

Pengenalan Ucapan Menggunakan Algoritma *Back Propagation*

Riska Wulandari¹⁾, Ahmad Izzuddin²⁾, Tamam Asrori³⁾

¹⁾Mahasiswa Program Studi Teknik Elektro, Fakultas Teknik, Universitas Panca Marga

^{2),3)}Dosen Program Studi Teknik Elektro, Fakultas Teknik, Universitas Panca Marga

Jl. Yos Sudarso 107 Pabean Dringu Probolinggo 67271

Email : riskawulandari445@gmail.com

Terima Naskah : 22 April 2016

Terima Revisi : 30 April 2016

ABSTRAK

Teknologi pengenalan ucapan saat ini telah mengalami perkembangan yang cukup pesat. Teknologi ini memungkinkan suatu perangkat untuk mengenali dan memahami kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan pola tertentu yang tersimpan dalam suatu perangkat. Pada proses ekstraksi parameter suara peneliti menggunakan metode *Linear Predictive Coding* (LPC). Dari proses parameter suara di hasilkan nilai normalisasi 4160 buah dengan asumsi bahwa waktu yang dibutuhkan untuk pengucapan satu kata butuh waktu kurang dari 0,5 detik. Jumlah frame yang dihasilkan sebanyak 50 buah frame dengan data sebanyak 240 byte data dan jarak antar frame sebanyak 80 byte data dengan overlapnya 160 byte data. Dari proses analisis LPC dilakukan penyimpanan data untuk digunakan sebagai inputan atau masukan diproses pada jaringan saraf tiruan untuk melakukan proses pengenalan. Seratus delapan puluh sampel suara dari sepuluh pembicara yang berbeda digunakan sebagai input proses pelatihan jaringan saraf tiruan. Dua puluh sampel suara dari sepuluh pembicara yang berbeda digunakan sebagai input proses pengujian data baru. Hasil pengujian proses pengenalan suara menunjukkan keberhasilan 75%.

Kata kunci : Pengenalan Ucapan, Metode *Linear Predictive Coding* (LPC), Algoritma Propagasi Balik

ABSTRACT

The current speech recognition technology has been progressing quite rapidly. This technology allows a device to recognize and understand spoken words by way of digitizing the word and match the digital signal with a specific pattern stored in a device. In the extraction process sound parameters of researchers using *Linear Predictive Coding* (LPC). From the sound parameters generated normalized value 4160 fruit on the assumption that the time required for the pronunciation of the word butuh less than 0.5 seconds. The number of frames produced as many as 50 pieces of the frame with as much data as 240 bytes of data and the distance between the frames of 80 bytes of data with 160 bytes of data overlap. From the LPC analysis performed data storage to be used as input or inputs are processed on an artificial neural network to perform the process of recognition. One hundred eighty voice samples from ten different speaker is used as the input of artificial neural network training process. Twenty samples of sounds from ten different speaker is used as input new data testing process. The test results demonstrate the success of speech recognition process 75%.

Keywords: *Speech Recognition, Methods Linear Predictive Coding (LPC), Backpropagation Algorithm*

PENDAHULUAN

Teknologi pengenalan ucapan atau suara saat ini telah mengalami perkembangan yang cukup pesat. Banyak aplikasi yang telah dihasilkan seiring dengan metode dan proses dari pengenalan ucapan atau suara tersebut. Pengenalan ucapan

adalah suatu pengembangan teknik dan system yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan. Teknologi ini memungkinkan suatu perangkat untuk mengenali dan memahami kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan

sinyal digital tersebut dengan pola tertentu yang tersimpan dalam suatu perangkat. Kata yang diucapkan diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode-kode tertentu untuk mengidentifikasi kata tersebut.

Jaringan syaraf tiruan merupakan salah satu algoritma pembelajaran mesin yang meniru proses yang terjadi pada kehidupan alami. Algoritma jaringan syaraf tiruan merupakan algoritma yang memungkinkan mesin untuk belajar dari pengalaman. Algoritma ini cocok untuk diaplikasikan ke permasalahan yang hanya diketahui masukan serta keluarannya, dan proses antara masukan dan keluaran tidak diketahui. Dengan alasan tersebut, algoritma jaringan syaraf tiruan ini dapat dipergunakan untuk menyelesaikan permasalahan pengenalan suara pada sistem ini.

Perambatan galat mundur (*Back Propagation*) adalah sebuah metode sistematis untuk pelatihan *multiplayer* jaringan saraf tiruan. Metode ini memiliki dasar matematis yang kuat, obyektif dan algoritma ini mendapatkan bentuk persamaan dan nilai koefisien dalam formula dengan meminimalkan jumlah kuadrat galat *error* melalui model yang dikembangkan (*training set*).

Pada penelitian ini Algoritma *Back Propagation* digunakan untuk pengenalan ucapan berupa kata, yang diupayakan untuk pengenalan ucapan menggunakan bahasa Indonesia. Berdasarkan latar belakang diatas, penulis mengambil judul "Pengenalan Ucapan Menggunakan Algoritma *Back Propagation*" sebagai judul penelitian.

Agar penelitian ini tidak keluar dari pokok bahasan, penulis memberikan batasan-batasan sebagai berikut :

1. Data masukan berupa file .wav
2. Proses pengolahan ucapan menggunakan ekstraksi ciri *Linear Predictive Coding* (LPC).
3. Proses pelatihan dan pengenalan ucapan menggunakan Algoritma *Back Propagation*.
4. Pengenalan ucapan hanya berupa kata dasar bukan kalimat berdasarkan Kamus Besar Bahasa Indonesia. Kata dasar seperti Akses, Aktif, Akun, Disket, Ekstrak, Filter, Ketik, Kode, Langkah, Objek, Obyek, Pindah, Proses, Salin, Sambung, Sensor, Simpan, Simbol, Sistem, dan Situs.

