

# IMPLEMENTASI ALGORITMA *HIDDEN MARKOV* MODEL SEBAGAI PENGENALAN PERINTAH SUARA PADA APLIKASI *WINAMP*

Dedy Abdullah<sup>1</sup>, Rizki Ramadhan<sup>2</sup>

<sup>1,2</sup>Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Bengkulu  
Jl. Bali PO BOX 118. Telp (0736) 227665, Fax (0736) 26161, Bengkulu 38119

<sup>1</sup>dedy\_abdullah@umb.ac.id

<sup>2</sup>rizk1.ramadh4n@gmail.com

**Abstrak:** Salah satu teknologi yang sedang berkembang saat ini adalah *speech recognition* yang mengolah suara menjadi suatu perintah yang dapat dikenal oleh suatu mesin sehingga memungkinkan seseorang melakukan *remote* pada aplikasi yang ada pada komputer dengan menggunakan suara yang masuk melalui *microphones*. Tujuan dari penelitian ini yaitu membuat *Media Player Winamp* dapat dioperasikan menggunakan suara dengan Algoritma *Hidden Markov Model*. Sinyal suara dicuplik sehingga menjadi sinyal digital kemudian dinormalisasi dan diproses dengan *preprocessing signal* menggunakan metode *LPC (Linear Predictive Coding)* selanjutnya dianalisis dan dicari nilai probabilitas yang maksimum sehingga bisa dikenali menggunakan *Hidden Markov Model (HMM)*. Hasil penelitian menunjukkan algoritma *Hidden Markov Model* dapat digunakan untuk mengeksekusi fungsi aplikasi *winamp* melalui perintah suara.

**Kata kunci:** *Speech Recognition*, Sinyal Suara, *Linear Predictive Coding*, *Hidden Markov Model*.

**Abstract:** *One technology that is being developed at this time is Speech recognition that processes sound into a command that can be recognized by a machine that allows a person to be remotely on existing applications on the computer by using an incoming voice via microphones. The purpose of this research is to make Winamp Media Player can be operated using voice with Hidden Markov Model algorithm. The voice signal is sampled into a digital signal then normalized and processed using the LPC (Linear Predictive Coding). It is analyzed and searched the probability that the maximum value that can be identified using a Hidden Markov Model (HMM). The results showed the algorithm Hidden Markov models can be used to execute the functions that exist in winamp application through voice commands.*

**Keywords:** *Speech Recognition, Linear Predictive Coding, Hidden Markov Model.*

## I. PENDAHULUAN

Semakin berkembangnya teknologi dewasa ini kita seakan dimanja dengan segala alat bantu

yang dapat mempermudah kita dalam melakukan suatu pekerjaan yang harus kita lakukan, karena kita tidak perlu mengeluarkan banyak tenaga untuk melakukan pekerjaan kita itu, misalnya untuk mengubah *channel* TV kesayangan kita-kita tidak perlu lagi menekan tombol yang ada pada televisi kita, cukup dengan menggunakan pengendali jarak jauh yang dinamakan "*Remote Control*". Dengan menggunakan *Remote Control* kita dapat mengubah satu *channel* ke *channel* lain dari tempat duduk kita sehingga akan menghemat tenaga dan waktu kita dibanding kita harus menekan tombol berulang kali pada televisi kita yang biasa jarak nya kurang lebih 3 meter dari tempat duduk kita.

Salah satu teknologi yang sedang berkembang saat ini adalah *speech recognition*.

*Speech recognition* adalah suatu teknologi yang dapat mengolah suara menjadi suatu perintah yang dapat dikenal oleh suatu mesin. Dimana dengan munculnya teknologi ini memungkinkan seseorang melakukan remote pada aplikasi yang ada pada komputer dengan menggunakan suara yang masuk melalui *microphones*.

Dengan membangun suatu aplikasi pengontrol yang menggunakan teknologi *speech recognition* akan membantu pengguna komputer dalam memberikan perintah pada komputer misalnya untuk mematikan komputer atau digunakan untuk menjalankan aplikasi yang diinginkan dengan menggunakan suara.

## II. LANDASAN TEORI

### A. Pengenalan Suara

Pengenalan suara atau *speech recognition* adalah proses otomatis mengenali kata-kata yang diucapkan kemudian mengubahnya ke dalam bentuk teks [1]. Dengan kata lain *speech recognition* merupakan suatu proses yang memungkinkan komputer untuk mengenali apa yang diucapkan oleh seseorang lalu mengubahnya ke bentuk tulisan.

*Input* data yang diterima berupa suara atau ucapan manusia, kemudian sistem akan mengidentifikasi kata atau kalimat yang diucapkan dan menghasilkan keluaran atau *output* berupa teks yang sesuai dengan apa yang diucapkan. *Input* yang berupa kata-kata tersebut diubah menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka lalu disesuaikan dengan kode-kode tertentu dan dicocokkan dengan suatu pola yang tersimpan dalam suatu perangkat. Untuk setiap ucapan yang berbeda akan dihasilkan pola ciri yang berbeda.

