**PENERAPAN METODE MEL FREQUENCY CEPTRAL COEFFICIENT DAN LEARNING VECTOR QUANTIZATION UNTUK TEXT-DEPENDENT SPEAKER VERIFICATION**

**TUGAS AKHIR**

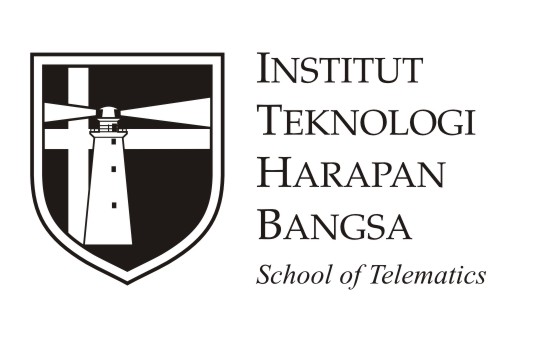
Diajukan sebagai syarat untuk menyelesaikan

Program Studi Strata-1 Departemen Teknik Informatika

Oleh :

**Sukoreno Mukti Widodo**

**1112051**



**DEPARTEMEN TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI HARAPAN BANGSA**

**BANDUNG**

2016BAB I

**PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang, lingkup permasalahan, tujuan dan manfaat penelitian, batasan masalah, dan sistematika penulisan.

**1.1 Latar Belakang**

Layanan keamanan pada umumnya menggunakan kata sandi untuk membatasi dan mengontrol akses layanan tersebut. Kata sandiyang biasa digunakan sering kali berbentuk teks, sehingga setiap kali pengguna ingin melakukan akses terhadap layanan tersebut, pengguna harus memasukan kata sandidengan cara mengetik kata sandi tersebut ke dalam *password field* yang tersedia. Penggunaan kata sandi dengan bentuk teks dianggap masih kurang aman karena sering kali terjadi kebocoran. Maka dari itu dibutuhkanlah bentuk lain dari kata sandi, untuk meningkatkan keamanan dalam mengakses layanan atau data tertentu. Salah satunya adalah dalam bentuk suara.

Manusia dapat mengenali seseorang hanya dengan mendengarkan orang tersebut berbicara. Maka, beberapa detik ucapan sudah cukup untuk mengidentifikasi pembicara. *Speaker* *Recognition* adalah proses untuk mengidentifikasi pembicara secara otomatis berdasarkan karakteristik suara [KSH12]. Melalui konsep ini, *Speaker Recognition* memungkinkan untuk menggunakan karakteristik suara pembicara sebagai kata sandi untuk memverifikasi identitas pembicara dan mengakses layanan tersebut.

Ada dua jenis *Speaker Recognition,* yaitu [ZHI13]: *Text Independent Speaker Verification* (TI-SV)dan *Text Dependent Speaker Verification* (TD-SV). TI-SV adalah jenis proses verifikasi suara pembicara tanpa membatasi pengenalan dengan kata-kata tertentu. TI-SV membutuhkan pelatihan data yang banyak untuk mendapatkan akurasi yang baik. Sedangkan TD-SV adalah jenis proses verifikasi suara pembicara dengan menggunakan kata-kata yang sama, tipe ini lebih cocok untuk digunakan untuk teknik identifikasi dan verifikasi suara pembicara karena untuk proses verifikasi pembicara bisa didapatkan hanya dengan mengektraksi ciri suara melalui beberapa kata sebagai *sample*.

Metode-metode yang dapat digunakan untuk mengekstraksi fitur suara diantaranya adalah LPC, LPCC, dan MFCC. Berdasarkan penelitian yang dilakukan oleh Utpal Bhattacharjee pada tahun 2013, metode MFCC yang digunakan untuk pengenalan suara dengan kondisi *noise* sebesar 20dB memberikan akurasi sebesar 97.03% sedangkan metode LPCC hanya memberikan akurasi sebesar 73.76% [BHA13]. Untuk metode LPC, tidak direkomendasikan untuk pengenalan pembicara karena lebih cocok untuk komputasi linear, sedangkan suara manusia pada dasarnya adalah non-linear [DAV13].

Sedangkan metode yang digunakan untuk pengenalan pola suara adalah LVQ, selain metode LVQ, penelitian *Speaker Recognition* dan *Speech Recognition* biasanya juga menggunakan metode HMM atau GMM. Metode LVQ lebih cocok digunakan untuk TD-SVkarena pengenalan pembicara jenis initidak memerlukan banyak pelatihan data seperti TI-SVsehingga lebih efisien dalam hal data latih dan waktu pelatihan data untuk penelitian ini [PEN15]. Sedangkan metode GMM dan HMM lebih banyak digunakan pada model stokastik, dimana lebih cocok digunakan untuk kasus TI-SV [GEE14]*.*

Pada penelitian ini, metode yang digunakan untuk ekstraksi fitur suara adalah metode MFCC. Metode ini merupakan metode yang paling terkenal dan paling sering digunakan dalam melakukan ekstraksi fitur suara dalam penelitian *Speaker Recognition* dan *Speech Recognition,* karena mampu untuk mengekstraksi karakteristik sinyal suara secara jelas, yang relatif berbeda dari setiap sifat saluran suara pembicara dan lebih efektif digunakan untuk pengenalan suara yang mengandung *noise.* Metode LVQ digunakan sebagai pengenalan pola untuk mengidentifikasi pembicara sesuai dengan fitur suara yang telah diekstraksi menggunakan metode MFCC. Menurut penelitian yang dilakukan oleh Geeta Nijhawan pada tahun 2014, kombinasi metode MFCC dan LVQ ini dianggap sangat baik untuk kasus TI-SVdengan menghasilkan akurasi sebesar 95%, sehingga penulis mendapat ide dari saran penelitian tersebut untuk melakukan penelitian TD-SV dengan menggunakan kombinasi metode MFCC dan LVQ.