Pengenalan Ucapan

Pengenalan ucapan adalah proses identifikasi yang dilakukan komputer untuk mengenali kata yang diucapkan oleh seseorang tanpa mempedulikan identitas orang terkait dengan melakukan konversi sebuah sinyal akustik, yang ditangkap oleh *audio device* (perangkat input suara).

Pola kerja Pengenalan ucapan adalah mencocokkan sinyal akustik yang diterima dengan data yang tersimpan dalam *template* ataupun *database*. Proses pencocokan memiliki dua model utama yaitu Model Akustik yang terdiri dari fonem yang memiliki nilai tertentu yang diambil dari sinyal akustik dan Model Bahasa berupa metode yang mengestimasi satu kata diikuti oleh serangkaian kata lainnya.

Audacity 2.0

Audacity adalah alat atau rekaman yang dapat mengedit hasil rekaman atau file yang berformat seperti mp3. Aplikasi ini dibangun dengan pustaka *WxWidgets* sehingga dapat berjalan pada berbagai sistem operasi. Dengan *Audacity*, pengguna bisa mengoreksi berkas suara tertentu, atau sekedar menambahkan berbagai efek yang disediakan. Selain itu, pengguna juga dapat berkreasi dengan suara yang dimiliki sendiri.

Kelebihan dari aplikasi ini adalah fitur dan kestabilan. Pustaka yang digunakan juga tidak terlalu banyak dan waktu tunggu juga tidak terlalu lama. Kekurangan dari aplikasi ini adalah antarmuka pengguna (*user interface*) yang sedikit kaku apabila dibandingkan dengan aplikasi sejenis di sistem operasi lain.

Microsoft Visual Basic 6.0

Sejarah Visual Basic diawali dari pengembangan bahasa BASIC (*Beginners All-Purpose Symbolic Instruction Code*) di Dartmouth College, Amerika Serikat pada awal tahun 1960. Sejak semula Basic dirancang untuk mudah dipelajari. Pada awal tahun 1975 muncul komputer pribadi (*Personal Computer/PC*) pertama, bermerek MITS Altair. Namun karena hanya memiliki RAM 4KB, satu-satunya bahasa yang bisa digunakan untuk pemrograman adalah Assembly, bahasa ini sangat sulit untuk dipahami karena perintahnya mendekati bahasa mesin. Tentu hal ini sangat menghambat perkembangan komputer pribadi. Bill Gate dan Paul Allen melihat

potensi yang sangat besar pada komputer pribadi dimasa depan. Keduanya menciptakan Basic untuk Altair. Ternyata sambutan masyarakat sangat baik dan akhirnya keduanya mendirikan perusahaan sendiri yang bergerak dibidang software yang bernama Microsoft.

Pada tahun 1982 IBM/PC diperkenalkan kepada masyarakat, Microsoft pun membuat sistem operasi MS-DOS untuk komputer ini. Dimana didalamnya disertakan pula bahasa Basic yang dikenal sebagai Quick Basic (QBASIC). Zamanpun berganti, pada tahun 1990 era DOS berlalu dan digantikan oleh era Windows.

Dari perkembangan ini akhirnya Microsoft membuat Basic versi Windows, bahasa pemrograman Basic ini dikenal dengan nama Visual Basic. Versi terakhirnya adalah MS Visual Basic 6.0 for Windows 98. Saat ini perkembangan visual basic telah mengarah ke pemrograman .NET, yaitu dikenal dengan nama Visual Basic .NET.

Pre-Processing Metode Linear Predictive Coding (LPC)

Linear Predictive Coding (LPC) adalah salah satu teknik analisa suara yang paling kuat, dan salah satu metode yang paling berguna untuk pengkodean suara dengan kualitas yang baik pada bit rate rendah. LPC memberikan perkiraan yang sangat akurat pada parameter suara dan pectrum efisien untuk perhitungan.

LPC membuktikan suatu model yang baik untuk pengenalan suara, yaitu memberikan parameter model yang tepat untuk sinyal suara, dapat dilihat pada spectrum koefisien peramalan yang mirip dengan spectrum sinyal aslinya.

Langkah-langkah analisa LPC untuk pengenalan suara adalah :

1. Normalisasi

Proses normalisasi amplitude diperoleh dengan membagi semua nilai sampel digital dengan nilai absolute maksimum dari sampel sinyal tersebut (lihat persamaan (1)).

