



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

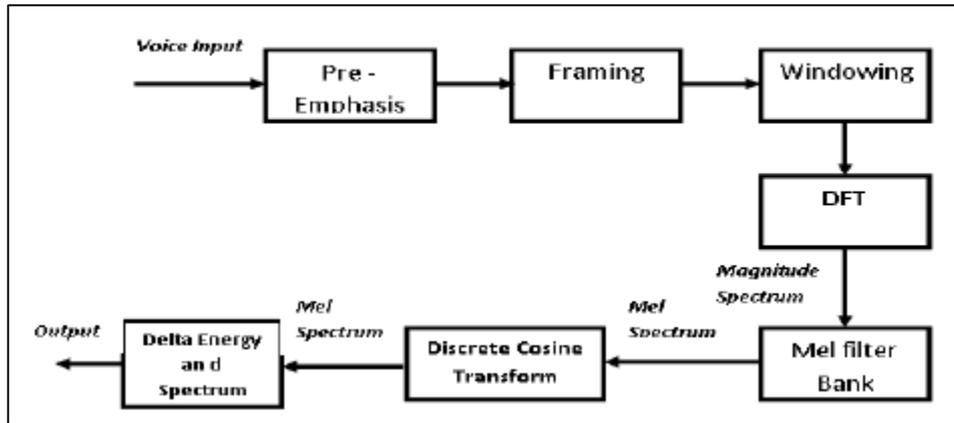
### LANDASAN TEORI

#### 2.1 Semangka

Semangka (*Citrullus vulgaris*, Schard) termasuk salah satu jenis tanaman buah-buahan semusim dan termasuk jenis tanaman menjalar atau merambat dengan perantaraan alat pemegang berbentuk pilih. Semangka juga merupakan tanaman terna semusim yang tumbuh menjalar di atas tanah atau memanjat dengan sulur-sulur atau alat pembelit. Batangnya lunak, bersegi dan berambut, dan panjangnya mencapai 1,5 – 5 m. Sulur tumbuh dari ketiak daun, bercabang 2-3. Daun berseling, bertangkai, helaian daun lebar dan berbulu, berbagi menjari, dengan ujung runcing. Panjang daun berkisar 3 – 25cm dengan lebar 1,5 – 15cm. Bagian tepi daun bergelombang, kadang bergigi tidak teratur. Permukaan bawahnya berambut rapat pada tulangnya.

#### 2.2 MFCC (Mel Frequency Cepstral Coefficients)

MFCC adalah sebuah teknik untuk melakukan ekstraksi fitur pada *file* suara yang hasilnya akan menghasilkan angka yang dapat diproses untuk kebutuhan selanjutnya. MFCC didasarkan pada persepsi pendengaran manusia yang tidak dapat mendengar frekuensi lebih dari 1Khz. Dengan kata lain, MFCC didasarkan pada variasi yang diketahui dari telinga manusia dengan frekuensi [8khz-10khz]. MFCC memiliki dua jenis *filter* yang berjarak linear pada frekuensi rendah di bawah 1000 Hz dan logaritmik atas 1000Hz (Muda, 2010).



Gambar 2.1 Langkah-langkah MFCC (Muda, 2010)

Pada teknik MFCC ada tujuh langkah yang dilakukan seperti yang ada pada gambar 2.1 langkah-langkah tersebut adalah

1. *Pre-Emphasis*. Langkah ini memproses *signal* yang masuk menggunakan *filter* untuk menekan frekuensi yang lebih tinggi pada *signal*. Proses ini akan meningkatkan energi *signal* pada frekuensi yang lebih tinggi.

