

# APLIKASI PENCIRIAN DENGAN LINEAR PREDICTIVE CODING UNTUK PEMBELAJARAN PENGUCAPAN NAMA HEWAN DALAM BAHASA INGGRIS MENGGUNAKAN JARINGAN SARAF TIRUAN PROPAGASI BALIK

Sigit Nur Rohman, Achmad Hidayatno, and Ajub Ajulian Zahra

Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro  
Jalan Prof. Sudharto, Tembalang, Semarang, Indonesia

## Abstract

In this research designed a recognition system for learning the pronunciation of the word animal names in English. Original speech signal sample at 8000 Hz pick out a small portion For voice parameter extraction process used method Linear Predictive Coding (LPC) to obtain cepstral coefficients. LPC cepstral coefficients are transformed into the frequency domain with Fast Fourier Transform (FFT). For decision making process of the introduction and use Neural Networks (NN) back propagation. Testing is done using the data train, according to a database of test data and test data do not fit database. While the networks do a variation of 3, 4 and 5 hidden layers respectively for 1, 2 and 3 the number of syllables said. Based on the results of testing training data, the recognition rate for each variation of each network the number of syllables showed no difference in test results, the percentage was 99% for the 1 syllable, 98.5% for the 2 syllables and 100% for 3 syllables. Test data suitable for testing the database, the highest recognition rate for type 1 syllable is a network with 4 hidden layers using a variation of the percentage is 85%, whereas type 2 syllables highest recognition rate using a variation of 5 hidden layers with the correct percentage of 75% and 81.67 % for type 3 syllables using 5 hidden layers. While the test results do not fit the test database, the highest recognition rate for type 1 syllable is a network with 4 hidden layers using a variation of the percentage is 15.83% while the type 2 syllables highest recognition rate using a variation of 3 hidden layers with percentage correct, 20.83% and 33.33% for type 3 syllables using 3 and 4 hidden layers.

*Keywords : Linear Predictive Coding, Fast Fourier Transform, Neural Network, Backpropagation.*

## 1. Pendahuluan

### 1.1 Latar Belakang

Bahasa Inggris merupakan bahasa *universal* yang digunakan untuk berkomunikasi di sebagian besar masyarakat di seluruh belahan dunia. Seperti halnya di Indonesia yang telah mulai diterapkan pelajaran Bahasa Inggris dari bangku Sekolah Dasar.

Perbedaan cara pengucapan dengan tulisan yang tertera membuat Bahasa Inggris susah dipahami bagi masyarakat Indonesia yang terbiasa menggunakan Bahasa Indonesia untuk berkomunikasi. Seperti kata "cow" dalam tulisan Bahasa Inggris yang mempunyai arti "sapi" dalam Bahasa Indonesia dibaca "kaw" secara lisan dalam pengucapannya. Untuk itu diperlukan cara yang dapat memudahkan manusia untuk melatih pengucapannya dalam Bahasa Inggris.

Dalam dunia perangkat lunak dapat dirancang suatu aplikasi sebagai alat bantu menata ejaan dalam berbahasa Inggris. Dengan menggunakan metode ekstraksi pola

suara *Linear Predictive Coding* (LPC) akan memudahkan dalam mendapatkan ciri tiap sinyal sebagai proses awal perancangan program. Dari metode ini akan diperoleh nilai koefisien *cepstral* LPC yang merupakan *feature* (ciri) dari suara yang diucapkan. Kemudian koefisien *cepstral* LPC tersebut diproses dengan *Fast Fourier Transform* (FFT) untuk mendapatkan sinyal pada domain frekuensi. Untuk proses pengenalan dan pengambilan keputusan dapat digunakan suatu algoritma Jaringan Saraf Tiruan (JST) propagasi balik.

### 1.2 Tujuan

Tujuan dari penelitian ini adalah membuat aplikasi pengenalan kata untuk pembelajaran pengucapan nama hewan dalam Bahasa Inggris dengan menggunakan jaringan saraf tiruan propagasi balik sebagai algoritma pembelajarannya.

### 1.3 Batasan Masalah

Untuk menyederhanakan pembahasan pada Penelitian ini,

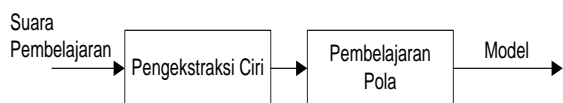
masalah dibatasi sebagai berikut :

1. Data masukan (pada *database*) berupa sinyal suara yang di ambil dari 4 orang responden (3 pria dan 1 wanita).
2. Pengenalan kata untuk 30 nama hewan yakni : *ant* (semut), *bear* (beruang), *bee* (lebah), *bird* (burung), *buffalo* (kerbau), *butterfly* (kupu-kupu), *camel* (unta), *cat* (kucing), *chicken* (ayam), *cow* (sapi), *crocodile* (buaya), *deer* (rusa), *dog* (anjing), *dolphin* (lumba-lumba), *donkey* (keledai), *dragonfly* (capung), *eagle* (elang), *elephant* (gajah), *fly* (lalat), *gorilla* (gorilla), *grasshopper* (belalang), *iguana* (iguana), *kangaroo* (kanguru), *lion* (singa), *lizard* (kadal), *monkey* (monyet), *mosquito* (nyamuk), *rabbit* (kelinci), *sheep* (domba), *tiger* (macan).
3. Derau (*noise*) yang turut terekam pada proses perekaman diabaikan.
4. Metode pencirian menggunakan LPC (*Linear Predictive Coding*).
5. Model pelatihan JST adalah algoritma perambatan balik (*backpropagation algorithm*).
6. Pengujian dilakukan dengan merekam suara yang diucapkan dalam keadaan normal dan tidak dibuat-buat sesuai dengan contoh suara pengucapannya.

