

## Terakreditasi SINTA Peringkat 4

Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 28/E/KPT/2019  
masa berlaku mulai Vol.3 No. 1 tahun 2018 s.d Vol. 7 No. 1 tahun 2022

Terbit online pada laman web jurnal:  
<http://publishing-widyagama.ac.id/ejournal-v2/index.php/jointecs>



Vol. 5 No. 3 (2020) 155 - 166

# JOINTECS

## (Journal of Information Technology and Computer Science)

e-ISSN:2541-6448

p-ISSN:2541-3619

### Pengenalan Suku Kata Bahasa Indonesia Menggunakan Metode *LPC* Dan *Backpropagation Neural Network*

Yenni Fatman<sup>1</sup>, Islamiyati<sup>2</sup>

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Islam Nusantara

<sup>1</sup>yennifatman@gmail.com, <sup>2</sup>islamiyati.uninus@gmail.com

#### Abstract

*Sound is the most important component in the development of digital technology today, to facilitate human life. Various speech recognition systems or Automatic Speech Recognition (ASR) have been developed in various countries with various languages. Voice recognition can be applied in various fields of life, one of which is a voice-based security system, in the form of a password. In Indonesia, there are many studies on speech recognition using the Indonesian language with a variety of methods, but it is still in limited numbers and only functions to command a particular application. Therefore, in this study, the authors conducted speech recognition based on Indonesian syllables because Indonesian itself has a syllable which is somewhat higher compared to other foreign language syllables. This system consists of 4 processes: sound recording, pre-processing, feature extraction using the Linear Predictive Code (LPC) method, and the sound classification process using the Backpropagation Neural Network method. There are 115 syllables and 74 syllables that are different from the 50 Indonesian words that are spoken. Accuracy results on the Indonesian syllable recognition system that is 100% able to recognize 74 training data from every 6 respondents and 115 testing data obtained the best accuracy of 69%.*

*Keywords: voice recognition; indonesian language; syllables; LPC; backpropagation neural network.*

#### Abstrak

Suara menjadi komponen terpenting dalam perkembangan teknologi digital saat ini, untuk mempermudah kehidupan manusia. Berbagai sistem pengenalan suara atau *Automatic Speech Recognition* (ASR) telah banyak dikembangkan di berbagai negara dengan berbagai bahasa. Pengenalan suara dapat diaplikasikan di berbagai bidang kehidupan salah satunya pada sistem keamanan berbasis suara, berupa *password*. Di Indonesia sendiri banyak penelitian mengenai pengenalan suara menggunakan bahasa Indonesia dengan berbagai metode, tetapi masih dalam jumlah yang terbatas dan hanya berfungsi untuk perintah suatu aplikasi tertentu. Oleh karena itu, pada penelitian ini, penulis melakukan pengenalan suara berdasarkan suku kata bahasa Indonesia karena bahasa Indonesia sendiri memiliki suku kata yang terbilang banyak dibandingkan dengan suku kata bahasa asing lainnya. Sistem ini terdiri dari 4 proses yaitu proses perekaman suara, proses *pre-processing*, proses ekstraksi ciri menggunakan metode *Linier Predictive Code* (LPC), dan proses klasifikasi suara menggunakan metode *Backpropagation Neural Network*. Terdapat 115 suku kata dan 74 suku kata yang berbeda dari 50 kata bahasa Indonesia yang diucapkan. Total suku kata bahasa Indonesia yang digunakan berjumlah 690 suku kata dari 6 responden. Hasil akurasi pada sistem pengenalan suku kata bahasa Indonesia yaitu 100% mampu mengenali 74 data pelatihan dari setiap 6 responden dan 115 data pengujian belum dilatih didapatkan akurasi terbaik sebesar

Diterima Redaksi : 20-04-2020 | Selesai Revisi : 25-06-2020 | Diterbitkan Online : 30-09-2020

69% dari 6 responden. Berdasarkan hasil pengujian yang telah dilakukan, semakin banyak data pelatihan yang diproses dalam jaringan maka semakin tinggi akurasi keberhasilan yang diperoleh (Sinyal suara dapat dikenali).

Kata kunci: pengenalan suara; bahasa indonesia; suku kata; LPC; *backpropagation neural network*.

© 2020 Jurnal JOINTECS

## 1. Pendahuluan

Di zaman perkembangan teknologi yang semakin pesat ini, banyak teknologi-teknologi baru yang memudahkan manusia dalam beraktivitas, terutama dalam bidang AI (*Artificial Intelligence*) atau kecerdasan buatan yang dalam pengembangan sistem komputer mampu melakukan tugas seperti halnya manusia, yang dahulu dilakukan dengan sistem analog kini terkomputerisasi.

Suara merupakan anugerah dari Allah SWT, yang dimana setiap manusia dianugerahkan suara yang berbeda-beda, memiliki karakter ciri khas masing-masing. Dengan adanya suara kita dapat berkomunikasi, mengerti dan memahami apa yang sedang dibicarakan serta salah satu cara yang paling efektif dalam menyampaikan maksud dan tujuan seseorang dalam menyampaikan suatu informasi.

Suara menjadi komponen terpenting dalam perkembangan teknologi digital saat ini, untuk mempermudah kehidupan manusia. Oleh karena itu, Berbagai sistem pengenalan suara atau lebih dikenal dengan *Automatic Speech Recognition* (ASR) telah banyak dikembangkan di berbagai negara dengan berbagai bahasa. *Speech Recognition* adalah suatu proses identifikasi suara berdasarkan kata yang diucapkan, dengan melakukan konversi sebuah sinyal akustik yang ditangkap oleh *audio device*. *Speech recognition* merupakan sistem yang digunakan untuk mengenali perintah kata dari suara manusia, dan kemudian diterjemahkan menjadi suatu data yang dimengerti oleh komputer [2].

Pengenalan suara dapat diaplikasikan di berbagai bidang kehidupan, seperti pada sistem keamanan berbasis suara, sistem keamanan rumah berbasis suara dan sistem pembelajaran berbasis suara. Sistem keamanan berbasis suara lebih efektif dan akurat dibanding dengan angka atau huruf, karena hal itu mudah untuk disadap oleh orang lain. Begitupun dengan sistem pembelajaran berbasis suara, memudahkan seorang pelajar dalam proses belajar mandiri.

Di Indonesia sendiri banyak penelitian mengenai pengenalan suara menggunakan bahasa Indonesia dan pengenalan jenis suara pria dan wanita dengan berbagai metode antara lain [2], [4], [9] dan [15] begitu juga dengan sistem analisa deteksi huruf, klastering suara serta menggunakan kata bahasa asing sebagai objek penelitian telah banyak diteliti antara lain [3], [7] dan [10] tetapi masih dalam jumlah yang terbatas dan hanya berfungsi untuk perintah suatu aplikasi tertentu. Tingkat pengenalan dipengaruhi oleh metode ekstraksi dan klasifikasi yang digunakan [9]. Dalam penelitian

ini, menggunakan metode *Linier Predictive Code* (LPC) untuk ekstraksi ciri dan metode *Backpropagation Neural Network* untuk klasifikasi suara.

Metode *Linier Predictive Code* (LPC) merupakan salah satu teknik ekstraksi ciri yang sering digunakan dalam mengekstraksi ciri sinyal digital suara. Ekstraksi ciri suara adalah untuk mengubah gelombang suara menjadi beberapa tipe representasi parametrik yang dapat diproses. Tahap-tahap yang dilakukan adalah *Pre-emphasis*, *Frame Blocking*, *Windowing*, *Analisis Autokorelasi* dan *Analisis LPC* [13].