Dua jenis atau tipe *speech recognition* jika dilihat dari ketergantungan pembicara [1], antara

lain :

#### a. *Independent Speech Recognition*

Sistem pengenalan ucapan tanpa terpengaruh dengan siapa yang berbicara, tetapi mempunyai keterbatasan dalam jumlah kosakata. Model ini akan mencocokkan setiap ucapan dengan kata yang dikenali dan memilih yang "sepertinya" cocok. Untuk mendapatkan kecocokan kata yang diucapkan maka digunakan model statistic yang dikenal dengan nama *Hidden Markov Model* (HMM).

#### b. *Dependent Speech Recognition*

*Dependent Speech Recognition* yaitu sistem pengenalan ucapan yang memerlukan pelatihan khusus dari pembicara, di mana hasil pelatihan dari masing-masing pembicara akan disimpan dalam sebuah profil. Profil inilah yang nantinya digunakan untuk berinteraksi dengan sistem pengenalan ucapan dan sistem akan bergantung siapa yang berbicara. Sistem ini biasanya lebih mudah untuk dikembangkan, dimana contoh suara sudah dibuat sebelumnya dan disimpan dalam database atau basis data dan jumlah kosakatanya lebih besar dibandingkan dengan *independent speech recognition*. Proses pengenalan ucapan dengan cara membandingkan ucapan pembicara dengan contoh suara yang sudah ada.

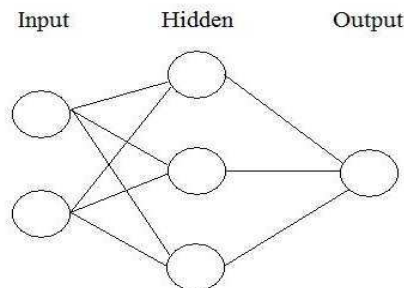
### B. *Hidden Markov Model*

*Hidden Markov Model* atau HMM adalah suatu proses stokastik ganda (ketidakpastian yang menggunakan data-data statistik) dari sebuah *system* yang diasumsikan sebuah proses *Markov* dengan salah satu parameter yang tidak dapat diobservasi (*hidden*). Proses yang tidak dapat diobservasi ini hanya dapat diobservasi melalui proses yang dapat diobservasi.

Teori dasar rantai *Markov* telah diketahui oleh seorang ahli Matematika dan Insinyur sekitar tahun 1980, tetapi hanya dalam dekade terkahir

yang telah diterapkan secara eksplisit untuk masalah dalam *speech processing*. Salah satu alasan utama mengapa *speech* model berdasarkan pada *Markov chains* atau rantai *Markov* belum dikembangkan sampai saat ini adalah kurangnya metode untuk mengoptimalkan parameter dari *Markov* model untuk menyesuaikan pola sinyal yang diamati. Perbaikan lanjutan dalam teori dan implementasi dari teknik *Markov modelling* telah dikembangkan dan ditingkatkan, sehingga telah sering digunakan atau diterapkan untuk masalah seperti *speech recognition* [2].

*Hidden Markov Model* menggabungkan dua atau lebih rantai *Markov* dengan hanya satu rantai yang terdiri dari state yang dapat diobservasi dan rantai lainnya membentuk *state* yang tidak dapat diobservasi (*hidden*), yang mempengaruhi hasil dari *state* yang dapat diobservasi.



Gambar 1. Ilustrasi HMM

Jika  $X = \{X_1, X_2, \dots, X_T\}$  adalah sebuah proses Markov, dan  $O = \{O_1, O_2, \dots, O_T\}$  adalah sebuah fungsi dari  $X$ , maka  $X$  adalah sebuah HMM yang dapat diobservasi melalui  $O$ , atau dapat ditulis  $O = f(X)$  untuk sebuah fungsi  $f$ . Parameter  $X$  menyatakan *state process* yang tersembunyi (*hidden*), sementara parameter  $O$  menyatakan *observation process* yang dapat diobservasi [2].

### C. Pengolahan Sinyal

*Preemphasis* adalah suatu proses produksi suara manusia, radiasi pada bibir dan lidah ketika proses *phonation* mengakibatkan komponen

frekuensi tingginya.

Hal ini dapat berpengaruh pada proses spectrum frekuensi sinyal menjadi merata antara frekuensi tinggi dilakukan pemfilteran dengan menggunakan *high pass filter*. Dimana *High pass filter* adalah jenis *filter* yang melewatkan *frekuensi* tinggi, tetapi mengurangi amplitudo *frekuensi* yang lebih rendah dari pada *frekuensi cutoff*.

Nilai-nilai pengurangan untuk *frekuensi* berbeda-beda untuk tiap-tiap *filter* ini. Terkadang *filter* ini disebut *low cut filter*, *bass cut filter* atau *rumble filter* yang juga sering digunakan dalam aplikasi audio. cuplikan sinyal dengan persamaan preemphasizer:

$$s(n) = s(n) - a s(n-1) \quad \dots (1)$$

dengan  $s(n)$  adalah sampel ke- $n$  dan harga  $a$  yang paling sering digunakan ialah 0.95.

