

# RANCANG BANGUN KAMUS VISUAL BAHASA INGGRIS TENTANG PETERNAKAN BERBASIS FLASH DENGAN MENGUNAKAN VOICE RECOGNITION

Moch. Agus  
Eko Fachtur Rochman

Teknik Informatika, Universitas Kanjuruhan Malang, augustus.synystrosaurus@gmail.com  
Teknik Informatika, Universitas Kanjuruhan Malang, eko.fachtur@gmail.com

## ABSTRACT

*In 2015, Indonesia will face the ASEAN free trade. One of the potential business sector is livestock. However, 95% of Indonesian farmers who are independent farmers are still don't know about English. According to that problem we need learning support tools such as visual dictionary application. Flash is a program which is currently used to study media application. This dictionary application which integrated with Flash can train pronunciation of the user and measure the rightness of what user said according to the samples using voice recognition. The method used for voice recognition is linear predictive coding to extract characteristics, and Hidden Markov Model for modeling and voice recognition. From the test results, the average of recognition reached 84%.*

**Key word** : Voice Recognition, LPC, HMM, Dictionary.

## ABSTRAK

Pada tahun 2015 Indonesia akan menghadapi pasar bebas ASEAN. Salah satu sektor usaha yang berpotensi adalah peternakan. Namun peternak Indonesia yang 95% peternak rakyat ini masih buta dengan bahasa Inggris. Maka dari itu diperlukan alat penunjang pembelajaran seperti aplikasi kamus visual. Flash merupakan program khusus yang dewasa ini dipakai untuk aplikasi media pembelajaran.

Aplikasi kamus yang terintegrasi Flash ini dapat melatih pengucapan dan memberi nilai seberapa benar pengucapan pengguna dengan menggunakan Voice Recognition. Metode yang dipakai untuk pengenalan suara adalah Linear Predictive Coding untuk mengekstraksi ciri, dan Hidden Markov Model untuk pemodelan dan pengenalan suara. Dari hasil pengujian rata-rata pengenalan mencapai 84%.

Kata kunci : Pengenalan ucapan, LPC, HMM, Kamus.

## 1. Pendahuluan

*Voice Recognition* adalah suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan. Teknologi ini memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan pola tertentu yang tersimpan dalam suatu perangkat. Salah satu pengaplikasian *voice recognition* ini adalah aplikasi kamus yang bisa melatih pengucapan pengguna.

Flash merupakan program khusus untuk membuat animasi dan User Interface yang atraktif. Flash sering dipakai dalam pembangunan website yang interaktif dan dinamis. Dewasa ini, Flash digunakan untuk media pembelajaran seperti CD Interaktif. Maka dari itu, peneliti beranggapan untuk membuat program ini dengan Flash dan akhirnya terhubung dengan *Voice Recognition*. Namun Flash sendiri mempunyai keterbatasan, yaitu Flash tidak bisa memanipulasi sinyal suara yang nantinya akan dipakai dalam proses pengenalan. Untuk bisa mengatasi hal ini,

diperlukan program yang bisa memanipulasi sinyal suara yaitu MATLAB.

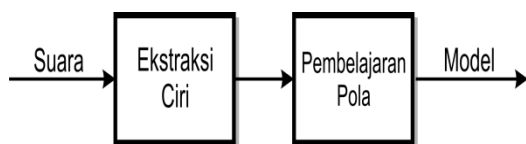
Tujuan dari tugas akhir ini adalah untuk merancang dan membangun aplikasi Kamus Visual Bahasa Inggris berbasis Flash tentang peternakan yang terintegrasi dengan *Voice Recognition* sebagai sarana penunjang pembelajaran untuk peternak sekaligus bisa mengerti dan melatih pengucapan bahasa Inggris dengan baik dan benar.