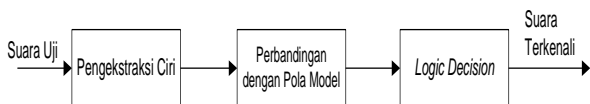
## 2. Landasan Teori

### 2.1 Pengenalan Suara

Salah satu bentuk pendekatan pengenalan suara adalah pendekatan pengenalan pola yang terdiri dari dua langkah yaitu pembelajaran pola suara dan pengenalan suara melalui perbandingan pola. Tahap perbandingan pola adalah tahap saat suara yang akan dikenali dibandingkan polanya dengan setiap kemungkinan pola yang telah dipelajari dalam fase pembelajaran, untuk kemudian diklasifikasikan dengan pola terbaik yang cocok. Blok diagram pembelajaran pola dan pengenalan suara ditunjukkan pada gambar di bawah ini.



(a) Blok diagram pembelajaran pola.



(b) Blok diagram pengenalan suara.

Gambar 1. Blok diagram pembelajaran pola dan pengenalan suara.

Pengenalan suara secara umum dapat dibagi menjadi tiga tahap, yaitu tahap ekstraksi ciri, tahap pemodelan atau

pembelajaran dan tahap pengenalan suara.

### 2.2 Ekstraksi Ciri dengan LPC

Langkah-langkah analisis LPC untuk mendapatkan koefisien LPC pada proses ekstraksi ciri suara adalah sebagai berikut :

#### 1. Preemphasis

Sinyal suara  $s(n)$  dimasukkan ke dalam sistem digital orde rendah (biasanya berupa filter FIR orde satu) yang digunakan untuk meratakan spektrum sinyal. Keluaran dari rangkaian *preemphasis*  $\tilde{s}(n)$  adalah :

$$\tilde{s}(n) = s(n) - \tilde{a}s(n-1), 0,9 \leq \tilde{a} \leq 1,0 \dots \dots \dots (1)$$

#### 2. Frame blocking

Sinyal suara hasil dari proses *preemphasis*  $\tilde{s}(n)$  diblok atau dibagi ke dalam beberapa *frame* yang terdiri dari  $N$ -sampel suara, dengan jarak antara *frame* yang berdekatan dipisahkan oleh  $M$ -sampel. Jika  $M \leq N$ , beberapa *frame* yang berdekatan akan saling *overlap* dan hasil estimasi spektral LPC akan berkorelasi dari *frame* ke *frame*. Sebaliknya, jika  $M > N$ , tidak akan ada *overlap* antara *frame* yang berdekatan sehingga beberapa isyarat sinyal suara akan hilang total.

#### 3. Windowing (penjendelaan)

*Windowing* digunakan untuk mengurangi *discontinuitas* sinyal pada awal dan akhir *frame*. Jika *window* didefinisikan sebagai  $w(n)$ , maka hasil dari penjendelaan sinyal adalah :

$$\tilde{x}_l(n) = \tilde{s}(n)w(n), 0 \leq n \leq N-1 \dots \dots \dots (2)$$

Jenis *window* yang biasa digunakan adalah *Hamming window* yang mempunyai bentuk umum :

$$w(n) = 0,54 - 0,46 \cos \left( \frac{2\pi n}{N-1} \right), 0 \leq n \leq N-1 \dots \dots \dots (3)$$

#### 4. Analisis autokorelasi

Setiap *frame* dari sinyal setelah melalui proses *windowing*, kemudian dilakukan analisis autokorelasi sebagai berikut :

$$r_l(m) = \sum_{n=0}^{N-1-m} \tilde{x}_l(n)\tilde{x}_l(n+m), m = 0,1,\dots,p \dots (4)$$

dengan nilai autokorelasi tertinggi  $p$  adalah orde LPC. Nilai  $p$  biasanya antara 8 sampai 16.

5. Analisis LPC

Proses selanjutnya adalah analisis LPC, yang mengubah setiap *frame* autokorelasi  $p+1$  ke dalam bentuk parameter-parameter LPC atau yang biasa disebut dengan koefisien LPC. Metode yang biasa digunakan dalam analisis LPC ini adalah metode Durbin yang mempunyai algoritma sebagai berikut :

$$E^{(0)} = r(0) \dots \dots \dots (5)$$

$$k_i = \left\{ r(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} r(i-j) \right\} / E^{(i-1)}, \quad 1 \leq i \leq p \dots \dots \dots (6)$$

$$\alpha_i^{(i)} = k_i \dots \dots \dots (7)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}, \quad 1 \leq j \leq i-1 \dots \dots \dots (8)$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \dots \dots \dots (9)$$

Dengan menyelesaikan persamaan 2.5 sampai 2.9 secara rekursif untuk  $i = 1, 2, \dots, p$ , koefisien LPC diperoleh sebagai berikut :

$$a_m = \alpha_m^{(p)}, \quad 1 \leq m \leq p \dots \dots \dots (10)$$

6. 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 rekursi sebagai berikut :

$$c_0 = \ln \sigma^2 \dots \dots \dots (11)$$

$$c_m = a_m + \sum_{k=1}^{m-1} \left( \frac{k}{m} \right) c_k a_{m-k}, \quad 1 \leq m \leq p \dots \dots \dots (12)$$

$$c_m = \sum_{k=1}^{m-1} \left( \frac{k}{m} \right) c_k a_{m-k}, \quad m > p \dots \dots \dots (13)$$

dengan  $\sigma$  adalah *gain* dari LPC. Koefisien *cepstral* yang digunakan direpresentasikan dengan  $Q$ , dimana koefisien yang digunakan biasanya adalah  $Q > p$ , yaitu  $Q = (3/2)p$ .