*Blocking into frames* adalah suatu proses pada tahapan sinyal yang telah dipreemphasis, diblok menjadi beberapa bagian dengan jumlah sample  $N$ , dan tiap bagian dipisahkan dengan sejumlah  $M$  sample.

Membagi hasil *preemphasis*  $s(n)$  ke dalam *frame-frame* yang masing-masing memuat  $N$  buah sampel yang dipisahkan sejauh  $M$  buah *sample*. Semakin semakin baik perkiraan spektral LPC dari *frame* ke *frame*.

$$s(n) = M < N \quad \dots (2)$$

*Frame Windowing* adalah suatu proses pada setiap *window* pada bagian sinyal yang telah dibuat sebelumnya. Hal ini dilakukan untuk meminimalkan pada bagian awal dan akhir sinyal. Jika mendefinisikan sebuah *window*  $w(n)$  dan sinyal tiap bagian adalah  $x(n)$  maka sinyal hasil proses *windowing*.

Melakukan *windowing* terhadap setiap *frame*

yang telah dibentuk untuk meminimalkan diskontinuitas pada ujung awal dan ujung akhir setiap *frame* dengan persamaan *Hamming Window* untuk sampel ke-*n* adalah :

$$W(n) = 0.54 - 0.46 \cos(2\pi n / (N-1)), 0 \leq n \leq N-1 \dots (3)$$

*Auto Correlation Analisis* adalah proses tahapan yang setiap bagian yang telah diberi *window* kemudian akan dibentuk *autokorelasinya*. Analisis *autokorelasi* terhadap setiap *frame* hasil *windowing*  $x_1(n)$  dengan persamaan :

$$r_1(m) = \sum_{n=0}^{N-1-m} x_1(n)x_1(n+m) \dots (4)$$

Analisa LPC adalah suatu proses analisis sinyal dimana semua nilai autokorelasinya yang telah dihitung pada tahap sebelumnya akan diubah menjadi parameter LPC.

Mengubah  $p+1$  buah hasil autokorelasi pada masing-masing *frame* menjadi koefisien LPC  $a_m = a_m(p)$  untuk  $m = 1, 2, \dots, p$  dengan persamaan dibawah ini:

$$\begin{aligned} E(0) &= r(0) \\ k_m &= \{r(m) - \sum_{j=1}^{m-1} a_j(m-1)r(m-j)\} / E(m-1), \\ 1 &\leq m \leq p \\ a_m(m) &= k_m \\ a_j(m) &= a_j(m-1) - k_m a_{m-j}(m-1), 1 \leq j \leq m-1 \\ E(m) &= (1 - k_m^2) E(m-1) \dots (5) \end{aligned}$$

*koefisien cepstral* adalah suatu proses pengubahan parameter LPC yang sangat penting yang bisa diturunkan dari koefisien LPC adalah koefisien cepstral LPC,  $c(m)$ . Mengubah parameter LPC  $a_m$  ke *koefisien cepstral*  $c_m$  untuk mendapatkan kinerja yang lebih baik dan tahan terhadap *noise*, yaitu dengan persamaan:

$$\begin{aligned} c_m &= a_m + \sum_{k=1}^{m-1} (k/m) c_k a_{m-k}, 1 \leq m \leq p \\ c_m &= \sum_{k=1}^{m-1} (k/m) c_k a_{m-k}, m > p \dots (6) \end{aligned}$$

Setelah sinyal suara disampling dan dikonversi ke dalam bentuk sinyal digital

dilanjutkan dengan proses *pre-emphasis* yang bertujuan untuk meratakan spektral sinyal, menghilangkan nilai nilai tinggi pada spektrum sinyal sehingga memudahkan dalam menentukan batas batas ketelitian sinyal pada pemrosesan sinyal berikutnya., *frame blocking*. Pada tahap *frame blocking*, sinyal yang telah di *preemphasis*, dibagi menjadi beberapa *frame* waktu yang sangat singkat untuk memperoleh kondisi sinyal yang *quasi-stationer* dalam domain waktu dengan jumlah  $N$  sampel dan tiap bagian dipisahkan dengan sejumlah  $M$  sampel, dan *windowing* untuk menghaluskan *spektral* sinyal. Setiap sinyal dicirikan oleh bentuk spectral *frekuensinya* dan disimpan sebagai *codebook* untuk masing-masing jenis perintah untuk mendapatkan panjang sinyal, jumlah segmen dan ciri khas sinyal.

#### D. Konsep Perancangan Data Base

Basis data adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya [3].

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat untuk memanipulasinya. Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi dalam suatu organisasi. Adapun istilah yang digunakan dalam perancangan database adalah sebagai berikut:

- a. *Entity* adalah tempat, orang atau kejadian serta, konsep yang informasinya direkam.