Sejauh ini metode LPC banyak digunakan dalam penelitian terkait identifikasi suara, seperti pada beberapa penelitian sebelumnya yang dilakukan oleh Yamina Azmi, LPC digunakan untuk identifikasi emosi seseorang dari sinyal suara secara *real time* [1], Heriyanto, LPC digunakan untuk mengidentifikasi ucapan warna bagi anak-anak dalam mengenal berbagai warna [8], Musliha Syam, LPC digunakan untuk mengidentifikasi asal daerah berdasarkan suara manusia [16], Ricki Juniansyah, LPC digunakan untuk mengenali suara manusia sebagai *password* [12], dan Ririen Kusumawati, LPC digunakan untuk menganalisis kesesuaian pada sinyal suara pengucapan kata *arabic* pada penutur Indonesia [13].

*Backpropagation* merupakan salah satu algoritma dengan pendekatan *supervised learning*, yang digunakan untuk mengurangi *error* melalui penyesuaian dari bobotnya berdasarkan perbedaan *output* dan target yang diinginkan melalui perambatan mundur sehingga nilai dapat diperbaiki dalam melakukan pengenalan pola. Terdapat 3 fase *Backpropagation* yaitu fase *feed forward*, fase *backpropagation* dan fase menyesuaikan bobot [6].

Adapun metode *Backpropagation Neural Network* yang telah diimplementasikan oleh beberapa penelitian terdahulu yang dilakukan oleh Desyanndana, *Backpropagation* digunakan untuk klasifikasi perintah suara [4], Faradiba, *Backpropagation* digunakan untuk pengenalan pola sinyal suara manusia [5], John Pierre, *Backpropagation* digunakan untuk pengenalan pola notasi balok [6], dan Erina Nursholihatun, *Backpropagation* digunakan untuk mengenali identitas penutur dengan membandingkan fitur suara masukkan [14]. Kedua metode tersebut memiliki kekurangan dan kelebihan masing-masing.

Pada dasarnya bahasa Indonesia merupakan bahasa yang sering digunakan dalam berkomunikasi dan menjadi bahasa induk bagi warganegara Indonesia. Tetapi setiap orang memiliki pembawaan suara sendiri

sesuai dengan karakter masing-masing serta logat daerah yang sangat berpengaruh dalam pengucapan bahasa Indonesia.

Dalam penuturan sebuah kata bahasa Indonesia setiap suku katanya akan diiringi oleh sebuah hembusan nafas. Hal ini memungkinkan adanya jeda antar suku kata dan menghasilkan representasi sinyal suara kata bahasa Indonesia yang berbeda untuk tiap suku katanya. Sehingga suara tutur bahasa Indonesia dapat dikenali pada tingkat suku kata [9].

Hal tersebut menjadi salah satu metode untuk belajar bahasa Indonesia yang baik dan benar, agar mudah dikenali oleh suatu sistem. Tidak banyak penelitian mengenai pengenalan suku kata bahasa Indonesia karena bahasa Indonesia sendiri memiliki suku kata yang terbilang banyak dibandingkan dengan suku kata bahasa asing lainnya.

Oleh karena itu, dalam penelitian ini akan merancang suatu sistem pengenalan ucapan berdasarkan suku kata bahasa Indonesia dengan mengimplementasikan metode *Linier Predictive Code (LPC)* dan metode *Backpropagation Neural Network* untuk mengenali sinyal suara beserta nilai akurasi yang didapatkan dalam pengenalan ucapan suku kata bahasa Indonesia.

**2. Metode Penelitian**

Pada penelitian ini, digunakan metode dengan pendekatan kualitatif dan metode *waterfall* sebagai model pendekatan pengembangan *software* yang memiliki beberapa tahapan terstruktur yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian dan pemeliharaan [8].

Algoritma yang digunakan untuk membangun sistem pengenalan suku kata bahasa Indonesia yaitu metode *Linier Predictive Code* dan metode *Backpropagation Neural Network*.

**2.1. Pengumpulan Data**

Tahap pertama yang dilakukan pada saat pengumpulan data yaitu studi literatur, tahapan ini bermaksud untuk mencari sumber pustaka dan mendapatkan informasi yang jelas untuk mendukung pembuatan dasar teori yang kuat serta metode yang akan digunakan pada sistem pengenalan ucapan.

Tahap kedua, studi lapangan adalah pengumpulan data secara langsung ke lapangan dengan cara observasi untuk mencari responden dan data kata, studi dokumentasi dengan melakukan perekaman suara dan *focus group discussion*.

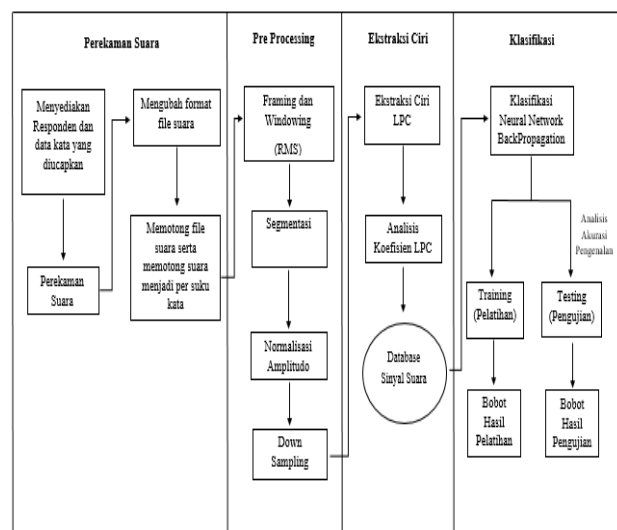
Tahap ketiga, pengumpulan sampel data suara berupa kata kerja, kata benda dan kata sifat dalam bahasa Indonesia. Terdapat 50 kata bahasa Indonesia yang difokuskan pada 1 suku kata, 2 suku kata dan 3 suku kata. Berjumlah 115 suku kata dan 74 suku kata yang berbeda dari 50 kata. Total suku kata 690 suku kata

dari 6 responden yaitu 3 laki-laki dan 3 perempuan yang merupakan Mahasiswa Fakultas Teknik Informatika Universitas Islam Nusantara.

**2.2. Perancangan Sistem**

Perancangan sistem merupakan tahap awal dalam membangun sistem pengenalan suku kata bahasa Indonesia. Perancangan tersebut meliputi struktur proses yang terdiri dari proses perekaman data sinyal suara, proses *pre processing*, proses ekstraksi ciri, dan proses klasifikasi yang terbagi dalam 2 tahap proses yaitu proses pelatihan dan proses pengujian yang akan di representasikan dengan *Software Matlab R2016*.

*Software* pendukung seperti *Format Factory* dan *Adobe Audition* merupakan *software* yang digunakan untuk mengubah format file suara dan memotong file suara yang hanya terjadi pada tahap awal saja yaitu proses perekaman. Struktur proses sistem pengenalan suku kata bahasa Indonesia secara umum yang dilakukan pada penelitian ini, dapat dilihat pada Gambar 1.



Gambar 1. Struktur Proses Sistem Pengenalan Suku Kata Bahasa Indonesia

Berdasarkan Gambar 1 diatas, terdapat struktur proses dari sistem pengenalan suku kata bahasa Indonesia yaitu pertama, proses perekaman suara dimulai dari menyediakan responden dan data kata yang diucapkan. Pada penelitian ini, terdapat 6 responden yang mengucapkan 50 kata berdasarkan pengucapan suku kata.

Tabel 1. Responden

| Laki-laki | Perempuan  |
|-----------|------------|
| Arif      | Nani       |
| Jejen     | Devi       |
| Rafsan    | Islamiyati |

Pada Tabel 1, terdapat mahasiswa yang terpilih sebagai responden dalam penelitian ini. Responden melakukan perekaman suara sebanyak satu kali dengan mengucap-

kan 50 kata berdasarkan persuku kata yang sudah disiapkan.