7. Pembobotan Parameter

Pembobotan dilakukan pada parameter karena sensitifitas koefisien *cepstral* orde rendah pada keseluruhan slope spektral dan sensitifitas koefisien *cepstral* orde tinggi pada derau. Pembobotan dilakukan dengan memberikan jendela tapis pada koefisien *cepstral*. Bentuk koefisien *cepstral* setelah pembobotan adalah :

$$\tilde{c}_m = w_m c_m, \quad 1 \leq m \leq Q \dots \dots \dots (14)$$

dengan  $w_m$  adalah jendela pembobotan atau disebut *bandpass lifter* ( tapis pada domain *cepstral* ). Besarnya  $w_m$  adalah :

$$w_m = \left[ 1 + \frac{Q}{2} \sin \left( \frac{\pi m}{Q} \right) \right], \quad 1 \leq m \leq Q \dots \dots \dots (15)$$

8. Turunan Temporal Koefisien *Cepstral*

Turunan koefisien *cepstral* (*delta cepstral*) meningkatkan keterwakilan sifat-sifat spektral sinyal yang dianalisis pada parameter. Turunan koefisien *cepstral* dapat dituliskan seperti berikut:

$$\frac{\partial c_m(t)}{\partial t} = \Delta c_m(t) \approx \left( \sum_{k=1}^K k c_m(t+k) \right) / \left( \sum_{k=1}^K k^2 \right) \dots (16)$$

dengan  $(2K + 1)$  adalah jumlah frame.

2.3 Analisa Frekuensi Sinyal dengan Transformasi Fourier Cepat

Transformasi Fourier Cepat atau *Fast Fourier Transform* (FFT) merupakan penyederhanaan dari *Discrete Fourier Transform* (DFT).

$$X_1(k) = \sum_{n=0}^{N-1} x_n e^{-j2\pi n k / N}, \quad k = 0, 1, \dots, N-1 \dots \dots (17)$$

Dan juga faktor  $e^{-j2\pi n / N}$  akan ditulis sebagai  $W_N$ , maka :

$$W_N = e^{-j2\pi / N} \dots \dots \dots (18)$$

Sehingga persamaan (17) menjadi :

$$X_1(k) = \sum_{n=0}^{N-1} x_n W_N^{kn}, \quad k = 0, 1, \dots, N-1 \dots \dots (19)$$

Proses *Fast Fourier Transform* (FFT) ini dilakukan setelah didapat koefisien LPC sebanyak 299 data untuk 1 suku kata, 780 data untuk 2 suku kata dan 1196 data untuk 3 suku kata. FFT ini dilakukan untuk meningkatkan unjuk kerja sistem karena dengan digunakannya proses FFT maka perbedaan antara pola kata yang satu dengan pola kata yang lain makin terlihat jelas.

FFT yang digunakan memakai 256 poin untuk 1 suku kata, 512 point untuk 2 suku kata dan 1024 point untuk 3 suku kata. Karena hasil FFT simetris maka keluaran FFT tersebut hanya diambil setengahnya saja. Keluaran dari FFT tersebut merupakan masukan bagi jaringan saraf tiruan.

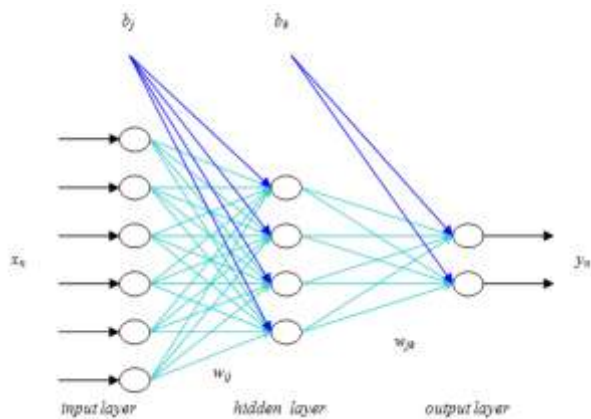
2.4 Jaringan Saraf Tiruan Propagasi Balik

Jaringan saraf tiruan perambatan balik merupakan salah satu model dari jaringan saraf tiruan umpan maju dengan

menggunakan pelatihan terbimbing yang disusun berdasar pada algoritma *error back propagation*. Pola masukan dan target diberikan sebagai sepasang data. Bobot-bobot awal dilatih dengan melalui tahap maju untuk mendapatkan galat keluaran yang selanjutnya galat ini digunakan dengan tahap mundur untuk memperoleh nilai bobot yang sesuai agar dapat memperkecil nilai galat sehingga target keluaran yang dikehendaki tercapai. Tujuan dari model ini adalah untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama proses pelatihan berlangsung serta kemampuan jaringan memberikan respon yang benar terhadap pola masukan yang berbeda dengan pola masukan selama pelatihan.