- b. Atribut adalah elemen data yang menunjukkan suatu *entity*.
- c. *Data value* adalah aktual yang disimpan pada atribut.
- d. *Record* adalah kumpulan elemen yang saling berkaitan yang menginformasikan suatu *entity* secara lengkap.
- e. *File* adalah kumpulan *record* yang sejenis dimana panjang elemen sama, atribut yang sama dan data *value* yang berbeda, datanya.
- f. *Database* kumpulan data *file* yang membentuk suatu bangunan.

#### E. Konsep Perancangan Data Flow Diagram

*Data Flow Diagram* (DFD) merupakan diagram yang digunakan untuk menggambarkan proses-proses yang terjadi pada sistem yang akan dikembangkan. Dengan model ini, data-data yang terlibat pada masing-masing proses dapat diidentifikasi [4]. Untuk membaca suatu DFD kita harus memahami dulu, elemen-elemen yang menyusun suatu DFD. Ada empat elemen yang menyusun suatu DFD, yaitu:

##### a. Proses

Aktivitas atau fungsi yang dilakukan untuk alasan bisnis yang spesifik, biasa berupa manual maupun terkomputerisasi.

##### b. Data flow

Suatu data tunggal atau kumpulan logis suatu data, selalu diawali atau berakhir pada suatu proses.

##### c. Data Store

Kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam data *store*. Aliran data di-update atau ditambahkan ke data *store*.

##### d. External entity

Masing-masing elemen akan diberi lambang tertentu untuk membedakan satu dengan

yang lain. Adapun beberapa metode untuk menggambarkan elemen-elemen tersebut.

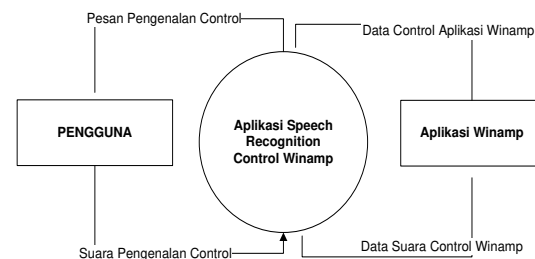
### III. METODOLOGI PENELITIAN

Jenis penelitian yang digunakan dalam penelitian ini adalah penelitian terapan. Teknik pengumpulan data pada penelitian terapan ini menggunakan teknik studi pustaka (*Library research*), yaitu dengan mempelajari konsep-konsep dasar mengenai *Hidden Markov Model* yang terdapat pada beberapa sumber literatur.

### IV. ANALISA DAN PERANCANGAN

#### A. Diagram Konteks

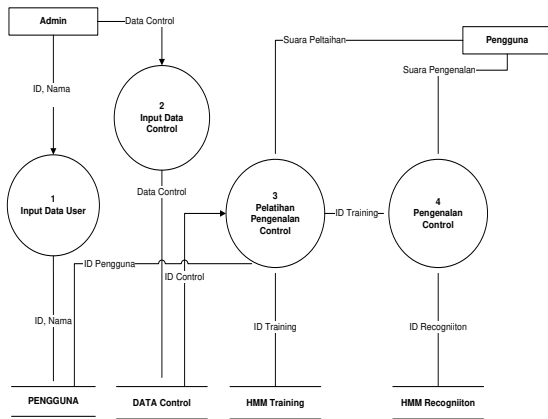
Diagram Konteks adalah suatu diagram alir yang tingkat tinggi yang menggambarkan seluruh jaringan, masukan dan keluaran. Adapun bentuk dari diagram konteks dapat dilihat pada gambar 2.



Gambar 2. Diagram Konteks

#### B. Diagram level 0

Diagram data level 0 merupakan gambaran sistem secara garis besar yang menerangkan hubungan antara *entity* dengan proses data dan pembuatan *file-file* yang menghasilkan informasi yang diinginkan. Adapun bentuk dari diagram data level 0 dapat dilihat pada gambar 3.



Gambar 3. Diagram Level 0

C. Perancangan Tabel

1. Tabel pengguna

Digunakan untuk menyimpan pengguna yang akan menginputkan suara pada penelitian ini.

Tabel 1. Tabel Pengguna

No	Field	Type	Size	Ket
1	User_ID	Varchar	5	ID Pengguna
2	Username	Varchar	20	Nama Pengguna

2. Tabel Data Perintah

Digunakan untuk menyesuaikan perintah yang ada di winamp dengan suara.

Tabel 2. Tabel Data Perintah

No	Field	Type	Size	Ket
1	W_ID	Varchar	5	ID Kata
2	W_Content	Varchar	5	Kata

3. Tabel Data HMM Training

Digunakan untuk menyimpan hasil pelatihan yang dilakukan user untuk mendeksi suara yang akan digunakan untuk pengoperasian winamp.