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

Keterangan :

$x'(n)$ = Hasil Normalisasi

$x(n)$ = Panjang Data

2. Pre-Emphasis

Proses *Pre-Emphasis* adalah proses yang dirancang untuk mengurangi dampak buruk dari transmisi dan suara latar. Proses *Pre-Emphasis*

sangat baik dalam mengurangi efek distorsi, atenuasi, dan saturasi dari media rekaman. Perhitungan *Pre-Emphasis* dilakukan pada sinyal digital dalam domain waktu dan menggunakan persamaan (2).

$$x(n) = x'(n) - \alpha \cdot x'(n - 1) \quad (2)$$

Keterangan :

$x(n)$ = *Pre-Emphasis*

$x'(n)$ = Data Normalisasi

α = Alpha dimana nilainya adalah 0,95

3. Framing dan Windowing

Frame blocking, pada tahap ini sinyal kata yang teremphasi, $x(n)$ dibagi menjadi *frame-frame* dengan masing-masing *frame* memuat N cuplikan kata dan *frame-frame* yang berdekatan dipisahkan sejauh M cuplikan, semakin $M \ll N$ semakin baik perkiraan spektral LPC dari *frame* ke *frame*.

$$\text{Frame} = \frac{x'(n) - \text{overlap}}{\text{jarak antar frame}} \quad (3)$$

Windowing, pada tahap ini dilakukan fungsi *weighting* pada setiap *frame* yang telah dibentuk pada langkah sebelumnya dengan tujuan untuk meminimalkan *discontinuities* pada ujung awal dan ujung akhir setiap *frame* yaitu dengan men-*taper* sinyal menuju nol pada ujung-ujungnya. Tipikal *window* yang digunakan pada metode autokorelasi LPC adalah *Hamming Window* yang memiliki persamaan berikut (lihat persamaan (4 dan 5)).

$$w(n) = 0.54 - 0.46 \cos \left[\frac{2n\pi}{N-1} \right] \quad (4)$$

$$s(n) = x(n)w(n) \quad (5)$$

Keterangan :

$w(n)$ = *Weighting*

$s(n)$ = *Hamming Window*

N = Jumlah *Frame*

$x(n)$ = Data Normalisasi

4. Proses Autokorelasi

Pada tahap ini masing-masing *frame* yang telah di *windowing* di autokorelasi untuk mendapatkan :

$$r(p) = \sum_{n=1}^{N-p} (s(n) * s(n+p)) \quad (6)$$

Dimana nilai autokorelasi yang tertinggi pada p adalah orde dari analisa LPC, biasanya orde LPC tersebut 8 sampai 16. Autokorelasi ke-0 melambangkan energi dari *frame* yang bersangkutan dan ini merupakan salah satu keuntungan dari metode autokorelasi. Orde dari analisa LPC yang peneliti gunakan adalah 12.

5. Analisis LPC

Langkah selanjutnya adalah melakukan analisis LPC, dimana pada tahap ini p+1 autokorelasi pada setiap frame diubah menjadi satu set LPC parameter yaitu koefisien LPC. Nilai autokorelasi kemudian disusun dalam suatu matriks toeplitz seperti tampak pada persamaan (7). Penyelesaian dari matriks toeplitz dilakukan dengan menggunakan algoritma durbin seperti yang di uraikan pada persamaan (8) sampai persamaan (12).

$$\begin{bmatrix} r(0) & r(1) & \dots & r(P-1) \\ r(1) & r(0) & \dots & r(P-2) \\ r(2) & r(1) & \dots & r(P-3) \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad (7)$$

$$k_j = \frac{-\{r(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} r(i-j)\}}{E^{(i-j)}}, 1 \leq i \leq P \quad (9)$$

$$a_j^{(i)} = k_j \quad (10)$$

$$a_j^{(i)} = a_j^{(i-1)} + k_j * a_{i-j}^{(i-1)} \quad (11)$$

$$E^{(i)} = (i - k_i^2) E^{(i-1)} \quad (12)$$

Keterangan :

$E^{(0)}$ = Analisis LPC ke-0

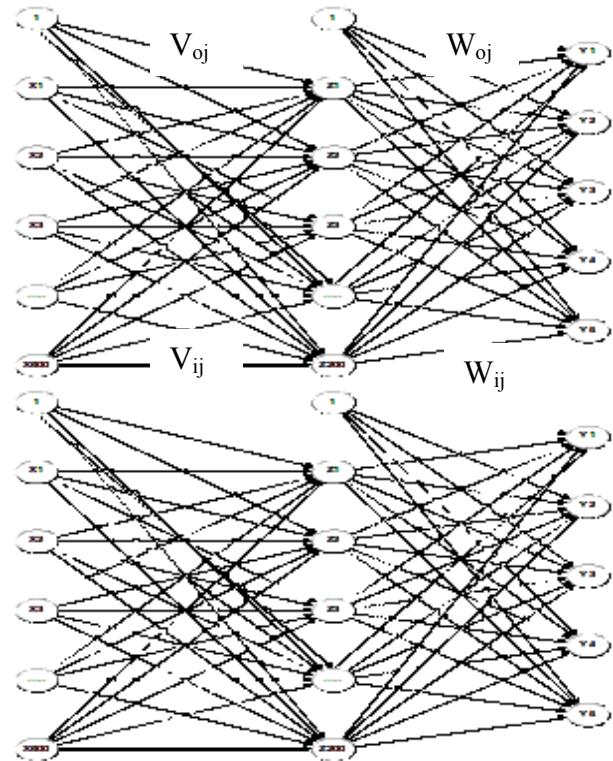
k_j = lambda

Jaringan Saraf Tiruan

Jaringan saraf tiruan ialah suatu sistem pengolah informasi yang mempunyai karakteristik menyerupai jaringan saraf biologis tubuh manusia. Jaringan saraf tiruan telah dikembangkan dengan menggunakan model matematis untuk menirukan cara kerja jaringan saraf biologis, dengan berdasarkan asumsi :

1. Pengolah informasi terdiri dari elemen-elemen sederhana yang disebut *neuron*.
2. Sinyal dilewatkan dari satu *neuron* ke *neuron* yang lain melalui hubungan koneksi.
3. Tiap hubungan koneksi mempunyai nilai bobot tersendiri.
4. Tiap *neuron* mempergunakan fungsi aktivasi (biasanya tidak *linear*) terhadap masukan yang diterimanya untuk menentukan sinyal keluarannya.

