

Teknik Pengenalan Sinyal Wicara Dengan Metoda Hidden Markov Model - Neural Network (Saraf Tiruan)

Dzulkifli¹, Mohammad Ahied²

¹Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Unesa
Kampus Ketintang Surabaya 60231, Telp. (031)-8289070
e-mail: dzulkiflihaa@yahoo.com

²Jurusan Pendidikan IPA
Fakultas Ilmu Pendidikan
Universitas Trunojoyo Madura
e-mail: ahied@trunojoyo.ac.id

Abstrak

Penelitian ini menjelaskan tentang teknik pengenalan sinyal wicara dengan metode Hidden Markov Model – Neural Network. Metode HMM tersebut digunakan untuk menentukan urutan data sinyal wicara berdasarkan inisial dan ekstraksi ciri dari masing-masing kata yang berbeda, sedangkan Neural network digunakan untuk menentukan keberhasilan dalam proses pengenalan sinyal wicara.

Penelitian ini diucapkan dengan pengucap tunggal dengan kata dan jumlah kata yang berbeda dan dilakukan dengan kumpulan 5 kata sampai 50 kata dalam bahasa Indonesia.

Eksperimen dilakukan dengan ketentuan : jumlah kata 50, struktur Neural Network yang digunakan berjumlah 5 Hidden layer, yang masing-masing 20 node, 10 node, 5 node, 10 node dan 20 node dan memperoleh hasil kesalahan pengenalan rata-rata dari kesalahan keseluruhan, mencapai 4.72 % atau prosentase keberhasilan 95.28 %, maka dari hasil eksperimen yang dicapai menunjukkan bahwa laju kesalahan dari sistem akan meningkat, seiring dengan bertambahnya jumlah kata.

Kata kunci : HMM-NN, vector kuantisasi, Backpropagasi, forward-backward, cepstral, LPC

ABSTRACT

This research describes the technique of the speech signal recognition method Markov Hidden Model - Neural Network. The HMM method is used to determine the order of the speech signal data based on the initials and extraction characteristics of each different, while the network was used to determine the success of the process of introducing the speech signal.

This research reciter pronounced with single words and the number of different words and performed by a collection of five words to 50 words in Indonesian.

Experiments performed with the following conditions: the number of words 50, the structure of Neural Network used were 5 Hidden layers, each 20 nodes, 10 nodes, five nodes, 10 nodes and 20 nodes and obtaining error recognition the average of the overall error, achieve 4.72% or 95.28% the percentage of success, then the experiment results achieved indicate that the error rate of the system will increase, as the number of words.

Keywords: HMM-NN, vector quantization, Backpropagation, forward-backward, cepstral, LPC

PENDAHULUAN

Secara garis besar teknik pengenalan sinyal wicara merupakan suatu usaha untuk menghasilkan suatu mesin cerdas yang mampu mengenali dan menentukan sinyal wicara yang diucapkan. Kesulitan yang paling mendasar adalah bagaimana melakukan ekstraksi terhadap sinyal ucapan menjadi beberapa parameter yang dapat digunakan untuk klasifikasi kata secara tepat.

Untuk mengatasi kesulitan tersebut, muncul dan berkembang beberapa metoda pendekatan, diantara pendekatan yang berhasil saat ini adalah HMM dengan Metode *LPC dan Cepstralnya* yang merupakan metode parametrik yang digunakan untuk merepresentasikan suatu segmen sinyal wicara.

Setelah HMM dengan metoda *LPC dan Ceptralnya* berhasil, maka untuk meningkatkan mutu pola pengenalan sinyal wicara dengan karakteristik sinyal wicara dengan faktor kesalahan pengenalan yang relatif kecil, maka penulis melakukan metoda yang sekarang sedang dikembangkan, yaitu metoda HMM-NN

Metoda ini berfungsi untuk memproses perubahan parameter sinyal yang diakibatkan dengan perbedaan sinyal wicara yang diucapkan dengan pengujian jumlah hidden layer, momentum, learning rate, lamanya iterasi dan jumlah kata, maka HMM-NN sangatlah membantu permasalahan yang ada.

Digital Signal Processing

Pengenalan pola ciri sinyal adalah kumpulan beberapa algoritma yang digunakan untuk mengklasifikasi data, untuk membuat satu atau lebih ciri-ciri sinyal dari yang banyak dan variatif. Untuk memperoleh ciri sinyal dibutuhkan metoda analisa sinyal, melalui proses normalisasi sebelum dilakukan ekstraksi ciri, proses normalisasi ini menggunakan pengolahan sinyal digital yang sudah umum dipakai. Urutan proses pengolahan sinyal digital dalam mengekstraksi ciri sinyal adalah :

1) Front End

Proses ini dimaksudkan memotong sinyal yang di depan dan belakang yang bukan sinyal informasi yang terjadi pada proses perekaman sinyal wicara, sehingga data yang tercakup merupakan data yang

ingin diolah lebih lanjut untuk menuju proses Pre-Emphasis

2) Pre-Emphasis

Proses ini dimaksudkan agar spektrum sinyal merata di semua frekuensi dengan menaikkan level frekwensi tinggi sebesar 6 dB/oktaf, fungsi transfer ditunjukkan pada persamaan (2.1).