**1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, rumusan masalah dari latar belakang tersebut adalah sebagai berikut:

1. Apakah metode MFCC mampu mengekstraksi ciri suara hingga mendapatkan warna suara yang unik dari masing-masing pembicara ?
2. Bagaimana hasil dari identifikasi dan verifikasi suara pembicara dengan menggunakan kombinasi metode MFCC dan LVQ ?

**1.3 Tujuan**

Tujuan dari penelitian ini adalah selain mampu untuk mengidentifikasi pembicara melalui pengenalan suara, juga untuk mengetahui performa dari kombinasi metode MFCC dan LVQ untuk identifikasi dan verifikasi suara pembicara.

**1.4 Batasan Masalah**

Untuk mempersempit masalah yang ditelit maka disusunlah batasan masalah seperti berikut :

1. Kata yang akan digunakan untuk melakukan verifikasi dibatasi hanya berupa buka, kunci, unlock dan lock.
2. Aplikasi yang dibuat hanya menerima masukan berupa suara dan keluarannya merupakan hasil identifikasi pembicara dan kata-kata terdaftar.

**1.5 Kontribusi Penelitian**

Membuat pengembangan aplikasi yang mampu mengidentifikasi dan memverifikasi pembicara dengan pengenalan suara serta mendapatkan perbaikan akurasi yang lebih baik yang pada penelitian sebelumnya yang dilakukan oleh Daniel Christian T. dengan Judul “Penerapan Metode Mel-Frequency Cepstral Coefficients Dan K-Means Clustering Untuk Pengenalan Pembicara” yang sebelumnya menggunakan metode MFCC dan K-Mean Clustering dan akan dikembangkan dengan menggunakan kombinasi metode MFCC dan LVQ untuk identifikasi dan verifikasi pembicara.

**1.6 Metodologi Penelitian**

Tahap-tahap yang penulis lakukan untuk pembuatan aplikasi adalah sebagai berikut:

1. Studi Literatur

Pada tahap ini penulis mencari data dan informasi mengenai pengenalan suara pembicara, *decoding* sinyal suara, perekaman data suara, serta metode *Mel Frequency Ceptral Coefficient* (MFCC) dan *Learning Vector Quantization* (LVQ).

1. Data Sampling

Data sampling merupakan perekaman suara pembicara dengan mengucapkan kata buka, kunci, unlock dan lock.

1. Analisis Permasalahan

Penulis melakukan analisis masalah, batasan dan kebutuhan untuk melakukan penelitian.

1. Perancangan

Penulis melakukan perancangan aplikasi seperti melakukan desain *database,* dan *user interface*.

1. Implementasi

Penulis melakukan implementasi dari perancangan yang dilakukan di tahap sebelumnya dan metode untuk mengidentifikasi suara pembicara dengan menggunakan metode *Mel Frequency Ceptral Coefficient* (MFCC) dan *Learning Vector Quantization* (LVQ).

1. Pengujian

Penulis melakukan pengujian terhadap aplikasi yang sudah diimplementasikan oleh penulis dengan melakukan *input* data yang beragam.

**1.7 Sistematika Penulisan**

Laporan tugas akhir ini dibagi menjadi 5 bab, yaitu :

1. Bab I Pendahuluan yang berisi berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian.

2. Bab II Landasan Teori yang berisi landasan teori dalam pembuatan tugas akhir ini.

3. Bab III Analisis dan Perancangan yang berisi analisis masalah dan pemodelan dari tugas akhir ini.

4. Bab IV Implementasi dan Pengujian yang berisi implementasi sistem dan pengujian terhadap sistem yang telah dibangun tersebut.

5. Bab V Penutup yang berisi kesimpulan dari tugas akhir ini dan saran untuk pengembangan lebih lanjut dari tugas akhir ini.

**BAB II**

**LANDASAN TEORI**

Pada bab ini akan di bahas mengenai review studi literatur yang akan digunakan untuk penelitian, Metode yang akan digunakan dan juga Kerangka pemikiran awal untuk metode.

2.1 Tinjauan Studi

Dalam penelitian ini peneliti menggunakan referensi dari 3 jurnal yaitu

Penghua Li, Shunxing Zhang, Huizong Feng, dan Yuanyuan Li (2015), “Speaker Identiﬁcation using Spectrogram and Learning Vector Quantization”, JCIS Journal of Computational Information Systems, Vol. 11, Issue 9. Dalam jurnal ini peneliti melakukan pengenalan pembicara melalui suaramenggunakan metode *Spectogram* sebagai ekstraksi fitur dan *Learning Vector Quantization* (LVQ).