Karakteristik jaringan saraf tiruan ditentukan oleh pola hubungan antar *neuron* (disebut *architecture*), metode untuk menentukan nilai bobot tiap hubungan (disebut *training*) dan ditentukan oleh fungsi aktivasi. Berikut adalah gambar struktur jaringan saraf tiruan:



Gambar 1. Struktur Jaringan Saraf Tiruan

Dimana :

X = Masukan (input).

Vij = Bobot pada lapisan tersembunyi.

Wij = Bobot pada lapisan keluaran.

Voj = Bias pada lapisan tersembunyi.

Woj = Bias pada lapisan tersembunyi dan lapisan keluaran.

J = 1, 2, 3, ..., n.

n = Jumlah unit pengolah pada lapisan tersembunyi.

k = Jumlah unit pengolah pada lapisan keluaran.

Y = Keluaran hasil.

Back Propagation

Perambatan galat mundur (*Back Propagation*) adalah sebuah metode sistematis untuk pelatihan *multiplayer* jaringan saraf tiruan. Metode ini memiliki dasar matematis yang kuat, obyektif dan algoritma ini mendapatkan bentuk persamaan dan nilai koefisien dalam formula dengan meminimalkan jumlah kuadrat galat *error* melalui model yang dikembangkan (*training set*).

Algoritma pelatihan untuk jaringan saraf tiruan *Back Propagation* ini adalah sebagai berikut :

Langkah 0 : Inisialisasi nilai bobot dengan nilai acak yang kecil.

Langkah 1 : Selama kondisi berhenti masih tidak terpenuhi, laksanakan langkah 2 sampai 9.

Langkah 2 : Untuk tiap pasangan pelatihan, kerjakan langkah 3 sampai 8.

Feedforward :

Langkah 3 : Untuk tiap unit masukan (X_b , $i=1, \dots, n$) menerima sinyal masukan x_i dan menyebarkan sinyal itu keseluruh unit pada lapis atasnya (lapis tersembunyi)

Langkah 4 : Untuk tiap unit tersembunyi (Z_j , $j=1, \dots, p$) dihitung nilai masukan dengan menggunakan nilai bobotnya :

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (13)$$

Kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi yang dipilih :

$$z_j = f(z_in_j) \quad (14)$$

dimana fungsi aktivasi yang digunakan ialah fungsi sigmoid biner yang mempunyai persamaan :

$$f_1(x) = \frac{1}{1 + \exp(-x)} \quad (15)$$

Hasil fungsi tersebut dikirim ke semua unit pada lapis di atasnya.

Langkah 5 : Untuk tiap unit keluaran (Y_k , $k=1, \dots, m$) dihitung nilai masukan dengan menggunakan nilai bobotnya :

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (16)$$

Kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi :

$$y_k = f(y_in_k) \quad (17)$$

Perhitungan nilai kesalahan :

Langkah 6 : Untuk tiap unit keluaran (Y_k , $k=1, \dots, m$) menerima pola target yang bersesuaian dengan pola masukan, dan kemudian dihitung informasi kesalahan :

$$\delta_k = (t_k - y_k) f'(y_in_k) \quad (18)$$

Kemudian dihitung koreksi nilai bobot yang kemudian akan digunakan untuk memperbaharui nilai bobot w_{jk} :

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (19)$$

Hitung koreksi nilai bias yang kemudian akan digunakan untuk memperbaharui nilai w_{0k} :

$$\Delta w_{0k} = \alpha \delta_k \quad (20)$$

dan kemudian nilai δ_k dikirim ke unit pada layer sebelumnya.

Langkah 7 : Untuk tiap unit tersembunyi (Z_j , $j=1, \dots, p$) dihitung delta masukan yang berasal dari unit pada layer di atasnya :

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk} \quad (21)$$

Kemudian nilai tersebut dikalikan dengan nilai turunan dari fungsi aktivasi untuk menghitung informasi kesalahan :

$$\delta_j = \delta_in_j f'(z_in_j) \quad (22)$$

Hitung koreksi nilai bobot yang kemudian digunakan untuk memperbaharui nilai v_{ij} :

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (23)$$

dan hitung nilai koreksi bias yang kemudian digunakan untuk memperbaharui v_{0j} :

$$\Delta v_{0j} = \alpha \delta_j \quad (24)$$

Memperbaharui nilai bobot dan bias :

Langkah 8 : Tiap nilai bias dan bobot ($j=0, \dots, p$) pada unit keluaran (Y_k , $k=1, \dots, m$) diperbaharui :

$$w_{jk} (new) = w_{jk} (old) + \Delta w_{jk} \quad (25)$$

$$v_{ij} (new) = v_{ij} (old) + \Delta v_{ij} \quad (26)$$

Langkah 9 : Menguji apakah kondisi berhenti sudah terpenuhi. Kondisi berhenti ini terpenuhi jika nilai kesalahan yang dihasilkan lebih kecil dari nilai kesalahan referensi.

METODE

Tahapan dalam penelitian ini terdiri dari beberapa tahapan, yaitu:

1. Pengumpulan Data

Pada tahap pengumpulan data dilakukan studi pustaka, pengamatan secara langsung dengan cara mengumpulkan data untuk mengidentifikasi suara. File suara berupa .wav.

2. Analisis Sistem

Pada tahap ini penulis melakukan analisis sistem sesuai data dan permasalahan yang telah

dikumpulkan sebelumnya, guna sebagai acuan yang digunakan untuk mengembangkan sistem yang sedang dikerjakan.

