KUANTISASI SKALAR NILAI BISPEKTRUM UNTUK PENCIRI SINYAL PADA SISTEM IDENTIFIKASI PEMBICARA DENGAN HMM SEBAGAI PENGENAL POLA

Agus Buono [1] Benyamin Kusumoputro [2] Wisnu Jatmiko [3]

[1] Departemen Ilmu Komputer FMIPA IPB
Kampus IPB Darmaga-Bogor
email : pudesha@yahoo.co.id
[2] Fakultas Teknik Universitas Indonesia
Fakultas Teknik Kampus UI Depok
email : nynykusumo@yahoo.com
[3] Fakultas Ilmu Komputer Universitas Indonesia
Fakultas Ilmu Komputer Kampus UI Depok

ABSTRACT

Pada paper ini disajikan teknik kuantisasi skalar untuk merepresentasikan nilai bispketrum sinyal suara pada sistem identifikasi pembicara (SIP). Jumlah channel yang dicoba adalah 128, 250, 400 dan 600, dan jenis statistik nilai bipektrumnya adalah rata-rata, median dan rata-rata di atas kuartil 3. Output kuantisasi ini diekstrak menggunakan Mel-Frequency Cepstrum Coefficients (MFCC) dengan jumlah koefisien 13 dan dilanjutkan dengan pengenalan pola menggunakan left-right Hidden Markov Model (HMM) dengan jumlah state 3.

Data yang dipergunakan melibatkan 10 pembicara yang mengucapkan ujaran "Pudesha" sebanyak 80 kali tanpa pengkondisian, dan disampling dengan frekuensi 1.1 kHz. Sebanyak 75% data digunakan untuk training dan sisanya sebagai data uji. Dalam hal ini ada 5 set data uji, yaitu sinyal asli, sinyal asli yang telah ditambah Gaussian noise (20 dB, 10 dB, 5 dB, dan 0 dB).

Hasil percobaan menunjukkan bahwa teknik kuantisasi skalar menghasilkan SIP dengan akurasi diatas 98% untuk semua channel. Namun untuk sinyal bernois 20 dB, terjadi penurunan dengan kisaran 69% hingga 83%. Untuk noise yang lebih berat, sistem gagal melakukan pengenalan dengan baik. Juga terlihat bahwa rata-rata bispektrum diatas kuartil 3 memberikan akurasi yang lebih baik di banding dua statistik lainnya.

Key words

Higher Order Statistic(HOS), Bispektrum, Mel-Frekuensi Cepstrum Coefficients (MFCC), Hidden Markov Model (HMM), Sistem Identifikasi Pembicara (SIP)

1. Pendahuluan

Pada [1] telah ditunjukkan bahwa tehnik Mel-Frequency Cepstrum Coefficients (MFCC) yang berbasis power spektrum untuk ekstraksi ciri sinyal suara dapat bekerja dengan baik khususnya untuk sinyal tanpa penambahan noise. Jika digabungkan dengan HMM sebagai pengenal pola pada SIP memberian akurasi rata-rata 99%. Namun demikian, untuk sinyal bernois 20 dB, sistem yang dihasilkan gagal, dan akurasi drop hingga 56%. Hal ini disebabkan nilai power spektrum sebagai penciri sinyal dan merupakan input dari proses MFCC bersifat sensitif terhadap noise. Sementara itu pada [2-4], ditunjukkan secara empiris bahwa statistik orde tinggi (HOS) mampu menekan pengaruh Gaussian noise, sehingga akurasi sistem dapat diperbaiki. Namun demikian, feature masukan ke sistem diperoleh dengan merata-ratakan seluruh frame yang ada, sehingga kurang memperhatikan aspek temporalnya dan penerapannya ke aplikasi lainnya menjadi terbatas.

Rabiner, 1989, [5], menyebutkan bahwa HMM merupakan proses stokastik yang memodelkan hubungan antar state serta state dengan observasinya dari waktu ke waktu. Oleh karena itu, model HMM secara konseptual sesuai dengan proses alami suara dihasilkan. Pemakaian HMM pada pemrosesan suara telah banyak dikupas dan memberikan akurasi di atas 95 %. Dari dua fakta empiris

i atas, maka Buono, Kusumoputro dan Jatmiko pada [6] ielakukan penggabungan HOS orde 3 (Bispektrum) engan HMM untuk membentuk SIP. Pendekatan yang ilakukan adalah dengan memperluas teknik MFCC dari D menjadi 2D, dengan tujuan agar nilai bispektrum dapat iekstrak menjadi feature-feature dengan dimensi yang iuh lebih kecil, sehingga HMM dapat bekerja dengan aik. Akurasi sistem yang dihasilkan untuk sinyal tanpa enambahan noise adalah sekitar 99% dan 89% untuk nyal dengan penambahan noise 20dB. Namun demikian ntuk noise yang lebih besar, sistem gagal melakukan engenalan dengan baik. Oleh karena itu, pada penelitian i dilakukan kuantisasi nilai bispektrum terlebih dahulu ebelum masuk ke proses ekstraksi yang menggunakan ietodologi MFCC.

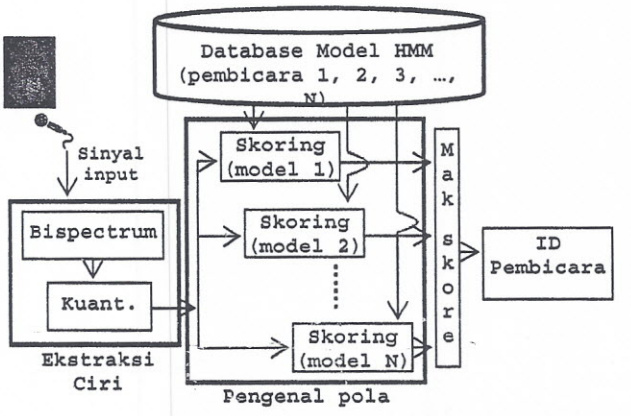
Selanjutnya, paper ini disajikan dengan susunan ebagai berikut : Bagian 2 mengenai kuantisasi bispektrum an integrasinya dengan HMM dengan pembahasan mulai ari prinsip sistem identifikasi pembicara, Higher Orde tatistic orde 3 (Bispektrum), metode kuantisasi skalar ispektrum, Hidden Markov Model, dan data serta ncangan percobaan. Hasil serta pembahasan disajikan ada bagian 3. Akhirnya, kesimpulan serta saran untuk enelitian selanjutnya disajikan pada bagian 4.

. Kuantisasi Bispektrum dan Integrasinya dengan HMM

.1 Sistem Identifikasi Pembicara

Identifikasi pembicara merupakan proses untuk ienentukan pembicara berdasar input suara yang iberikan [7]. Secara umum, sistem identifikasi pembicara rdiri dari dua subsistem, yaitu subsistem ekstraksi ciri an subsistem pengenal pola, seperti disajikan pada ambar 1. Subsistem ekstraksi ciri melakukan proses ansformasi sinyal input ke dalam satu set vektor ciri ebagai representasi dari sinyal suara suatu pembicara ntuk proses selanjutnya. Subsistem pencocokan pola ierupakan bagian untuk melakukan identifikasi suatu embicara yang belum diketahui dengan cara iembandingkan sinyal suaranya yang telah diekstrak ke alam vektor ciri dengan set vektor ciri dari pembicara ng telah diketahui dan tersimpan dalam sistem.

Dari aspek pengembangan sistem, ada dua fase pada stem identifikasi pembicara. Fase pertama adalah tahap elatihan. Pada fase ini sistem melakukan pelatihan untuk ienentukan parameter model untuk setiap pembicara erdasar data suara pembicara tersebut. Pada penelitian ini, ase yang digunakan adalah "pudesha" dan dimodelkan engan *Hidden Markov Model* (HMM). Dari sampel data engan frase "pudesha" ini, model setiap pembicara dilatih dengan menggunakan algoritma Baum Welch seperti yang disajikan pada [5]. Fase kedua adalah tahapan pengujian, yaitu sinyal input yang diberikan kepada sistem dicocokan dengan dengan model setiap pembicara yang ada pada sistem. Keputusan untuk menentukan pembicara didasarkan pada skor tertinggi untuk setiap model. Untuk penghitungan skor ini digunakan algoritma *Forward* [5].