Tabel 2. Tabel Data HMM Training

No	Field	Type	Size	Ket
1	User_ID	Varchar	5	ID Pengguna

2	W_ID	Varchar	5	ID Daftar Kata
3	HT_ID	Auto	-	ID Training
4	HT_Traindata	Memo		Data Training

4. Tabel data HMM

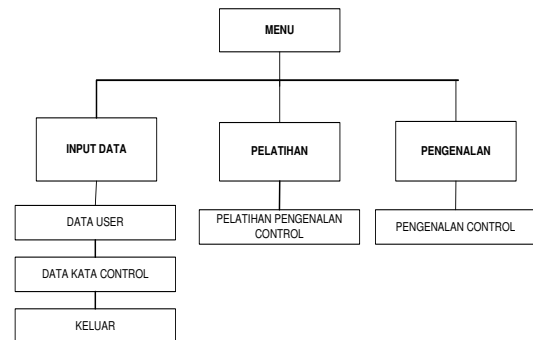
Merupakan untuk menyimpan data hasil dari training oleh pengguna yang akan di ujicobakan untuk mengoperasikan program.

Tabel 3. Tabel HMM

No	Field	Type	Size	Ket
1	User_ID	Varchar	5	ID Pengguna
2	W_ID	Varchar	25	ID Kata
3	HMM_Model	Memo	-	Data HMM Model

D. Perancangan Struktur Menu

Struktur menu yang akan digunakan sebagai interface antara pengguna dan aplikasi dapat dilihat pada gambar 4 berikut.



Gambar 4. Struktur Menu

V. HASIL DAN PEMBAHASAN

A. Hasil Dan Pembahasan

1. Tampilan Menu Utama

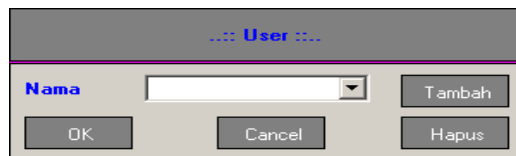
Menu utama muncul saat sistem dijalankan. Terdapat menu Perintah, Pengenalan, Pelatihan, bila user belum terdaftar di wajibkan untuk masuk ke menu user, untuk menambah user baru, setelah itu baru pengguna masuk ke menu pelatihan.



Gambar 5. Menu utama

## 2. Tampilan Tambah User

Menu tambah digunakan bila nama *user* belum terdaftar dalam menu pilihan nama, bila nama *user* telah terdaftar, *user* hanya perlu memilih nama dan meilih tombol OK untuk melanjutkan ke proses berikutnya.



Gambar 6. Menu Pengguna

## 3. Tampilan Menu Daftar Kata

Untuk memberikan perintah agar dapat mengontrol aplikasi *winamp* *user* wajib melakukan pelatihan dan pengenalan suara sesuai daftar kata yang ada, seperti pada gambar berikut.



Gambar 7. Menu Daftar Kata

## 4. Tampilan Menu Pelatihan

Setelah memilih daftar kata yang akan dilatih *user* harus memasukkan suara agar sistem dapat memproses sinyal suara untuk mewakili daftar kata yang telah dipilih, tidak jarang *user* harus memasukkan suara berulang kali karena suara yang diterima terlalu pendek karena saat *user* mengeluarkan suara rentang waktu penerimaan suara sudah melewati batas waktu perakaman yang sudah ditentukan yaitu 1,5 detik. *User* harus memasukkan suara berulang kali bisa juga disebabkan sistem menerima suara lain selain dari mulut pengguna misalnya suara *noise* atau suara yang tidak diinginkan seperti suara kendaraan atau suara binatang yang ikut masuk saat *user* melakukan pelatihan, bila suara *noise* tersebut mengganggu suara *user* yang akan disimpan, *User* harus membatalkan proses penyimpanan dan mengulang proses pelatihan dan *sesitifitas microphone* diatur melalui *audio recording* yang ada pada sistem operasi.

Bila sistem menerima masukkan suara maka sistem akan menghitung panjang sinyal, panjang sinyal didapat dari perhitungan panjang sinyal saat perekaman dikurang lama diam sinyal yaitu ukuran pajang sinyal yang terlalu kecil, jumlah segmen dan ciri khas sinyal adalah representasi dari sejumlah himpunan parameter yang didapat dari proses *Linier Predictive Coding* yaitu proses ekstraksi untuk mendapatkan sederetan besaran nilai pada bagian sinyal masukan yang didapat dari proses fungsi *encode* dan *decode* sinyal menggunakan *codebook* yang berukuran 128 bit.

Proses pengambilan sinyal ucapan menggunakan fungsi *recording* dari *windows* yaitu *wavein* dan *waveout* yang berformat *wave* dengan *channel mono* dan *sampel data 11khz per detik*.