3. Perancangan Sistem

Merupakan tahap penyusunan proses, data, aliran proses dan hubungan antar data yang paling optimal dan memenuhi kebutuhan pihak yang terkait sesuai dengan hasil analisa kebutuhan.

4. Implementasi Sistem

Pada tahap ini, penulis akan membuat aplikasi berdasarkan rancangan yang telah dibuat sebelumnya sesuai dengan kebutuhan pihak yang terkait.

5. Pengujian Program

Pada tahapan ini, penulis akan melakukan pengujian *blackbox (blackbox testing)* adalah salah satu metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada input dan output aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum). Tahap pengujian atau testing merupakan salah satu tahap yang harus ada dalam sebuah siklus pengembangan perangkat lunak (selain tahap perancangan atau desain).

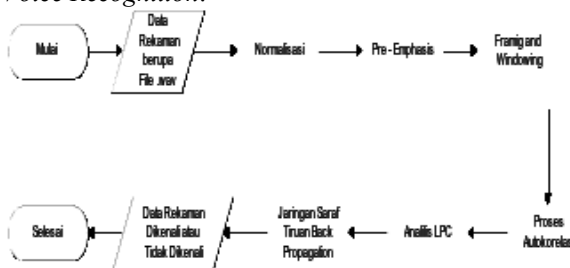
6. Menyusun Laporan/Dokumentasi

Apabila semua tahapan yang dimulainya penelitian hingga pada uji coba dan evaluasi sistem sudah sesuai dan tidak terjadi kesalahan lagi, maka tahapan akhir yang dilakukan adalah penyusunan laporan pada hasil penelitian.

HASIL DAN PEMBAHASAN

Analisa dan Desain Sistem

Pada penelitian ini peneliti menggunakan metode *Linear Predictive Coding (LPC)* untuk mengolah data rekaman suara untuk digunakan sebagai *input* pada Jaringan Saraf Tiruan *Back Propagation* yang diterapkan menggunakan Bahasa Pemrograman *Microsoft Visual Basic 6.0*. berikut ini adalah gambar blok diagram system *Voice Recognition*.



Gambar 2. Blok Diagram Sistem *Voice Recognition*

Implementasi Proses Pencuplikan Sinyal

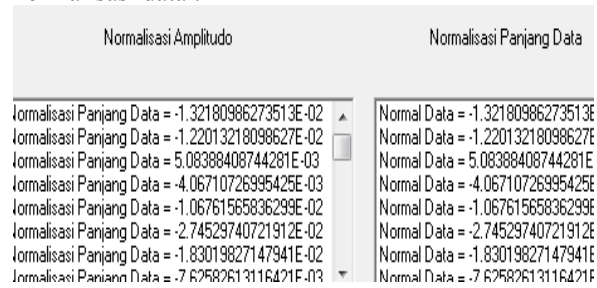
Berikut ini adalah tampilan hasil nilai panjang data dan nilai maksimal dari panjang data :



Gambar 3. Nilai Panjang Data dan Nilai Maks dari Panjang Data

Normalisasi

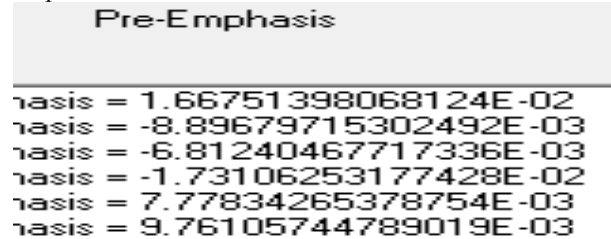
Berikut ini adalah tampilan hasil proses normalisasi data :



Gambar 4. Proses Normalisasi

Pre-Emphasis

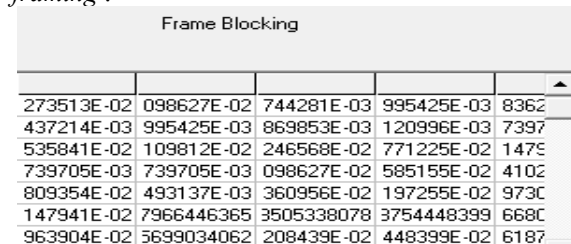
Berikut ini adalah tampilan hasil Proses *Pre-Emphasis* :



Gambar 5. Hasil Proses *Pre-Emphasis*

Framing dan Windowing

Berikut ini adalah tampilan hasil proses *framing* :



Gambar 6. Hasil Proses *Framing*

Berikut ini adalah tampilan hasil proses *windowing* :

	901881E-03	789016E-04	995425E-04	159634E-04	690392E-04	257753
	497712E-04	159634E-04	295882E-04	896797E-04	591764E-04	74885E
	828673E-03	168785E-03	197254E-03	981698E-03	718353E-03	591764
	591764E-04	591764E-04	789016E-04	068124E-03	528215E-03	892222
	847483E-03	239451E-04	288765E-03	957804E-03	178444E-03	990843
	718353E-03	3373157092	270462E-02	558719E-02	934418E-03	45551E
	371123E-03	272496E-03	166751E-03	558719E-03	694967E-03	493137
	598882E-02	7966446365	178444E-03	297916E-03	068124E-02	150991

Gambar 7. Hasil Proses *Windowing*

Proses Autokorelasi

Berikut ini adalah tampilan hasil proses autokorelasi :