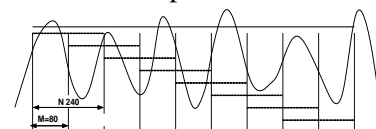
$$H(z) = 1 - \tilde{a}z^{-1}, 0.95 \leq a \leq 1 \tag{2.1}$$

Dengan model filter seperti diatas, maka persamaan (2.2) merupakan keluaran Preemphasis, yaitu:

$$\tilde{s}(n) = s(n) - \tilde{a}s(n-1) \tag{2.2}$$

3) Framing

Sinyal wicara kemudian diblok kedalam frame dengan *N* sampel, frame yang bersebelahan dipisahkan sejauh *M* sampel. Dimana $M=N/3$. dan terjadi penumpukan sebanyak $N-M$ sampel. Frame ke-2 mulai $2M$ sampel setelah frame ke-1 (M sampel setelah frame ke-2) dan terjadi penumpukan $N-2M$. Proses ini berlanjut mencapai nilai sinyal maksimum perframe..



Gambar 2.1. Framing

jika $M \ll N$ maka estimasi spektral LPC akan halus. Jika $M \gg N$, maka tidak terjadi penumpukan pada frame yang bersebelahan sehingga sinyal suara tidak muncul pada layar analisa frame dan korelasi antara hasil estimasi spektral LPC dari frame yang bersebelahan akan berisi komponen noise yang akan membesar dan jika *M* membesar semakin banyak komponen sinyal suara yang dihilangkan.

4) Windowing

Windowing digunakan untuk meminimalisasi efek diskontinuitas sinyal pada setiap awal dan akhir bingkai. Selain itu juga untuk memperkecil sinyal hingga menjadi nol pada awal dan akhir setiap bingkai. Jika window dinyatakan sebagai $w(n)$, maka hasil dari proses windowing adalah :

$$\tilde{s}_i(n) = s_i(n) * w(n), 0 \leq n \leq N-1 \quad (2.3)$$

Window yang digunakan adalah Window Hamming dengan bentuk persamaan :

$$w(n) = \left(0.54 - .46 \cos \left[\frac{2\pi n}{N-1} \right] \right), \quad 0 \leq n \leq N-1 \quad (2.4)$$

Ekstraksi ciri

Setelah sinyal wicara mengalami proses windo-wing, maka langkah berikutnya adalah melakukan proses ekstraksi ciri, ekstraksi ciri melalui 2 tahapan, yaitu :

a) LPC “Linear Predictive Coding”.

Mengekstraksi infomasi yang terkandung dalam sinyal Wicara dan analisa spektral yang digunakan untuk mengkarakterisasi sifat dari sinyal wicara terhadap waktu. Metoda LPC dapat memodelkan sinyal suara relatif lebih baik dari pada metode lain. Hal ini berlaku pada kondisi hampir lunak (*quasi steady state*) pada daerah bersuara dimana model “*all-pole*” LPC memiliki pendekatan yang baik pada *vokal tract spectral envelope*. Untuk daerah tidak bersuara dan daerah transien, LPC masih dapat memodelkan sinyal suara walaupun tidak sebaik pada daerah bersuara.

LPC memiliki kemampuan dalam pengolahan pola suara yang lebih baik daripada metode FFT (*fast fourier transformator*). Pada analisa LPC terhadap sinyal suara dapat memisahkan sumber dari **vokal tract** (*vokal yang terbentuk dari pernafasan*), sehingga representasi dari karakteristik vocal tract yang berkaitan pada wicara yang diproduksi dapat diketahui.

Metode LPC adalah Markov Model yang secara analitik dapat secara tepat dan sederhana untuk dapat diimplementasikan baik pada perangkat lunak

sehingga proses komputasi yang diperlukan lebih sedikit.

b) Koefisien Cepstral

Koefisien Cepstral $c(\tau)$ didefinisikan sebagai suatu transformasi balik Fourier dari logaritmik spektrum $|X(\omega)|$ pada selang waktu yang sangat pendek. Jika suatu sinyal suara $x(t)$ dimodelkan dan dianggap sebagai suatu respon dari suatu filter artikulasi vokal track yang digerakkan oleh pseudoperiodic source $g(t)$, maka :

$$x(t) = \int_0^t g(\tau)h(t-\tau) d\tau$$

$$X(\omega) = G(\omega)H(\omega)$$

$$(5.6) \log |X(\omega)| = \log |G(\omega)| + \log |H(\omega)|$$

$$c(\tau) = F^{-1} \log |X(\omega)| = F^{-1} \log |G(\omega)| + F^{-1} \log |H(\omega)|$$

Secara prinsip, persamaan dapat menggambarkan bahwa koefisien Cepstral $c(\tau)$ mengandung 2 informasi penting, yaitu elemen yang berkaitan dengan daerah frekuensi rendah (*low frequency element*) dan elemen yang berkaitan dengan frekuensi tinggi (*high frequency element*).