Gambar 8. Menu Pelatihan

Bila sinyal ucapan masuk melalui *micropophone* suara akan ditangkap dengan fungsi *record wavein* dari fungsi yang ada pada sistem informasi *windows*, sinyal ucapan akan diekstrak menggunakan metode ekstraksi ciri *Linear Predictive Coding* (LPC). LPC salah satu metode analisis sinyal suara yang menyatakan ciri-ciri penting dari sinyal suara tersebut dalam bentuk koefisien-koefisien LPC. yang mana hasil dari nilai koefisien dari LPC akan dilakukan pemodelan menggunakan metode *Hidden Marcov Model* untuk menetapkan pola *Hidden marcov Model* dan untuk mendapatkan parameter-parameter *Hidden Marcov Model*, *Hidden Marcov Model* yaitu merupakan salah satu bentuk pemodelan suara dimana sinyal suara dianalisis dan dicari nilai probabilitas yang maksimum sehingga bisa dikenali, dari hasil pemodelan tersebut akan didapatkan parameter yang selanjutnya digunakan dalam proses pengenalan kata, parameter tersebut disimpan kedalam database *HMM\_Training*, contoh tampilan apabila komputer dapat menerima sinyal masukan untuk Kata *ON* terlihat pada gambar 9.

*Preemphasis* adalah suatu proses produksi suara manusia, radiasi pada bibir dan lidah ketika proses *phonation* mengakibatkan komponen frekuensi tingginya.



Gambar 9. Menu Pengenalan Perintah Suara

Hal ini dapat berpengaruh pada proses spectrum *frekuensi* sinyal menjadi merata antara frekuensi tinggi dilakukan pemfilteran dengan menggunakan *high pass filter*. Dimana *High pass filter* adalah jenis *filter* yang melewatkan frekuensi tinggi, tetapi mengurangi *amplitudo frekuensi* yang lebih rendah dari pada *frekuensi cut off*.

Nilai-nilai pengurangan untuk *frekuensi* berbeda-beda untuk tiap-tiap *filter* ini. Terkadang *filter* ini disebut *low cut filter*, *bass cut filter* atau *rumble filter* yang juga sering digunakan dalam aplikasi *audio*. cuplikan sinyal dengan persamaan *preemphasizer*

$$s(n) = s(n) - a s(n-1) \dots (7)$$

dengan  $s(n)$  adalah sampel ke- $n$  dan harga  $a$  yang paling sering digunakan ialah 0.95.

*Blocking into frames* adalah suatu proses pada tahapan sinyal yang telah di *preemphasis*, *diblok* menjadi beberapa bagian dengan jumlah sample  $N$ , dan tiap bagian dipisahkan dengan sejumlah  $M$  sample.

Membagi hasil *preemphasis*  $s(n)$  ke dalam frame-frame yang masing-masing memuat  $N$  buah sampel yang dipisahkan sejauh  $M$  buah sample. Semakin semakin baik perkiraan spektral LPC dari *frame ke frame*.

$$s(n) = M < N \dots (8)$$



*Frame Windowing* adalah suatu proses pada setiap *window* pada bagian sinyal yang telah dibuat sebelumnya. Hal ini dilakukan untuk meminimalkan pada bagian awal dan akhir sinyal. Jika mendefinisikan sebuah *window*  $w(n)$  dan sinyal tiap bagian adalah  $x(n)$  maka sinyal hasil proses *windowing*.

Melakukan *windowing* terhadap setiap *frame* yang telah dibentuk untuk meminimalkan diskontinuitas pada ujung awal dan ujung akhir setiap *frame* dengan persamaan *Hamming Window* untuk sampel ke- $n$  adalah:

$$W(n) = 0.54 - 0.46 \cos(2\pi n / N - 1), 0 \leq n \leq N - 1 \quad \dots (9)$$

*Auto Correlation Analysis* adalah proses tahapan yang setiap bagian yang telah diberi *window* kemudian akan dibentuk *autokorelasinya*. Analisis *autokorelasi* terhadap setiap *frame* hasil *windowing*  $x_1(n)$  dengan persamaan :

$$r_1(m) = \sum_{n=0}^{N-1-m} x_1(n)x_1(n+m) \quad \dots (10)$$

Analisa LPC adalah suatu proses analisis sinyal dimana semua nilai *autokorelasinya* yang telah dihitung pada tahap sebelumnya akan diubah menjadi parameter LPC.

Mengubah  $p + 1$  buah hasil *autokorelasi* pada masing-masing *frame* menjadi koefisien LPC  $a_m = a_m(p)$  untuk  $m = 1, 2, \dots, p$  dengan persamaan dibawah ini:

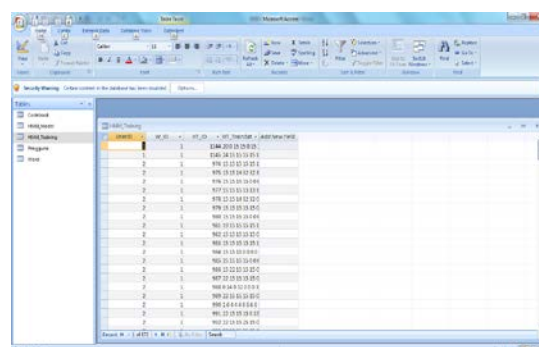
$$\begin{aligned} E(0) &= r(0) \\ k_m &= \{r(m) - \sum_{j=1}^{m-1} \alpha_j(m-1)r(|m-j|) / E(m-1), 1 \leq m \leq p \\ \alpha_m(m) &= k_m \\ \alpha_j(m) &= \alpha_j(m-1) - k_m \alpha_{m-j}(m-1), 1 \leq j \leq m-1 \\ E(m) &= (1 - k_m^2) E(m-1) \quad \dots (11) \end{aligned}$$