	154127E-05	154127E-05	154127E-05	154127E-05	154127E-05
	850086E-05	850086E-05	850086E-05	850086E-05	850086E-05
	091002E-04	091002E-04	091002E-04	091002E-04	091002E-04
	376578E-04	376578E-04	376578E-04	376578E-04	376578E-04
	999931E-03	999931E-03	999931E-03	999931E-03	999931E-03
	440965E-03	440965E-03	440965E-03	440965E-03	440965E-03
	076455E-03	076455E-03	076455E-03	076455E-03	076455E-03
	443389E-03	443389E-03	443389E-03	443389E-03	443389E-03

Gambar 8. Hasil Proses Autokorelasi

Analisis LPC

Berikut ini adalah tampilan hasil proses analisis LPC :

	2798477885	734826E-05	615847E-05	615847E-05	562269E-04	1
	1786573015	044685E-04	153202E-04	153202E-04	252538E-04	4
	3329854909	089016E-04	330752E-04	330752E-04	328851E-04	4
	3382986538	467649E-03	865587E-03	865587E-03	518927E-03	2
	3277409385	904184E-03	165431E-03	165431E-03	065986E-03	1
	5259403441	725791E-03	263823E-03	263823E-03	674159E-02	7
	3409892355	020965E-03	220378E-03	220378E-03	997759E-03	8
	2137805566	362765E-03	552771E-03	552771E-03	572576E-02	1

Gambar 9. Hasil Proses Analisis LPC

Data yang diperoleh dari hasil analisis LPC akan disimpan berupa file .txt untuk kemudian digunakan sebagai *input*-an dari jaringan saraf tiruan *Back Propagation*. Berikut tampilan hasil semua data rekaman yang telah disimpan dan di tampilkan ke dalam MSFlexGrid Visual Basic 6.0. Berikut ini adalah tampilan semua data rekaman dari Proses Analisis LPC :

	0.99996	-0.00007	-0.00007	-0.00007	-0.00011
	0.99996	-0.00008	-0.00008	-0.00008	-0.00013
	0.99999	-0.00003	-0.00003	-0.00003	-0.00004
	0.99994	-0.00013	-0.00013	-0.00013	-0.00019
	0.99999	-0.00002	-0.00002	-0.00002	-0.00004
	0.99999	0.99999	-0.00002	-0.00002	-0.00003
	0.99998	-0.00003	-0.00003	-0.00003	-0.00005
	1.00001	0.00001	0.00001	0.00001	0.00002
	1.00000	0.00000	0.00000	0.00000	-0.00001
	0.99992	-0.00016	-0.00016	-0.00016	-0.00024

Gambar 10. Tampilan Semua Data Rekaman dari Proses Analisis LPC

Jaringan Saraf Tiruan *Back Propagation*

Berikut adalah tampilan hasil dari proses pengenalan jaringan saraf tiruan *Back Propagation* :

Gambar 11. Hasil Proses Pengenalan Jaringan Saraf Tiruan *Back Propagation*

Pada tahap *binerisasi* sejak awal peneliti telah melakukan perekaman yang disesuaikan berdasarkan abjad pada setiap kata. Misalnya kata Aktif merupakan kata pertama dalam proses perekaman dan nilai *binerisasinya* adalah 00001 sampai kata Situs yang nilai *binerisasinya* adalah 10100. Tahap *binerisasi* dilakukan terhadap 20 kata yang telah direkam.

Dari hasil pengenalan atau *training* di atas didapat nilai Bobot V (bobot pada lapisan tersembunyi) dan nilai Bobot W (bobot pada lapisan keluaran).

Berikut adalah gambar hasil dari Bobot W (Bobot Pada Lapisan Keluaran) :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-3125.60	-3125.29	-3125.35	-3125.87	-3124.8	-3125.09	-3124.78	-3124.87	-3125.47	-3124.04	-3125.54	-3125.45	-3125.43	-3124.38	-3124.56
2	-3125.16	-3125.7	-3125.99	-3125.38	-3124.22	-3124.61	-3125.61	-3125.39	-3125.86	-3124.75	-3125.32	-3125.02	-3124.85	-3125.31	-3124.98
3	-3124.71	-3125.94	-3125.79	-3125.81	-3125.18	-3125.3	-3124.11	-3125.02	-3125.69	-3124.89	-3125.63	-3125.43	-3124.85	-3125.39	-3125.1
4	-3125.34	-3124.4	-3125.78	-3124.15	-3125.59	-3124.07	-3124.32	-3124.88	-3125.5	-3124.29	-3125.33	-3124.5	-3125.89	-3125.98	-3124.46
5	-3125.47	-3125.47	-3125.37	-3124.89	-3125.06	-3125.57	-3124.46	-3125.7	-3125.81	-3124.28	-3125.7	-3124.71	-3125.32	-3125.81	-3124.95

	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115
1	-624.549	-625.862	-624.128	-625.218	-624.471	-624.004	-625.704	-624.077	-625.309	-625.343	-624.634	-625.973	-625.661	-625.716	-625.022
2	-625.839	-625.426	-625.123	-625.015	-625.063	-625.829	-624.434	-625.135	-625.188	-625.912	-625.087	-624.118	-625.599	-625.202	-624.035
3	-625.664	-625.393	-624.842	-624.166	-624.02	-624.121	-625.814	-625.293	-625.397	-624.122	-625.264	-624.847	-625.245	-625.146	-625.774
4	-625.534	-625.61	-624.718	-625.771	-625.808	-625.051	-625.984	-625.38	-624.427	-625.351	-625.21	-624.739	-625.437	-624.264	-624.637
5	-625.865	-624.638	-624.715	-625.937	-625.942	-624.602	-624.792	-624.616	-625.522	-624.949	-625.412	-625.195	-624.033	-624.668	-624.761