Konversi Parameter LPC ke Koefisien Cepstrum

Himpunan parameter LPC yang dapat diturunkan dari himpunan koefisien LPC adalah koefisien cepstrum LPC, $c(m)$, untuk itu dipakai rekursif

$$C_0 = \ln \sigma^2$$

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

$$C_m = \sum_{k=1}^{m-1} \left(\frac{K}{m} \right) C_k a_{a-m}, \quad m > p$$

dimana σ^2 adalah gain dari model LPC koefisien cepstrum, koefisien ini merupakan representasi *transformasi Fourier* dari logaritma *magnitudo spektrum*, menunjukkan lebih realiable dari pada ke-3 koefisien diatas. Reprrensatasi cepstrum untuk $Q > p$ digunakan dimana $Q \cong (3/2)p$.

c) Vector Quantization “VQ”

VQ adalah suatu metode untuk mengklasifikasi ciri dengan memanfaatkan metode vektor sebagai pembanding jarak antara ciri. Sejumlah vektor kode buku m_i dengan nilai bebas di pakai untuk memprediksi berbagai vektor data masukan x , dengan nilai yang telah terkuantisasi. Biasanya vektor kode buku sudah didesain untuk tiap kelas dari nilai x yang menunjuk kelas sama dengan menghitung jarak terdekat dengan menggunakan vektor m_i , perhitungan jarak menggunakan persamaan :

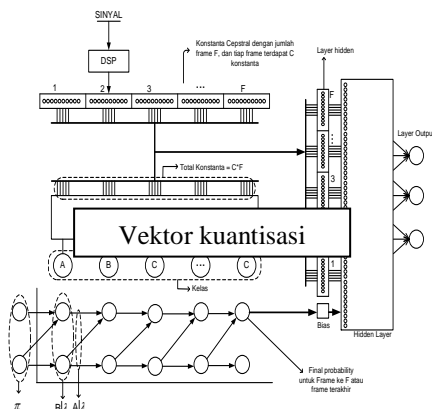
$$c = \arg \min \{ \|x - m_i\| \} \tag{2.7}$$

Dalam algoritma dasar VQ, hanya *reference vector* (vektor yang menunjuk pada kelas target) yang terdekat dengan *input vector* yang diupdate. Arah penggeserannya bergantung pada apakah *reference vector* tersebut (“winner”) berada pada kelas yang sama dengan *input vector*. Dalam algoritma yang diperbaiki, dua vector (*winner* dan *runner-up*) sama-sama belajar (berubah nilainya) jika beberapa syarat terpenuhi. Dasar pemikirannya adalah jika input berada kurang-lebih pada jarak yang sama dari *winner* maupun *runner-up*, maka keduanya harus berubah.

METODE PENELITIAN

3.1.Rancangan Model Proses

Rancangan model simulasi yang dirancang merupakan program pengenalan sinyal wicara. Diagram blok proses kerja simulasi dijelaskan oleh gambar dibawah ini :



Gambar 3.1 Blok diagram proses HMM-NN

Dalam proses tersebut diatas terbagi 3 segmen (1) Perekaman suara (2). DSP, (3).Pengenalan Sinyal wicara dengan Metoda HMM-NN, yaitu :

3.2.1. Perekaman suara atau pencuplikan suara

Fungsi dari program ini adalah mencuplik suara manusia, sehingga dapat dibaca oleh komputer. Aturan pencuplikan suara mengikuti format PCM (*pulse code modulation*). Dalam tesis ini menggunakan frekuensi sebesar 11.025Hz karena fasilitas ini yang disediakan oleh soundcard sebagai media pencuplikan suara. Untuk menyimpan sinyal wicara mengikuti format .wav dengan mode *mono* dengan jumlah bit per sample 16 bit.

3.2.2. Proses DSP (Digital Sinyal Processing)

Fungsi dari pengolahan sinyal digital ini adalah untuk mendapatkan karakteristik matematis dari sinyal wicara.

3.2.3. Pengenalan ciri sinyal

Pada pengenalan ciri dalam tesis ini menggunakan metode HMM (*forward dan forward-Backward*). dengan target berupa fungsi gaussian. Karena target yang diinginkan arah maju dan mundur,

HASIL DAN PEMBAHASAN

4 PENGUJIAN HASIL SIMULASI

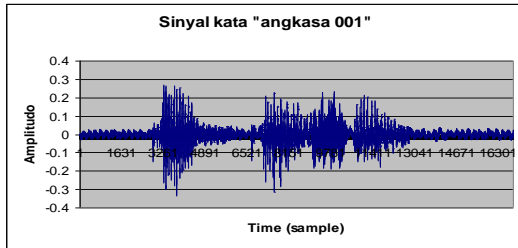
Pengujian awal Perangkat lunak yang digunakan dalam tesis ini meliputi 6 proses, yakni :

1. Pencuplikan suara manusia menjadi sinyal digital
2. Pengolahan sinyal digital untuk mendapatkan ciri matematis sinyal
3. Pengklasifikasian ciri dari beberapa sinyal
4. Mentraining data sinyal wicara yang dihasilkan

5. Menentukan nilai error Training
6. Menentukan hasil pengenalan sinyal wicara "Mapping"

▪ **Proses pencuplikan sinyal**

Dalam proses ini, dengan sample sinyal wicara "angkasa" sebagai salah satu data untuk uji coba. Dari hasil proses ini diperoleh sinyal digital yang disimpan dalam file format wave.