## 2. Tinjauan Pustaka

### 2.1 Pengenalan Suara Digital

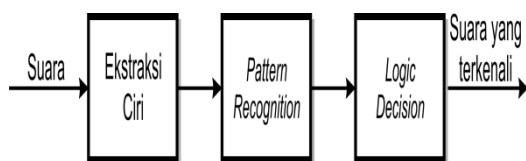
Pengenalan suara merupakan salah satu upaya agar suara dapat dikenali atau diidentifikasi sehingga dapat dimanfaatkan. Pengenalan suara dapat dibedakan ke dalam tiga bentuk pendekatan, yaitu pendekatan akustik-fonetik (*the acoustic-phonetic approach*), pendekatan kecerdasan buatan (*the artificial intelligence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

Pendekatan pengenalan pola terdiri dari dua langkah yaitu pembelajaran pola suara dan pengenalan pola melalui perbandingan pola. Tahap perbandingan pola adalah tahap bagi ucapan yang akan dikenali, dibandingkan polanya dengan setiap kemungkinan pola yang telah dipelajari dalam fase pembelajaran, untuk kemudian diklasifikasi dengan pola terbaik yang cocok. Blok diagram pembelajaran pola dapat dilihat pada gambar 1.



Gambar 1 Blok diagram pembelajaran pola

Sedangkan blok diagram pengenalan kata dapat dilihat pada blok diagram 2.



Gambar 2 Blok diagram pengenalan kata

Berikut ini merupakan penjelasan dari masing-masing blok:

#### 1. Pengekstraksi Ciri.

Merupakan proses mendapatkan sederetan besaran pada bagian sinyal masukan untuk menetapkan pola pembelajaran atau pola uji.

#### 2. Pembelajaran Pola

Satu atau lebih pola pembelajaran yang berhubungan dengan bunyi suara dari kelas yang sama, digunakan untuk membuat pola representatif dari ciri-ciri kelas tersebut. Hasilnya yang biasa disebut dengan pola referensi.

#### 3. Perbandingan dengan Pola Model

Pola uji yang akan dikenali, dibandingkan dengan setiap kelas pola referensi. Kesamaan besaran antara pola uji dengan setiap pola referensi akan dihitung.

#### 4. Logic Decision

Menentukan kelas pola referensi mana yang paling cocok untuk pola uji berdasarkan klasifikasi pola.

Pengenalan suara secara umum dapat dibagi menjadi tiga tahap, yaitu : ekstraksi ciri, pemodelan, dan pengenalan(Hapsari J.P. 2007).

### 2.2 Analisis LPC (*Linear Predictive Coding*)

Secara garis besarnya, prosesor model LPC dibagi kedalam 8 bagian seperti diperlihatkan pada gambar 3.

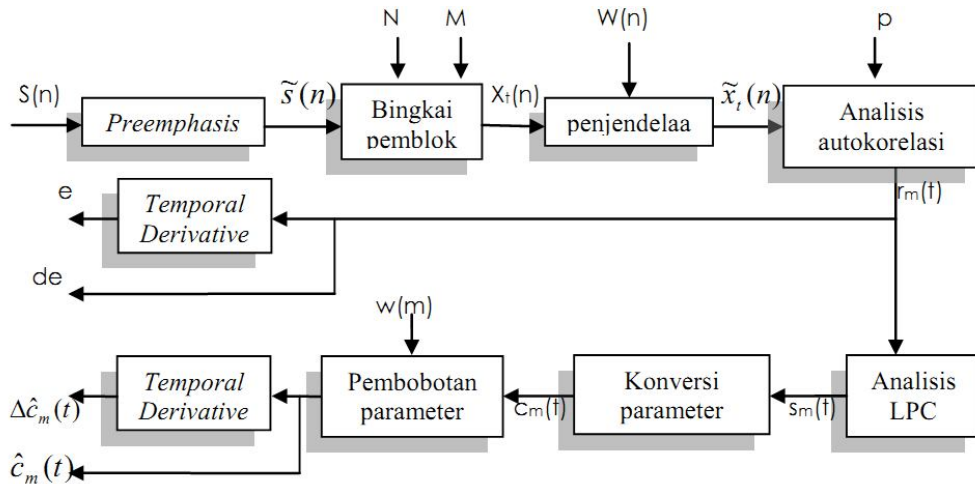
Bagian-bagian tersebut adalah seperti berikut:

#### a. Pre-emphasis

Preemphasis; sinyal suara digital  $s(n)$ , dimasukkan kedalam sistem digital orde rendah (biasanya berupa filter FIR orde satu) untuk mereratakan spektrum sinyal.