Dalam penelitian tersebut, terdapat 3 langkah yang dilakukan oleh peneliti, yaitu *preprocessing, Feature extraction,* dan *Classifier*. Pada tahap *preprocessing*, dilakukan *emphasizing* dan *windowing* pada *input* suara. Setelah itu, *input* suara yang telah melalui *preprocessing* ditransformasikan menggunakan *short-time* *Fast Fourier Transform* untuk mendapatkan *spectrogram* dari suara tersebut. Kemudian dilanjutkan dengan *Gabor Filter* untuk meningkatkan tekstur suara dan menggunakan algoritma *Local Binary Pattern* untuk vektor fitur suaranya. Lalu pada tahap klasifikasi, vektor fitur suara tersebut di *input* ke jaringan LVQ untuk data latih dan percobaan identifikasi pembicara.

Data suara dikumpulkan dari 5 responden (3 pria dan 2 wanita) masing-masing responden mengulangi ucapan sebanyak 10 kali. Spesifikasi suara yang didapat yaitu ukuran *frame* sebesar 30ms dan panjang *overlapping* nya adalah 15ms, sedangkan frekuensi maksimumnya adalah 8 kHz.

Dari hasil penelitian yang dilakukan, dengan menggunakan metode *Spectogram* dan *Learning Vector Quantization* menunjukkan akurasi yang cukup baik yaitu sebesar 90%.

Kemudian dalam penelitian yang dilakukan oleh Kshamamayee Dash, Debananda Padhi, Bhoomika Panda dan Sanghamitra Mohanty (2012). “Speaker Identification using Mel Frequency Cepstral Coefficient and BPNN”, IJARCSSE International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4. Metode yang digunakan adalah *Mel Frequency Ceptral Coefficient* (MFCC) dan *Back Propagation Neural Network* (BPNN).

Pada penelitian tersebut langkah pertama yang dilakukan adalah *Front-end processing* yang dimana bagian ini akan mengubah sinyal suara menjadi kumpulan vektor-vektor dari fitur suara yang memiliki karakteristik masing-masing dari suara yang di *input*. Lalu selanjutkan dilakukan *Speaker Modeling*, bagian ini membagi-bagi data karakteristik suara menjadi beberapa model sebagai data pembanding untuk identifikasi pembicara. Bagian terakhir adalah *decision logic* yaitu menentukan identitas pembicara dengan cara membandingkan vektor fitur suara yang baru dengan yang ada di *database* dan memilih model terbaik.

Hasil yang didapatkan dari penelitian adalah dari 100 suara *sampling* didapatkan akurasi sebesar 85% dengan *error rate* diantara -1 sampai 2.5. Jenis pengenalan pembicara yang digunakan dalam penelitiannya adalah *Text Dependent Speaker Verification* dengan kata-kata yang digunakan adalah “*welcome to the world of speaker identification*”.

Dalam penelitian yang dilakukan oleh Rupali S Chavan dan Ganesh S. Sable (2013). “Text Dependent Speaker Independent Isolated Word Speech Recognition Using HMM”, IJECCCE International Journal of Electronics Communication and Computer Engineering, Volume 4, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209. Metode yang digunakan pada penelitian tersebut adalah MFCC untuk mengekstraksi fitur suara dan HMM/GMM untuk pengenalan.

Pada penelitan tersebut, langkah pertama yang dilakukan adalah *preprocessing*, suara diambil menggunakan *microphone* spesifikasi suara yang diambil adalah suara selama 2 detik dengan kedalaman 16 bit, menggunakan *sampling rate* sebesar 11025Hz. Selanjutnya dilakukan *feature extraction* langkah-langkah yang dilakukan pada *feature extraction* adalah *pre-emphasis, framing, windowing, FFT, Mel Filterbank,* dan *DCT*. Pelatihan dan pengenalan pada penelitian tersebut sama-sama menggunakan HMM.

Dalam pembuatan data latih, penelitian tersebut mengumpulkan 5 responden pembicara dan diminta untuk mengucapkan 4 kali untuk kata yang sama pada masing-masing responden, sehingga didapatkan 20 total sampel suara untuk 1 kata. Setiap *input* suara diambil 2 detik dengan kedalaman 16 bit, dengan *sampling rate* sebesar 11025Hz.

Hasil dari penelitian ini menunjukan bahwa metode MFCC dan HMM yang digunakan untuk mengidentifikasi pembicara dengan kata-kata yang digunakan adalah bahasa Inggris menghasilkan performa yang baik dalam melakukan pengenalan pembicara, performa yang baik ini dihasilkan karena terdapat kemiripan antara arsitektur dari HMM data latih dan berbagai data bicara.