Tabel 2. Sampel Data

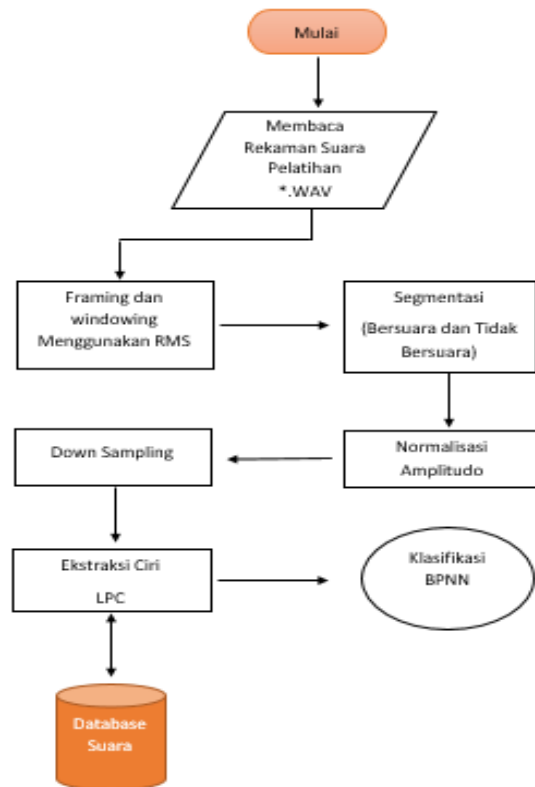
| 1 Suku Kata | 2 Suku Kata | 3 Suku Kata |
|-------------|-------------|-------------|
| Cat         | I bu        | Be la jar   |
| Bom         | Di a        | Be ker ja   |
| Lap         | Do a        | Men ca ri   |
| Bor         | A ir        | Me re ka    |
| Om          | A ku        | Me li hat   |
|             | Ka mu       | Mem be ri   |
|             | Ka mi       | Ber ma in   |
|             | Per gi      | Ber ja lan  |
|             | Du duk      | Ber la ri   |
|             | Mi num      | Me nu lis   |
|             | Ma kan      | Mem ba ca   |
|             | Ba ru       | Ber te mu   |
|             | Can tik     | Ber di ri   |
|             | Sa kit      | Me mang gil |
|             | Pa nas      | Ber ba gi   |
|             | Ba ik       | Me nyim pan |
|             | Ma nis      | Me na ngis  |
|             | Mu rah      | Me nya nyi  |
|             | Mu dah      | Ter se nyum |
|             | Ru mah      | Me na rik   |
|             | Da tang     |             |
|             | To long     |             |
|             | Hi lang     |             |
|             | Ri ngan     |             |
|             | Be nar      |             |

Sebelum melakukan perekaman suara, dilakukan pengumpulan sampel data seperti pada Tabel 2. Sampel data kata terdiri dari 50 kata dengan 115 suku kata yang akan diucapkan saat proses perekaman. Perekaman suara dilakukan pada kondisi tenang dan kedap suara. Direkam dengan menggunakan *HandPhone* ASUS Zenfone 2-ZE550ML dan disimpan dalam format \*.3GPP. Suara yang dihasilkan dari perekaman ini yaitu kata yang akan menjadi *database* dalam sistem pengenalan suku kata bahasa Indonesia.

Setelah dilakukan perekaman suara, selanjutnya mengubah format file suara menjadi format \*.Wav dengan menggunakan aplikasi *Format Factory* dengan tujuan untuk menyamakan format file data suara yang akan dimasukkan ke dalam sistem. Data hasil proses perekaman suara berformat \*.Wav akan dipotong sesuai dengan data kata yang sudah disediakan sebelumnya. Pada penelitian ini, proses pemotongan file suara diperlukan untuk mengambil satu persatu kata yang terdapat dalam file rekaman. Pemotongan file suara terbagi menjadi 2 bagian yaitu pemotongan suara secara manual dan pemotongan suara secara otomatis. Untuk tahapan ini menggunakan pemotongan suara secara manual menggunakan aplikasi *Adobe Audition*.

Proses kedua, pada Gambar 1 diatas yaitu proses *Pre-processing* merupakan proses pengolahan awal terhadap data rekaman dengan mengurangi efek dari perekaman dan melakukan penguatan sinyal digital

hasil perekaman, serta mempersiapkan data pada bentuk yang tepat untuk proses ekstraksi ciri. Ekstraksi Ciri atau *feature extraction* dilakukan pada dua proses, yaitu ekstraksi ciri untuk pembuatan *database* dan ekstraksi ciri masukkan data pengujian.



Gambar 2. Flowchart Skema Pre-processing dan Ekstraksi Ciri

*Flowchart* skema *Pre-processing* dan ekstraksi ciri dari alur sistem yang dibuat digambarkan dengan sederhana menggunakan simbol yang mudah untuk dimengerti. *Flowchart* tersebut ditunjukkan pada Gambar 2, dimulai dari membaca rekaman suara pelatihan \*.WAV yang dilanjutkan dengan proses *Framing dan windowing* menggunakan RMS (*Root Mean Square*) untuk melihat energi pada sinyal suara agar dalam proses pemotongan suara dapat memotong dengan tepat dan mudah.

*Root Mean Square* adalah akar dari nilai rata-rata dari suatu fungsi yang dikuadratkan. Untuk menghitung nilai RMS atau efektif suatu fungsi, maka yang pertama kali dilakukan adalah mengkuadratkan fungsi tersebut. Seperti pada rumus 1.

$$x_{RMS} = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}} \quad (1)$$

dengan  $x_{RMS}$  adalah nilai RMS,  $n$  adalah banyaknya data dan  $x_i$  adalah nilai data ke- $i$  [18].

Pada penelitian ini, melakukan percobaan dengan memasukkan nilai *window* pada sinyal suara inputan diantara nilai 50, 75 dan 100 untuk menentukan nilai

*window* yang optimal untuk digunakan pada sampel sinyal suara yang lain.

Jika rumus 1 RMS diimplementasikan dalam aplikasi MATLAB, maka akan seperti dibawah ini.

---

#### Program Jurnal

---

```
x1=x(:,1)
ldata = length(x1)-100
for y = 1:ldata
    a = 0;
    for i = 1:75
        a = a + x1(y+i)^2;
    end
    rms6_cat(y) = sqrt(a);
end
```

---

Langkah selanjutnya pada Gambar 2, yaitu proses Segmentasi merupakan proses memisahkan bagian bersuara dan bagian tidak bersuara atau memisahkan suara dari *noise*. Segmentasi ini dilakukan berdasarkan suku kata.

Jika proses segmentasi diimplementasikan dalam aplikasi MATLAB, maka akan seperti dibawah ini.

---

#### Program Jurnal

---

```
threshold (i) = 0;
threshold (i) = 1;
for i = 1:ldata
    if rms6_cat (i) > 0.009
        threshold (i) = 1;
    else
        threshold (i) = 0;
    end
end

flag = 0;
n = 1;

for i = 1 :ldata
    if threshold(i) == 1;
        segm6_cat (n) = x1 (i);
        n = n+1;
        flag = 1 ;
    else
        if flag == 1 ;
            i = ldata ;
        end
    end
end
```

---

Berdasarkan pada Gambar 2, terdapat proses Normalisasi Amplitudo yaitu proses yang digunakan untuk menormalkan degradasi nilai sampel sinyal digital yang diakibatkan perbedaan jarak antara mulut dan mikrofon perekam. Proses normalisasi amplitudo diperoleh dengan membagi semua nilai sampel sinyal digital dengan nilai *absolut* maksimum dari sampel sinyal digital tersebut. Dapat dilihat pada rumus 2.

$$x'(n) = \frac{x(n)}{\max(|x|)}, \quad 0 \leq n \leq N - 1 \quad (2)$$

dengan  $x'(n)$  adalah nilai hasil normalisasi amplitudo,  $x(n)$  adalah nilai sampel sinyal digital,  $\max(|x|)$  adalah nilai absolut maksimum dan N adalah panjang signal [17].

Jika rumus 2 normalisasi diimplementasikan dalam aplikasi MATLAB, maka akan seperti dibawah ini.