#### b. Bingkai Pemblok

Sinyal suara hasil *pre-emphasis* diblok kedalam beberapa bingkai N sampel suara, dengan jarak antara bingkai yang berdekatan dipisahkan oleh M sampel untuk membentuk *overlap* sinyal, sehingga hasil estimasi spektral LPC akan berkorelasi dari bingkai ke bingkai.



Gambar 3 Diagram blok ekstraksi ciri dengan analisis LPC

**c. Penjendelaan**

setiap frame kemudian dijendelakan (proses *windowing*) untuk meminimalkan diskontinuitas sinyal pada awal dan akhir bingkai. Jendela yang biasa digunakan untuk metode autokorelasi LPC adalah jendela *Hamming* dengan bentuk:

$$w(n) = 0,54 - 0,46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1, \dots (1)$$

**d. Analisis Autokorelasi**

Setiap bingkai sinyal yang telah dijendelakan kemudian dianalisis autokorelasinya untuk memberikan:

$$r_l(m) = \sum \tilde{x}_l(n) \tilde{x}_l(n+m), \quad m = 0, 1, \dots, p, \dots (2)$$

dengan nilai autokorelasi tertinggi, p, adalah orde analisis LPC. Autokorelasi ke-nol  $R_l(0)$  merupakan energi bingkai yang ke-l.

**e. Analisis LPC**

Pada tahap ini setiap bingkai dengan autokorelasi ke-(p+1) akan dikonversi ke bentuk parameter-parameter LPC yang dapat berupa koefisien LPC, koefisien refleksi, *cepstral coefficients* atau transformasi yang lainnya sesuai kebutuhan.

**f. Konversi Parameter LPC menjadi Koefisien Cepstral**

Rangkaian parameter yang sangat penting, yang dapat diturunkan secara langsung dari rangkaian koefisien LPC adalah koefisien cepstral,  $c(m)$ , yang

ditentukan secara rekursif sebagai berikut:

$$c_0 = \ln \sigma^2 \dots \dots \dots (3)$$

$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad 1 \leq m \leq p \dots \dots (4)$$

$$c_m = \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad m > p, \dots (5)$$

dengan  $\sigma^2$  adalah gain dalam model LPC.

**g. Pembobotan Parameter**

Karena *cepstral coefficients* orde rendah sensitif terhadap kemiringan spektrum dan *cepstral coefficients* orde tinggi sensitif terhadap derau, maka dilakukan pembobotan *cepstral coefficient* dengan jendela penyadap sehingga meminimalkan sensitivitas tersebut.

**h. Turunan Cepstral Terhadap Waktu**

Representasi *cepstral* dari spektrum suara memberikan representasi yang bagus atas sinyal spektrum lokal untuk analisis bingkai. Peningkatan representasi dapat diperoleh melalui aproksimasi  $\partial c_m(t) / \partial t$  dengan polinomial ortogonal pada jendela dengan panjang berhingga, yaitu :

$$\frac{\partial c_m(t)}{\partial t} = \Delta c_m(t) \approx \mu \sum_{i=-K}^K k c_m(t+k), \dots \dots (6)$$

dengan  $\mu$  adalah konstanta normalisasi yang tepat dan  $(2K + 1)$  adalah jumlah bingkai pada waktu komputasi dikerjakan. Untuk setiap bingkai t, hasil

analisis LPC adalah vektor koefisien cepstral terbobot Q, yaitu :

$$o'_t = (\hat{c}_1(t), \hat{c}_2(t), \dots, \hat{c}_Q(t), \Delta c_1(t), \Delta c_2(t), \dots, \Delta c_Q(t), \dots) \dots \dots \dots (7)$$

dengan  $o_t$  adalah komponen-komponen vektor  $2Q$  dan menunjukkan matriks transpose. Hasil analisis prosesor LPC berupa besaran parameter-parameter koefisien cepstral ( $\hat{c}_m(t)$ ), turunan koefisien cepstral ( $\Delta \hat{c}_m(t)$ ), besaran energi bingkai (e), dan besaran turunan energi frame (de) akan menjadירuntun observasi bagi HMM, dituliskan dalam bentuk:

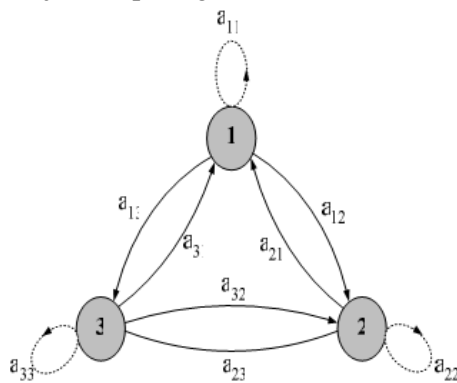
$$o = [\hat{c}_m(t) \ \Delta \hat{c}_m(t) \ e \ de] \dots \dots \dots (8)$$

(Subito 2010).

### 2.3 Pemodelan dengan HMM (Hidden Markov Model)

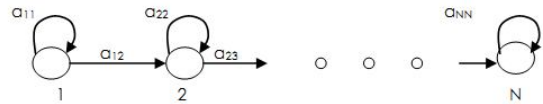
Algoritma HMM didasari oleh model matematik yang dikenal dengan rantai Markov. Beberapa hal yang dapat dijelaskan tentang rantai Markov yaitu:

1. Transisi keadaan dari suatu keadaan tergantung pada keadaan sebelumnya.  $P[q_t = j | q_{t-1} = i, q_{t-2} = k, \dots] = P[q_t = j | q_{t-1} = i]$
  2. Transisi keadaan bebas terhadap waktu.  $a_{ij} = P[q_t = j | q_{t-1} = i]$
- Rantai Markov secara umum ditunjukkan pada gambar 4



Gambar 4 Rantai Markov

Sedangkan bentuk dari HMM yang sering dipakai untuk pengenalan suara yaitu HMM kiri-kanan dapat dilihat pada gambar 5



Gambar 5 HMM kiri-kanan dengan N cacah keadaan

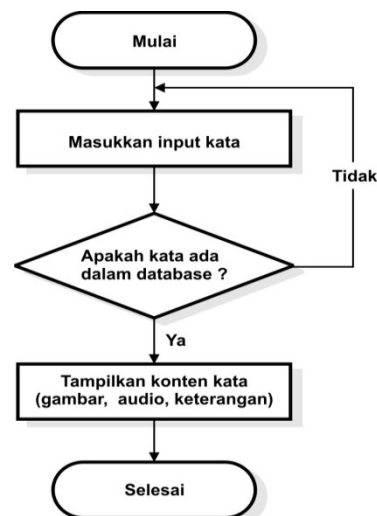
Elemen yang terdapat pada HMM yaitu :

1. Jumlah *state* dalam model (N).
  2. Jumlah simbol observasi yang berbeda tiap *state* (M).
  3. Distribusi *state* transisi  $A = \{a_{ij}\}$  dengan  $a_{ij} = P[q_{t+1} = j | q_t = i], 1 \leq i, j \leq N$
  4. Distribusi probabilitas simbol observasi,  $B = \{b_j(k)\}$  dengan  $b_j(k) = P(o_t = v_k | q_t = j), 1 \leq j \leq N$
  5. Distribusi *state* awal  $\pi = \{\pi_i\}$   $\pi_i = P[q_1 = i], 1 \leq i \leq N$
- (Hapsari J.P. 2007).

### 3. Pembahasan

#### 3.1 Perancangan Sistem

Dalam perancangan sistem digunakan dua bahasa pemrograman yaitu Matlab dan Action Script 3. Matlab berperan dalam proses pengenalan ucapan. Sedangkan Action Script 3 pada Flash, berperan untuk menampilkan gambar dan suara serta pengaksesan *database*. Alir utama jalannya program dapat dilihat pada Gambar 6.



Gambar 6 Diagram jalannya program kamus

Proses diawali dengan *user* menginputkan kata (berbahasa Indonesia) yang akan dicari bahasa Inggrisnya pada *menu* utama. Flash akan melakukan *search* pada *database*. *Form* pada Flash akan menunjukkan konten berupa gambar, maupun *audio* bagaimana mengucapkan *pronunciation* kata tersebut dengan benar. Dan kemudian *user* akan menginputkan suaranya melalui *microphone*, untuk mengetahui seberapa besar pengucapannya.