### 2.4.1 Arsitektur Jaringan Saraf Tiruan Perambatan Balik

Hubungan antar neuron dalam suatu jaringan dan membentuk lapisan disebut arsitektur jaringan. Arsitektur jaringan saraf tiruan perambatan balik terdiri dari beberapa lapisan, yaitu lapisan masukan, lapisan tersembunyi dan lapisan keluaran. Masing-masing neuron dalam tiap lapisan saling terhubung dengan lapisan di atasnya. Neuron-neuron pada lapisan masukan terhubung dengan neuron-neuron pada lapisan tersembunyi melalui sinapsis yang disebut bobot, begitu juga pada lapisan tersembunyi neuron-neuronnya terhubung melalui sinapsis dengan neuron-neuron pada lapisan keluaran. Terdapat masing-masing 1 buah bias pada lapisan masukan dan lapisan tersembunyi yang bernilai = 1. Arsitektur jaringan saraf tiruan perambatan balik dapat dilihat pada Gambar 2.



Gambar 2. Arsitektur jaringan saraf tiruan perambatan balik

### 2.4.2 Algoritma Jaringan Saraf Tiruan Perambatan Balik

Algoritma jaringan saraf tiruan perambatan balik terdiri dari dua bagian:

#### 1. Algoritma Pelatihan Jaringan saraf tiruan perambatan balik

Sebelum melakukan proses pelatihan, terdapat beberapa parameter jaringan yang harus ditentukan terlebih dahulu, yaitu :

- Laju pembelajaran (*learning rate*), harus diberikan dan mempunyai nilai positif kurang dari 1. Semakin tinggi nilainya, maka semakin cepat kemampuan jaringan untuk belajar. Akan tetapi hal ini kurang baik, karena galat yang dihasilkan tidak merata.
- Toleransi galat, semakin kecil kesalahan maka jaringan akan memiliki nilai bobot yang lebih akurat, tetapi akan memperpanjang waktu pelatihan.
- Jumlah maksimal proses pelatihan yang dilakukan (maksimum iterasi), biasanya bernilai besar dan diberikan untuk mencegah terjadinya perulangan tanpa akhir.

Di dalam proses pelatihan perambatan balik terdapat tiga tahap. Tahap pertama ialah tahap maju. Pada tahap ini seluruh proses awal inialisasi bobot-bobot masukan dilakukan. Pada tahap ini juga ditentukan laju pembelajaran ( $\alpha$ ), nilai toleransi galat dan jumlah iterasi (siklus setiap pola pelatihan) yang diperlukan selama proses komputasi berlangsung. Setelah semua proses inialisasi dilakukan, maka langkah selanjutnya ialah proses maju. Setiap unit masukan akan mengirimkan sinyal masukan ke lapisan tersembunyi. Setelah dihitung dengan menggunakan fungsi aktivasi maka keluarannya akan dikirimkan ke lapisan di atasnya, yaitu lapisan keluaran. Setelah nilai keluaran diperoleh, maka dibandingkan dengan target keluaran sebenarnya. Selisih nilai keluaran dengan target keluaran sebenarnya disebut dengan galat. Jika nilai galat lebih kecil atau sama dengan dari nilai ambang maka proses iterasi dihentikan, tetapi jika tidak maka nilai galat tersebut digunakan untuk memodifikasi bobot-bobot untuk mengoreksi kesalahan yang terjadi.

Tahap kedua adalah tahap mundur. Pada tahap ini, nilai galat yang diperoleh pada di lapisan keluaran digunakan untuk mengoreksi bobot-bobot yang ada pada lapisan tersembunyi yang berhubungan langsung dengan lapisan keluaran. Setelah itu nilai galat di setiap unit pada lapisan tersembunyi juga dihitung untuk mengoreksi bobot-bobot yang menghubungkan lapisan masukan dengan lapisan tersembunyi.

Tahap ketiga adalah tahap pengoreksian bobot. Setelah seluruh bobot pada lapisan masukan dan lapisan tersembunyi dimodifikasi sesuai dengan besar faktor galatnya, maka ketiga fase ini diulang secara terus menerus sampai kondisi berhenti dipenuhi. Kondisi berhenti yang dimaksud adalah jika jumlah iterasi yang ditetapkan tercapai atau jika nilai galat jaringan telah sama dengan atau lebih kecil dari nilai toleransi galat yang ditetapkan sebelumnya. Pada tahap pelatihan,

jaringan diharapkan dapat melatih seluruh data pelatihan yang diberikan untuk mendapatkan bobot akhir jaringan yang akan digunakan pada tahap pengujian.

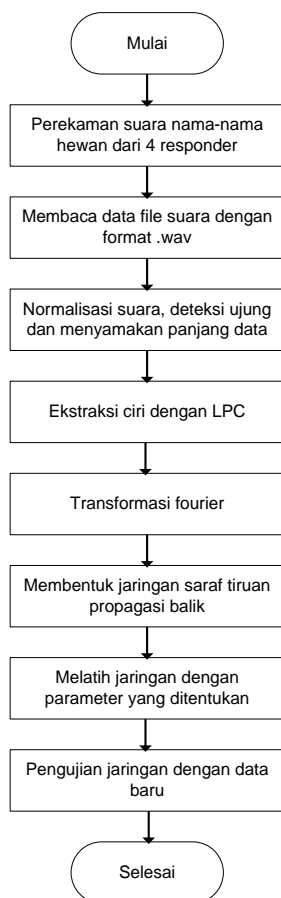
## 2. Algoritma Pengujian Jaringan Saraf Tiruan Perambatan Balik

Setelah proses pelatihan, tahap selanjutnya adalah proses pengujian. Pada proses pengujian, tahap yang dilakukan hanya sampai tahap maju saja, tidak ada tahap mundur apalagi tahap modifikasi bobot. Seluruh bobot masukan diambil dari nilai bobot terakhir yang diperoleh dari proses pelatihan. Pada tahap pengujian ini, jaringan diharapkan dapat mengenali pola berdasarkan data baru yang diberikan.