Gambar 1. Blok diagram sistem identidikasi pembicara dengan HMM sebagai pengenal pola

2.2 Higher Order Statistic Orde 3 (Bispektrum)

Jika {X(k)}, k = 0, ±1, ±2,..., adalah proses stokastik yang bernilai real, maka *cumulant* order 3 adalah $c_3^X(\tau 1,\tau 2)$, yang dirumuskan sebagai, [8]:

$$c_3^X(\tau 1,\tau 2)=\sum_{p=1}^{3}\sum_{m=1}^{R}(-1)^{p-1}(p-1)!E\left(\prod_{X_i\in S_1}X_i\right)E\left(\prod_{X_j\in S_2}X_j\right)E\left(\prod_{X_k\in s_3}X_k\right) \quad (1)$$

R adalah banyaknya cara menyekat set {Xk, Xk+τ1, Xk+τ2} menjadi p sekatan, dengan p = 1, 2, 3. Sebagai ilustrasi, untuk p = 2, maka diperoleh 3 kemungkinan sekatan (R = 3), yaitu: s1 = {Xk, Xk+τ1}, s2 = {Xk+τ2}; s1 = {Xk}, s2 = {Xk+τ1, Xk+τ2}; dan s1 = {Xk+τ1}, s2 = {Xk, Xk+τ2}. Bispektrum, yang disebut juga sebagai spektrum cumulant, adalah transformasi Fourier dari barisan cumulant tersebut, dan diformulasikan sebagai [8]:

$$C_3^x(\omega_1,\omega_2)=\sum_{\tau_1=-\infty}^{+\infty}\sum_{\tau_2=-\infty}^{+\infty}c_3^x(\tau_1,\tau_2)\exp\{-j(\omega_1\tau_1,\omega_2\tau_2)\} \quad (2)$$

Untuk proses stasioner, *cumulant* order 3 dapat diformulasikan sebagai:

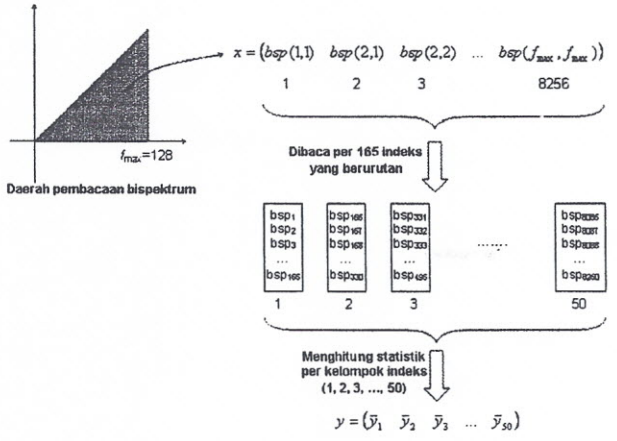
$$c_3^x(\tau_1,\tau_2)=E\{x(t)x(t+\tau_1)x(t+\tau_2)\} \quad (3)$$

Paramater τ_1 dan τ_2 pada Pers. (1-3) di atas adalah lag yang secara teoritis bernilai bilangan real. Pada prakteknya,

nilai bispektrum ini diduga dari sejumlah samples data. Secara umum ada dua pendekatan dalam menduga bispektrum, yaitu pendekatan parametrik dan pendekatan konvensional. Pendekatan konvensional dikelompokkan menjadi tiga, yaitu teknik tidak langsung (*indirect technique*), teknik langsung (*direct technique*), dan modulasi kompleks (*complex demodulates*). Pada penelitian ini digunakan metode konvensional dengan teknik tidak langsung untuk menduga nilai bispektrum. Hal ini dikarenakan teknik ini lebih sederhana dibanding lainnya. Algoritma secara lengkap dapat dilihat pada [8].

2.3 Metode Kuantisasi Skalar nilai Bispektrum

Oleh karena nilai bispektrum bersifat simetrik, maka pembacaan hanya dilakukan pada daerah segitiga dari ruang domain bispektrum. Pada absis i, pembacaan ordinatnya dilakukan dari 1 hingga i, untuk i=1, 2, 3, ..., f_{max} dengan f_{max} adalah frekuensi maksimum dari domain bispektrum. Oleh karena itu, daerah pembacaan bispektrum yang berbentuk segitiga tersebut diubah menjadi vektor dengan indeks 1, 2, 3, ..., $f_{max}(f_{max}+1)/2$. Pada penelitian ini, nilai f_{max} adalah 128, sehingga vektor daerah pembacaan tersebut mengandung 8256 elemen. Kuantisasi skalar dilakukan dengan membaca sejumlah indeks yang berurutan dan mengubahnya menjadi satu nilai dengan salah satu cara, yaitu cara merata-ratakan, median, atau rata-rata setelah persentil 75%. Hasil dari kuantisasi skalar dari sebuah frame suara ini adalah satu vektor yang disebut sebagai vektor pewakil. Banyaknya unsur pada vektor pewakil disebut sebagai jumlah channel yang nilainya tergantung dari jumlah indeks yang dibaca pada pembacaan. Gambar 2. memberikan ilustrasi pembentukan vektor pewakil mengandung 50 channel.

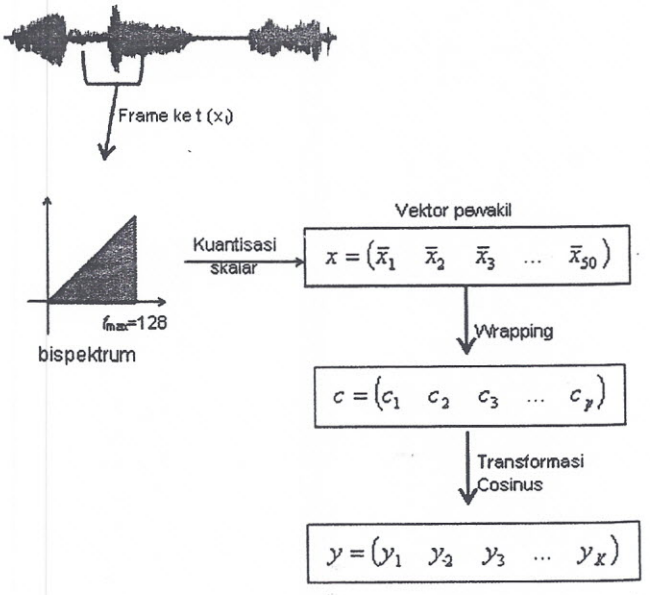


Gambar 2. Kuantisasi Skalar dengan Jumlah Channel 50 untuk Bispektrum dengan f_{max}=128 (jumlah indeks per kelompok [8256/50]=165)

Proses kuantisasi skalar tersebut diimplementasikan dengan Algoritme berikut, [2] :

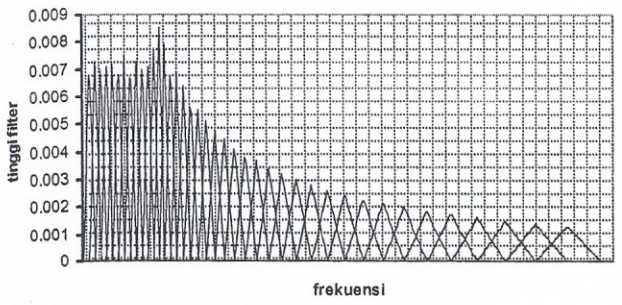
```
Algoritme Proses Kuantisasi Skalar
Input : BSP[128:128];
Output: Pewakil[1:channel];
%membaca ½ domain BSP
k=0;
for i=1:128
    for j=1:i
        k=k+1;
        MAG(k)=BSP(i,j);
    end
end
%Menghitung kuantisasi skalar
channel=p;
k=floor(length(MAG)/channel);
offset=0;
for i=1:channel
    t=0;
    for j=(1+offset):(k+offset)
        t=t+1;
        tem(t)=MAG(j);
    end
    Pewakil(i)=mean(tem);
end
```

Hasil kuantisasi skalar ini adalah sebuah vektor pewakil dengan sejumlah tertentu channel, yaitu 128, 250, 400 atau 600. Untuk mereduksi jumlah channel digunakan teknik seperti yang dilakukan pada MFCC, yaitu *wrapping* dan transformasi kosinus. Gambar 3 menyajikan proses lengkap proses ekstraksi ciri.



Gambar 3. Alur Proses Ekstraksi Ciri Mengunakan Teknik Skalar Kuantisasi – *Wrapping* dan Transformasi Kosinus (WC)

Proses wrapping menggunakan sejumlah filter seperti ditunjukkan pada Gambar 4 yang terdiri dari 13 filter linear dan 27 filter logaritma, [9].