Gambar 4.1 Sinyal kata "angkasa"

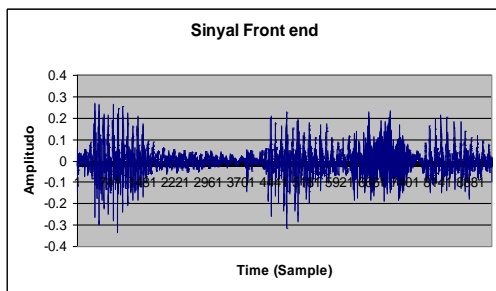
Didapatkan sinyal dengan panjang 19650 sample Frekuensi 11.025 Hz, interval waktu 0:0:1:782 mili detik.

4.1. Pengolahan Sinyal Digital

Setelah data di cuplik maka dilakukan pengolahan sinyal digital yang meliputi :

1) Deteksi Front-End

Hasil program untuk proses deteksi front-end terlihat pada gambar dibawah :



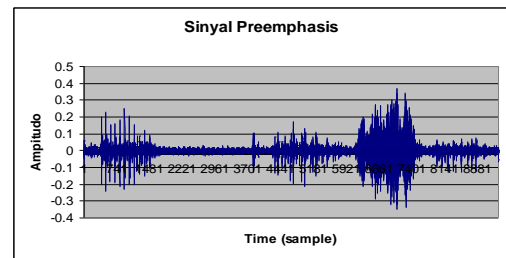
Gambar 4.2 Sinyal hasil deteksi front-end dari kata "angkasa"

amplitudo sesuai yang dikehendaki dalam hal ini 5 % dari amplitudo minimal, jika sinyal wicara menggunakan 16 bit maka nilai sample maksimal adalah 2^{15} hasilnya 32768, jadi untuk 5 % dari 32768 adalah sekitar 1638,4 , dengan maksud jika sinyal dengan amplitudo kurang dari 1638,4 maka akan terpotong dengan sendirinya, dapat dilihat dari hasil yang dicapai dalam proses pemotongan amplitudo,

yang mana diberikan satu contoh bentuk sinyal wicara kata "angkasa" selama 1,2

2) Pre-Emphasis

Hasil preemphasis menggunakan parameter \hat{a} sebesar 0.94 terlihat Hasil dari proses ini tidak mengubah jumlah sample sehingga masih mempunyai interval yang sama seperti proses sebelumnya, yakni 8312 sample dengan interval 0:0:0:779 mili detik.

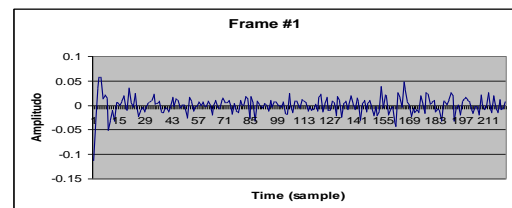


Gambar 4.3 Sinyal hasil Pre-Emphasis dari kata "angkasa"

3) Framing

Dalam uji coba digunakan pembingkai dengan aturan $N=240$ sample sebagai banyaknya sample dari satu bingkai, dan $M=80$ sample sebagai jarak antara bingkai. Hasil proses pembingkai terlihat pada gambar dibawah ini

:

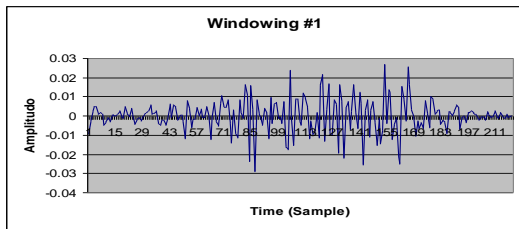


Gambar 4.4 Bingkai sinyal dari kata "angkasa"

Hasil dari proses ini adalah bingkai sinyal sebanyak 115 bingkai, dengan panjang tiap bingkai 220 sample dan jarak antara bingkai 80 sample, karena menggunakan frekuensi 11.025 Hz didapatkan interval waktu 0:0:0:22 mili detik.

4) Windowing

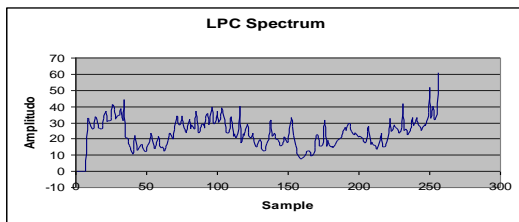
Proses windowing menggunakan windowing jenis hamming, hasil dari proses windowing ini terlihat pada gambar dibawah ini :



Gambar 4.5 Sinyal hasil proses windowing hamming pada bingkai ke-1 dari kata "angkasa"

5) LPC

Proses LPC menggunakan metoda autokorelasi dengan hasil terlihat pada gambar 4.6. Sinyal pada gambar tersebut didapat dari perkalian konstanta LPC dengan sinyal windowing.