2.1.1 *State* of *the Art* dari rujukan penelitian

Dalam penelitian ini, terdapat pula penelitian – penelitian sebelumnya yang membahas hal yang sama. Penelitian sebelumnya adalah sebagai berikut :

Tabel 2.1 State of the Art

| Peneliti | Tahun | Topik | Metode |
| --- | --- | --- | --- |
| Penghua Li, Shunxing Zhang, Huizong Feng, Yuanyuan Li | 2015 | Speaker Identiﬁcation using Spectrogram and Learning Vector Quantization | 1. Spectogram 2. Learning Vector Quantization |
| Kshamamayee Dash, Debananda Padhi, Bhoomika Panda, Sanghamitra Mohanty | 2012 | Speaker Identification using Mel Frequency Cepstral Coefficient and BPNN | 1. *Mel Frequency Ceptral Coefficients* 2. *Back Propagated Neural Network* |
| Rupali S Chavan, Ganesh S. Sable | 2013 | Text Dependent Speaker Independent Isolated Word Speech Recognition Using HMM | 1. *Mel Frequency Ceptral Coefficients* 2. *Hidden Markov Model* |

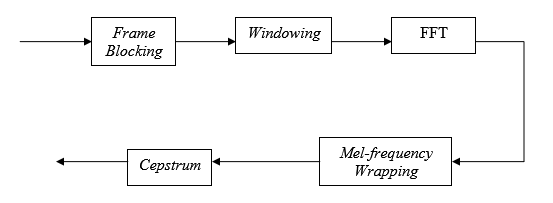
2.2 Tinjauan Pustaka

2.2.1 *Mel Frequency Ceptral Coefficient (MFCC)*

Ekstraksi fitur merupakan proses perhitungan dari sinyal suara untuk merepresentasikan karakteristik dari sinyal tersebut dalam bentuk data diskrit. Salah satu metode yang paling popular untuk mengekstraksi fitur adalah MFCC. MFCC merupakan metode untuk mengekstraksi fitur yang menghitung koefisien *cepstral* berdasarkan variasi frekuensi kritis pada sistem pendengaran manusia.

Beberapa keunggulan metode MFCC adalah [BHA15]:

1. Mampu menghasilkan data dengan akurasi tinggi dalam kondisi suara banyak *noise.*
2. Mampu mengekstraksi fitur suara selengkap mungkin dengan data seminimal mungkin.
3. Dapat mereplikasi sistem pendengaran suara manusia, sehingga akurasinya tinggi.

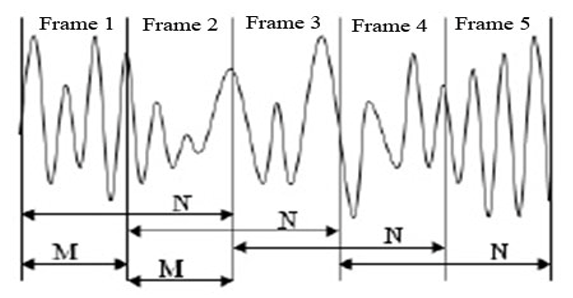


Gambar 2.1 Tahap-tahap MFCC

2.2.2 *Frame Blocking* [ANG11]

Pada tahap *frame blocking*, sinyal suara dibagi menjadi beberapa *frame* dengan panjang pada umumnya sebesar 20-30ms yang berisi N sampel masing-masing frame dipisahkan oleh M (M<N) dimana M adalah banyaknya pergeseran antar *frame*.

*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. *Frame blocking* diperlukan karena sinyal suara mengalami perubahan dalam jangka waktu tertentu.



Gambar 2.2 Proses Frame Blocking

2.2.3 *Windowing*

Proses *Windowing* dilakukan untuk meminimalisir diskontinuitas yang terjadi pada sinyal, yang disebabkan oleh kebocoran spektral pada saat proses *frame blocking* dilakukan dimana sinyal yang baru, memiliki frekuensi yang berbeda dengan sinyal aslinya.

Konsep dari *windowing* adalah meruncingkan ujung sinyal menjadi nol pada bagian awal dan akhir setiap *frame.* Proses *windowing* dilakukan dengan cara mengalikan tiap *frame* dari dengan jenis *window* yang digunakan, yang dapat dituliskan dalam persamaan berikut [ANG11]:

(2.1)

Dimana,

= sinyal hasil *windowing* sampel ke- 𝑛

= nilai sampel ke- 𝑛 sebelum di *windowing*

= nilai *window* ke- 𝑛

= jumlah sampel setiap *frame*

= indeks sampel dalam suatu *frame*

Karena dalam penelitian suara banyak menggunakan *hamming* window dengan kesederhanaan rumus serta hasil dari *windowing* yang baik maka jenis *window* yang digunakan adalah *hamming* *window* yang dituliskan ke dalam persamaan berikut [LYO11]:

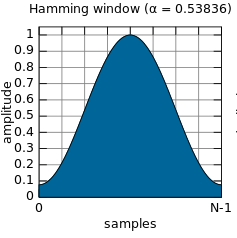
(2.2)

Dimana,

= nilai *window* ke- 𝑛

= jumlah sampel setiap *frame*

= indeks sampel dalam suatu *frame*

*Sumber: https://en.wikipedia.org/wiki/Window\_function*

Gambar 2.3 Hamming Window

2.2.4 *Discrete Fourier Transform*

Untuk mendapatkan analisis frekuensi dari sinyal waktu diskrit maka perlu didapatkan representasi frekuensi dari sinyal yang biasanya dinyatakan dalam domain waktu. *Discrete Fourier Transform* (DFT) adalah prosedur yang digunakan dalam pemrosesan sinyal digital dan filterisasi digital. DFT memungkinkan kita untuk menganalisa, memanipulasi dan mensintesis sinyal yang tidak mungkin dilakukan pada sinyal analog [LYO11]. DFT berasal dari *Continue Fourier Transform* yang didefinisikan sebagai berikut:

(2.3)

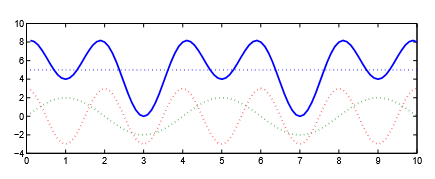
Dimana adalah sinyal kontinu dalam domain waktu. Persamaan tersebut digunakan untuk mentransformasikan fungsi sinyal dalam domain waktu ke dalam domain frekuensi . DFT sendiri dapat dinyatakan ke dalam persamaan matematis berikut bentuk eksponensial:

(2.4)

Dimana adalah urutan nilai diskrit sampel dalam domain waktu dari suatu sinyal dan adalah . Persamaan tersebut dapat dinyatakan dalam bentuk *rectangular* dengan persamaan sebagai berikut:

(2.5)

Dimana,

 komponen output DFT ke-m

Gambar 2.4 Contoh Sinyal DFT

indeks *output* DFT dalam domain frekuensi, m = 0, 1, 2, 3, …, N-1

= urutan *input* sampel,

= indeks input sampel dalam domain waktu, n = 0, 1, 2, 3 ... , N-1

=

= jumlah urutan input sampel dan jumlah titik frekuensi dalam output DFT

Nilai frekuensi dari setiap sinusoid yang berbeda tergantung kepada sampel sinyal asli () dan jumlah sampel . Nilai frekuensi sinusoid dapat dihitung dengan . Sedangkan analisis frekuensi adalah perkalian dengan frekuensi sinusoid, yang bisa dihitung dengan rumus sebagai berikut:

(2.6)

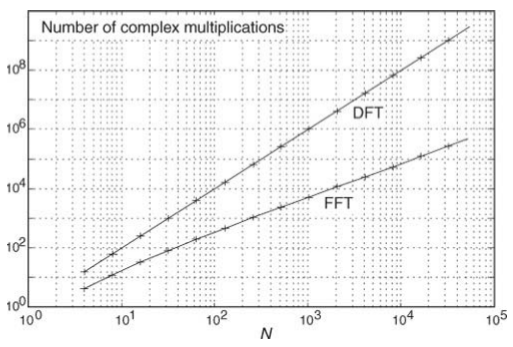
Dari persamaan tersebut kita mendapati bahwa menghasilkan nilai *magnitude* dari komponen sinyal *input* yang didapat dari hasil akar dari penjumlahan bilangan real dan bilangan imajiner yang dikuadratkan dan dapat dituliskan dalam persamaan berikut:

(2.7)

2.2.5 *Fast Fourier Transform* [LYO11]

*Fast Fourier Transform* (FFT) adalah tahap untuk mengubah setiap *frame* yang terdiri dari N sampel dari domain waktu ke dalam domain frekuensi, sama dengan DFT, yang membedakan adalah operasi DFT membutuhkan operasi O(*N*2) sedangkan FFTmengurangi operasi komputasinya menjadi O(*N* log2 (*N*)). FFTdilakukan untuk mendapatkan besaran frekuensi pada setiap *frame*. Keluaran dari proses FFTini adalah berupa spektrum atau periodogram.

Salah satu algoritma dari FFT yang paling popular adalah radix-2. Sebagai pembanding dengan DFT, untuk jumlah sampel N yang besar seperti contoh N = 512, dengan menggunakan perhitungan DFT membutuhkan perhitungan sebesar 114 kali lebih banyak daripada yang dibutuhkan oleh perhitungan FFT. Semakin besar jumlah sampel N, maka perhitungan semakin komplek jika menggunakan DFT.

*Sumber: Understanding Digital Signal Processing*

Gambar 2.5 Grafik perbandingan kompleksitas perhitungan DFT dan FFT

Langkah pertama untuk menginterpretasikan FFT adalah dengan menghitung nilai frekuensi dari setiap sampel tengah dari FFT. Jika sampel waktu yang diterima FFT dalam bentuk real, maka hanya keluaran dari sampai yang independen. Dalam kasus ini, kita hanya perlu menghitung nilai frekuensi FFT untuk selama . Jika sampel waktu yang diterima berbentuk kompleks, kita harus menghitung semua nilai frekuensi FFT untuk selama . Untuk menghitung amplitudo dari FFT digunakan rumus sebagai berikut:

(2.8)

Dan untuk menghitung nilai *magnitude* dari FFT, digunakan persamaan:

(2.9)

Turunan langsung dari hasil FFT adalah pembagian urutan data *input* menjadi 2 bagian, yaitu elemen indeks ganjil dan elemen indeks genap yang dapat dilihat dari persamaan berikut:

(2.10)

untuk sampai m = , dan

(2.11)