Gambar 4. Empat Puluh Filter pada Proses *Wrapping*

Proses *wrapping* terhadap vektor pewakil, x, yang berdimensi p (p adalah banyaknya channel) menggunakan formula, [9] :

$$c(i) = \log\left[\sum_{f=1}^{p} x(f) * h_i(f)\right], \text{ i=1, 2, 3, ..., 40} \qquad (4)$$

Dalam hal ini $h_i(f)$ adalah nilai filter ke i untuk dimensi ke f pada vektor pewakil. Oleh karena itu, setiap vektor pewakil yang berdimensi p akan ditransformasi menjadi vektor baru yang berdimensi 40. Berikutnya vektor hasil *wrapping* ini akan ditransformasi menggunakan tranfsormasi kosinus dengan formula :

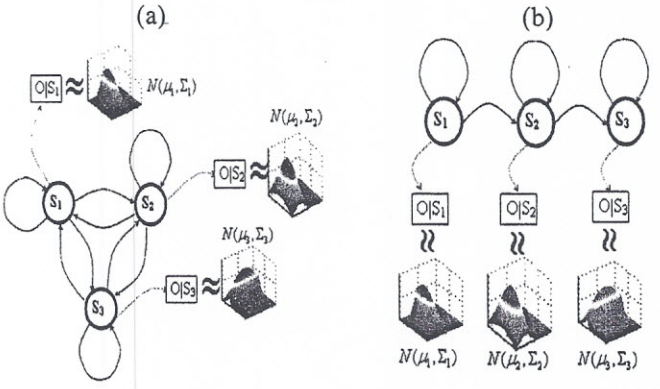
$$y(k) = \sum_{i=1}^{40} \cos\left[2(i-1)k\pi / 40\right], \text{ k=1,2,3,...,13} \qquad (5)$$

2.4 Hidden Markov Model (HMM)

Hidden Markov Model (HMM) merupakan model markov orde satu yang mempunyai dua jenis state, yaitu hidden state dan observable state. Setiap *hidden state* dapat menghasilkan suatu *outcome* yang teramati pada setiap periode t, yaitu O_t. *Outcome* dari *hidden state* ini disebut sebagai *observable state* atau *emitten*. Oleh karena itu, dari periode t=1 hingga t=T diperoleh barisan peubah teramati (*observation state*) $O=O_1, O_2, O_3, \ldots, O_T$, yang merupakan *outcome* dari barisan peubah tak teramati $Q=q_1, q_2, q_3, \ldots, q_T$. Berdasar hubungan antar *state*, dikenal dua jenis HMM, yaitu *ergodic* dan *left-right* HMM. Pada *Ergodic* HMM, antar dua state selalu ada *link*, sehingga disebut juga sebagai *fully connected* HMM. Sedangkan pada *left-right* HMM, state dapat disusun dari kiri ke kanan sesuai dengan *link*-nya. Gambar 5. memberikan contoh *ergodic* dan *left-right* HMM dengan tiga *hidden state* dengan distribusi peubah *emitten*-nya adalah *Gaussian*. Suatu HMM dinotasikan dengan, [10] :

$$\lambda = (A, B, \Pi)$$

A adalah matriks peluang transisi, B adalah matriks peluang observasi dan Л adalah vektor peluang awal.



Gambar 5. Contoh HMM dengan Tiga Hidden State dan Distribusi Emitten *Gaussian*, (a) *Ergodic*, (b) *Left-Right* HMM

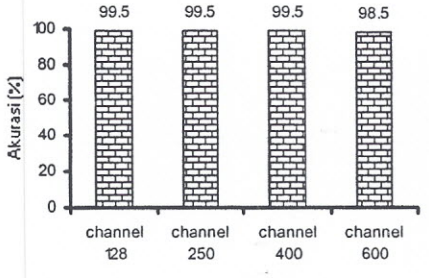
Pada penelitian ini digunakan left-right HMM dengan 3 state dan dilatih dengan algoritma Baum-Welch. Sedangkan untuk pengujian menggunakan algoritma forwad.

2.4 Data Percobaan

Penelitian ini menggunakan data dari 10 pembicara yang mengucapkan ujaran "pudesha" tanpa pengkondisian masing-masing sebanyak 80 kali yang disampling dengan frekuensi 1.1 kHz. Untuk pelatihan model, maka dari 75 % dipilih sebagai data latih dan sisanya sebagai data uji. Untuk berikutnya, dibuat lima set data uji, yaitu sinyal asli dan sinyal asli dengan penambahan noise (20 dB, 10 dB, 5 dB, dan 0 dB). Proses kuantisasi dicobakan dengan empat jumlah channel, yaitu 128, 250, 400 dan 600. Untuk menghitung bispektrum digunakan tiga jenis statistik, yaitu rata-rata, median dan rata-rata bispektrum di atas kuartil 3.

3. Hasil Percobaan

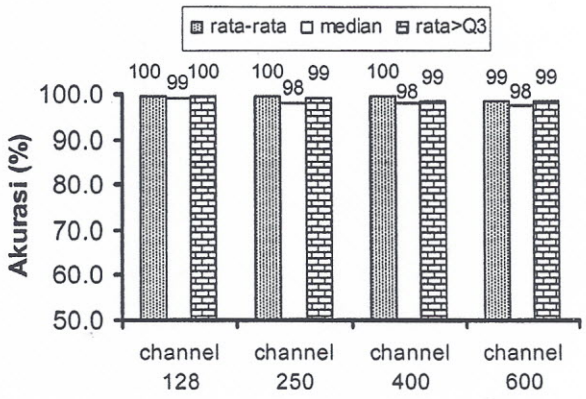
Gambar 6 menyajikan hasil pengenalan terhadap data uji tanpa penambahan noise.



Gambar 6. Akurasi Sistem untuk Data Asli pada Berbagai Jumah Channel

Terlihat bahwa dengan metode yang dikembangkan dapat melakukan pengenalan dengan baik (>98%) untuk sinyal tanpa penambahan noise, baik pada jumlah channel 128, 250, 400 maupun 600. Begitu juga dari segi jenis statistik

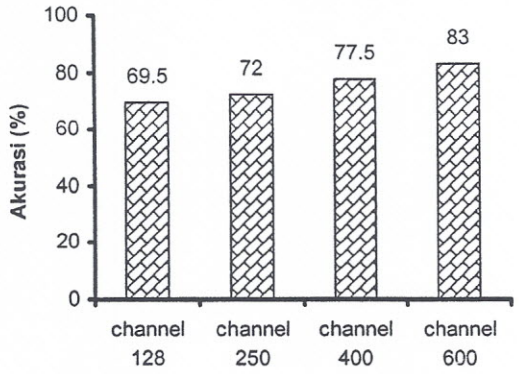
nilai bispektrum, terlihat bahwa akurasi sistem berkisar pada 99% untuk ketiga jenis statistik yang dipergunakan, seperti ditunjukkan Gambar 7.



Gambar 7. Perbandingan Akurasi antara Statistik Rataan, Median dan Rataan BSP di atas Persentil 75% pada Berbagai Channel untuk Sinyal Asli

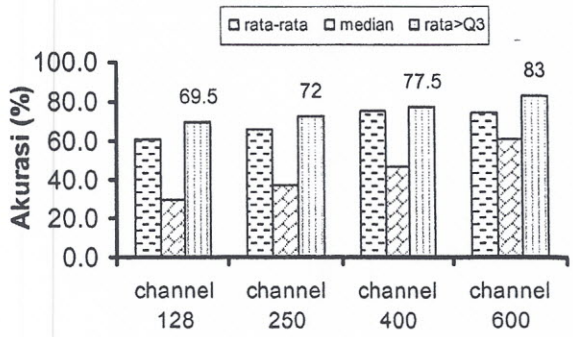
Dua fakta di atas menunjukkan bahwa untuk sinyal asli, metode yang diusulkan dapat melakukan pengenalan dengan baik, berapun jumlah channel maupun jenis statistik bispektrum yang dipergunakan.

Sedangkan untuk sinyal dengan penambahan noise 20 dB, terjadi penurunan akurasi, seperti pada Gambar 8.



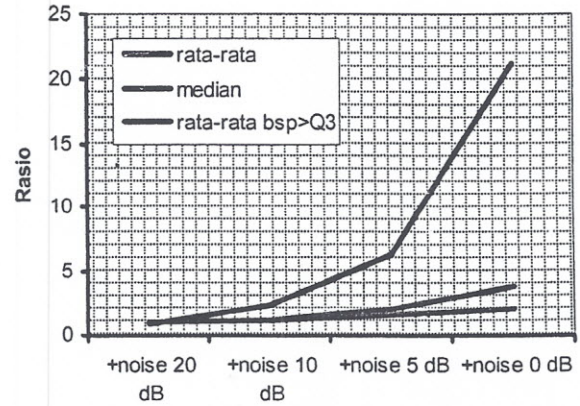
Gambar 8. Perbandingan Akurasi antar Berbagai Jumlah Channel dengan Kuantisasi Skalar-WC untuk Sinyal Suara yang Ditambah *Noise* 20 dB

Terlihat akurasi tertinggi adalah 83% dengan jumlah channel 600 dan menjadi 69.5% kalau jumlah channel 128. Untuk melihat pengaruh jenis statistik, perhatikan Gambar 9 yang menyajikan perbandingan akurasi antar ketiga statistik dan jumlah channel untuk data bernois 20 dB. Dari ketiga statistik tersebut, median memberikan akurasi yang paling rendah di antara ke tiga statistik di atas, baik pada channel 128, 250, 400, hingga 600. Jenis statistik terbaik adalah rataan bispektrum di atas kuartil 3.