---

#### Program Jurnal

---

```
norm6_cat(i) = segm6_cat(i)/max(segm6_cat);
```

---

Pada Gambar 2 diatas, proses *Down Sampling* bertujuan untuk menurunkan atau mengurangi jumlah sampel sinyal suara yang terlalu banyak, namun hasil proses *down sampling* ini masih bisa mewakili sampel-sampel yang dihilangkan dengan menentukan nilai *sampling* yang diinginkan. Nilai *down sampling* yang digunakan yaitu 500 *sampling* yang diambil dari setengah sampel sinyal suara asli.

Jika proses *down sampling* diimplementasikan dalam aplikasi MATLAB, maka akan seperti dibawah ini.

---

#### Program Jurnal

---

```
x = norm6_cat
pjdata = length(x);
ldata = pjdata/500 ;

a = floor (ldata);
for i = 1 : 500
    b = i*a;
    dsc6_cat(i) = x(b);
end
```

---

Berdasarkan pada Gambar 2 diatas, setelah proses *down sampling*, selanjutnya ke proses utama dalam melakukan ekstraksi ciri LPC, yaitu proses analisis koefisien LPC. Analisis koefisien LPC merupakan proses untuk menentukan nilai parameter *Linear Prediction Coding* (LPC) dengan menganalisis setiap nilai masukkan untuk memperoleh nilai yang terbaik. Hasil dari analisis koefisien LPC merupakan hasil sinyal suara yang diprediksi.

Pada penelitian ini, menggunakan *toolbox* LPC dan menganalisis nilai koefisien dari nilai 50 dan nilai 100 sebagai parameter untuk menentukan nilai koefisien yang terbaik. Dengan menganalisis nilai koefisien LPC, maka akan terlihat perbedaan ciri sinyal digital suara dengan nilai koefisien 50 dan nilai koefisien 100. Sinyal digital suara yang terlihat signifikan perbedaannya pada nilai koefisien yang sudah ditentukan, maka nilai koefisien tersebut optimal untuk digunakan pada sinyal suara lainnya.

Jika diimplementasikan dengan memanfaatkan *toolbox Linear Predictive Coding* (LPC) dalam aplikasi MATLAB, maka akan seperti dibawah ini.

---

#### Program Jurnal

---

```
% pada command window
cat1_lpc=lpc(cat1,50);
Plot (cat1_lpc)
```

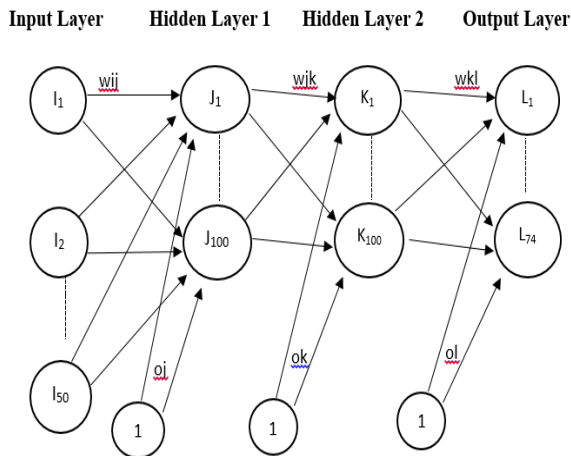
---

Proses klasifikasi suara *Backpropagation Neural Network* berdasarkan pada Gambar 2, digunakan untuk mengklasifikasi suatu pola sinyal masukkan tertentu dengan melakukan perbaikan bobot penghantar antar lapisan. Pada proses klasifikasi terbagi menjadi 2 bagian yaitu proses *training* (pelatihan) dan proses *testing* (pengujian). Dalam *backpropagation* jika data yang digunakan terlalu besar maka sistem akan membutuhkan waktu yang lama untuk menemukan pola mencapai target. Sebaliknya, jika data terlalu sedikit maka sistem tidak dapat mengenali data dengan baik sehingga sulit untuk mencapai target. Oleh karena itu, dalam penelitian ini dilakukan pembagian data yaitu data untuk pelatihan (74 suku kata) dan data untuk pengujian (115 suku kata).

Tabel 3. Sampel Data Pelatihan 74 Suku Kata

| Sampel Data Pelatihan |     |      |     |      |
|-----------------------|-----|------|-----|------|
| Cat                   | Per | Ik   | La  | Lis  |
| Bom                   | Gi  | Nis  | Jar | Te   |
| Lap                   | Du  | Rah  | Ker | Mang |
| Bor                   | Duk | Dah  | Ja  | Gil  |
| Om                    | Num | Mah  | Men | Nyim |
| I                     | Ma  | Da   | Ca  | Pan  |
| Bu                    | Kan | Tang | Me  | Na   |
| Di                    | Ba  | To   | Re  | Ngis |
| A                     | Ru  | Long | Li  | Nya  |
| Do                    | Can | Hi   | Hat | Nyi  |
| Ir                    | Tik | Lang | Mem | Ter  |
| Ku                    | Sa  | Ri   | Ber | Se   |
| Ka                    | Kit | Ngan | In  | Nyum |
| Mu                    | Pa  | Be   | Lan | Rik  |
| Mi                    | Nas | Nar  | Nu  |      |

Sampel data pada Tabel 3, merupakan sampel data yang diambil dari 115 suku kata sehingga didapatkan 74 suku kata yang berbeda seperti pada tabel diatas. Untuk sampel data pengujian 115 suku kata dapat dilihat pada Tabel 2.



Gambar 3. Arsitektur Jaringan Backpropagation (Pada Pengenalan Suku Kata Bahasa Indonesia)

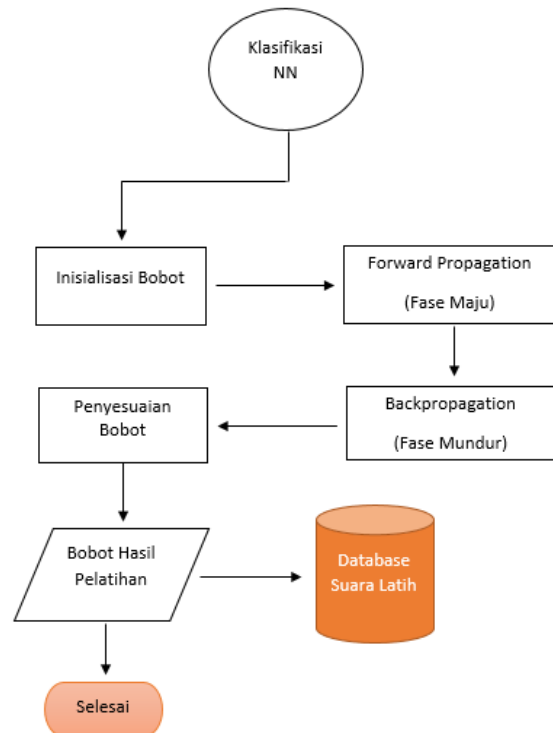
Pada Gambar 3, merupakan Arsitektur Jaringan *Backpropagation* yang digunakan pada penelitian ini yang terdiri dari empat layer yaitu *input layer*, *hidden layer 1*, *hidden layer 2* dan *output layer*. Inputan divariabelkan dengan (In), *Hidden layer 1* divariabelkan (Jn) dengan bobot ( $w_{ij}$ ) dan bias ( $o_j$ ), *Hidden layer 2* divariabelkan (Kn) dengan bobot ( $w_{jk}$ ) dan bias ( $o_k$ ) dan pada *Output layer* divariabelkan (Ln) dengan bobot ( $w_{kl}$ ) dan bias ( $o_l$ ).

Pada *Input layer* tidak terjadi proses komputasi, hanya terjadi pengiriman sinyal input I ke *hidden layer*. Pada *hidden layer* dan *output layer* terjadi proses komputasi terhadap bobot dan bias serta dihitung besarnya *output* dari *hidden* dan *output layer* berdasarkan fungsi aktivasi sigmoid biner karena *output* yang diharapkan bernilai antara 0 sampai 1.