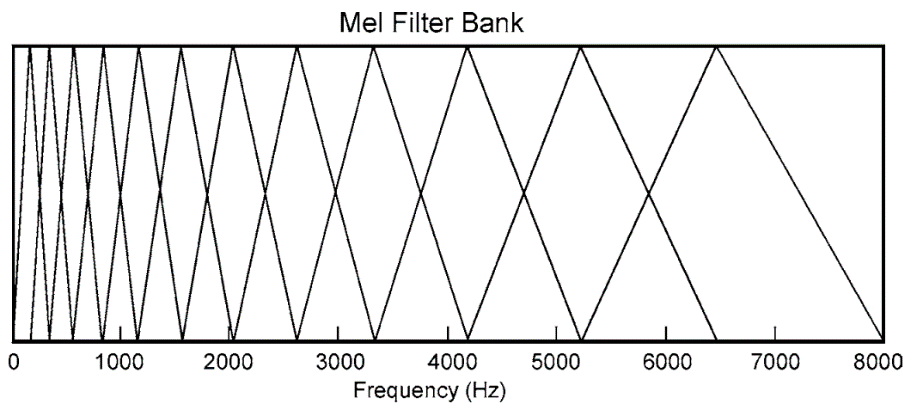
untuk sampai m = , dimana:

(2.12)

dalam rentang 0 sampai karena masing-masing dari dua DFT di sisi kanan pada persamaan 2.7 periodik dalam m dengan periode .

2.2.6 ­*Mel-Frequency Wrapping*

*Mel-Frequency Wrapping* menggunakan *Filterbank* untuk menyaring sinyal suara yang telah diubah menjadi ke dalam bentuk domain frekuensi. *Filterbank*adalah sistem yang membagi *input* sinyal ke dalam kumpulan analisis sinyal, yang masing-masing sesuai dengan wilayah yang berbeda sesuai spektrum [RYA08]. Biasanya, daerah di spektrum yang diberikan oleh analisis sinyal kolektif menjangkau seluruh suara yang terdengar oleh pendengaran manusia, yaitu dari sekitar 20 Hz sampai 20 kHz. *Filterbank* yang digunakan dalam metode MFCC khususnya untuk proses *Mel-Frequency Wrapping* adalah *mel-filterbank­*. ­*mel-filterbank* yang terdiri dari rangkaian *Triangular Window* yang saling *overlap* akan menyaring sinyal sebanyak N sampel.



Gambar 2.6 Mel Filterbank

Nilai *mel* dapat dihitung dengan persamaan berikut:

(2.13)

Sedangkan untuk mendapatkan nilai frekuensi dari nilai *mel* dapat dihitung dengan persamaan berikut:

(2.14)

Dimana,

Nilai *mel,* konversi dari nilai frekuensi

Nilai frekuensi, konversi dari nilai mel

Nilai frekuensi

Batas atas frekuensi maksimal adalah setengah dari frekuensi sampling dari file audio. Setelah itu menentukan nilai mel, nilai mel didefinisikan sebagai .

Nilai *mel* didapat dirubah ke dalam Hertz (Hz) dengan menggunakan persamaan 2.14, tetapi untuk dapat membuat *filterbank,* nilai frekuensi harus dikonversikan ke dalam nilai sampel FFT terdekat dengan menggunakan persamaan berikut [HUA01]:

(2.15)

Dimana,

= Nilai sample FFT

= Jumlah FFT tiap frame

= Frekuensi sampling

= Frekuensi batas atas

= Frekuensi batas bawah

= Jumlah filter

Untuk menghitung *filterbank* dapat digunakan persamaan berikut [HUA01]:

(2.16)

Dimana,

= Nilai filterbank

= Indeks filter

= Indeks input sampel FFT (k = 0, 1, ... , N - 1)

= Nilai sampel FFT ke-m

Proses pemfilteran sinyal dilakukan untuk mendapatkan *log energy* pada tiap filter dengan menggunakan persamaan [HUA01]:

(2.17)

Dimana,

= Nilai log energy

= Nilai magnitude (Hz)

= Nilai filterbank

= Jumlah nilai FFT tiap frame

= Indeks filter

= Indeks input sampel FFT (k = 0, 1, ... , N - 1)

2.2.7 *Ceptrum* [HUA01]

Pada tahap terakhir ini, nilai mel akan dikonversikan kembali ke dalam domain waktu, yang hasilnya disebut *Mel Frequency Ceptral Coefficient.* Konversi ini dilakukan dengan menggunakan *Discrete Cosine Transform* (DCT). adalah nilai rata-rata dalam dB yang dapat digunakan untuk estimasi energi yang berasal dari filterbank. Koefisien DCT adalah nilai amplitudo dari spektrum yang dihasilkan. Perhitungan DCT dapat dilihat pada persamaan berikut:

(2.18)

Dimana,

= Nilai koefisien C (*cepstrum)* ke-i

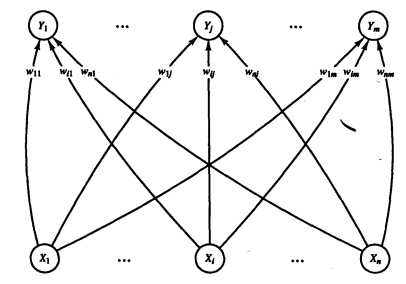
= Nilai log energy

= Jumlah filter

= Indeks filter , = Indeks koefisien DCT

2.2.8 Learning Vector Quantization [FAU94]

*Learning Vector Quantization* (LVQ) adalah satu metode untuk melakukan klasifikasi pola yang dimana setiap keluarannya merepresentasikan sebuah kelas atau kategori tertentu. LVQ merupakan salah satu algoritma dari *supervised neural network*.