### 3.1.1 Inter-Process Communication

Agar Flash dan Matlab dapat terhubung diperlukan mekanisme agar dua aplikasi ini bisa berkomunikasi. Dalam perancangan Tugas Akhir ini metode yang dipakai adalah *Inter-Process Communication* (IPC). Pengiriman pesan antara Flash dan Matlab menggunakan *file .txt* atau *plain text*.

Flash akan menyimpan file yang berisi kata yang dicari, kemudian menyimpannya dengan nama *test.txt*. Kemudian, jendela Matlab terbuka dan membaca *file* tersebut. Matlab akan melakukan prosedur pengenalan kata dan menyimpan persentase pengenalan dalam bentuk *result.txt* yang nantinya akan disajikan oleh Flash.

### 3.1.2 Voice Recognition

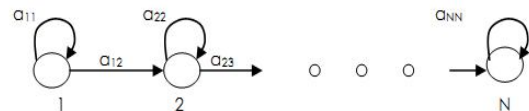
Desain dari program pengenalan kata menggunakan dua metode yaitu *Linear Predictive Coding* (LPC) dan *Hidden Markov Model* (HMM). LPC digunakan untuk mengekstraksi ciri dan HMM digunakan untuk metode pengenalan. Sistem dibagi menjadi dua bagian. Yang pertama adalah sistem pelatihan suara dan yang kedua adalah sistem pengenalan suara.

#### 3.1.2.1 Ekstraksi Ciri

Tahap kedua adalah proses ekstraksi ciri sinyal suara. Dari sinyal suara yang terekam akan didapatkan ciri-ciri besaran dari setiap kata yang diucapkan menggunakan program *HMMExtract.m*. Parameter LPC yang digunakan adalah  $p = 8$ ,  $N = 400$  yaitu jumlah sampel tiap *frame*, dan  $M = 100$  (jarak antara *frame* yang berurutan). Proses ekstraksi ciri akan menghasilkan matriks  $O$  dengan  $AxB$ ,  $A$

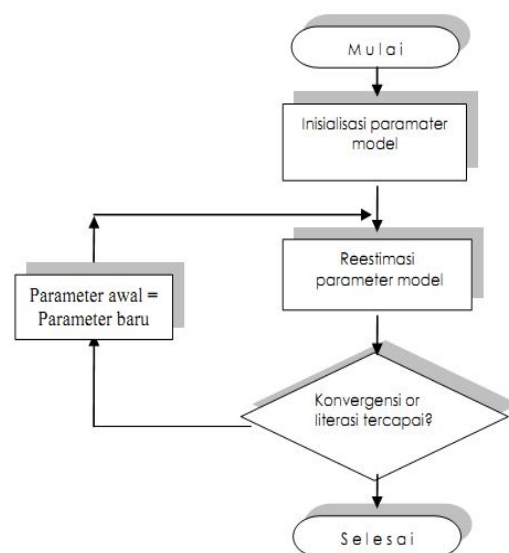
adalah banyaknya *frame*, sedangkan  $B$  menunjukkan bahwa setiap baris terdiri dari koefisien *cepstral* dan turunan koefisien *cepstral* terhadap waktu. Hasil dari ekstraksi ciri akan digunakan untuk proses pemodelan dan pengenalan.

#### 3.1.2.2 Pemodelan



Gambar 7 Left-right HMM dengan N keadaan

Tahap ketiga, melakukan proses pelatihan setiap kata menggunakan model HMM kiri-kanan dengan program *trainHMM.m*. Runtun observasi hasil ekstraksi ciri akan mengalami proses pelatihan berurutan sesuai dengan urutan jenis kata yang diucapkan yang tersimpan dalam satu berkas (*modKata*) *.mat*. Gambar model HMM kiri-kanan dengan  $N$  cacah keadaan dapat dilihat pada gambar 7. Sedangkan diagram alir proses pelatihan dapat dilihat pada gambar 8.