Tabel 4. Parameter Arsitektur Jaringan Backpropagation

| Input Layer | Hidden Layer 1 | Hidden Layer 2 | Output Layer |
|-------------|----------------|----------------|--------------|
| 50          | 100            | 100            | 74           |

Sebelum melakukan proses pelatihan, terlebih dahulu menentukan nilai parameter yang tepat pada jaringan untuk dapat mengenali suara dengan baik, seperti pada Tabel 4 dimana *input layer* memiliki nilai 50 diambil dari nilai koefisien LPC, *hidden layer 1* dan *hidden layer 2* memiliki nilai 100 diambil dari nilai terbesar yang melebihi nilai *output layer* dan nilai *output layer* 74 diambil berdasarkan data sampel suara yang dilatih.



Gambar 4. Flowchart Sistem Pengenalan Skema Pelatihan

Pada proses pelatihan, sistem akan menerima masukan berupa sampel yang akan dijadikan sebagai data pelatihan. Data pelatihan akan disimpan dalam *database* yang akan dijadikan acuan dalam proses pengujian.

Langkah selanjutnya berdasarkan *flowchart* Gambar 4 diatas yaitu proses fase maju (*Forward Propagation*) bertujuan untuk menelusuri besarnya *error*. Sesuai dengan arsitektur jaringan *backpropagation* maka didapatkan rumus perhitungan seperti pada rumus 3.

$$y_j = \sigma \left( \sum_{i=1}^n w_{ij} x_i + b_j \right) \quad (3)$$

dengan  $y_j$  adalah output,  $\sigma$  (sigma) adalah simbol dari fungsi aktivasi,  $n$  adalah banyaknya neuron di layer input,  $w_{ij}$  adalah nilai bobot input layer dan hidden layer,  $x_i$  adalah input  $x$  dan  $b_j$  adalah nilai bias [1].

Fungsi aktivasi yang digunakan yaitu fungsi aktivasi sigmoid biner. Fungsi aktivasi sigmoid biner memiliki nilai pada range 0 sampai 1 [11]. Didefinisikan seperti pada rumus 4.

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (4)$$

dengan  $\sigma()$  adalah simbol dari fungsi aktivasi,  $x$  adalah nilai sinyal keluaran dari satu *neuron* yang akan diaktifkan dan  $e$  adalah nilai konstanta dengan nilai = 2.718281828 [1].

Jika rumus 3 dan rumus 4 diimplementasikan dalam aplikasi MATLAB, maka akan seperti dibawah ini.

---

#### Program Jurnal

---

```

clc;
clear all;

%proses pelatihan

load('arif_mat.mat');
eta = 0.05;
wij = random('normal',0,1,51,101);
wjk = random('normal',0,1,101,101);
wk1 = random('normal',0,1,101,74);

a = 0;
RMSE = 10;
while a < 5000
    for p = 1:74
        for j = 1:100
            netj(j) = 0;
            i(51)=1;
            for i = 1:51
                netj(j) = netj(j)+(wij(i,j)*arif_matrix(p,i));
            end
            oj(j) = logsig(netj(j));
        end
        oj(101)=1;

        for k = 1:100
            netk(k) = 0;
            for j = 1:101
                netk(k) = netk(k)+(wjk(j,k)*oj(j));
            end
            ok(k) = logsig(netk(k));
        end
        ok(101)=1;
    end
end

```

Menghitung *Error* (RMSE) dapat dilihat pada rumus 5.

$$error = \frac{1}{n} \sum_{i=1}^n (target_i - prediksi_i)^2 \quad (5)$$

dengan  $n$  adalah jumlah data,  $i$  adalah urutan data pada *database*,  $target_i$  adalah nilai hasil target dan  $prediksi_i$  adalah nilai hasil prediksi [1].

Jika rumus 5 diimplementasikan dalam aplikasi MATLAB, maka akan seperti dibawah ini.

---

#### Program Jurnal

---

```

temprmse = 0;
for l = 1:74
    netl(1) = 0;
    for k = 1:101
        netl(1) = netl(1)+(wk1(k,1)*ok(k));
    end
    ol(1) = logsig(netl(1));

    if p == 1
        err(1) = 1-ol(1);
    else
        err(1) = 0-ol(1);
    end
    temprmse = temprmse+(err(1)^2);
end

temprmse = 0.5*temprmse;
if (RMSE>temprmse)
    RMSE = temprmse
end

errk = zeros(1,101);
errj = zeros(1,101);
for k=1:101
    for l=1:74
        errk(k) = errk(k)+(wk1(k,l)*err(l));
    end

    for j=1:101
        for k=1:101
            errj(j) = errj(j)+(wjk(j,k)*errk(k));
        end
    end
end

```

Fase mundur (*Backpropagation*) pada Gambar 4 diatas, bertujuan untuk mengupdate bobot dengan melakukan perhitungan balik dari *neuron* keluaran agar memiliki nilai bobot yang sesuai. Pada penelitian ini, dalam mengupdate bobot dilakukan tanpa menggunakan momentum. Dan menyesuaikan bobot, jika keluaran dari jaringan berbeda dengan target yang diharapkan maka jaringan melakukan penyesuaian terhadap bobot yang ada. Proses tersebut akan terus berjalan sampai keluaran pada jaringan dan target sama.

Memperbaiki suatu bobot ( $w$ ) berdasarkan *error* ( $E$ ) dengan rumus 6.

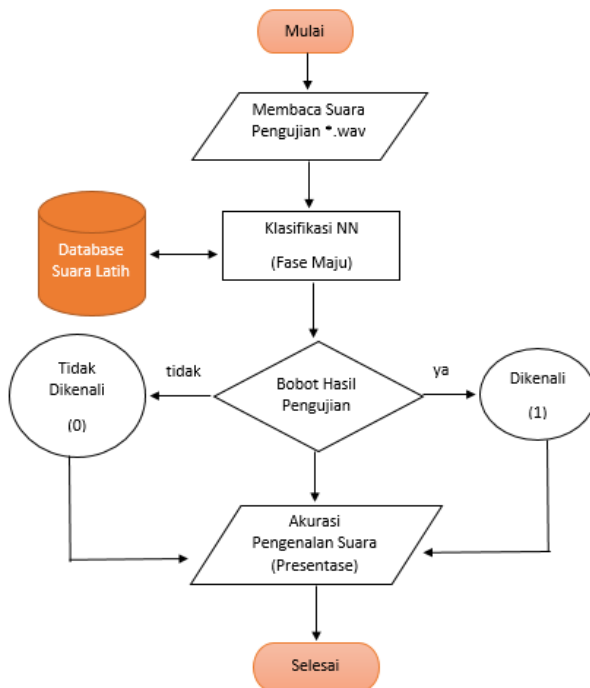
$$w_{new} = w_{old} - \alpha \frac{\partial E}{\partial w} \quad (6)$$

dengan  $w_{new}$  adalah bobot baru,  $w_{old}$  adalah bobot lama,  $\alpha$  adalah learning rate dan  $\frac{\partial E}{\partial w}$  adalah mencari nilai turunan  $E$  terhadap variabel yang akan diperbaharui ( $w$ ) [1].

Jika rumus 6 diimplementasikan dalam aplikasi MATLAB, maka akan seperti dibawah ini.