## 3. Perancangan Sistem

### 3.1 Gambaran Umum

Perancangan sistem merupakan tahap yang penting dalam proses pembuatan program. Perancangan bertujuan agar dalam pembuatannya dapat berjalan secara sistematis, terstruktur dan rapi sehingga hasil program dapat berjalan sesuai dengan apa yang dikehendaki. Secara umum pembuatan program ini mengikuti alur sesuai yang ditunjukkan pada Gambar berikut.



Gambar 3. Diagram perancangan system

Pembuatan sistem pengenalan kata terdiri dari proses pengelolaan *database*, proses ekstraksi ciri, proses pemodelan dan pelatihan. Ketiga proses tersebut sering disebut dengan proses pembelajaran. Proses pengenalan kata merupakan proses perbandingan antara suara uji dengan model suara yang didapat dari proses pembelajaran.

### 3.2 Pengelolaan Database

Dalam pembuatan *database* dilakukan perekaman suara dari 4 orang responden dengan frekuensi cuplik 8000 Hz selama 1 detik. Tiap responder mengucapkan kata sebanyak 150 kali untuk 30 nama hewan. Sehingga total data latih sebanyak 600 data.

Suara memiliki amplitude yang bervariasi tergantung pada keras lemahnya suara tersebut pada saat diucapkan. Normalisasi suara dilakukan untuk menyamakan rentang amplitude suara dengan membagi nilai-nilai amplitudo dengan nilai amplitude tertinggi. Sehingga diperoleh nilai hasil normalisasi suara dengan nilai maksimum adalah 1 dan nilai minimum adalah -1.

Dalam suatu masukan data suara, nilai yang akan diproses pada umumnya terletak di tengah-tengah. Untuk mengatasi kelebihan bagian bukan suara (bagian kosong yang terdapat pada awal dan akhir sebuah berkas suara) perlu dilakukan proses deteksi ujung suara. Penentuan awal suara adalah data pertama yang mempunyai amplitude mutlak lebih dari 0,115. Data kurang dari 0,115 sebelum awal suara dihilangkan. Sedangkan akhir suara adalah data terakhir yang mempunyai amplitude mutlak lebih dari 0,115. Data yang kurang dari 0,115 setelah akhir suara dihilangkan.

Proses selanjutnya yaitu menyamakan panjang data pada tiap data suara sebelum diumpungkan ke proses ekstraksi ciri LPC. Proses ini dibedakan menjadi 3 golongan yaitu untuk panjang data 1 suku kata, 2 suku kata dan 3 suku kata sesuai dengan jumlah suku kata ucap tiap nama hewan. Panjang data untuk 1 suku kata adalah 3200, 5040 untuk panjang data 2 suku kata dan 7520 untuk panjang data 3 suku kata. Sedangkan nilai 0,00001 ditambahkan pada data suara untuk menambah nilai data yang kurang dari nilai panjang data yang ditentukan.

### 3.3 Proses Ekstraksi Ciri

LPC merupakan metode ekstraksi ciri yang sering digunakan dalam proses pengolahan suara. Proses pengekstraksian ciri dengan menggunakan metode LPC akan diperoleh nilai koefisien *cepstral* LPC yang merupakan ciri dari suara pembicaraan.

Proses pencirian LPC terdapat beberapa tahap. Data dari hasil pembacaan sinyal suara akan melewati proses *preemphasis*. Setelah melalui proses *preemphasis*, sinyal

suara dibagi dalam beberapa *frame* sepanjang  $N$  *sample* suara dengan jarak antara *frame* yang berdekatan dipisahkan oleh  $M$  *sample*. Setiap *frame* kemudian dijendelakan menggunakan *Hamming window*, dianalisis autokorelasi dan dianalisis LPC sehingga didapatkan koefisien LPC dari sinyal suara. Selanjutnya koefisien LPC dikonversi menjadi koefisien *cepstral* sebelum dilakukan pembobotan dan penurunan.

Proses *frame blocking* yang dilakukan pada sistem ini ditetapkan tiap 30 mili detik dengan jarak antar *frame* 10 mili detik. Jadi dengan kecepatan cuplik sebesar 8000 Hz maka tiap *frame* akan berisi 240 *byte* data dengan jarak antar *frame* 80 *byte* data atau dengan kata lain *overlap* yang terbentuk sebesar 160 *byte* data.

Dengan ketentuan *frame* seperti di atas, maka jumlah *frame* dapat ditentukan dan dibedakan sesuai dengan jumlah suku katanya :

- 1 Suku Kata

Untuk data hasil cuplik sebanyak 3200 data akan terbentuk  $(3200-160)/80 = 38$  *frame*. Pada perhitungan koefisien LPC, digunakan orde LPC 12 sehingga diperoleh data output sebanyak  $(12+1) \times 38 = 494$  data. Selanjutnya koefisien LPC ini di konvers menjadi matrik berukuran  $494 \times 1$  data.

- 2 Suku Kata

Untuk data hasil cuplik sebanyak 5040 data akan terbentuk  $(5040-160)/80 = 61$  *frame*. Pada perhitungan koefisien LPC, digunakan orde LPC 12 sehingga diperoleh data output sebanyak  $(12+1) \times 61 = 793$  data. Selanjutnya koefisien LPC ini di konvers menjadi matrik berukuran  $793 \times 1$  data.

- 3 Suku Kata

Untuk data hasil cuplik sebanyak 7520 data akan terbentuk  $(7520-160)/80 = 92$  *frame*. Pada perhitungan koefisien LPC, digunakan orde LPC 12 sehingga diperoleh data output sebanyak  $(12+1) \times 92 = 1196$  data. Selanjutnya koefisien LPC ini di konvers menjadi matrik berukuran  $1196 \times 1$  data.