Gambar 9. Perbandingan Akurasi antara Statistik Rataan, Median dan Rataan BSP di atas Persentil 75% pada Berbagai Channel untuk Sinyal Asli dengan Penambahan *Noise* 20 dB

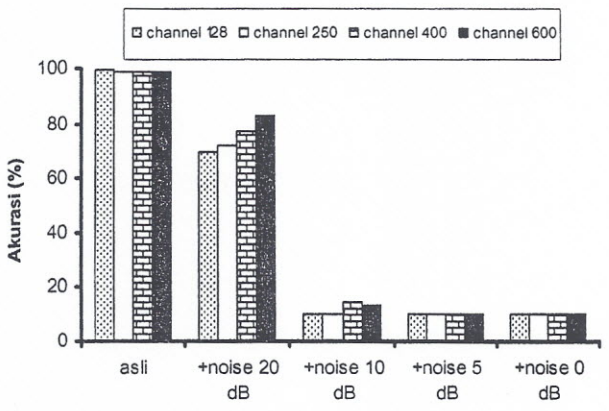
Hasil percobaan yang ditunjukkan pada Gambar 8 dan Gambar 9 memberikan bukti empiris bahwa nilai rata-rata bispektrum di atas persentil 75% bersifat lebih *robust* (kurang sensitif) terhadap pengaruh *noise*, dibandingkan dengan rataan maupun median, seperti ditunjukkan juga pada Gambar 10. Gambar 10 menampilkan perbandingan nilai statistik pada berbagai *noise* terhadap nilai statistik besaran tersebut saat tidak diberi tambahan *noise*. Pada *noise* 20 dB, ketiga statistik mempunyai nilai yang relatif sama dengan nilainya untuk sinyal tanpa penambahan *noise*, dengan rasio 1.000059, 0.899651 dan 0.999753, masing-masing untuk rata-rata, median dan rata-rata bsp di atas persentil 75%. Dengan bertambahnya *noise*, rasio ini meningkat. Hal ini menunjukkan bahwa *noise* yang diberikan menaikkan nilai bsp, yang pada akhirnya berpengaruh pada statistik yang dipakai. Pada penambahan *noise* sebesar 0 dB, statistik median meningkat tajam, yaitu 21.1 kali, rata-rata meningkat 3.9 kali, dan rata-rata bsp setelah Q3 relatif lebih baik, yaitu hanya meningkat menjadi 2.1 kali nilainya saat kondisi tanpa penambahan noise.



Gambar 10. Rasio Nilai Statistik pada Berbagai Penambahan *Noise* terhadap Nilainya pada Kondisi Tanpa Penambahan *Noise*

Gambar 11. menyajikan akurasi sistem untuk berbagai *noise* dan berbagai channel dengan menggunakan statistik rata-rata bsp di atas Q3. Meskipun secara empiris, statistik

rata-rata bsp di atas Q3 relatif tidak sensitif terhadap *noise*, namun akurasi sistem untuk noise 10 dB hingga 0 dB turun secara drastis.



Gambar 11. Perbandingan Akurasi antar Channel untuk Berbagai *Noise*

Hal ini menunjukkan bahwa teknik kuantisasi yang dilakukan masih belum bisa dengan baik untuk merepresentasikan data sinyal yang terkontaminasi noise. Salah satu kelemahan yang ada adalah bahwa penentuan pusat channel pada kuantisasi skalar ini dilakukan dengan membagi rata dari semua sampel bispektrum yang ada. Oleh karena itu pusat channel tidak mencerminkan distribusi spasial data bispektrum.

4. Kesimpulan

Dua hal yang bisa diutarakan berdasar hasil percobaan identifikasi dengan kuantisasi skalar adalah :

1. Sistem yang dikembangkan dengan teknik kuantisasi skalar dikombinasikan dengan transformasi *wrapping* dan kosinus mampu melakukan pengenalan dengan akurasi yang baik untuk sinyal tanpa penambahan *noise* (>98%). Namun pada sinyal yang ditambah *noise*, akurasi sistem turun secara drastis untuk *noise* 10 dB hingga 0 dB.
2. Statistik rata-rata nilai bispektrum di atas persentil 75% relatif lebih *robust* terhadap noise dibanding dengan statistik rata-rata maupun median.

Dari hasil percobaan terlihat bahwa salah satu kelemahan kuantisasi skalar adalah pada pemilihan channel yang bersifat tetap, sehingga hal ini mengabaikan distribusi empiris dari bispektrum. Oleh karena itu, riset selanjutnya sebaiknya menggunakan kuantisasi yang mengakomodasi distribusi empiris data bispektrum dalam domain frekuensi.

REFERENSI

[1] Buono, A. and B. Kusumoputro., February 2008, "A Problem in Data Variability on Speaker Identification System Using Hidden Markov Model", Prociding of the Conference on Artificial Intelligence and Application (AIA), IASTED , Innsbruck-Austria.

[2] Fanany, M.I. dan B. Kusumoputro., 1998, "Bispectrum Pattern Analysis and Quantization to Speaker Identification", Thesis Master Ilmu Komputer, Fasilkom Universitas Indonesia.

[3] Hidayat, N. dan B. Kusumoputro., 1999, "Pengembangan Sistem Pengenal Suara Menggunakan Estimasi Trispektrum dan Kuantisasi Skalar", Thesis Master Ilmu Komputer Fasilkom Universitas Indonesia.

[4] Triyanto, A. dan B. Kusumoputro, 2000, "Ekstraksi Ciri Pada Data Suara Menggunakan Spektra Orde Tinggi dan Kuantisasi Vektor untuk Identifikasi Pembicara Menggunakan Jaringan Neural Buatan", Thesis Program Master Ilmu Komputer, Fasilkom Universitas Indonesia.

[5] Rabiner, L.R., 1989, "A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition", Proceeding IEEE, Vol 77 No. 2.

[6] Buono, A., W. Jatmiko, and B. Kusumoputro, April 2009, "Perluasan Metode MFCC 1D ke 2D Sebagai Ekstraksi Ciri Pada Sistem Identifikasi Pembicara Menggunakan HMM", Jurnal Makara, Sains, Vol. 13, No. 1, Universitas Indonesia.

[7] C. Cornaz, U. Hunkeler, 2005, "An Automatic Speaker Recognition System", Ecole Polytechnique, Federale De Lausanne, http://www.ifp.uiuc.edu/~minhdo/teaching/speaker_recognition.

[8] C. L. Nikeas, A. P. Petropulu, 1993,"Higher Order Spectra Analysis: A Nonlinear Signal Processing", Framework, Prentice-Hall, Inc., New Jersey.

[9] Todor D. Ganchev, 2005, *"Speaker Recognition"*, Ph.D. Thesis. Wire Communications Laboratory, Department of Computer and Electrical Engineering, University of Patras Greece.

[10] Dugad, R. Dan U.B. Desai, 1996, "A Tutorial on Hidden Markov Model", Technical Report, Departement of Electrical Engineering, Indian Institute of Technology, Bombay.

Agus Buono, memperoleh gelar Sarjana dan Master bidang statistik di IPB pada tahun 1992 dan 1996. Gelar Master dan Doktor bidang Ilmu Komputer diperoleh dari Universitas Indonesia pada tahun 2000 dan 2009. Saat ini sebagai Staf Pengajar Departemen Ilmu Komputer Institut Pertanian Bogor.

Benyamin Kusumoputro, memperoleh gelar Sarjana bidang fisika dari Institut Teknologi bandung dan Doktor Optoelektronika dari Tokyo Institute of Technology-Jepang. Gelar Profesor diperoleh pada tahun 2002 dari Universitas Indonesia. Saat ini sebagai Staf Pengajar Fakultas Teknik Universitas Indonesia.

Wisnu Jatmiko, memperoleh gelar Sarjana Elektro dan Magister Ilmu Komputer dari Universitas Indonesia. Ph.D bidang komputer diperoleh dari Jepang pada tahun 2008. Saat ini sebagai Dosen Fakultas Ilmu Komputer Universitas Indonesia.