**Program Jurnal**

```
%perubahan bobot
for i=1:51
    for j = 1:100
        wij(i,j)=wij(i,j)+(eta*errj(j)*arif_matrix
        (p,i)*(logsig(netj(j))*(1-logsig(netj(j)))));
    end
end
for j=1:101
    for k = 1:100
        wjk(j,k)= wjk(j,k)+(eta*errk(k)*oj(j)*
        (logsig(netk(k))*(1-logsig(netk(k)))));
    end
end
for k=1:101
    for l = 1:74
        wk1(k,l)=wk1(k,l)+(eta*errl(l)*ok(k)*(logsig(ne
        t1(l))*(1-logsig(net1(l)))));
    end
end
end
a = a+1;
terr(a)=RMSE
end
```



Gambar 5. Flowchart Sistem Pengenalan Skema Pengujian

Proses terakhir yaitu proses *testing* (pengujian) yang ditunjukkan pada *flowchart* Gambar 5 diatas, sistem akan diuji dengan cara memasukkan sampel dan dibandingkan dengan data pelatihan yang ada, kemudian hasil akan keluar berdasarkan kemiripan dari data pelatihan.

Data suara yang digunakan merupakan sampel data pengujian 115 suku kata dan sampel data pelatihan 74

suku kata untuk dapat dikenali sesuai klasifikasi yang sebenarnya atau tidak dapat dikenali sama sekali serta keakurasian dari pengenalan suara dihitung dalam skala presentase. Pada tahap pengujian hanya melakukan satu kali pengujian yaitu pada fase maju yang akan menghasilkan bobot pengujian.

Jika proses pelatihan diimplementasikan dalam aplikasi MATLAB, maka akan seperti dibawah ini.

**Program Jurnal**

```
clc;
clear all;

%proses pengujian

load('arif_bobot')
load('arif_matriks115.mat');
a = 0;
for p = 1:115
    for j = 1:100
        netj(j) = 0;
        i(51)=1;
        for i= 1:51
            netj(j)=netj(j)+(wij(i,j)*arif_matriks115(p,i)
            );
        end
        oj(j) = logsig(netj(j));
    end
    oj(101)=1;
    for k = 1:100
        netk(k) = 0;
        for j = 1:101
            netk(k) = netk(k)+(wjk(j,k)*oj(j));
        end
        ok(k)= logsig(netk(k));
    end
    ok(101)=1;
    for l = 1:74
        net1(l) = 0;
        for k = 1:101
            net1(l) = net1(l)+(wk1(k,l)*ok(k));
        end
        ol(p,l) = logsig(net1(l));
    end
end
end
```

Presentase keberhasilan sistem dapat diketahui dengan rumus 8 sebagai berikut :

$$Hasil Akurasi = \frac{\sum data sukses}{\sum data input} \times 100\% \quad (8)$$

dengan  $\Sigma$  data sukses adalah jumlah data uji yang berhasil dikenali dan  $\Sigma$  data input adalah jumlah data masukkan keseluruhan yang akan di uji [5]. Dengan rumus tersebut, kita dapat mengetahui tingkat keberhasilan sistem.

**3. Hasil dan Pembahasan**

Tujuan yang ingin dicapai pada sistem ini yaitu sebuah sistem yang dapat mengenali suara masukkan berupa suku kata dengan berbagai macam jenis dan variasi yang diucapkan oleh masing-masing responden. Sistem hanya dapat mengenali suara dari orang yang telah dilatih, sehingga jika diberikan suara masukkan yang



tidak tersimpan dalam *database*, suara tidak dapat dikenali.

Langkah-langkah proses pengenalan suara yang telah dilakukan sesuai dengan prosedur yang digambarkan pada Gambar 1 ataupun *flowchart* Gambar 2, Gambar 3 dan Gambar 5, diimplementasikan pada program Matlab dan hasil dari proses tersebut akan terlihat setelah program Matlab dijalankan.

Berikut ditampilkan hasil dari proses perekaman suara yang sudah dilakukan. Sampel suara ini sebagai inputan yang direpresentasikan menjadi sinyal ucapan dalam bentuk matriks dengan cara memberikan perintah *audioread* di MATLAB seperti pada berikut ini.

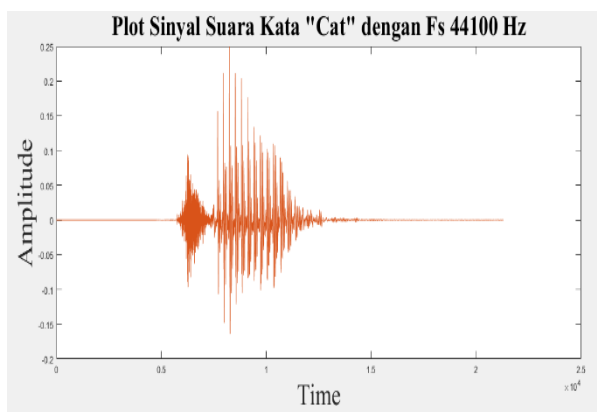
---

#### Program Jurnal

---

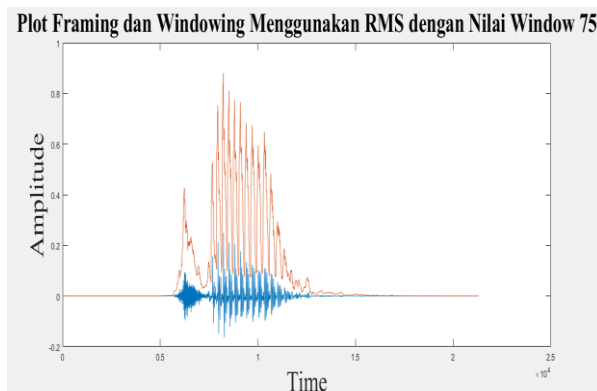
```
[x,fs]=audioread('D:\SuaraStudio\Arif\1sukukat
a\arif1a.wav');
```

---



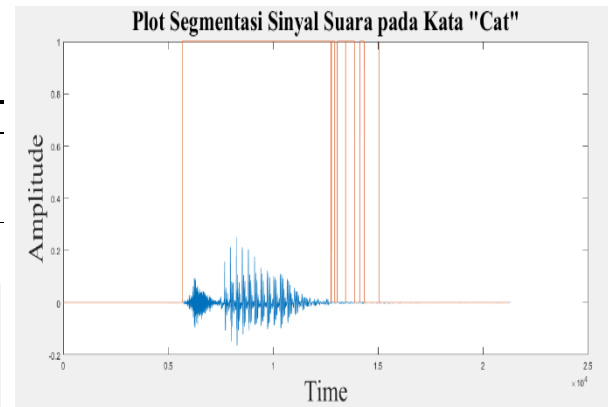
Gambar 6. Representasi .wav dalam Bentuk Grafik Pada Kata 'Cat'

Grafik pada Gambar 6 diatas, menunjukkan hasil dari perintah program matlab yang direpresentasikan dalam bentuk grafik. Sampel sinyal suara diambil dari salah satu responden dalam bentuk file \*.WAV dengan salah satu kata yaitu "Cat" yang sudah dipotong sesuai per suku kata dengan frekuensi 44100 Hz.



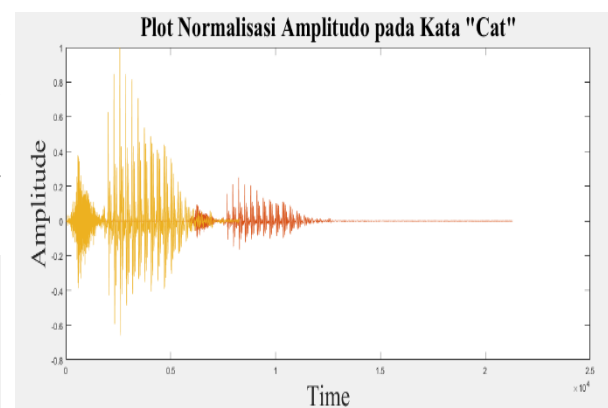
Gambar 7. Grafik Framing dan Windowing Menggunakan RMS dengan Nilai Window 75

Grafik pada Gambar 7, merupakan salah satu hasil dari proses *framing dan windowing* menggunakan RMS dengan kata "Cat" yang direpresentasikan dalam bentuk grafik. Memasukkan nilai *windowing* pada sinyal suara inputan yaitu 75 yang telah ditentukan. Berdasarkan gambar tersebut terlihat bentuk energi pada sinyal suara yang dimana nilai window 75 terlihat optimal, tidak terlalu rapat ataupun longgar untuk digunakan pada sampel sinyal suara yang lainnya. Memberikan nilai *window* ini bertujuan untuk mempermudah proses pemotongan pada tahap segmentasi.