Gambar 8 Diagram alir proses pelatihan

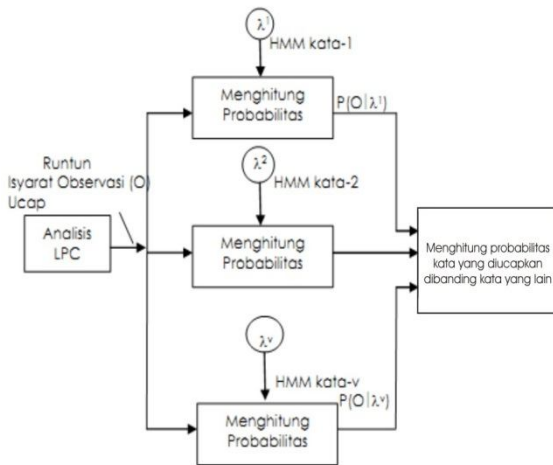
Setelah iterasi maksimum tercapai atau syarat konvergen tercapai maka akan diperoleh hasil pelatihan berupa parameter-parameter model  $\lambda_{opt} = (A, \pi, \mu, U)$ , ditambah dengan nilai logaritma probabilitas untuk setiap iterasi dan peubah

nama kata yang dimodelkan. Pada akhir proses pemodelan, akhirnya  $variableA$ ,  $\mu$ , dan  $U$  akan disimpan dengan format nama `modKata.mat`.

### 3.3.2.2 Pengenalan Kata

Tahap terakhir adalah melakukan pengujian/pengenalan atas sampel kata-kata yang telah Untuk proses pengujian atau pengenalan setiap kata menggunakan program `testHMM.exe` yang akan terbuka dari tombol *test pronunciation* pada menu *Result* yang sebelumnya dibuat menggunakan Flash. Hasil komputasi `testHMM.exe` adalah persentase keberhasilan akurasi kata, dan akhirnya akan disimpan dalam *plain text* dengan nama `result.txt`.

Terhadap sampel data hasil rekaman dilakukan proses ekstraksi ciri untuk mendapatkan runtun observasi, untuk kemudian dihitung probabilitas model runtun observasi menggunakan algoritma Viterbi, kemudian kata yang sebelumnya diucapkan akan dicari berapa probabilitasnya di banding kata yang lain.

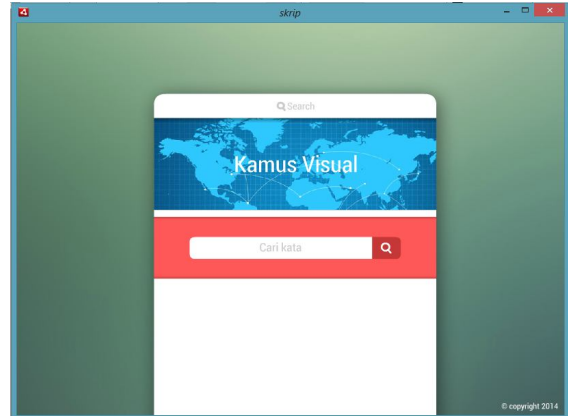


Gambar 9 Diagram blok pengenalan kata

## 3.2 Pengujian Sistem

### 3.2.1 Jalannya Program

Program utama dijalankan dari `skrip.exe`. Saat dijalankan, program akan menuju jendela utama. Pada jendela utama terdapat 1 *textbox* dan 1 tombol *search*. Tampilan jendela utama dapat dilihat pada Gambar 10.



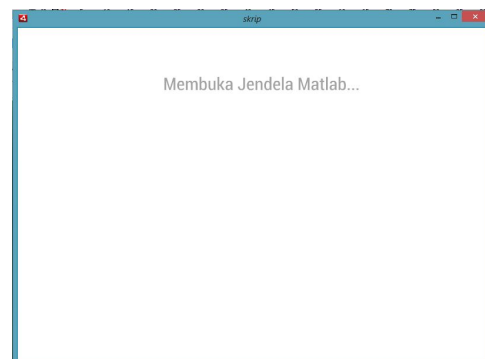
Gambar 10 Tampilan Menu Search

Setelah pengguna menginputkan kata dan memilih tombol *search*, pengguna akan dibawa pada tampilan antar muka *result*.



Gambar 11 Tampilan Antar Muka Result

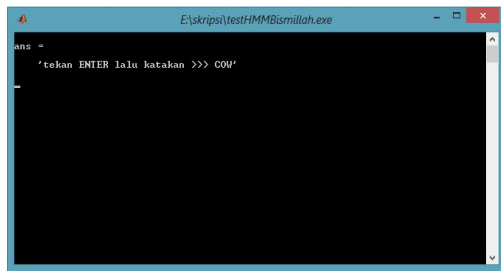
Ketika tombol *microphone* dipilih maka jendela yang akan keluar adalah jendela *voice recognition* seperti pada gambar 12.