$$Y[n] = X[n] - 0.95 X[n-1] \quad \dots \quad \text{Rumus 2.1}$$

2. *Framing*. Pada proses ini *signal* bunyi disegmentasi menjadi sejumlah *frame* yang lebih kecil dengan panjang sekitar 20 sampai 40 msec. *signal* bunyi dibagi ke dalam *frame* dari  $N$  *sample* dan *frame* yang berdekatan dipisahkan oleh  $M$  ( $M < N$ ). *Value* yang sering dipakai adalah  $M = 100$  dan  $N = 256$ .
3. *Hamming Windowing*. *Hamming Windowing* digunakan sebagai *window shape* dengan mengkonsiderasikan blok berikutnya pada *feature extraction processing chain* dan mengintegrasikan semua garis frekuensi terdekat.
4. *Discrete Fourier Transformation*. DFT digunakan untuk melakukan konversi  $N$  *sample* yang ada dari domain waktu ke domain frekuensi.

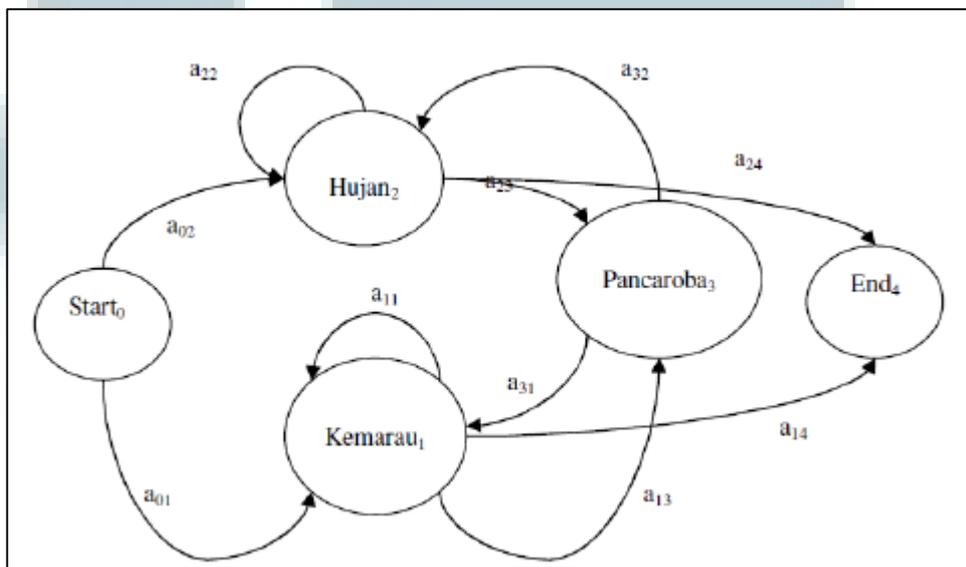
Proses DFT dilakukan menggunakan algoritma FFT (*fast forier transformation*).

5. *Mel Filter Bank Processing*. Pada tahap FFT frekuensi *range* dalam FFT *spectrum* sangatlah besar dan *signal* suara tidak mengikuti *linear scale*. Pada proses ini dilakukan suatu proses penghitungan jumlah dari komponen *filter spectral* agar output yang dihasilkan mendekati *Mel scale*.
6. *Discrete Cosine Transform*. Proses ini mengubah *log Mel spectrum* ke dalam domain waktu menggunakan *Discrete Cosine Transform* (DCT). Hasil dari proses ini disebut sebagai *Mel Frequency Cepstrum Coefficient*.
7. *Delta Energy* dan *Delta Spectrum*. Proses ini menambahkan fitur yang berguna untuk menghitung perubahan pada *voice signal* dan perubahan *frame*.

### **2.3 Hidden Markov Model**

Sebelum membahas tentang *Hidden Markov Model*. Penulis membahas dulu tentang *Markov Model* atau yang sering disebut sebagai *Markov Chain*. Model ini ditemukan oleh Andrey Markov dan merupakan bagian dari proses stokastik yang memiliki properti *Markov* (Prasetyo, 2010). Dengan memiliki properti tersebut berarti, apabila diberikan *input* keadaan saat ini, keadaan akan datang dapat diprediksi dan *markov chain* tersebut lepas dari keadaan di masa lampau. Artinya, deskripsi kondisi saat ini menangkap semua informasi yang mempengaruhi evolusi dari suatu sistem di masa depan. Dengan kata lain, kondisi masa depan dituju dengan menggunakan probabilitas bukan dengan determinasi.