	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
1	-0.9388	0.48602	-0.16208	0.659556	-0.37132	-0.06644	-0.18133	0.841984	0.46397	0.589801	0.150957	-0.69409	-0.39436	0.21778	0.053053
2	-0.0345	0.929242	-0.05424	-0.18913	-0.29924	-0.40475	0.072489	0.817012	0.940039	-0.05273	-0.04359	-0.49693	0.030496	-0.0933	0.562612
3	-0.09821	0.338017	0.633838	0.927793	0.170532	-0.08393	-0.5315	0.485573	-0.18086	0.257054	0.467988	0.256771	0.972121	0.660446	-0.6995
4	0.00692	0.31775	-0.25326	-0.50775	-0.71214	0.71775	0.568878	-0.95758	0.581359	-0.62216	0.934203	-0.64668	-0.99138	-0.078464	0.979505
5	0.559249	-0.85041	-0.25745	0.729402	0.848456	-0.1402	-0.37261	-0.58776	-0.40557	0.404024	0.388883	-0.45863	-0.11737	0.040401	-0.95844

Gambar 12. Nilai Bobot W (Bobot Pada Lapisan Keluaran)

Dimana kolom menunjukkan jumlah lapisan tersembunyi (Jumlah *Neuron Hidden*) dan baris menunjukkan jumlah lapisan keluaran (Jumlah *Output*).

Berikut adalah gambar hasil dari Bobot V (Bobot Pada Lapisan Tersembunyi) :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
1	2879799	7367.76	48997.8	2879799	77044.44	48267.9		591	592	593	594	595	596	597	598	599	600
2	2809054	12419.7	97512.1	229954.4	42091.1	17028.9		48018.5	2879444	7368.17	48264.7	24890.5	22187.64	20974.1	28000.6	12101.9	4712
3	12295.4	6432498	94674.29	22171.5	1.42498	79421.3		47229	220446.6	12023.4	47272.3	17521.7	10992.5	14884	22098.3	6432498	39492.3
4	-12925.5	6432498	94674.2	22181	1.42498	79421.37		391802.34	-12927.3	6432498	391802.34	12927.3	10992.5	14884	22098.3	6432498	39492.3
5	-38011.8	2917807	5586519	18027.9	2917806	5586519		391802.34	22098.3	6432498	391802.34	12927.3	10992.5	14884	22098.3	6432498	39492.3
6	-14481.8	272881.3	5586519	18027.9	2917806	5586519		391802.34	22098.3	6432498	391802.34	12927.3	10992.5	14884	22098.3	6432498	39492.3
7	-12774.1	181818.6	1.18E+05	-12774.1	181818.6	1.18E+05		391802.34	22098.3	6432498	391802.34	12927.3	10992.5	14884	22098.3	6432498	39492.3
8	-12774.1	181818.6	1.18E+05	-12774.1	181818.6	1.18E+05		391802.34	22098.3	6432498	391802.34	12927.3	10992.5	14884	22098.3	6432498	39492.3
9	-12774.1	181818.6	1.18E+05	-12774.1	181818.6	1.18E+05		391802.34	22098.3	6432498	391802.34	12927.3	10992.5	14884	22098.3	6432498	39492.3
10	-12774.1	181818.6	1.18E+05	-12774.1	181818.6	1.18E+05		391802.34	22098.3	6432498	391802.34	12927.3	10992.5	14884	22098.3	6432498	39492.3

Gambar 13. Nilai Bobot V (Bobot Pada Lapisan Tersembunyi)

Dimana kolom menunjukkan jumlah *input*-an (Jumlah data rekaman) yang akan dicari nilai bobotnya dan baris menunjukkan jumlah lapisan tersembunyi (Jumlah *Neuron Hidden*).

Setelah nilai Bobot V (bobot pada lapisan tersembunyi) dan nilai Bobot W (bobot pada lapisan keluaran) disimpan maka proses pengujian dapat dilakukan untuk mengetahui akurasi hasil pelatihan tersebut.

Pengujian

Pada tahap pengujian ini peneliti melakukan uji coba aplikasi sebanyak tiga kali untuk menghasilkan akurasi pengenalan yang maksimal. Cara melakukan uji coba adalah pertama peneliti membuka data rekaman yang akan diuji, kemudian langsung dilakukan proses pengujian. Jika data rekaman suara sesuai dengan *binerisasi* yang telah

ditentukan maka data rekaman tersebut dikenali. Jika tidak sesuai dengan *binerisasi* yang telah ditentukan maka data rekaman tidak dikenali.

Uji Coba 1

Pada uji coba 1 peneliti menggunakan Bobot V dan Bobot W yang telah dicantumkan pada Gambar 4.28 dan Gambar 4.29 dimana nilai *epoch*-nya adalah 1000, maksimal *error*-nya adalah 0.0001, dan *learning rate*-nya 0,5. Pada proses uji coba 1 ditemukan hasil rekaman yang dikenali dengan nilai *error* 0.25. Data rekaman dikenali karena *output* yang dihasilkan sesuai dengan proses *binerisasi* yang telah ditentukan.

Berikut ini merupakan hasil dari proses pengujian pengenalan menggunakan jaringan saraf tiruan *Back Propagation*.