Gambar 4.6 Sinyal hasil proses LPC pada bingkai ke-1 untuk kata "angkasa"

Konstanta LPC hasil pengolahan untuk bingkai ke-1 dengan 8 konstanta dapat dilihat pada keterangan dibawah

Koef LPC	Frame ke 1
1	0.01765
2	-0.48542
3	-0.36920
4	-0.35882
5	0.12216
6	0.10992
7	0.14160
8	-0.08157

Karena 1 bingkai menghasilkan p=8 konstanta, maka untuk satu sinyal dengan 153 bingkai didapatkan $8*153 = 1224$ koefisien konstanta.

6) Cepstral

Hasil sinyal pada proses cepstral ini adalah berupa konstanta cepstral sebanyak 11 untuk tiap satu bingkai. Konstanta cepstral hasil pengolahan untuk bingkai ke-1 dapat dilihat pada keterangan dibawah

Koef Cepstral	Frame ke 1
1	-0.86894
2	0.98736
3	1.01370
4	-0.19664
5	0.06430
6	-0.43377
7	0.53254
8	-0.28736
9	0.15919
10	-0.22051
11	0.22988

Karena 1 bingkai menghasilkan Q=11 konstanta, maka untuk satu sinyal dengan 184 bingkai didapatkan $11*184 = 2024$ koefisien konstanta. Dan data konstanta inilah yang dipakai untuk inputan ke proses pengklasifikasian ciri.

4.2. Pembuatan buku kode

Berikut ini hasil dari vektor ciri dari kata "angkasa" yang berupa deretan angka 2 yang menandakan indek output vektor cepstrum yang diperolehnya

Tabel- 4.3. NilaiKode Buku kata "Angkasa"

2,2,2,2,8,2,2,2,8,2,8,2,2,2,2,2,8,2,2,8,2,2
,2,2,8,2,8,2,2,8,2,8,2,2,2,2,2,2,
2,2,2,2,2,2,8,2,2,2,2,2,2,8,2,2,2,2,2,2,8,2
,2,8,8,2,2,8,2,2,2,2,2,2,2,8,2,2,
2,2,2,2,2,2,2,2,8,2,2,2,2,2,2,2,2,2,2,2,
,2,2,8,2,8,2,2,2,2,8,2,2,2,8,2,2,8,

Dengan demikian, maka masukan dari HMM berupa deretan indek dari buku kode dari masing-masing sinyal yang berbeda berdasar vektor indek dari buku kode tersebut, sehingga nampak bahwa jika diantara indek tersebut terdapat angka yang sama, berarti diantara vektor-vektor ciri antar segmen tidak memiliki kandungan ciri yang sama, dan sebaliknya. Sedangkan Algoritma Baum-Welch yang merupakan gabungan dari algoritma-algoritma

Tabel - 4.4. Nilai Probabilitas State untuk kata “Angkasa”

Tgt	Source				
	State 1	State 2	State 3	State 4	State 10
1	0.00389	0.01799	0.04538	0.0074	0.01039
2	0.09611	0.00624	0.06808	0.05893	0.02772
3	0.01112	0.02296	0.01363	0.0097	0.00901
4	0.18594	0.02421	0.09424	0.31213	0.34322
5	0.07226	0.01893	0.03101	0.06284	0.04774
6	0.52741	0.8373	0.64859	0.48537	0.41459
7	0.00801	0.00956	0.00972	0.02172	0.00708
8	0.02776	0.03163	0.05014	0.03612	0.03906
9	0.06549	0.02901	0.03703	0.00378	0.09888
10	0.00201	0.00217	0.00218	0.00201	0.0023

4.3. Pembentukan Parameter Model $\lambda (\pi, A, B)$

Dalam pembentukan parameter model π, A, B akan melibatkan beberapa parameter yaitu parameter arah maju α (alfa), parameter arah mundur β (beta), parameter ξ (Xi) dan parameter γ (gamma).

Tabel - 4.2. Nilai Probabilitas State untuk kata “Angkasa”

Tgt	Source				
	State 1	State 2	State 3	State 4	State 10
1	0.0097	0.01112	0.01363	0.00901	0.02296
2	0.05893	0.09611	0.06808	0.02772	0.00624
3	0.0074	0.00389	0.04538	0.01039	0.01799
4	0.31213	0.18594	0.09424	0.34322	0.02421
5	0.07226	0.06284	0.03101	0.04774	0.01893
6	0.52741	0.41459	0.64859	0.48537	0.8373
7	0.05578	0.09888	0.03703	0.00378	0.02901
8	0.00211	0.0023	0.00218	0.00201	0.00217
9	0.01434	0.00708	0.00972	0.02172	0.00956
10	0.02916	0.03906	0.05014	0.03612	0.03163