LVQ terdiri dari lapiran *input*, lapisan kompetitif, dan lapisan *output*. Lapisan input adalah masukan data, lapisan kompetitif adalah pada saat terjadinya kompetisi pada *input* untuk masuk dalam suatu kelas berdasarkan kedekatan jaraknya. Dalam lapisan kompetitif, proses pembelajaran dilakukan secara terawasi. *Input* akan bersaing untuk dapat masuk ke dalam suatu kelas.

Gambar 2.7 Arsitektur LVQ

Dalam LVQ terdapat dua jenis proses, yaitu proses pembelajaran dan proses pengujian. Proses pembelajaran dilakukan melalui beberapa iterasi sampa batas iterasi maksimal tercapai. Algoritma dalam melakukan LVQ adalah sebagai berikut:

1. Inisialisasi vektor referensi; inisialisasi *learning rate* , dimana .
2. Selama kondisi berhenti adalah *false*, lakukan langkah 2-4.
3. untuk setiap *input* pelatihan vektor x, lakukan langkah 3 – 4.
4. Cari J hingga adalah nilai minimal.
5. Perbarui dimana kondisinya adalah:

Jika maka

(2.19)

Jika maka

(2.20)

1. Kurangi *learning rate*.
2. Uji kondisi berhenti, seperti kondisi yang mungkin menetapkan sebuah jumlah tetap dari iterasi.

Keterangan:

vektor pelatihan

Kategori atau kelas yang tepat untuk vektor pelatihan.

bobot vektor untuk *output* ke –

kategori atau kelas yang direpresentasikan dari unit *output* ke –

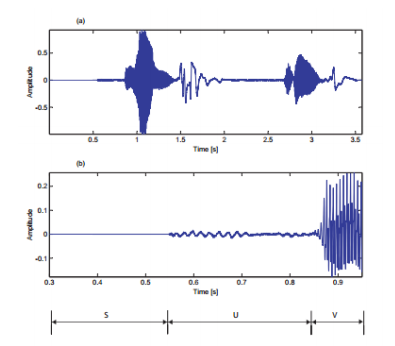
jarak Euclidean antara vektor *input* dan bobot vektor untuk *output* ke –

*Learning Rate,* .

2.3 Tinjauan Objek Penelitian

Berdasarkan penelitian yang dilakukan penulis, berikut merupakan objek yang dibahas dalam penelitian ini.

2.3.1 Suara

Suara adalah bunyi yang dikeluarkan dari mulut manusia, seperti pada waktu berbicara, menyanyi, tertawa, menangis, dan lain sebagainya [SET16]. Pada umumnya suara merambat melalui udara. Suara tidak dapat merambat melalui ruang hampa. Manusia pada umumnya dapat memproduksi suara dengan frekuensi 70Hz hingga 10.000 Hz sedangkan suara yang dapat didengar oleh manusia adalah suara yang memiliki frekuensi dari rentang 20 sampai 20.000 Hz [MAN11].

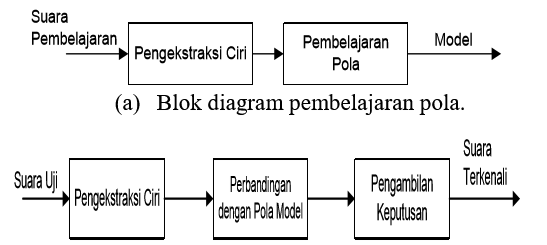
Gambar 2.8 Representasi Sinyal Suara

*Sumber: Speech Recognition using Hidden Markov Model: performance evaluation in noisy environtment*

2.3.2 Pengenalan Suara

Pengenalan suara adalah sebuah teknik yang memungkinkan sebuah sistem komputer untuk mengenali huruf, kata atau kalimat yang diucapkan. Dalam dunia komputer, pengenalan suara biasa dikenal sebagai *Automatic Speech* *Recognition* atau *Computer Speech Recognition*. Teknologi ini memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan suatu pola tertentu yang tersimpan dalam suatu perangkat. Secara umum pengenalan suara terdiri dari empat tahapan, yaitu [KSH12]:

1. **Ekstraksi Fitur**. Proses mengekstrak karakter suara dari data sinyal suara yang telah diubah menjadi data digital.
2. **Pembentukan Model Suara.** Proses pembelajaran yang membentuk model referensi agar sistem dapat mengenali pembicara. proses ini membutuhkan data berupa vektor-vektor fitur yang telah diekstraksi.
3. **Pencocokan Pola**. Proses pencocokan pola dengan menerima data ekstraksi fitur sebagai *input* dan model referensi yang sebelumnya didapatkan dan memberikan skor-skor kesesuaian dengan pola.
4. **Pembuatan Keputusan**. Proses pembuatan keputusan untuk hasil akhir, pada pengenalan pembicara tahapan ini menghasilkan identifikasi dan verifikasi. Proses ini akan menerima skor-skor hasil keseuaian dari pencocokan pola, dan akan menghasilkan identitas pembicara serta verifikasi pembicara.



Gambar 2.9 Tahapan Pengenalan Suara

2.3.3 Pengenalan Pembicara

Pengenalan pembicara secara umum terbagi menjadi dua bagian, yaitu identifikasi pembicara dan verifikasi pembicara. Identifikasi pembicara adalah proses untuk menentukan identitas pembicara sedangkan verifikasi secara otomatis berdasarkan karakteristik suara [KSH12]. Pengenalan pembicara memungkinkan sebuah sistem komputer untuk menerima *input* berupa suara dari seseorang lalu dapat mengenali atau mengidentifikasi orang yang menghasilkan suara tersebut.