Selanjutnya koefisien LPC yang diperoleh tersebut diproses dengan *Fast Fourier Transform* (FFT) untuk mendapatkan sinyal pada domain frekuensi. Hal ini bertujuan agar perbedaan antar pola kata yang satu dengan yang lain terlihat lebih jelas sehingga ekstraksi parameter sinyal memberikan hasil yang lebih baik.

Proses *Fast Fourier Transform* (FFT) yang digunakan memakai 256 poin untuk 1 suku kata, 512 point untuk 2 suku kata dan 1024 point untuk 3 suku kata. Karena hasil FFT simetris maka keluaran FFT tersebut hanya diambil

setengah saja yang dianggap sudah dapat merepresentasikan ciri dari sinyal ucapan dalam domain frekuensi. Dari data FFT ini kemudian dijadikan masukan bagi jaringan saraf tiruan.

### 3.4 Pelatihan Jaringan Saraf Tiruan *Back Propagation*

Setelah melalui tahap transformasi Fourier selanjutnya parameter-parameter yang dihasilkan dimasukkan ke dalam jaringan saraf tiruan dengan menggunakan metode pembelajaran *Back Propagation*. Proses pelatihan jaringan saraf tiruan dilakukan dengan mengambil *input* hasil proses FFT dari seluruh suara hasil perekaman *database*. Jumlah data pada *database* adalah 600 potong kata. Terdiri dari 200 potong kata untuk tiap-tiap jenis golongan suku katanya. Sehingga input pelatihan bagi JST sebanyak data FFT  $\times$  200 data.

Pada jaringan dilakukan variasi 3, 4 dan 5 hidden layer untuk masing-masing jenis suku kata ucap. Sehingga total terbentuk 9 variasi jaringan dengan nilai laju pembelajaran 0,001. Sedangkan fungsi aktivasi yang digunakan ialah *sigmoid bipolar* yang memiliki range keluaran antara -1 sampai 1, sehingga keluaran yang telah ditentukan juga harus bernilai antara -1 sampai 1.

### 3.5 Pengambilan Keputusan

Pengambilan keputusan dilakukan dalam 2 tahap. Tahap pertama setelah pendeteksian ujung data suara. Sedangkan tahap yang kedua setelah simulasi data suara dengan jaringan yang telah tersimpan. Pada tahap pertama data suara yang mempunyai panjang data antara 1000-3200 untuk 1 suku kata, 2000-5040 untuk 2 suku kata dan 3000-7520 untuk 3 suku kata akan diproses ke proses selanjutnya. Jika panjang data melebihi atau kurang dari batasan yang telah ditentukan tersebut maka suara dikenali salah.

Pada tahap kedua keluaran dari JST propagasi balik akan disimulasikan dengan jaringan yang telah dilatih. Hasil simulasi akan dibandingkan dengan target yang telah ditentukan. Jika hasil simulasi sesuai dengan target maka suara dikenali benar. Namun jika tidak sesuai dengan target akan dikenali salah.

## 4. Pengujian Dan Analisis

Pengujian dilakukan menggunakan data latih, data uji sesuai database dan data uji tak sesuai database.

### 4.1 Pengujian Data Latih

Pada pengujian data latih, data suara yang digunakan untuk melakukan pengujian merupakan data suara dari 4 responden yang telah direkam sebagai *database*. Perekaman dilakukan selama 1 detik pada frekuensi 8000

Hz dan resolusi 16 bit. Tiap responden menuturkan 5 kali pengucapan untuk 1 nama hewan. Sehingga setiap responder mengucapkan nama hewan sebanyak 150 kali untuk 30 nama hewan yang berbeda. Total terdapat 600 data suara untuk data latih.

Hasil pengenalan pada data latih dapat dilihat pada table berikut.

**Tabel 1 Hasil pengujian data latih**

Jumlah Suku Kata	Nama Jaringan	Persentase Benar
1 Suku Kata	net_satu_1	99%
	net_satu_2	99%
	net_satu_3	99%
2 Suku Kata	net_dua_1	98,5%
	net_dua_2	98,5%
	net_dua_3	98,5%
3 Suku Kata	net_tiga_1	100%
	net_tiga_2	100%
	net_tiga_3	100%

Hasil pengujian menggunakan data latih untuk masing-masing jenis jumlah suku kata dari seluruh variasi jaringan, tidak menunjukkan perbedaan hasil pengujian. Persentase benar untuk jenis 1 suku kata bernilai 99%, 2 suku kata bernilai 98,5% sedangkan 3 suku kata bernilai 100%. Hal ini disebabkan karena data yang digunakan untuk menguji jaringan adalah data latih. Selisih nilai MSE akhir dari tiap-tiap jaringan yang sangat kecil juga ikut mempengaruhi hasil pengujian jaringan.

**4.2 Pengujian Data Uji Sesuai Database**

Pada pengujian data uji sesuai database, data suara yang digunakan untuk melakukan pengujian merupakan data suara dari 4 responden yang telah direkam dengan mengucapkan nama-nama hewan sesuai dengan database. Keempat responder merupakan 2 responder dari dalam database dan 2 responder diluar database.

Perekaman dilakukan selama 1 detik pada frekuensi 8000 Hz dan resolusi 16 bit. Tiap responden menuturkan 3 kali pengucapan untuk 1 nama hewan. Sehingga setiap responder mengucapkan nama hewan sebanyak 90 kali untuk 30 nama hewan yang berbeda. Data uji sesuai database untuk setiap nama hewan berjumlah 12 data suara. Total keseluruhan terdapat 360 data suara untuk data uji sesuai database.