Gambar 12 Jendela Antar Muka Voice Recognition

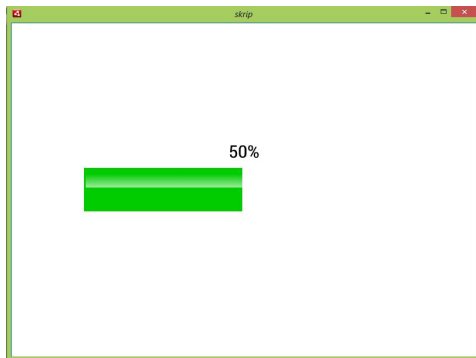


Diikuti oleh jendela Matlab seperti pada gambar 13



**Gambar 13**Antar Muka Aplikasi testHMM

Pada gambar 13, pengguna diminta untuk menekan tombol ENTER untuk mulai merekam kata “COW”.



**Gambar 14** Tampilan Score Result

Kemudian jendela MATLAB akan otomatis tertutup. Lalu jendela aplikasi FLASH akan menampilkan *score* seperti pada Gambar 14. FLASH akan membaca file result.txt yang berisi *integer* yang akhirnya akan diproses.

### 3.2.2 Hasil dan Pembahasan

Pada saat pengujian, dua orang responden memasukkan suaranya kedalam *database* suara. setiap responden memasukkan suara sebanyak tiga puluh satu kata yang sama untuk semua responden. Pengujian dilakukan dengan mengucapkan kata yang telah dimasukkan sebanyak sepuluh kali untuk tiap kata dengan memakai basisdata suara responden masing-masing. Responden adalah orang yang fasih berbicara bahasa Inggris agar sampel yang digunakan untuk pengenalan kata benar-benar sampel kata dalam bahasa Inggris yang baik dan benar.

#### 3.2.2.1 Pengujian tanpa *Noise*

Pengujian dilakukan dengan mengucapkan katadengan kondisi ruangan *noise* rendah. Hasil pengujian ini ditampilkan pada Tabel 1.

**Tabel 1.** Hasil pengujian basisdata suara responden yang memiliki basisdata suara tanpa *noise*.

Responden	Kata yang dikenali	Persentase
1	266	85,8%
2	255	82,2%

Hasil rata-rata pengujian dari keseluruhan responden yang memiliki basisdata suara tanpa derau menunjukkan nilai 84%.

#### 3.2.2.2 Pengujian dengan *Noise*

Pengujian dilakukan dengan mengucapkan kata pada kondisi ruangan *noise* tinggi. Kondisi berderau atau *noise* adalah kondisi dimana variasi dari nilai *amplitude* yang dihasilkan oleh matrik maksim berbeda jauh untuk kata yang sama pada waktu yang berbeda. Perekaman dilakukan pada ruangan dengan penuh *noise* dari luar. Hasil pengujian ini ditampilkan pada Tabel 2.

**Tabel 2.** Hasil pengujian basisdata suara responden yang memiliki basisdata suara dengan *noise*.

Responden	Kata yang dikenali	Persentase
1	197	63,5%
2	189	60,9%

Hasil rata-rata pengujian dari keseluruhan responden yang memiliki basisdata suara dengan derau menunjukkan nilai 62,2%.

#### 3.2.2.3 Pengujian Responden yang tidak Memiliki Basisdata

Pada pengujian ini, kata yang diucapkan sebagai masukan diucapkan oleh dua responden yang belum memasukkan basisdata suara. Pengujian dilakukan dengan mengucapkan kata sebanyak 31 kata dengan kata yang

diucapkan sama seperti yang ada pada basisdata suara sebelumnya. Tiap-tiap responden mengucapkan sepuluh kali untuk tiap kata dengan memakai basisdata suara responden pertama. Hasil pengujian ditampilkan pada tabel 3.