4.4. Pembentukan Observasi symbol setiap state $P(B|\lambda)$

Setelah didapat nilai State $P(A|\lambda)$ antara i ke j pada $(t+1)$ kemudian dilanjutkan ke nilai observasi simbol $P(B|\lambda)$ pada state j jika pada probabilitas $P(A|\lambda)$ maka probabilitas observasi dalam setiap symbol dalam satu keadaan $P(B|\lambda)$ juga didapat, dari jumlah simbol yang begitu banyak maka setiap simbolnya terdapat 10 state dimana tiap-tiap statenya hampir sama nilainya ini diakibatkan dengan sistem yang random pada kata yang diucapkan rata-rata mempunyai nilai 0,00097 lebih jelasnya lihat pada tabel 4.3.

Tabel- 4.3. Nilai Probabilitas ObservasiSymbol untuk kata “Angkasa” dengan jumlah symbol 32 dan jumlah State 10

Siml	State				
	1	2	3	4	10
S1	0.00097	0.00097	0.00097	0.00097	0.96993
S2	0.96993	0.96993	0.96993	0.96993	0.00097
S3	0.00097	0.00097	0.00097	0.00097	0.00097
S4	0.00097	0.00097	0.00097	0.00097	0.00097
S5	0.00097	0.00097	0.00097	0.00097	0.00097
S6	0.00097	0.00097	0.00097	0.00097	0.00097
S32	0.00097	0.00097	0.00097	0.00097	0.00097

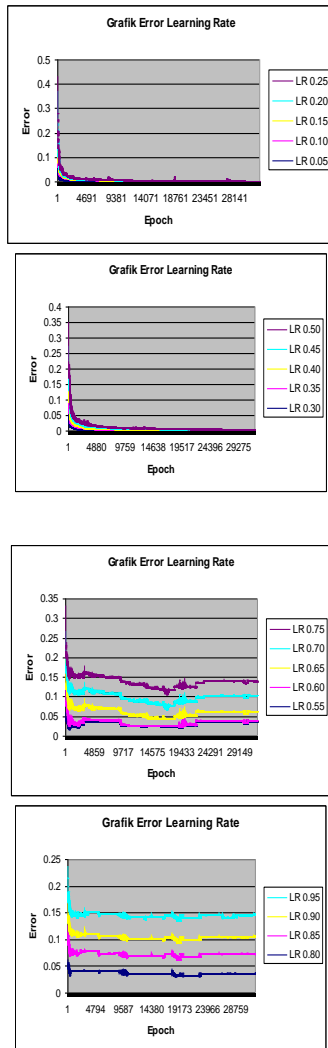
4.5. Pembentukan Inisial probabilitas

Tabel- 4.4.. Nilai Initial Probabilitas kata “

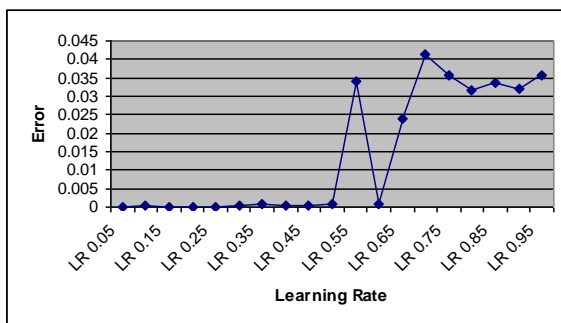
No	State ke	Inisial Prob	Prob Final
1	1	0.05614	- 2.4396982
2	2	0.10793	
3	3	0.01960	
4	4	0.03185	
5	5	0.08092	
6	6	0.61192	
7	7	0.00470	
8	8	0.04209	
9	9	0.04282	
10	10	0.00202	

7.Data hasil simulasi

7.1. Hasil simulasi dengan beda learning rate

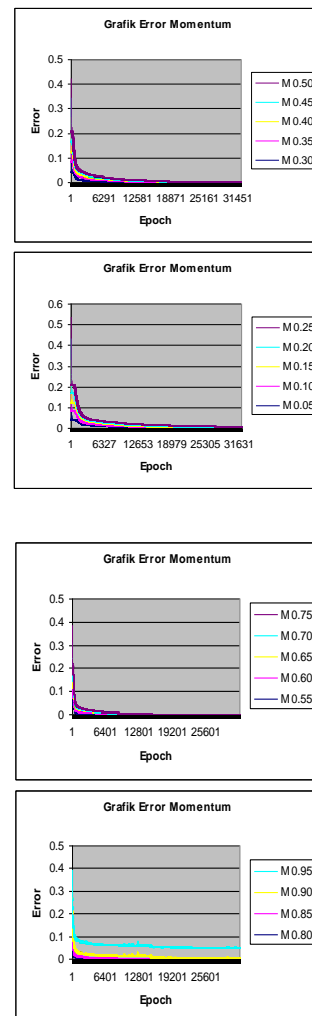


Gambar 4.7. Grafik nilai eror training pada momentum 0.85 dengan HL 3 (10, 20, 10) masing-masing learning rate