Hasil pengenalan pada data uji sesuai database dapat dilihat pada table berikut.

**Tabel 2 Hasil pengujian data uji sesuai database**

Jumlah Suku Kata	Nama Jaringan	Persentase Benar
1 Suku Kata	net_satu_1	81,67%
	net_satu_2	85%
	net_satu_3	78,33%
2 Suku Kata	net_dua_1	70%
	net_dua_2	68,33%
	net_dua_3	75%
3 Suku Kata	net_tiga_1	79,17%
	net_tiga_2	80,83%
	net_tiga_3	81,67%

Hasil pengujian menggunakan data uji sesuai database untuk masing-masing jenis jumlah suku kata dari seluruh variasi jaringan, menunjukkan perbedaan hasil pengujian. Hasil pengujian pada jaringan net\_satu\_2 untuk jenis 1 suku kata adalah yang tertinggi yaitu dengan persentasi benar 85%. Sedangkan untuk jenis 2 suku kata, persentasi benar tertinggi pada jaringan net\_dua\_3 dengan 75% dan jaringan net\_tiga\_3 untuk jenis 3 suku kata dengan persentasi benar 81,67%. Hal ini membuktikan penambahan jumlah hidden layer akan meningkatkan tingkat pengenalan suatu jaringan sampai pada titik tertentu sebelum akhirnya penambahan jumlah hidden layer tidak lagi berpengaruh terhadap tingkat pengenalan suatu jaringan.

**4.3 Pengujian Data Uji Tak Sesuai Database**

Pada pengujian data uji tak sesuai database, data suara yang digunakan adalah data suara yang kemungkinan diucapkan salah oleh seorang responder ketika pengucapan nama hewan dalam Bahasa Inggris. Terdapat 3 variasi pengucapan salah yang direkam untuk setiap nama hewan. Sehingga setiap responder melakukan pengucapan salah sebanyak 90 kali untuk 30 nama hewan yang berbeda. Jumlah responder untuk data uji tak sesuai database sejumlah 4 responder. 2 responder dalam database dan 2 responder diluar database. Data suara untuk setiap nama hewan berjumlah 12. Total keseluruhan terdapat 360 data suara untuk data uji tak sesuai database. Perekaman dilakukan selama 1 detik pada frekuensi 8000 Hz dan resolusi 16 bit.

Hasil pengenalan pada data uji tak sesuai database dapat dilihat pada table berikut.

**Tabel 3 Hasil pengujian data uji tak sesuai database**

Jumlah Suku Kata	Nama Jaringan	Persentase Benar
1 Suku Kata	net_satu_1	18,33%
	net_satu_2	15,83%

	net_satu_3	20,83%
	net_dua_1	20,83%
2 Suku Kata	net_dua_2	26,67%
	net_dua_3	25%
	net_tiga_1	33,33%
3 Suku Kata	net_tiga_2	33,33%
	net_tiga_3	38,33%

Hasil pengujian menggunakan data uji tak sesuai *database* untuk masing-masing jenis jumlah suku kata dari seluruh variasi jaringan, tidak terlalu menunjukkan perbedaan hasil pengujian. Hasil pengenalan pada jaringan *net\_satu\_2* untuk jenis 1 suku kata adalah yang terbaik yaitu dengan persentasi benar 15,83%. Sedangkan untuk jenis 2 suku kata, persentasi pengenalan terbaik pada jaringan *net\_dua\_1* dengan 20,83% serta jaringan *net\_tiga\_1* dan *net\_tiga\_2* untuk jenis 3 suku kata dengan persentasi benar 33,33%. Hal ini membuktikan penambahan jumlah *hidden layer* tidak terlalu berpengaruh terhadap hasil pengujian data uji tak sesuai *database*.

#### 4.4 Faktor yang Mempengaruhi Tingkat Pengenalan Pengujian Data

Berdasarkan hasil pengujian data baik data latih, data uji sesuai *database* dan data uji tak sesuai *database*, faktor-faktor yang mempengaruhi tingkat pengenalan pada aplikasi pencirian dengan *linear predictive coding* untuk pembelajaran pengucapan nama hewan dalam Bahasa Inggris menggunakan jaringan saraf tiruan propagasi balik adalah sebagai berikut.

##### 1. Kondisi Lingkungan

Karena kondisi ruangan saat perekaman tidak kedap suara sehingga memungkinkan adanya suara-suara lain yang ikut terekam. Dengan adanya derau maka akan menghasilkan sinyal suara yang tidak sesuai dengan sinyal asli. Untuk mengatasi hal ini diperlukan ruangan perekaman yang seminimal mungkin bebas derau.

##### 2. Kondisi suara responden

Kondisi suara responden sangat mempengaruhi hasil pengenalan. Setiap sinyal suara yang diucapkan oleh seseorang selalu memiliki karakteristik yang berbeda, baik itu panjang-pendek, keras-pelan dan lain-lain. Hal ini dapat diatasi dengan mengusahakan pengucapan sesuai kondisi normalnya.

##### 3. Letak Mikrofon

Karena perekaman dilakukan secara bertahap sehingga dalam peletakan mikrofon tidak sama. Jarak dan sudut

mikrofon sangat mempengaruhi pada sinyal suara yang dihasilkan. Untuk mengatasi hal ini dapat dilakukan dengan mengatur jarak mikrofon dan sudut mikrofon yang sebisa mungkin sama.