Model ini merupakan bagian dari *finite state* atau *finite automation*. *Finite automation* sendiri adalah kumpulan *state* yang transisi antar *state*-nya dilakukan berdasarkan masukan observasi. Pada *Markov Chain*, setiap busur antar *state* berisi probabilitas yang mengindikasikan kemungkinan jalur tersebut akan diambil. Jumlah probabilitas semua busur yang keluar dari sebuah simpul adalah satu. Gambar di bawah menggambarkan contoh *Markov Chain* yang menggambarkan kondisi cuaca.



Gambar 2.2 Representasi dari *Markov Chain*

Pada gambar 2.2,  $a_{ij}$  adalah probabilitas transisi dari *state*  $i$  ke *state*  $j$ . Misalkan, dari simpul  $start_0$  keluar dua kemungkinan  $a_{02}$  dan  $a_{01}$ . Maka jumlah probabilitas  $a_{01} + a_{02}$  adalah satu. Hal ini juga berlaku bagi simpul-simpul yang lain. *Markov chain* bermanfaat untuk menghitung probabilitas suatu kejadian teramati yang secara umum dapat dirumuskan sebagai berikut.

$$P(\sigma_t) = P(\sigma_t | \sigma_{t-1}, \sigma_{t-2}, \sigma_{t-3}, \dots) \quad \dots \quad \text{Rumus 2.2}$$

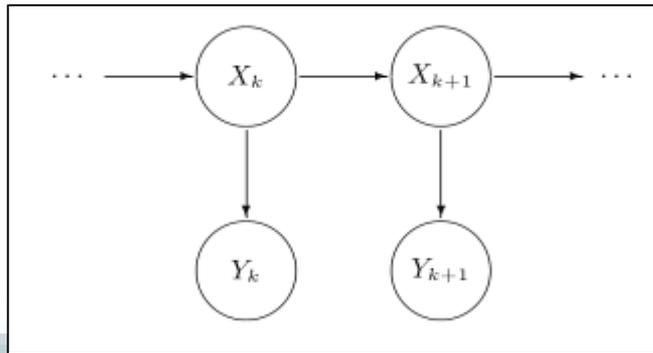
$\sigma_1$  adalah kondisi saat ini, dan  $\sigma_t$  adalah kondisi pada waktu tertentu yang berhubungan dengan  $\sigma_1$ . Sedangkan  $\sigma_{t-1}$  adalah kondisi sebelum  $\sigma_1$ . Dari

penjelasan di atas, dapat diketahui bahwa *Markov Chain* bermanfaat untuk menghitung probabilitas urutan kejadian yang dapat diamati. Terkadang ada urutan kejadian yang ingin diketahui tetapi tidak dapat diamati. Untuk itulah dikembangkan model baru yaitu *Hidden Markov Model*.

HMM (*Hidden Markov Model*) merupakan model statistik dimana suatu sistem yang dimodelkan diasumsikan sebagai *markov process* dengan kondisi yang tidak terobservasi. Suatu HMM dapat dianggap sebagai jaringan Bayesian dinamis sederhana (*simplest dynamic Bayesian network*).

Dalam *Markov Model* biasa, setiap keadaan dapat terlihat langsung oleh pengamat. Oleh karena itu, kemungkinan dari transisi antar kondisi menjadi satu-satunya parameter teramati. Dalam HMM, keadaan tidak terlihat secara langsung, tetapi *output* yang bergantung terhadap keadaan tersebut terlihat. Setiap kondisi memiliki distribusi kemungkinan di setiap *output* yang mungkin. Oleh karena itu, urutan langkah yang dibuat oleh HMM memberikan suatu informasi tentang urutan dari keadaan. Perlu dipahami, bahwa sifat *hidden* “tersembunyi” menunjuk pada kondisi langkah yang dilewati model, bukan kepada parameter-parameter dari model tersebut. Walaupun parameter model diketahui, model tetap tersembunyi.