MODEL JARINGAN SYARAF TIRUAN *RESILIENT BACKPROPAGATION* UNTUK IDENTIFIKASI PEMBICARA DENGAN PRAPROSES MFCC

Agus Buono [1] Irman Hermadi [2] Nurhadi Susanto [3]

[1, 2, 3] Departemen Ilmu Komputer FMIPA IPB
Kampus IPB Darmaga-Bogor
email : pudesha@yahoo.co.id

ABSTRACT

Pada penelitian ini, dikembangkan suatu model jaringan syaraf tiruan resilient backpropagation untuk identifikasi pembicara denganekstraksi ciri menggunakan teknik MFCC. Data suara yang digunakan dalam penelitian ini adalah data suara yang diambil secara unguided atau tanpa panduan dari 10 pembicara yang mengucapkan ujaran "komputer". Selain itu diamati pula pengaruh noise terhadap akurasi identifikasi dengan cara menambahkan white gaussian noise pada data yang digunakan. Untuk meningkatkan keyakinan pendeteksian, digunakan nilai threshold sebagai batas minimum dari seorang pembicara.

Hasil percobaan menunjukkan bahwa jumlah neuron terbaik adalah 100, dan untuk sinyal asli, akurasi rata-rata diperoleh sebesar 96%. Namun untuk sinyal bernois 30 dB dan 20 dB, akurasi rata-rata berkisar 60-70% dan 40-50%. Dengan memberikan threshold, mesipun akurasi turun menjadi 85%, namun tingkat keyakinan pengenalan menjadi lebih tinggi. Dalam hal ini tidak ada salah klasifikasi dari seorang pembicara ke pembicara lain.

Key words

Jaringan Syaraf Tiruan (JST) Resilient Backpropagation, Mel-Frekuensi Cepstrum Coefficients (MFCC), Sistem Identifikasi Pembicara (SIP)

1. Pendahuluan

Seperti disebutkan dalam [1] bahwa persyaratan ciri biometrik sebagai pengenal seseorang, adalah bersifat alami, mudah diukur, tidak terlalu berubah dari waktu ke waktu, tidak mudah ditiru, tidak dipengaruhi kondisi phisik, serta tidak terlalu terganggu dengan adanya gangguan lingkungan. Selain suara adalah besaran yang hampir memenuhi semua kriteria tersebut, sistem identifikasi berbasis suara juga lebih murah, karena sistem yang dikembangkan lebih bersifat software.

Dari riset yang sudah ada, teknik ekstraksi ciri menggunakan model MFCC mampu mengekstrak ciri suara dengan baik. Buono dan Kusumoputro, [2], melakukan identifikasi pembicara dengan ekstraksi teknik MFCC dan HMM sebagai pengenal pola memberikan akurasi rata-rata 99%. Oktavianto 2004, [3], menggunakan jaringan syaraf tiruan propagasi balik untuk pengenalan pembicara memberikan hasil yang di atas 90%. Beberapa modifikasi dari prosedur propagasi balik telah diajukan untuk menambah kecepatan pembelajaran. Martin Riedmiller dan Braun, 1993, dalam [4], telah mengembangkan suatu metode yang disebut *Resilient Backpropagation*. Metode ini telah terbukti memiliki kecepatan pembelajaran yang baik dan juga andal, [4]. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan model jaringan syaraf tiruan *resilient backpropagation* untuk mengidentifikasi pembicara pada data yang direkam tanpa pengarahan.

Selanjutnya, paper ini disajikan dengan susunan sebagai berikut : Bagian 2 mengenai teknik MFCC dan JST resilient untuk identifikasi pembicara dengan pembahasan mulai dari prinsip sistem identifikasi pembicara, teknik ekstraksi MFCC, JST (propagasi balik standar, inisialisasi, dan propagasi balik resilient), dan data percobaan. Hasil serta pembahasan disajikan pada bagian 3. Akhirnya, kesimpulan serta saran untuk penelitian selanjutnya disajikan pada bagian 4.

2. Ekstraksi MFCC dan JST Resilient untuk Identifikasi Pembicara

2.1 Sistem Identifikasi Pembicara

Identifikasi pembicara merupakan proses untuk menentukan pembicara berdasar input suara yang

diberikan [5]. Secara umum, sistem identifikasi pembicara terdiri dari dua subsistem, yaitu subsistem ekstraksi ciri dan subsistem pengenal pola. Subsistem ekstraksi ciri melakukan proses transformasi sinyal input ke dalam satu set vektor ciri sebagai representasi dari sinyal suara suatu pembicara untuk proses selanjutnya. Subsistem pencocokan pola merupakan bagian untuk melakukan identifikasi suatu pembicara yang belum diketahui dengan cara membandingkan sinyal suaranya yang telah diekstrak ke dalam vektor ciri dengan set vektor ciri dari pembicara yang telah diketahui dan tersimpan dalam sistem. Dari aspek pengembangan sistem, ada dua fase pada sistem identifikasi pembicara. Fase pertama adalah tahap pelatihan. Pada fase ini sistem melakukan pelatihan untuk menentukan parameter model untuk setiap pembicara berdasar data suara pembicara tersebut.

Menurut Campbell (1997), [6], Pengenalan pembicara berdasarkan jenis aplikasinya dibagi menjadi:

1. Identifikasi pembicara adalah proses mengenali seseorang berdasarkan suaranya. Identifikasi pembicara dibagi dua, yaitu:
   - Identifikasi tertutup *(closed-set identification)* yang mana suara masukan yang akan dikenali merupakan bagian dari sekumpulan suara pembicara yang telah terdaftar atau diketahui.
   - Identifikasi terbuka *(open-set identification)* suara masukan boleh tidak ada pada kumpulan suara pembicara yang telah terdaftar.
2. Verifikasi pembicara adalah proses menerima atau menolak permintaan identitas dari seseorang berdasarkan suaranya.

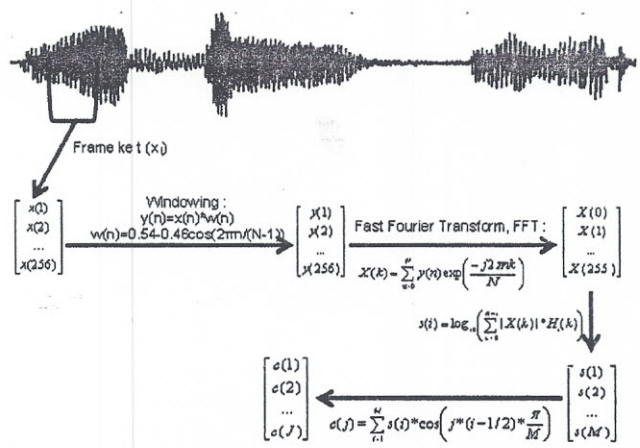
Sedangkan berdasarkan teks yang digunakan, pengenalan pembicara dibagi menjadi dua, [6] :

1. Pengenalan pembicara bergantung teks yang mengharuskan pembicara untuk mengucapkan kata atau kalimat yang sama, baik pada pelatihan maupun pengenalan.
2. Pengenalan pembicara bebas teks yang tidak mengharuskan pembicara untuk mengucapkan kata atau kalimat yang sama, baik pada pelatihan maupun pengenalan.

Penelitian yang dilakukan adalah identifiasi pembicara secara tertutup dan bersifat *text dependent*.

2.2 Mel-Frequency Cepstrum Coefficients (MFCC)

Ekstraksi ciri merupakan proses untuk menentukan satu nilai atau vektor yang dapat dipergunakan sebagai penciri obyek atau individu. Di dalam pemrosesan suara, ciri yang biasa dipergunakan adalah nilai koefisien cepstral dari sebuah frame. Satu teknik ekstraksi ciri sinyal suara yang umum dan menunjukkan kinerja yang baik adalah teknik *Mel-Frequency Cepstrum Coefficient*,(MFCC) yang menghitung koefisien cepstral dengan mempertimbangkan persepsi sistem pendengaran manusia terhadap frekuensi suara. Dibandingkan dengan metode ekstraksi ciri lainnya, Davis dan Mermelstein memperlihatkan bahwa MFCC sebagai teknik ekstraksi ciri memberikan hasil pengenalan yang tinggi, [7]. Diagram alur teknik MFCC dalam mengekstrak sinyal suara adalah seperti pada Gambar 1.



