikan sebuah barisan pengamatan O dimana masalah estimasi termasuk untuk menemukan parameter model yang tepat yang menentukan model yang paling optimal.

Terdapat dua kriteria optimasi yang ditemukan dalam literatur ASR, pertama maximum likelihood (ML) dan Maxomum Mutual Information (MMI).

Maximum likelihood (ML) merupakan algoritma untuk mencari probabilitas maksimum dari barisan pengamatan. Probabilitas ini adalah total likelihood (kemiripan) dari obeservasi dan dapat diekpresikan secara matematis sebagai $L_{tot} = \{ O \mid \lambda \}$. Namun, parameter model yang memiliki nilai maksimum lokal dapat dipilih mengunakan prosedur iteratif. Seperti Baum-Welch atau Metode berbasis gradien.

2.4.4 Pemodelan Unit Wicara

HMM dapat digunakan untuk merepresentasikan berbagai unit suara yang mana setiap model memiliki kelebihan dan kekurangan masing-masing sesuai dengan penggunaannya. Jenis-jenis unit suara (Hwang, 1993) yang digunakan sebagai model antara lain :

1. Kata

Model ini mampu mengenali satuan bunyi yang bervariasi karena bunyi yang sama dapat dikenali dengan model yang berbeda kata bunyi tersebut dapat terkandung dalam kata yang berbeda. Model berbasis kata ini biasanya digunakan dalam pengenalan suara yang memiliki kosa kata dengan jumlah yang kecil. Sedangkan untuk kosa kata berukuran besar, model ini tidak dapat menghasilkan pengenalan

yang baik karena diperlukan perulangan yang sangat banyak untuk melakukan pelatihan data.

2. Phone (fonem)

Salah satu pendekatan yang baik dalam pengenalan suara dengan kosa kata cukup besar adalah dengan menggunakan model sub kata seperti fonem. Dengan menggunakan model ini, proses pelatihan akan lebih mudah dilakukan karena objek pelatihan merupakan fonem-fonem yang terdapat dalam sebuah bahasa. Sehingga proses komputasi lebih ringan. Namun, sedikit kekurangan pada model ini adalah masalah pengucapan sebuah kata dimana bunyi fonem akan selalu dipengaruhi oleh fonem lainnya.

3. Multi-phone

Pengembangan dari model fonem adalah multi-phone yang terdiri atas sillabel, demi sillabel, dan yang lainnya.