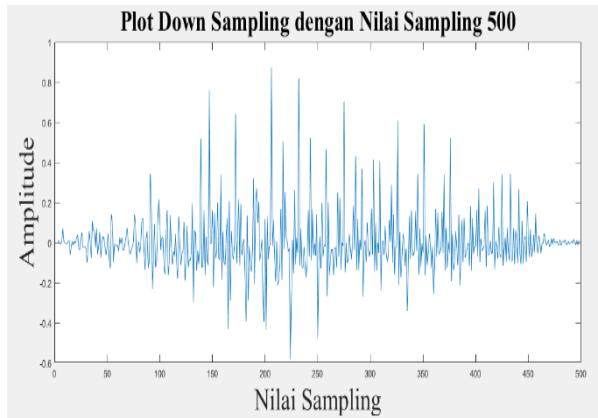
Gambar 8. Grafik Segmentasi Sinyal Suara Pada Kata 'Cat'

Pada Gambar 8 diatas, merupakan hasil dari segmentasi. Terlihat dengan menentukan nilai *threshold* akan secara otomatis menampilkan hasil dari memisahkan sinyal suara dari *noise* dengan mencari nilai *energy & Zero Crossing* dari sampel suara. Untuk nilai *threshold* pada setiap sampel suara berbeda-beda sesuai dengan kondisi suara yang akan dipotong.



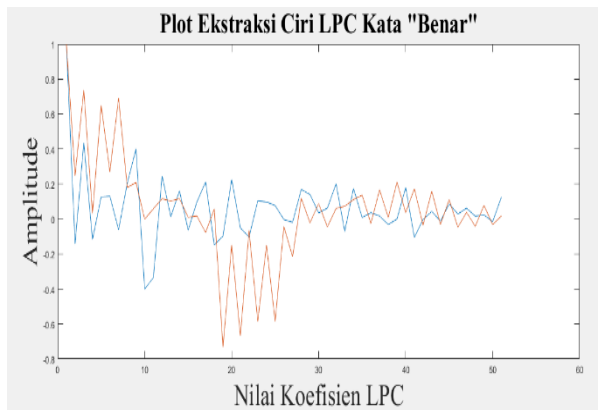
Gambar 9. Grafik Normalisasi Amplitudo Pada Kata 'Cat'

Pada Gambar 9 diatas, terlihat hasil normalisasi amplitudo dengan menormalkan degradasi nilai sampel sinyal suara agar memiliki nilai amplitudo yang sama yaitu 1. Sampel sinyal suara yang memiliki amplitudo yang berbeda seperti memiliki nilai amplitudo 0.25 ataupun 0.8 akan dinormalisasikan menjadi 1 sehingga sampel sinyal suara memiliki nilai amplitudo yang setara yaitu maksimal 1.



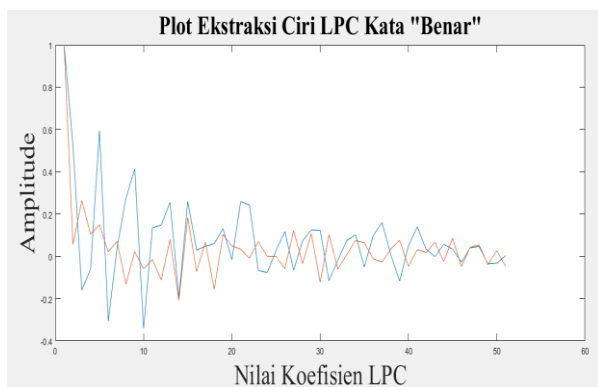
Gambar 10. Grafik Down Sampling dengan Nilai Sampling 500

Grafik pada Gambar 10 diatas, merupakan hasil menurunkan nilai *sampling* dari salah satu sinyal suara dengan kata “Cat” yang diambil dari responden perempuan. Nilai *sampling* diambil dari setengah nilai sinyal suara yaitu 500 *sampling* dengan nilai amplitudo 1. Sinyal suara terlihat tidak padat ataupun padat sekali sehingga dapat mempermudah dalam ekstraksi ciri. Nilai *sampling* yang sudah ditentukan tersebut bisa mewakili sampel-sampel yang hilang.



Gambar 11. Nilai Koefisien LPC 50, down sampling 500

(Responden 1 Laki-laki)



Gambar 12. Nilai Koefisien LPC 50, down sampling 500

(Responden 2 Perempuan)

Pada Gambar 11 dan Gambar 12 diatas, merupakan hasil ekstraksi ciri dengan memberikan nilai koefisien LPC 50 yang sudah ditentukan dan nilai *Down Sampling* 500 untuk mendapatkan ciri dari sinyal suara. Salah satu kata yang diucapkan yaitu kata “Benar” dengan 2 responden yang berbeda. Hasil ekstraksi ciri LPC yang direpresentasikan dalam bentuk grafik tersebut terlihat perbedaan yang signifikan dari kedua sinyal suara, masing-masing memiliki ciri sinyal suara yang berbeda. Hasil dari ekstraksi ciri LPC ini akan menjadi *database* dalam pelatihan dan pengujian.

### 3.1. Hasil Pelatihan

Hasil pelatihan disusun dalam sebuah tabel sehingga dapat dibandingkan untuk menentukan *learning rate* dan kondisi *epoch* tercepat. Adapun hasil perubahan dari *learning rate* dan *epoch* seperti pada Tabel 5.

Tabel 5. Hasil Percobaan Penentuan Leraning Rate Pada 1 Data

| Pelatihan |               |        |
|-----------|---------------|--------|
| Epoch     | Learning Rate | RMSE   |
| >= 3000   | 0,01          | 0,0244 |
|           | 0,02          | 0,0193 |
|           | 0,05          | 0,0149 |

Pada Tabel 5 diatas, terlihat perbedaan dari setiap nilai *learning rate* dan RMSE dengan nilai *epoch* yang sama. Menunjukkan bahwa nilai *learning rate* yang paling optimal terjadi pada nilai 0,05 dengan nilai *error* (RMSE) sebesar 0,0149 memiliki nilai *error* terkecil dari nilai yang lainnya. Hanya saja nilai pada *epoch* kurang optimal karena hasil *error* belum mencapai target. Semakin besar nilai *learning rate* semakin cepat proses pelatihan tergantung pada nilai *epoch* yang ditentukan.

Tabel 6. Hasil Data Pelatihan dari 6 Responden dengan Fungsi

| Aktivasi Sigmoid Biner |               |       |         |
|------------------------|---------------|-------|---------|
| Responden              | Learning Rate | Epoch | RMSE    |
| Nani                   | 0,05          | 5000  | 0,0101  |
| Jejen                  | 0,05          | 5000  | 0,098   |
| Devi                   | 0,05          | 5000  | 0,0103  |
| Rafsan                 | 0,05          | 5000  | 0,00104 |
| Islamiyati             | 0,05          | 5000  | 0,0105  |
| Arif                   | 0,05          | 5000  | 0,0095  |

Setelah melakukan percobaan *learning rate* pada 1 data pelatihan maka akan diketahui nilai *learning rate* yang tepat. Kita bisa melihat pada Tabel 6, hasil pelatihan terhadap 6 responden data pelatihan memiliki tingkat nilai *error* terkecil yang berbeda-beda. Untuk nilai *epoch* mengambil nilai lebih besar dari 3000 menjadi 5000 *epoch* karena pada *epoch* ke-5000 batas *error* yang digunakan tercapai yaitu 0,01. Parameter-parameter tersebut memberikan waktu pelatihan yang paling optimal dibandingkan dengan yang lain.