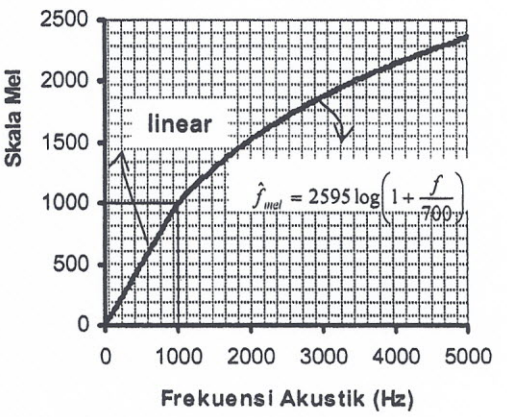
Gambar 1. Ilustrasi Ekstraksi dengan MFCC dengan Panjang Frame 256

Dari Gambar 1 terlihat bahwa sinyal dibaca frame demi frame, dan dilakukan windowing untuk setiap frame untuk berikutnya dilakukan transformasi Fourier. Dari nilai hasil transformasi Fourier ini selanjutnya dihitung spektrum mel menggunakan sejumlah (M) filter yang dibentuk sedemikian sehingga jarak antar pusat filter adalah konstan pada ruang frekuensi mel. Dari literatur yang ada, skala mel ini dibentuk untuk mengikuti persepsi sistem pendengaran manusia yang bersifat linear untuk frekuensi rendah dan logaritmik untuk frekuensi tinggi, dengan batas pada nilai frekuensi akustik sebesar 1000 Hz. Proses ini dikenal dengan nama *Mel-Frequency Wrapping*. Koefisien MFCC merupakan hasil transformasi *Cosinus* dari spektrum mel tersebut, dan dipilih J koefisien. Transformasi kosinus berfungsi untuk mengembalikan domain, dari frekuensi ke domain waktu. Du dalam [8], hubungan antara frekuensi akustik dengan skala mel (**Mel**ody) adalah sebagai berikut :

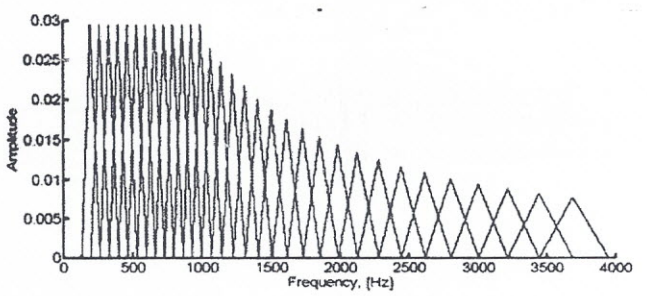
$$F_{mel} = \begin{cases} 2595 * \log_{10}\left(1 + \dfrac{F_{Hz}}{700}\right) & jika \quad F_{Hz} > 1000 \\ F_{Hz} & jika \quad F_{Hz} \le 1000 \end{cases} \quad (1)$$

dan dilukiskan seperti pada Gambar 2. Terlihat bahwa untuk frekuensi rendah, filter yang digunakan menggunakan skala linear, sehingga lebarnya konstan. Sedangkan untuk frekuensi tinggi (>1000 Hz), filter dibentuk dengan skala logaritma. Pada penelitian ini digunakan model filter Slaney, dalam [7], yang terdiri 40

filter segitiga (13 linear dan 27 logaritmik) seperti disajikan pada Gambar 3.



Gambar 2. Grafik Hubungan Frekuensi dengan Skala Mel



Gambar 3. Filter Slaney untuk Proses Wrapping

Dari 40 filter yang sudah dibentuk, maka dilakukan wrapping terhadap sinyal dalam domain frekuensi dan menghasilkan satu komponen untuk setiap filter dengan formula berikut :

$$X_i = \log_{10}\left(\sum_{k=0}^{N-1} |X(k)|\, H_i(k)\right) \qquad (2)$$

Dalam hal ini i=1, 2, 3, …, M (M adalah jumlah filter segitiga) dan $H_i(k)$ adalah nilai filter segitiga ke i untuk frekuensi akustik sebesar k. Nilai koefisien MFCC ke j akhirnya diperoleh menggunakan transformasi kosinus sesuai formula berikut :

$$C_j = \sum_{i=1}^{M} X_i \cos\left(j(i-1)/2\frac{\pi}{M}\right) \qquad (3)$$

dengan j=1,2,3,…,K, K adalah jumlah koefisien MFCC yang diinginkan dan M adalah jumlah filter.

2.3 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) merupakan suatu sistem pemroses informasi yang memiliki persamaan secara umum dengan cara kerja jaringan syaraf biologi, [9]. Metode komputasional dari JST diinspirasikan oleh cara kerja sel-sel otak manusia. Untuk berpikir, otak manusia mendapat rangsangan dari *neuron-neuron* yang terdapat pada indera manusia, kemudian hasil rangsangan tersebut diolah sehingga menghasilkan suatu informasi.

Menurut Fausett 1994, [9], suatu JST dicirikan oleh tiga hal sebagai berikut:

1. Arsitektur jaringan syaraf tiruan
   Arsitektur jaringan ialah pengaturan *neuron* dalam suatu lapisan, pola hubungan dalam lapisan dan di antara lapisan.
2. Teknik pembelajaran (penentuan pembobot koneksi)
   Metode pembelajaran digunakan untuk menentukan nilai pembobot yang akan digunakan pada saat pengujian.
3. Fungsi aktivasi
   Fungsi aktivasi merupakan fungsi yang menentukan level aktivasi, yaitu keadaan internal sebuah *neuron* dalam JST. Keluaran aktivasi ini biasanya dikirim sebagai sinyal ke *neuron* lainnya.

JST Propagasi Balik Standar

Menurut Fu 199, [10], jaringan propagasi balik (*propagation network*) merupakan jaringan umpan maju berlapis banyak (*multilayer feedforward network).* Aturan pembelajaran propagasi balik disebut *backpropagation* yang merupakan jenis dari teknik *gradient descent* dengan *backward error (gradient) propagation.* Fungsi aktivasi yang digunakan dalam propagasi balik ialah fungsi sigmoid. Hal ini disebabkan karena dalam jaringan propagasi balik fungsi aktivasi yang digunakan harus kontinu, dapat didiferensialkan, dan monoton naik, [9]. Salah satu fungsi aktivasi yang paling banyak digunakan ialah sigmoid biner, yang memiliki selang [0, 1] dan didefinisikan sebagai:

(2.26)

$$f_1(x) = \frac{1}{1+\exp(-x)} \qquad (4)$$

Dengan turunannya

$$f_1'(x) = f_1(x)[1 - f_1(x)] \qquad (5)$$

Jaringan ini menggunakan metode pembelajaran dengan pengarahan (*supervised learning*).

Setelah dilakukan inisialisasi bobot dan bias (2.27) (berpengaruh pada kecepatan JST dalam mencapai kekonvergenan [9]), pada pelatihan JST propagasi balik terdapat tiga tahapan, yaitu pelatihan input yang bersifat umpan maju, penghitungan galat, dan penyesuaian pembobot. Secara umum cara kerja JST propagasi balik ada beberapa langkah. Pertama, pola input dan target dimasukkan ke dalam jaringan. Selanjutnya pola input ini akan berubah sesuai dengan propagasi pola tersebut ke lapisan-lapisan berikutnya hingga menghasilkan output.

Output ini akan dibandingkan dengan target. Apabila dari hasil perbandingan ini dihasilkan nilai yang sama, proses pembelajaran akan berhenti. Tetapi apabila berbeda, maka jaringan mengubah pembobot yang ada pada hubungan antar *neuron* dengan suatu aturan tertentu agar nilai output lebih mendekati nilai target.

Proses pengubahan pembobot adalah dengan cara mempropagasikan kembali nilai korelasi galat output jaringan ke lapisan-lapisan sebelumnya (propagasi balik). Kemudian dari lapisan input, pola akan diproses lagi untuk mengubah nilai pembobot, hingga akhirnya memperoleh output jaringan baru. Proses ini dilakukan berulang-ulang sampai diperoleh nilai yang sama atau minimal sesuai dengan galat yang diinginkan. Proses perubahan pembobot inilah yang disebut proses pembelajaran.