Ada dua jenis pengenalan suara*,* yaitu [ZHI13]: *Text Independent Speaker Verification* (TI-SV)dan *Text Dependent Speaker Verification* (TD-SV). TD-SV merupakan pengenalan pembicara yang mengharuskan sumber pembicara untuk mengatakan kata-kata yang spesifik, seperti *password*. Sedangkan TI-SV tidak mengharuskan sumber pembicara untuk mengatakan kata-kata yang spesifik, melainkan pembicara dapat bebas mengatakan kata-kata apapun untuk melakukan pengenalan pembicara.

Pengenalan pembicara dapat dimanfaatkan untuk diaplikasikan dalam berbagai macam bidang, seperti [KSH12]:

1. **Untuk autentikasi**

Pengenalan pembicara untuk autentikasi memungkinkan pengguna menggunakan suara mereka untuk melakukan autentikasi identitas pengguna tersebut. Cara ini dianggap jauh lebih mudah dibanding dengan cara tradisional seperti menggunakan PIN atau *password*, karena pengguna harus mengingat PIN atau *password* tersebut.

1. **Untuk pengawasan**

Pengenalan pembicara memungkinkan pihak-pihak berwajib untuk melakukan pengawasan terhadap orang-orang tertentu yang dicurigai melalui suara mereka. Cara ini juga dapat digunakan untuk mendapatkan informasi data-data tentang orang tersebut.

1. **Untuk Identifikasi Forensik**

Rekaman suara dapat menjadi bukti-bukti untuk menghukum penjahat dengan adanya pengenalan suara.

**DAFTAR PUSTAKA**

[ANG11] Angga Setiawan, Achmad Hidayanto, R. Rizal Isnanto. 2011. “Aplikasi Pengenalan Ucapan dengan Ekstraksi Mel-Frequency Cepstrum Coefficients (MFCC) Melalui Jaringan Syaraf Tiruan (JST) Learning Vector Quantization (LVQ) untuk Mengoperasikan Kursor Komputer”, TRANSMISI, 13 (3), 2011, 82-86

[BHA13] Bhattacharjee, Utpal. Januari 2013. “A COMPARATIVE STUDY OF LPCC AND MFCC FEATURES FOR THE RECOGNITION OF ASSAMESE PHONEMES”, International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 1

[DAV13] Dave, Namrata. Juli 2013. “FEATURE EXTRACTION METHODS LPC, PLP AND MFCC IN SPEECH RECOGNITION”. International Journal For Advance Research In Engineering And Technology, Vol 1, Issue 4

[FAU94] Fausett, Laurene. 1994. “Fundamental of Neural Networks: Architectures, Algorithms, and Applications”.

[GEE14] Geeta Nijhawan, M.K Soni. Juli 2014. “SPEAKER RECOGNITION USING MFCC AND VECTOR QUANTISATION”. International Journal on Recent Trends in Engineering and Technology, Vol 11 No. 1

[HUA01] Huang, Xuedong, Alex Acero and Hsiao-Wuen Hon. 2001. Spoken Language Processing: A Guide To Theory, Algorithm And System Development. Prentice Hall, New Jersey.

[KSH12] Kshamamayee Dash, Debananda Padhi, Bhoomika Panda and Prof. Sanghamitra Mohanty. April 2012. “SPEAKER IDENTIFICATION USING MEL FREQUENCY CEPSTRAL COEFFICIENT AND BPNN”. International Journal of Advanced Research in Computer Science and Software Engineering, Vol 2, Issue 4

[LYO11] Lyons, Richard G. 2011. Understanding Digital Signal Processing 3rd Edition. Prentice Hall, Boston

[MAR03] Marinov, Stetoslav. Februari 2003. “Text Dependent and Text Independent Speaker Veriﬁcation Systems. Technology and Applications”.

[MAN11] Mandalia, Darshan and Gareta, Pravin. Mei 2011. “Speaker Recognition Using MFCC and Vector Quantization Model”. Department of Electrical Engineering Electronics & Communication Engineering Program Institute of Technology. NIRMA University.

[PEN15] Penghua LI, Shunxing ZHANG, Huizong FENG, Yuanyuan LI. 2015. “SPEAKER IDENTIFICATION USING SPECTROGRAM AND LEARNING VECTOR QUANTIZATION”. Journal of Computational Information Systems, Vol 11, Issue 9

[RYA11] Ryan J. Cassidy, 2008. “Auditory Filter Bank Lab”. [Center for Computer Research in Music and Acoustics](http://ccrma.stanford.edu/)m, CCRMA

[SET16] Setiawan, Ebta. 2016. Suara, Kamus Besar Bahasa Indonesia Online. http://kbbi.web.id/suara. Diakses Maret 2016

[ZHI13] Zhizheng Wu, Anthony Larcher, Kong Aik Lee**,** Eng Siong Chn, Tomi Kinnunen and Haizhou Li. Agustus 2013. “Vulnerability evaluation of speake rveriﬁcation under voice conversion spooﬁng: the effect of text constraints”. ISCA