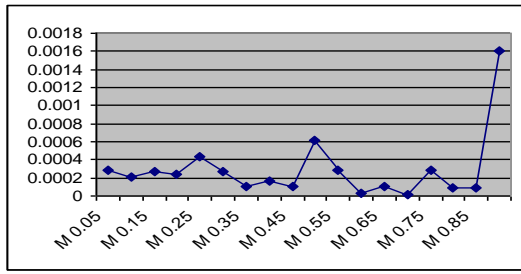


Uji dengan hidden layer 3 (10,20,10) iterasi 50.000, momentum 0.85, tujuannya untuk mengetahui bobot kali kecepatan sehingga jika bobot besar tentu membutuhkan learning rate yang kecil untuk lebih teliti mendeteksi laju error yang akan dicapainya dengan lamanya iterasi memungkinkan semua data dapat dikenali dengan jumlah eror yang kecil.

7.2. Hasil simulasi dengan beda Momentum

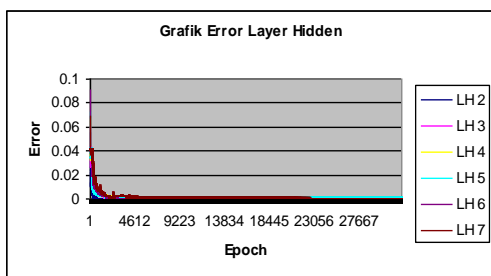


Gambar 4.8. Grafik nilai eror training pada learning rate 0.25 dengan HL 3 (10, 20, 10) masing-masing momentum



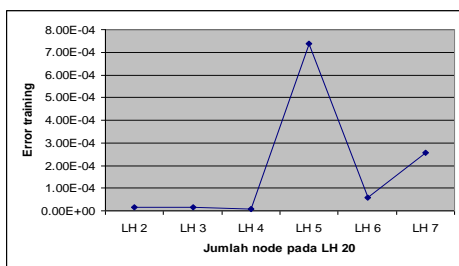
Uji dengan hidden layer 3 (10,20,10) iterasi 50.000, Learning rate 0.25 pada nilai momentum 0.70 dengan laju error 1.82 E-05

7.3. Hasil simulasi dengan beda Hidden layer

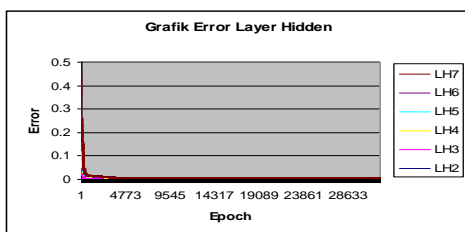


Gambar 4.9. Grafik pada LH dengan masing-masing node 20 learning rate 0.15 dan momentum 0.70

Uji dengan node 20 pada masing-masing HL iterasi 50.000, pada LH ke 4 mempunyai 6.82E-06 laju error yang kecil.

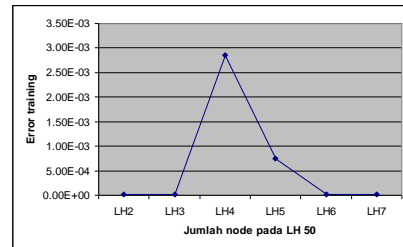


7.4. Hasil simulasi dengan beda Hidden Layer



Gambar 4.10.. Grafik pada M dengan masing-masing node 50 learning rate 0.15 dan momentum 0.70

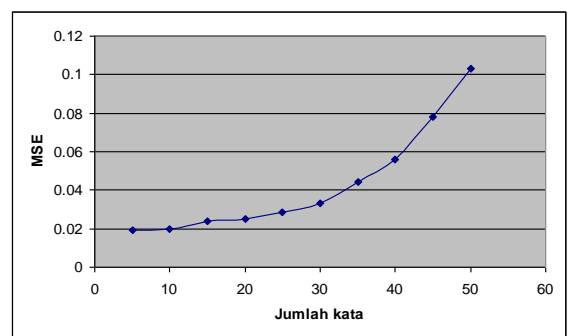
Uji dengan node 50 pada masing-masing HL iterasi 50.000, pada LH ke 3 mempunyai 9.51E-06 laju error yang kecil.



7.5. Hasil simulasi MSE

Tabel 4.5. Tabel kinerja jaringan syaraf tiruan terhadap jumlah kata

No.	Jumlah kata	MSE
1	5	0.01914
2	10	0.01955
3	15	0.02307
4	20	0.02511
5	25	0.02878
6	30	0.03343
7	35	0.04443
8	40	0.05016
9	45	0.07819
10	50	0.10297