### 3.2. Hasil Pengujian

Dalam sistem ini pengujian dilakukan sebanyak 2 tahap. Pengujian dengan suara pelatihan ini merupakan pengujian pertama yang dilakukan pada suara masukan yang telah dilatih sebelumnya. Pengujian ini bertujuan untuk mengetahui apakah sistem dapat mengenali dengan baik atau tidak dapat mengenali suara yang telah dilatih sebelumnya. Data pelatihan terdiri dari 74 suku kata. Hasil pengujian dapat dilihat pada Tabel 7.

Tabel 7. Hasil Akurasi Pengujian 74 Sampel Data

| No | Responden  | Akurasi |
|----|------------|---------|
| 1  | Nani       | 100%    |
| 2  | Jejen      | 100%    |
| 3  | Devi       | 100%    |
| 4  | Rafsan     | 100%    |
| 5  | Islamiyati | 100%    |
| 6  | Arif       | 100%    |

Tingkat keberhasilan pada proses pengujian terhadap 74 sampel data pelatihan dengan 6 responden sebesar 100%. Sampel data dapat dikenali dengan sempurna oleh sistem.

Tahap kedua, pengujian terhadap 115 sampel data pengujian yang belum dilatih dari 6 responden dapat dilihat pada Tabel 8.

Tabel 8. Hasil Akurasi Pengujian 115 Sampel Data

| No | Responden  | Akurasi |
|----|------------|---------|
| 1  | Nani       | 69%     |
| 2  | Jejen      | 65%     |
| 3  | Devi       | 65%     |
| 4  | Rafsan     | 66%     |
| 5  | Islamiyati | 66%     |
| 6  | Arif       | 68%     |

Berdasarkan Tabel 8 diatas, pengujian akurasi tertinggi yaitu pada nilai 69% dengan responden nani. Dan pengujian dengan akurasi terendah berada pada nilai 65% dengan responden jejen dan devi. Maka berdasarkan hasil pengujian dengan tingkat akurasi yang berbeda-beda menunjukkan bahwa dari kondisi responden saat mengucapkan suatu kata dan sampel data yang belum dilatih sangat berpengaruh terhadap nilai presentase yang didapatkan.

### 4. Kesimpulan

Berdasarkan hasil percobaan yang dilakukan, maka dapat diambil kesimpulan akurasi dari sistem pengenalan suku kata bahasa Indonesia menggunakan metode *Linier Predictive Code* (LPC) dan metode *Backpropagation Neural Network* memiliki tingkat keberhasilan pengujian suara untuk 74 sampel data yang telah dilatih mencapai 100% dari setiap 6 responden. Dan tingkat keberhasilan pengujian suara untuk 115 data pengujian belum dilatih mencapai nilai tertinggi 69% dengan responden nani, dibanding nilai

akurasi yang lainnya. Kondisi responden saat mengucapkan suatu kata dan sampel data yang belum dilatih sangat berpengaruh terhadap nilai presentase yang didapatkan karena semakin banyak data pelatihan yang diproses dalam jaringan maka semakin tinggi akurasi keberhasilan yang diperoleh (Sinyal suara dapat dikenali).

### Daftar Pustaka

- [1] Azmi, Y., Djamal, E. C., & Ilyas, R. (2018). Identifikasi Emosi Dari Sinyal Suara Secara *Real Time* Menggunakan *Linear Predictive Coding* dan *Backpropagation*. 16–21.
- [2] Broto, W. (2017). Metode *Artificial Intelligence* Sebagai Aplikasi Pengenalan Ucapan Disabilitas. *E-Journal*.  
<https://doi.org/http://doi.org/10.21009/03.SNF2017>
- [3] Chamidy, T. (2016). Metode *Mel Frequency Cepstral Coeffisients* (MFCC) Pada Klasifikasi *Hidden Markov Model* (HMM) Untuk Kata *Arabic* Pada Penutur Indonesia. *MATICS*.  
<https://doi.org/10.18860/mat.v8i1.3482>
- [4] Desyanndana, I. P., Informatika, T., Sains, F., Teknologi, D. A. N., Sanata, U., & Yogyakarta, D. (2019). *Voice Recognition System for Recognize Voice Command Using Backpropagation Neural Network*.
- [5] Faradiba. (2017). Pengenalan Pola Sinyal Suara Manusia Menggunakan Metode *Backpropagation Neural Network*. *Jurnal EduMatSains*.
- [6] Haumahu, J. P. (2019). Implementasi Jaringan Syaraf Tiruan Untuk Pengenalan Pola Notasi Balok Menggunakan Metode *Backpropagation*. 6(3), 255–259.
- [7] Heriyanto. (2015). Analisa Deteksi Huruf Hijaiyah Melalui *Voice Recognition* Menggunakan Kombinasi *Energy*. *Telematika*.
- [8] Heriyanto, H., & Simanjuntak, O. S. (2017). Identifikasi Ucapan Warna Menggunakan LPC (*Linier Predictive Code*) Dan Kelompok Pemilihan Bobot. *Telematika*.  
<https://doi.org/10.31315/telematika.v14i01.1968>
- [9] Hidayat, S., Hidayat, R., & Adji, T. B. (2015). Sistem Pengenal Tutur Bahasa Indonesia Berbasis Suku Kata Menggunakan MFCC, Wavelet Dan HMM. *The 7th National Conference On Information Technology And Electrical Engineering*.
- [10] Izzah, N. (2018). Klastering Suara Berdasarkan *Gender* Menggunakan Algoritma K-Means Dari Hasil Ekstraksi FFT (*Fast Fourier Transform*). *SOULMATH*.  
<https://doi.org/10.25139/sm.v6i1.790>
- [11] Julpan, Nababan, E. B., & Zarlis, M. (2015). Analisis Fungsi Aktivasi Sigmoid Biner Dan Sigmoid Bipolar Dalam Algoritma

- Backpropagation* Pada Prediksi Kemampuan Siswa. *Jurnal Teknovasi*.
- [12] Juniansyah, R. R. (2017). Perancangan Sistem Pengenalan Suara Dengan Metode *Linear Predictive Coding*. 4(1), 404–411.
- [13] Kusumawati, R. (2016). Metode *Linear Predictive Coding* (LPC) Pada Klasifikasi *Hidden Markov Model* (HMM) Untuk Kata Arabic Pada Penutur Indonesia. *MATICS*. <https://doi.org/10.18860/mat.v8i1.3481>
- [14] Nursholihatun, E., Mariyanto, S., & Sasongko, Al, Zainuddin, A. (2020). Identifikasi Suara Menggunakan Metode *Mel Frequency Cepstrum Coefficients* ( MFCC ) Dan Jaringan Syaraf Tiruan. 7(1), 48–55.
- [15] Permana, I. S., Indrawaty, Y., & Zulkarnain, A. (2019). Implementasi Metode MFCC Dan DTW Untuk Pengenalan Jenis Suara Pria Dan Wanita. *MIND Journal*. <https://doi.org/10.26760/mindjournal.v3i1.61-76>
- [16] Syam, M., Raharjo, I. J., & Patmasari, R. (2019). Identifikasi Asal Daerah Berdasarkan Suara Manusia Dengan Metode *Linier Predictive Coding* ( LPC ). 6(3), 10226–10233.
- [17] Yuliantari, R. V., Hidayat, R., & Wahyunggoro, O. (2016). Ekstraksi Ciri Dan Pengenalan Tutor Vokal Bahasa Indonesia Menggunakan Metode *Discrete Wavelet Transform* (DWT) Dan *Dynamic Time Warping* (DTW) Secara *Realtime*. *Prosiding SNST Ke-7*.
- [18] Zhang, B., & Sennrich, R. (2019). *Root Mean Square Layer Normalization*. *NeurIPS*, 1–12. <http://arxiv.org/abs/1910.07467>