Inisialisasi Pembobot Nguyen-Widrow

Inisialisasi pembobot bertujuan untuk meningkatkan kemampuan *neuron-neuron* tersembunyi untuk melakukan pembelajaran. Hal ini dilakukan dengan mendistribusikan pembobot dan bias awal sedemikian rupa sehingga dapat meningkatkan kemampuan lapisan tersembunyi dalam melakukan proses pembelajaran. Inisialisasi Nguyen-Widrow didefinisikan sebagai persamaan berikut, [9] :

- Hitung harga faktor penskalaan β

$$\beta = 0.7\, p^{1/n} \qquad (6)$$

dimana:

β = faktor penskalaan
n = jumlah *neuron* lapisan input
p = jumlah *neuron* lapisan tersembunyi

- Untuk setiap unit tersembunyi (j=1, 2,... ,p):
  - Hitung v_{ij} (lama) yaitu bilangan acak diantara -0.5 dan 0.5 (atau diantara -γ dan +γ). Pembaharuan pembobot v_{ij} (lama) menjadi v_{ij} baru yaitu:

$$v_{ij}(baru) = \frac{\beta v_{ij}(lama)}{\left\| v_j(lama) \right\|} \qquad (7)$$

  - Tetapkan bias.
    v_{ij} = Pembobot pada bias bernilai antara $-\beta$ dan β.

Resilient Backpropagation

Resilient backpropagation (RPROP) adalah salah satu algoritma yang digunakan untuk mempercapat laju pembelajaran pada pelatihan jaringan syaraf tiruan propagasi balik. RPROP melakukan penyesuaian nilai bobot secara langsung berdasarkan informasi dari gradien lokalnya. Untuk melakukannya, pada tiap nilai bobot diberikan suatu nilai perubahan bobot individual yang secara personal menentukan besarnya perubahan bobot. Nilai perubahan ini terus berubah selama proses pembelajaran berdasarkan pada pengamatan lokalnya terhadap fungsi galatnya (Riedmiller dan Braun, 1993, dalam [4]).

Secara sederhana, algoritma ini menggunakan tanda turunan untuk menentukan arah perbaikan bobot-bobot. Besarnya perubahan setiap bobot ditentukan oleh suatu faktor yang diatur pada parameter yang disebut *delt_inc* dan *delt_dec*. Apabila gradien fungsi *error* berubah tanda dari satu iterasi ke iterasi berikutnya, maka bobot akan berkurang sebesar *delt_dec*. Sebaliknya apabila gradien *error* tidak berubah tanda dari satu iterasi ke iterasi berikutnya, maka bobot akan berkurang sebesar *delt_inc*. Apabila gradien *error* sama dengan 0 maka perubahan sama dengan perubahan bobot sebelumnya. Pada awal iterasi, besarnya perubahan bobot diinisalisasikan dengan parameter *delta0*. Besarnya perubahan tidak boleh melebihi batas maksimum yang terdapat pada parameter *deltamax*, apabila perubahan bobot melebihi batas maksimum perubahan bobot, maka perubahan bobot akan ditentukan sama dengan maksimum perubahan bobot, Mathworks, 1999, [11].

2.4 Data Percobaan dan Arsitektur JST

Data suara yang digunakan direkam menggunakan fungsi `wavrecord` pada Matlab, dan disimpan menjadi *file* berekstensi WAV dengan fungsi `wavwrite`. Setiap pembicara (ada 10 pembicara) mengucapkan kata "komputer" sebanyak 60 kali sehingga didapat 600 data suara. Setiap suara direkam selama 1 detik tanpa pengarahan (*unguided*) dengan *sampling rate* 16000 Hz dan kemudian dikuantisasi dengan ke dalam representasi 16 bit, sehingga masing-masing menghasilkan ukuran file 31,25 KB.

Untuk mendapatkan data yang memiliki *noise*, data yang telah dikumpulkan sebelumnya disalin sebanyak dua kali kemudian ditambahkan *white gaussian noise* masing-masing dengan SNR 30 dB dan 20 dB. Setelah tahapan ini selesai dilakukan, didapatkan tiga tipe data suara yaitu: data tanpa penambahan *noise*, data dengan SNR 30 dB, dan data dengan SNR 20 dB dengan jumlah 600 data suara untuk tiap tipenya. Selanjutnya data yang telah dikumpulkan tadi dibagi menjadi dua kelompok dengan perbandingan 2:1 untuk tiap pembicara. Kelompok pertama, sebanyak 400 data suara, akan digunakan sebagai data latih dan kelompok kedua, sebanyak 200 data suara digunakan sebagai data uji.

Arsitektur JST Propagasi Balik yang digunakan adalah arsitektur *multilayer perceptron* dengan satu *hidden layer*. Jumlah *neuron input* disesuaikan dengan jumlah koefisien MFCC. Jumlah *neuron hidden* dibagi menjadi tiga puluh perlakuan yakni 10 sampai 300 dengan *increment* 10. Sedangkan jumlah *neuron output* disesuaikan dengan target pembicara. Inisialisasi yang digunakan adalah Nguyen-Widrow dengan alasan laju pembelajaran yang lebih baik, [9]. Struktur JST *Resilient Backpropagation* dapat dilihat pada Tabel 1.

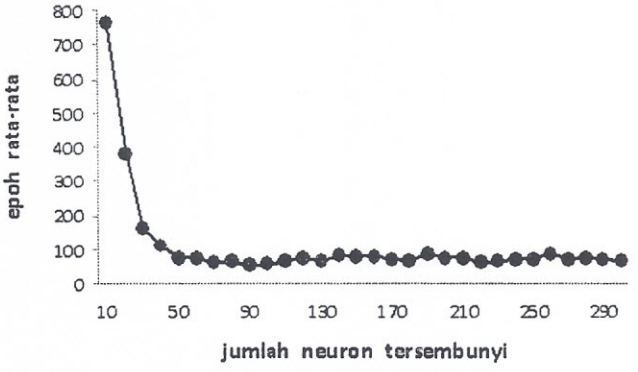
Tabel 1 Struktur JST *Resilient Backpropagation*

Karakteristik	Spesifikasi
Arsitektur	1 *hidden layer*
Jumlah *neuron input*	Dimensi hasil MFCC
Jumlah *neuron hidden*	10 sampai 300 dengan *increment* 10
Jumlah *neuron output*	10 (Definisi *target*)
Inisialisasi bobot	Nguyen-Widrow
Fungsi Pembelajaran	*Resilient Backpropagation*
Fungsi aktivasi	*Log-sigmoid*
Toleransi galat	0.0001

Parameter lainya dipilih nilai default dari Matlab, yaitu *delta0*, *deltamax*, *deltamin*, *delt_inc* dan *delt_dec* berturut-turut adalah 0,1; 50; 0,1; 1,2 dan 0,5.

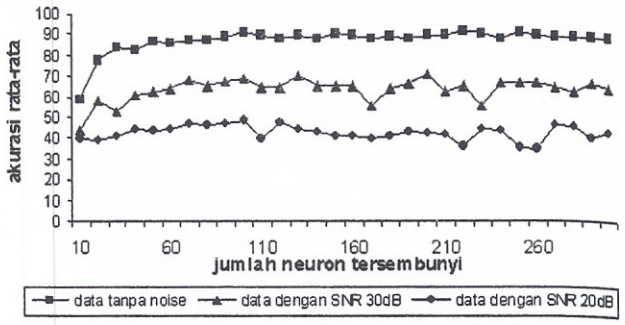
3. Hasil Percobaan

Perbandingan jumlah epoch hingga jaringan optimum antar berbagai jumlah neuron hidden dapat dilihat pada Gambar 4. Terlihat bahwa jumlah epoch hingga tercapainya generalisasi menurun secara drastis untuk jumlah neuron hidden hingga 50. Setelah itu jumlah epoch relatif tetap.



Gambar 4 Grafik perbandingan jumlah epoh rata-rata terhadap jumlah *neuron* tersembunyi pada pelatihan dengan data tanpa *noise*

Gambar 5 menyajian perbandingan akurasi rata-rata dari berbagai jumlah neuron hidden. Dari gambar di atas terlihat bahwa untuk sinyal dengan penambahan noise, nilai akurasi turun secara nyata, mulai drai noise 30 dB dan noise 20 dB, masing-masing dengan akurasi berkisar 60 hingga 70% serta 40 hingga 50%. Hal ini menunjukkan bahwa teknik yang dikembangkan telah gagal melakukan pengenalan dengan baik untuk sinyal bernoise, meskipun hanya 30 dB. Dari gambar tersebut terlihat bahwa akurasi rata-rata maksimum diperoleh untuk jumlah neuron hidden sebanyak 100, dan terjelek pada jumlah neuron hidden 10, dengan akurasi rata-rata untuk sinyal asli sebesar 59%.



Gambar 5. Grafik perbandingan nilai akurasi rata-rata terhadap jumlah *neuron* tersembunyi

Dengan menggunakan neuron hidden sebanyak 100 diperoleh akurasi rata-rata dari 10 pembicara sebesar 96% seperti disajikan pada tabel 2. Terlihat bahwa pembicara yang dapat diidentikasi dengan benar seluruhnya adalah pembicara 1, pembicara 2, pembicara 5, dan pembicara 8. Di samping itu, dapat dilihat juga bahwa pembicara yang paling sedikit diidentifikasi dengan benar adalah pembicara 9. Pada pembicara tersebut, data uji yang dapat diidentifikasi dengan benar hanya tujuh belas data atau 85% sedangkan sisanya dua data uji diidentifikasi sebagai suara pembicara 6 dan satu diidentikasi sebagai suara pembicara 7.