#### 4. Cara perekaman sinyal suara

Perekaman sinyal suara yang tepat juga ikut mempengaruhi akurasi pengenalan. Cara perekaman yang baik adalah ucapan diucapkan sewajar mungkin dan tidak dibuat-buat, suara diucapkan tidak terlalu keras dan tidak terlalu lemah.

#### 5. Parameter yang digunakan

Tingkat keberhasilan program dipengaruhi oleh parameter penyusun suatu jaringan. Salah satunya yaitu dengan melakukan penambahan pada jumlah *hidden layer*.

#### 6. Penggunaan *database*

*Database* sangat berpengaruh pada akurasi pengenalan, semakin banyak *database* yang digunakan maka ciri karakteristik sinyal suara yang diperoleh juga akan semakin beragam, sehingga peluang pengenalan sinyal yang diujikan akan semakin besar. Selain itu, panjang sinyal ucapan yang dijadikan *database* juga sangat mempengaruhi hasil pengenalan. Semakin panjang sinyal ucapan, maka ciri-ciri akustik untuk setiap responden akan lebih banyak disarikan sehingga pengenalan akan semakin baik.

## 5. Penutup

### 5.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil pengujian dan pembahasan adalah sebagai berikut.

1. Berdasarkan hasil pengujian data latih, tingkat pengenalan untuk masing-masing variasi jaringan dari tiap-tiap jumlah suku kata tidak menunjukkan perbedaan hasil pengujian, yaitu persentase benar 99% untuk 1 suku kata, 98,5% untuk 2 suku kata dan 100% untuk 3 suku kata.
2. Berdasarkan hasil pengujian data uji sesuai *database*, tingkat pengenalan tertinggi untuk jenis 1 suku kata adalah jaringan dengan menggunakan variasi 4 *hidden layer* dengan persentasi benar 85% sedangkan jenis 2 suku kata tingkat pengenalan tertinggi menggunakan variasi 5 *hidden layer* dengan persentasi benar 75% dan 81,67% untuk jenis 3 suku kata menggunakan 5 *hidden layer*.
3. Berdasarkan hasil pengujian data uji tak sesuai *database*, tingkat pengenalan terbaik untuk jenis 1 suku kata adalah jaringan dengan menggunakan variasi 4 *hidden layer* dengan persentasi benar 15,83% sedangkan jenis 2 suku kata tingkat pengenalan terbaik menggunakan variasi 3 *hidden layer* dengan persentasi benar 20,83% dan 33,33% untuk jenis 3 suku kata

menggunakan 3 dan 4 *hidden layer*.

4. Semakin kompleks suatu jaringan, yang ditandai dengan semakin besarnya jumlah *hidden layer*, akan menghasilkan kinerja jaringan yang semakin bagus. Hal ini ditandai dengan semakin tinggi tingkat keberhasilan pengenalan kata hingga mencapai titik dimana penambahan jumlah *hidden layer* tidak lagi berpengaruh terhadap tingkat pengenalan.
5. Semakin besar jumlah *hidden layer*, waktu pelatihan jaringan yang diperlukan juga semakin lama. Hal ini dikarenakan proses perhitungan yang semakin banyak dan rumit.

## 5.2 Saran

Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah sebagai berikut.

1. Perekaman *database* dan pengujian hendaknya dilakukan dengan ucapan responden yang asli atau menggunakan pelafalan yang baik dan benar serta diucapkan sewajarnya dan tidak dibuat-buat.
2. Karena sistem sangat peka terhadap sinyal derau, maka agar dapat dipertahankan keberhasilan pengenalannya, diperlukan suatu ruangan yang cukup tenang dan bersih terhadap sinyal derau.
3. Penelitian ini dapat dikembangkan lebih lanjut menjadi aplikasi untuk pembelajaran pengucapan berbagai macam kata dalam Bahasa Inggris dengan dilengkapi pemandu virtual.

## Daftar Pustaka

- [1]. Echols, John M. dan Shadily, Hasan, *Kamus Inggris Indonesia*, PT. Gramedia, Jakarta.
- [2]. Siang, J. J., "Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab", Penerbit Andi, Yogyakarta, 2005.
- [3]. Irfandy, Mahmud, *Aplikasi Pengenalan Ucapan Dengan Jaringan Syaraf Tiruan Propagasi Balik Untuk Pengendalian Robot Bergerak*, Skripsi S-1, Universitas Diponegoro, Semarang, 2009.
- [4]. Yudho, Theodorus, D. N., *Aplikasi Pencirian Dengan Transformasi Wavelet Untuk Pengenalan Pengucap Teks Tak Bebas Menggunakan Jaringan Saraf Tiruan*, Skripsi S-1, Universitas Diponegoro, Semarang, 2011.
- [5]. Fadhil, Febri, W. K., *Aplikasi Jaringan Saraf Tiruan Perambatan Balik Untuk Peramalan Harga Saham*, Skripsi S-1, Universitas Diponegoro, Semarang, 2011.
- [6]. Kurnianto, N. A., *Penentuan Jenis Kelamin Itik Dengan Metode Dynamic Time Warping (DTW)*, Skripsi S-1, Universitas Diponegoro, Semarang, 2006.
- [7]. Perangiangan, Kasiman, *Pengenalan Matlab*, Andi, Yogyakarta, 2006.
- [8]. Thiang dan Saputra, Hadi, *Sistem Pengenalan Kata Dengan Menggunakan Linear Predictive Coding Dan Nearest Neighbor Classifier*, Jurnal Teknik Elektro, Universitas Kristen Petra, 2005.