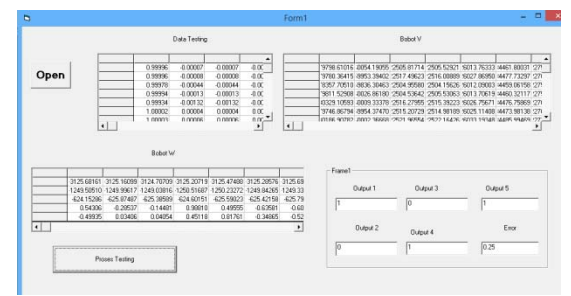


Gambar 14. Uji Coba 1

Uji Coba 2

Pada uji coba 2 peneliti menggunakan nilai bobot yang sama dengan uji coba 1 dimana nilai *epoch*-nya adalah 1000, maksimal *error*-nya adalah 0.0001, dan *learning rate*-nya 0.5. Pada proses uji coba 2 ditemukan hasil rekaman yang tidak dikenali meskipun nilai *error* 0.25. Data rekaman tidak dikenali karena ketika peneliti melakukan pengujian *output* yang dihasilkan tidak sesuai dengan data rekaman yang dipilih.

Berikut ini merupakan proses pengujian pengenalan menggunakan jaringan saraf tiruan *Back Propagation*.

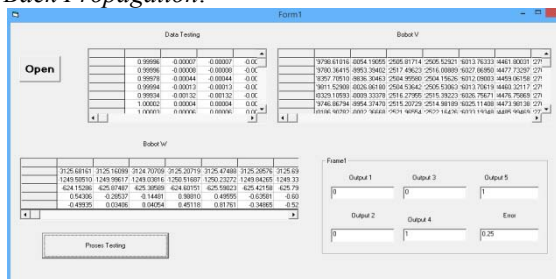


Gambar 15. Uji Coba 2

Uji Coba 3

Pada uji coba 3 peneliti menggunakan nilai bobot yang sama dengan uji coba 1 dan uji coba 2 dimana nilai *epoch*-nya adalah 1000, maksimal *error*-nya adalah 0.0001, dan *learning rate*-nya 0,5. Pada proses uji coba 3 ditemukan hasil rekaman yang dikenali dengan nilai *error* 0.25. Data rekaman dikenali karena *output* yang dihasilkan sesuai dengan *binerisasi* yang telah ditentukan.

Berikut ini merupakan proses pengujian pengenalan menggunakan jaringan saraf tiruan *Back Propagation*.



Gambar 16. Uji Coba 3

SIMPULAN

Dari hasil perancangan dan pengimplementasian jaringan saraf tiruan *Back Propagation* untuk pengenalan ucapan dapat diambil kesimpulan sebagai berikut :

1. Perancangan sistem pengenalan ucapan ini menggunakan Bahasa Pemrograman *Visual Basic* yang digunakan dalam proses ekstraksi ciri menggunakan metode *Linear Predictive Coding* (LPC) dan proses pelatihan jaringan saraf tiruan *Back Propagation*.
2. Stuktur jaringan saraf tiruan *Back Propagation* yang optimal untuk sistem pengenalan ucapan ini, peneliti menggunakan 200 *hidden layer*, nilai *learn rate* yang digunakan adalah 0.5 dan maksimal *error* 0.0001.
3. Agar mencapai tingkat keakurasian yang bagus peneliti melakukan *Training* sebanyak 1000 kali (Jumlah *Epoch*) yang menghasilkan nilai *error* sebesar 0.25, dimana keberhasilan pada pelatihan tersebut sebanyak 75%.
4. Kekurangan dalam sistem pengenalan ucapan ini terletak pada saat melakukan proses pengujian yang bergantung pada nilai *epoch* yang digunakan. Semakin

banyak nilai *epoch* yang digunakan semakin lama proses pengujian dihasilkan.

DAFTAR PUSTAKA

- [1] Arihputra. 2011. Fieldbus : voice/speech recognition (pengenalan ucapan), URL : <http://arihputra.blogspot.com/2011/11/voicesspeech-recognition-pengenalan.html?m=1>
- [2] Irfandy, Mahmud., 2004. Fieldbus : Aplikasi Pengenalan Ucapan Dengan Jaringan Syaraf Tiruan Propagasi Balik Untuk Pengendalian Robot Bergerak, URL : http://www.google.co.id/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CC0QFjAC&url=http%3A%2F%2Fprints.undip.ac.id%2F25298%2F1%2FM_L2F004490.pdf&ei=3b8QVfe_MM6JuATX0YDwAw&usq=AFQjCNG7zrOZr1SwTsT3Q72S2PHBjYZVcQ&bv m=bv.89184060.d.c2E
- [3] Resmana., Adipranata, Rudi., 1999. Fieldbus : Pengenalan Suara Manusia dengan Metode Jaringan Saraf Tiruan *Back Propagation* berbasis LPC; Universitas Kristen Petra
- [4] Rizal, Ahmad., Anggraeni, Lisa., Suryani, Vera., 2009. Fieldbus : Pengenalan Suara Paru-Paru Normal Menggunakan LPC dan Jaringan Syaraf Tiruan *Back-Propagation*, URL : http://scholar.google.co.id/scholar_url?url=http://achmadrizal.staff.telkomuniversity.ac.id/wp-content/uploads/sites/11/2014/06/fullpaper-klasifikasi-suara-paru-JST-BP.pdf&hl=en&sa=X&scisig=AAGBfm2bC8Iri-AIL5L3-8ArR7SifHMgAw&nossl=1&oi=scholar&ei=xNAQVaPSB9SKuATzhYCACg&ved=0CBwQgAMoATAA