Anggap bahwa deretan Markov yang disembunyikan adalah  $\{X_k\}_{k \geq 0}$ , kemudian proses yang dapat diamati adalah  $\{Y_k\}_{k \geq 0}$ , maka secara grafik dapat direpresentasikan seperti gambar di bawah ini.



Gambar 2.3 *Graphical Representation HMM*

Gambar 2.3 *Graphical Representation* dari struktur *hidden Markov model*, dimana  $\{Y_k\}$  adalah proses yang dapat diamati dan  $\{X_k\}$  adalah proses yang tersembunyi. Pada gambar 2.3 dapat dilihat bahwa garis dengan arah kanan pada  $X_k$  kepada  $X_{k+1}$  menggambarkan bahwa distribusi dari variabel  $X_{k+1}$  bergantung pada proses  $X_0, \dots, X_k$ .

HMM adalah variasi dari *finite state machine* yang memiliki kondisi tersembunyi  $Q$ , suatu nilai *output*  $O$  (observasi), kemungkinan transisi  $A$ , kemungkinan *output*  $B$ , sebuah kondisi awal  $\Pi$ . Kondisi saat ini tidak terobservasi (Prasetyo, 2010). Tetapi, setiap keadaan menghasilkan *output* kemungkinan  $B$ . Biasanya,  $Q$  dan  $O$  dimengerti, jadi HMM disebut triple  $(A, B, \Pi)$ . Penjabaran *variable* yang biasanya digunakan untuk pembuatan HMM adalah sebagai berikut

1. Himpunan *observed state*:  $O = o_1, o_2, \dots, o_N$
2. Himpunan *hidden state*:  $Q = q_1, q_2, \dots, q_N$
3. Probabilitas transisi:  $A = a_{01}, a_{02}, \dots, a_{n1} \dots a_{nm}$ ;  $a_{ij}$  adalah probabilitas untuk pindah dari *state*  $i$  ke *state*  $j$ .
4. Probabilitas emisi atau *observation likelihood*:  $B = b_i(O_t)$ , merupakan probabilitas observasi  $O_t$  dibangkitkan oleh *state*  $i$ .
5. *State* awal:  $q_0$ , yang tidak terkait dengan observasi.

## 2.4 Voice recognition

Pengenalan suara merupakan salah satu upaya agar suara dapat dikenali atau diidentifikasi sehingga dapat dimanfaatkan. Pengenalan suara dapat dibedakan ke dalam tiga bentuk pendekatan, yaitu pendekatan akustik-fonetik (*the acoustic-phonetic approach*), pendekatan kecerdasan buatan (*the artificial intelligence approach*), dan pendekatan pengenalan pola (*the pattern recognition approach*) (Lee, 1999).

Tujuan dari bicara (*speech*) adalah komunikasi. Ada beberapa cara untuk mengkararakteristik potensi bicara pada komunikasi. Menurut teori informasi, bicara atau *speech* dapat direpresentasikan dalam kondisi isi dari pesannya. Cara alternatif lain untuk mengkararakteristik *speech* adalah dengan menggunakan *signal* yang mengandung informasi mengenai pesan tersebut (*acoustic waveform*). Sebuah *signal speech* yang sukses membutuhkan pengetahuan dan keahlian yang baik dalam *signal processing*, *physics (acoustics)*, *pattern recognition*, *computer science*, dan lain-lain (Sigmund 2003). Basis dari semua *speech processing* adalah *speech* analisis diikuti oleh *sophisticated coding* dan *pattern recognition*. *Speech recognition* bisa tercapai setelah melakukan *speech processing* sehingga dapat disimpulkan bahwa *speech recognition* adalah suatu proses mengenali suara nyata yang dilakukan oleh komputer melalui *speech processing*.