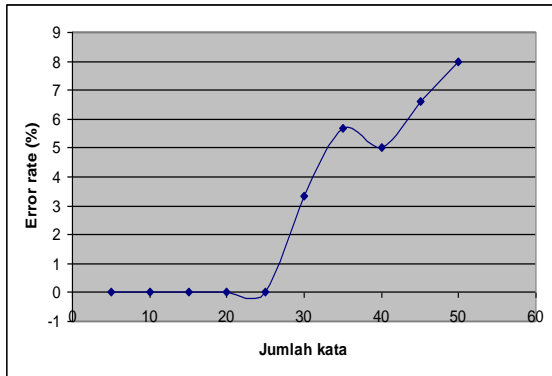


Gambar 4.11. Grafik MSE terhadap perubahan jumlah kata

7.6. Hasil eksperimen

Tabel 4.14. Rata-rata kesalahan pada jumlah kata yang berbeda

No	Jumlah kata	Rata-rata kesalahan Pengenalan (%)
1	5	0
2	10	0
3	15	0
4	20	0
5	25	0
6	30	3.33
7	35	5.7
8	40	5
9	45	6.6
10	50	8
	Rata-rata	4.72 %



Gambar 4.12. Grafik rata-rata kesalahan pengenalan sebagai fungsi kata

KESIMPULAN

Beberapa kesimpulan yang dapat diambil dari hasil tesis ini antara lain :

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat disimpulkan beberapa hal penting sebagai berikut :

- Proses Digital Sinyal Prosesing, merupakan pengenalan awal bentuk sinyal mulai dari Front end, Pre-Emphasis, Framing, Windowing, FFT, LPC dan Cepstral yang merupakan parameter

yang cukup representatif sebagai parameter ciri dari sinyal wicara.

- Dari proses beberapa percobaan parameter baik learning rate dan momentum maka nilai yang paling tepat memakai Learning rate 0.15, dan 0.70 untuk momentum untuk mendapatkan hasil yang akurat
- Metoda Hidden Markov Model – Neural Network cukup baik digunakan untuk proses pengenalan sinyal wicara dengan kesalahan pengenalan rata-rata dari prosentase kesalahan keseluruhan 4.72 % atau prosentase keberhasilan mencapai 95.28 %
- Pengoptimalan learning rate, jumlah hidden layer, node hidden dan jumlah kata dapat menentukan faktor keberhasilan yang diharapkan.

SARAN

- Untuk mendapatkan nilai error yang kecil dalam analisis pengenalan sinyal dengan metoda HMM-NN, maka diperlukan penelitian-penelitian lanjutan dengan algoritma yang berbeda dengan karakteristik pengambilan data yang bermacam-macam.
- Sistem hybrid tersebut dapat dibentuk dengan menghubungkan metoda HMM-NN, dengan posisi diletakkan sebelum HMM atau sesudah HMM ataupun sejajar kedua-duanya dengan algoritma yang beragam.

Daftar Pustaka

1. Alex Waibel, chair Raj Reddy, Jaime Corbonell, Richard Lippmann, MIT Lincoln Labs, "Speech Reconition using Neural Networks", Scholl of Computer Science Carnegie Mellon University Pittsbugh, Pennsylvania, May 1995.
2. A. Achmad, "Implementasi Sistem Pengenalan Suara Manusia dengan menggunakan Koefisien linear predictive coding (LPC)", Industrial Electronics Seminar (IES), Oktober 18-19, 2000.
3. Fausett Laurance, " Fundamental of Neural Network", Architecture, Algorithm and Aplications, Prentice Hall International Inc. USA, 1994.
4. Furui, Sadoaki., "Digital Speech Processing, Synthesis and Recognition", Marcel Dekker. Inc, 1989.
5. L.B. Rabiner, B.H. Juang, " Fundamental of Speech Recognition", Prentice Hall International, 1993.
6. Marzuki A, dalam tesisnya, "Implementasi Hidden Markov Model (HMM) pada sistem identifikasi speaker," dengan frekuensi sampling 12 Khz dengan lebar frame sama
7. Mochatr M, " Design of intelligent Approximator using Genetig and Hybrid Learning Basis Function Neural Network", Industrial Electronics Seminar (IES), Oktober 18-19, 2000.
8. Pelton., Gordon., "Voice Processing", Mc Graw Hill,1993.
9. R.C. Gonsalez and R.E. Woods, *Digital Image Processing*. Addison Wesley, 1993.
10. R.F. Vaz and D. Cyganski SPIE: *Aplication Of Digital Image Processing XIII*, Vol. 1,349, pp. 452-459, 1990.
11. Rabiner, Lawrence R., Bernard, "Theory and Aplication of Digital Signal Processing", Prentice Hall of India, New Delhi, 1990.
12. R.L. Rabiner, B.S.Atal, M.R.Sambur, "LPC Prediction Error-Analysis of Its Variation With the position of the Analysis Frame ", IEEE Transactions on Acoustics, speech and Signal Processing, vol ASSP-25, pp. 434-442 Oct 1977.
13. Suyanto,. "Pengenalan Sinyal Ucapan Suku kata Bahasa Indonesia dengan fonem Sebagai komponen dasarnya", Industrial Electronics Seminar (IES), Oktober 27-28 1999.
14. Tomkins Willems J., "Biomedical Digital Signal Processing", Prentice Hall International Inc., USA, 1993..