Pengubahan parameter LPC menjadi koefisien *cepstral* adalah suatu proses pengubahan parameter LPC yang sangat penting yang bisa diturunkan dari koefisien LPC adalah

koefisien cepstral LPC,  $c(m)$ . Mengubah parameter LPC am ke koefisien cepstral cm untuk mendapatkan kinerja yang lebih baik dan tahan terhadap *noise*, yaitu dengan persamaan:

$$\begin{aligned} c_m &= a_m + \sum_{k=1}^{m-1} (k/m) c_{m-k}, 1 \leq m \leq p \\ c_m &= \sum_{k=1}^{m-1} (k/m) c_{m-k}, m > p \quad \dots (12) \end{aligned}$$

Setelah sinyal suara disampling dan dikonversi ke dalam bentuk sinyal digital dilanjutkan dengan proses *pre-emphasis* yang bertujuan bertujuan untuk meratakan spektral sinyal, menghilangkan nilai nilai tinggi pada *spektrum* sinyal sehingga memudahkan dalam menentukan batas batas ketelitian sinyal pada pemrosesan sinyal berikutnya. Pada tahap *frame blocking*, sinyal yang telah di *preemphasis*, dibagi menjadi beberapa *frame* waktu yang sangat singkat untuk memperoleh kondisi sinyal yang *quasi-stationer* dalam domain waktu dengan jumlah  $N$  sampel dan tiap bagian dipisahkan dengan sejumlah  $M$  sampel, dan *windowing* untuk menghaluskan spektral sinyal, Setiap sinyal dicirikan oleh bentuk *spektral frekuensinya* dan disimpan sebagai *code book* untuk masing-masing jenis perintah untuk mendapatkan panjang sinyal, jumlah *segmen* dan ciri khas sinyal.

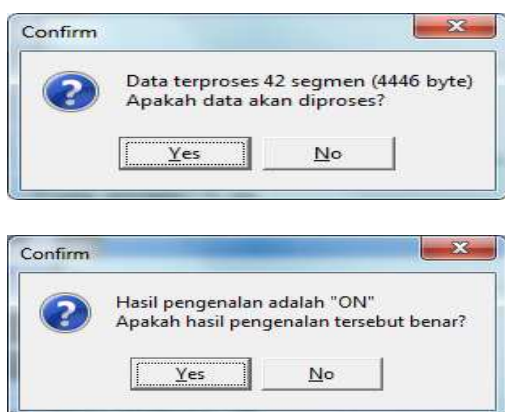


Gambar 10. Gambar Database HMM Training

### 5. Tampilan Menu Pengenalan

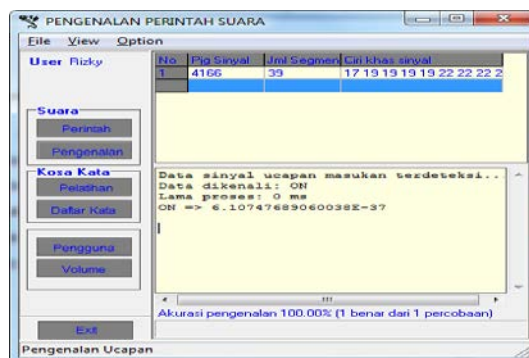
Setelah semua daftar kata telah melalui proses pelatihan, selanjutnya *user* wajib

melakukan proses pengenalan yang mana akan menentukan hasil dari proses pelatihan, pada proses pengenalan suara *user* yang telah melalui proses pelatihan akan di uji apakah telah sesuai mengenal daftar kata yang telah dilatih, *user* harus memilih *Yes* apabila kata yang dikenali telah sesuai atau *No* apabila kata yang dikenali belum sesuai, pada proses ini sistem akan membandingkan kembali pengenalan kata pelatihan yang telah tersimpan dengan suara masukan baru dimana sistem berinteraksi dengan *user* untuk mendapatkan hasil pengenalan yang lebih tepat, semakin sering *user* melakukan percobaan pengenalan tingkat akurasi akan semakin baik karena semakin banyak pola pengenalan suara yang tersimpan akan memudahkan tingkat pengenalan suara.