**Tabel 3** Hasil pengujian basisdata suara responden yang memiliki basisdata suara tanpa noise.

Responden	Kata yang dikenali	Persentase
1	226	72,9%
2	218	70,3%

Hasil rata-rata pengujian dari keseluruhan responden yang tidak memiliki basisdata suara menunjukkan nilai 66,9%.

#### 4. Kesimpulan

Dari penelitian, perancangan dan pembahasan, maka dapat disimpulkan bahwa :

1. Program Kamus Visual Bahasa Inggris Tentang Peternakan Berbasis Flash dengan Menggunakan *Voice Recognition* bisa dirancang dengan menggunakan pendekatan *Pattern Matching* atau pengenalan pola. Metode ekstraksi ciri menggunakan *Linear Predictive Coding* (LPC) dan pengenalan menggunakan *Hidden Markov Model* (HMM).
2. Hasil pengujian basisdata suara responden yang memiliki basisdata dengan mengucapkan lema yang terdapat pada basisdata tanpa derau adalah 84%, sedangkan hasil pengujian dengan derau adalah 62,2%.
3. Hasil pengujian 2 responden yang tidak memiliki basisdata adalah 71,6%.
4. Hasil rata-rata fitur *score* seluruh kata untuk empat responden adalah 73,5%.

#### 5. Saran

1. Diperlukan metode untuk *generate* runtun observasi seperti *Vector Quantization* untuk meningkatkan akurasi pengenalan kata.

2. Karena pertumbuhan aplikasi *mobile* sangat pesat, diharapkan program dapat terintegrasi dengan *mobile*, sehingga tidak diperlukan lagi headset maupun *microphone*. Dengan PC sebagai *server*, perekaman dan hasil pengenalan dapat dilihat melalui *mobile*.
3. Aplikasi perlu dikembangkan ke bidang yang lain selain peternakan, seperti pertanian dan sebagainya.

#### 5. Daftar Pustaka

- Binanto, M. 2010. *Multimedia Digital - Dasar Teori dan Pengembangannya*. Andi. Yogyakarta.
- Fadlisyah, Bustami, dan Ikhwanus, M. 2013. *Pengolahan Suara*. Graha Ilmu. Yogyakarta.
- Hapsari J.P. 2007. *Aplikasi Pengenalan Suara dalam Pengaksesan Sistem Informasi Akademik*. Universitas Diponegoro. Semarang.
- Ifeachor, Emmanuel and Barrie W. Jervis. 1993. *Digital Signal Processing*. Addison-Wesley Publishing Company. Boston.
- Iqbal, *Kamus Sebagai Sumber Diksi Dalam Sastra*. <http://pusatbahasaalazhar.wordpress.com/pesona-puisi/kamus-sebagai-sumber-diksi-dalam-sastra/>. Diunduh : 4 Januari 2014.
- Lawrence Rabiner, and Biing Hwang Juang. 1993. *Fundamentals of Speech Recognition*. Prentice Hall. New Jersey.
- Mulyadu, ST. 2010. *Membuat Aplikasi untuk ANDROID*. Multimedia Center. Yogyakarta.
- Puspitosari, Heni A. 2010. *Animasi Grafis Dengan Adobe Flash Pro Cs5 Tingkat Lanjut*. Skripta. Yogyakarta
- Septriansyah, Christin dan Arsyad. *Tantangan Insan Peternakan dan Kesehatan Hewan*. <http://lampost.co/berita/tantangan-insan-peternakan-dan-kesehatan-hewan.2012>. Diunduh : 3 Maret 2014.
- Sutopo, Ariesto Hadi. 2004. *Multimedia Interaktif dengan Flash*. Graha Ilmu. Yogyakarta.



Thiang, Dhanny Wijaya. 2010. *Speech Recognition Using LPC and HMM Applied For Controlling Movement of Mobile Robot*. Petra Christian University. Surabaya.

Wong, Michael, *Pengenalan Matlab pada Sistem Kontrol*.  
<http://wongmichael.wordpress.com/2012/10/21/178/>. 2012. Diunduh 1 April 2014.

Wikipedia *Komunikasi Antar Proses*.[http://id.wikipedia.org/wiki/Komunikasi\\_antar\\_proses](http://id.wikipedia.org/wiki/Komunikasi_antar_proses). Diunduh : 20 Desember 2014.