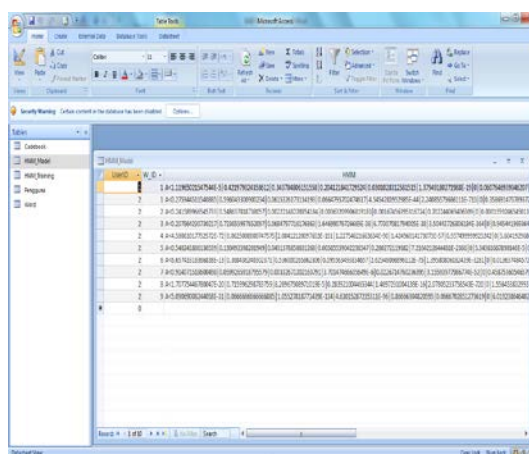


Gambar 11. Gambar Proses Pengenalan

Pada gambar diatas terlihat sistem memproses signal masukan dari *microphone* dan sinyal dikenal untuk kata “ON”, pada proses ini sistem memerlukan interaksi *user* untuk memastikan bahwa pengenalan tersebut benar, hasil pengenalan akan disimpan ke dalam *database* HMM Model.



Gambar 12. Gambar pengenalan.



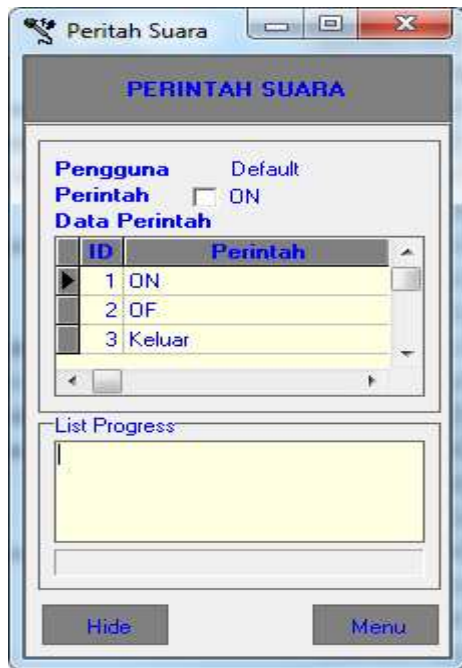
Gambar 13. Gambar Database HMM Model

## B. Pengujian Sistem

Tahap pengujian adalah mencoba kemampuan program dalam memproses sinyal suara untuk mengeksekusi program.

### 1. Proses Perintah Suara

Berikut tampilan menu untuk melakukan perintah suara untuk mengontrol aplikasi winamp, perintah yang pertama kali diberikan adalah perintah *ON*, bila perintah *ON* dikenali menu *ON* akan ter ceklis yang berarti sistem siap menerima perintah *winamp* untuk mengeksekusi atau mengotrol aplikasi winamp, proses ini dilakukan agar sistem tidak selalu memproses suara masukan diluar kata - kata yang telah dilatih sebelumnya.



Gambar 14. Menu Proses Perintah Suara Belum Terpilih



Gambar 15. Menu Proses Perintah Suara Sudah Terpilih

Pada proses ini sistem telah mengenali suara *user* untuk mewakili kata - kata yang telah dipilih pada daftar kata, dan Daftar kata perintah tersebut juga mewakili perintah *eksekusi* program dan *even even keyboard*, seperti untuk memanggil aplikasi winamp pada daftar kata Winamp diberikan perintah: `winexec('C:\ProgramFiles\Winamp\Winamp.exe',SW_NORMAL)`; setelah aplikasi winamp berjalan maka daftar kata perintah yang lain seperti *Play* mewakili *keyboard even* seperti `keybd_event(Ord('x'), 0, 0, 0)`; yang berarti *user* menekan huruf "X" untuk mengeksekusi menu *Play*, Kata *Pause* mewakili *keyboard even*

`keybd_event(Ord('c'), 0, 0, 0)`; yang berarti *user* menekan huruf "C" , Kata *Stop* mewakili *keyboard even* `keybd_event(Ord('v'), 0, 0, 0)`; yang berarti *user* menekan huruf "V" Kata *Next* mewakili *keyboard even* `keybd_event(Ord('z'), 0, 0, 0)`; yang berarti *user* menekan huruf "Z" yang berarti *user* menekan huruf "V" Kata *Lanjut* mewakili *keyboard even* `keybd_event(Ord('b'), 0, 0, 0)`; yang berarti *user* menekan huruf "B" *keyboard even* tersebut harus sesuai dengan *even Global hot key* yang ada pada Aplikasi Winamp yang terinstal pada komputer *user*.

## VI. KESIMPULAN

Dari hasil penelitian dapat dihasilkan kesimpulan bahwa algoritma *Hidden Markov Model* dapat digunakan untuk mengeksekusi fungsi aplikasi winamp melalui perintah suara.

## REFERENSI

- [1] Mains, Ron. *White Paper On Speech Recognition In The SESA Call Center* University of Maryland. 2001.
- [2] Handel, Ramon van. "Hidden Markov Models" Lecture Notes. 2008.
- [3] Kusumadewi, Sri; Purnomo, Hari. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Graha Ilmu. Yogyakarta. 2010.
- [4] Al Fatta, Hanif. *Analisis dan Perancangan Sistem Informasi*. Yogyakarta. Andi Offset. 2007.