Tabel 2 Hasil identifikasi model JST terbaik dari dua puluh data pembicara tanpa *threshold*

Pembicara	Diidentifikasi Sebagai Pembicara										Persentase
	1	2	3	4	5	6	7	8	9	10	
1	20										100 %
2		20									100 %
3			19		1						95 %
4				19	1						95 %
5					20						100 %
6						19			1		95 %
7							19		1		95 %
8								20			100 %
9						2	1		17		85 %
10								1		19	95 %

Selanjutnya, pada proses identifikasi ditambahkan satu tahapan lagi. Kali ini setelah ditemukan nilai maksimal dari keluaran model JST, dilakukan pembandingan terhadap nilai *threshold* dari pembicara tersebut. Sebuah data suara yang diuji diidentifikasi sebagai suara salah seorang pembicara hanya jika nilai maksimal keluaran dari model JST, yang menyatakan bahwa data tersebut suara dari salah seorang pembicara, lebih besar dari nilai *threshold*. Apabila nilai maksimal yang ditemukan masih lebih kecil dari pada nilai *threshold* maka data suara tersebut tidak dikategorikan sebagai satu pun pembicara. Dengan penambahan tahap *threshold* dalam proses identifikasi, model JST yang dibangun menjadi lebih "hati-hati" dalam mengidentifikasi suatu suara. Hasil identifikasi pembicara untuk dua puluh data pengujian tanpa *noise* dengan menggunakan *threshold* ditampilkan pada Tabel 3. Pada tabel tersebut ditambahkan satu pembicara baru yaitu

pembicara 0. Pembicara ini ditambahkan dengan maksud untuk menampung data suara yang hasil identifikasinya lebih kecil daripada nilai *threshold*.

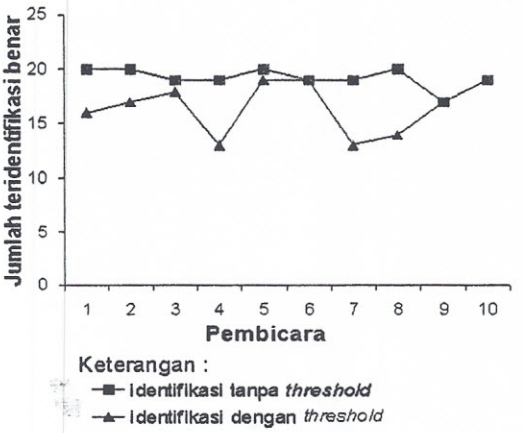
Tabel 3 Hasil identifikasi model JST terbaik dari dua puluh data pembicara dengan *threshold*

Pembicara	Diidentifikasi Sebagai Pembicara											Persentase
	0	1	2	3	4	5	6	7	8	9	10	
1	4	16										80 %
2	3		17									85 %
3	2			18								90 %
4	7				13							65 %
5	1					19						95 %
6	1						19					95 %
7	7							13				65 %
8	6								14			70 %
9	3									17		85 %
10	1										19	95 %

Dari Tabel 3 dapat dilihat bahwa setelah ditambahkan *threshold* tidak ada lagi data suara dari satu pembicara yang teridentifikasi sebagai pembicara lain. Tapi di lain pihak dapat dilihat juga bahwa tidak ada lagi data suara yang seluruhnya diidentifikasi dengan benar. Jumlah data suara yang teridentifikasi dengan benar terbanyak hanya sembilan belas data yaitu data suara dari pembicara 5, pembicara 6, dan pembicara 10. Satu data suara dari masing-masing pembicara tadi dikenali sebagai pembicara 0 yang berarti bahwa nilai keluaran model JST untuk data tersebut lebih kecil dari nilai *threshold*nya.

Jumlah data suara yang teridentifikasi dengan benar terendah terjadi pada pembicara 4 dan pembicara 7, yaitu tiga belas data suara atau hanya 65 % dari seluruh data suara yang diujikan. Jumlah data suara teridentifikasi dengan benar yang rendah juga terjadi pada pembicara 8. Dari dua puluh data yang diujikan, hanya empat belas data yang diidentifikasi dengan benar.

Bila dibandingkan dengan identifikasi tanpa *threshold*, jumlah data suara yang teridentifikasi dengan benar pada identifikasi dengan *threshold* secara umum mengalami penurunan yang cukup drastis. Hal ini dapat dilihat dengan jelas dalam grafik perbandingan jumlah data suara yang teridentifikasi dengan benar pada Gambar 6. Dari grafik terlihat bahwa pada identifikasi tanpa *threshold* jumlah data suara yang dikenali dengan benar secara umum mengalami penurunan dibandingkan dengan identifikasi tanpa *threshold*. Nilai akurasi keseluruhan pun turun menjadi hanya 82.5%. Hal ini disebabkan karena hasil keluaran dari model JST untuk data suara tersebut masih lebih kecil dari nilai *threshold* pembicara yang bersangkutan. Keadaaan tersebut mengakibatkan data suara yang diujikan tadi dianggap bukan merupakan suara dari pembicara yang bersangkutan dan kemudian diklasifikasikan sebagai data suara pembicara 0.



Gambar 6 Grafik perbandingan jumlah data suara yang teridentifikasi dengan benar pada data tanpa *noise*

Penurunan jumlah data suara teridentifikasi dengan benar yang cukup drastis ini kemungkinan disebabkan oleh dua hal. Pertama, data dan model JST yang digunakan masih kurang baik. Model yang masih kurang baik menyebabkan identifikasi kurang baik, yang digambarkan dengan nilai maksimal keluaran dari model yang kurang besar. Nilai maksimal keluaran yang kurang besar ini mengakibatkan data suara yang diujikan diangap bukan suara pembicara yang bersangkutan karena nilainya lebih kecil dari *threshold*. Kemungkinan kedua adalah kurang baiknya nilai *threshold* itu sendiri. Jika nilai *threshold* yang diambil terlalu besar, maka akan banyak data suara yang tidak teridentifikasi karena nilai maksimalnya lebih kecil dari *threshold*.

4. Kesimpulan

Dari penelitian yang telah dilakukan, dapat disimpulkan bahwa model jaringan syaraf tiruan *resilient backpropagation* dapat digunakan untuk identifikasi pembicara pada data yang direkam tanpa pengarahan. Dari tiga puluh model yang dibangun, nilai akurasi rata-rata terbaik didapatkan dari model dengan seratus *neuron* tersembunyi yaitu sebesar 96%. Nilai akurasi rata-rata terendah didapatkan dari model dengan sepuluh *neuron* tersembunyi, yaitu 59%. Untuk sinyal bernois, meskipun hanya 30 dB, sistem gagal melakukan pengenalan dengan baik.

Penambahan nilai threshold untuk pengenalan akan menurunkan akurasi sistem menjadi 83%. Namun demiian meningkatkan keyakinan hasil akurasi. Artinya, bahwa sinyal yang dideteksi sebagai pembicara tertentu, maka kita lebih yakin bahwa pendeteksian tersebut benar. Untuk kasus yang kurang pasti, maka akan terklasifikasi ke kelas 0.

Dari hasil percobaan yang sudah dilakukan, terlihat bahwa sistem yang dikembangkan belum secara optimum beerja dengan baik, khususnya untuk sinyal bernois. Untuk itu ada beberapa hal untuk penelitian selanjutnya, yaitu kajian terhadap teknik ekstraksi ciri yang robust terhadap noise, kajian metode pengenal pola yang optimum dan penentuan nilai threshold yang lebih baik.

REFERENSI

[1] Reynolds, D., 2002, "Automatic Speaker Recognition Acoustics and Beyond : Tutorial note", MIT Lincoln Laboratory, 2002.

[2] Buono, A. and B. Kusumoputro., 2008, "Sistem Identifikasi Pembicara Berbasis Power Spektrum Menggunakan Hidden Markov model", Jurnal Ilmiah Ilmu Komputer, ISSN 1693-1929, edisi Mei 2009, Departemen Ilmu Komputer IPB.

[3] Oktavianto, B., 2004, "Pengenalan Pembicara dengan Jaringan Syaraf Tiruan Propagasi Balik", Skripsi Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Pertanian Bogor.

[4] Saputro, DW., 2006, "Pengenalan Karakter Tulisan Tangan dengan Menggunakan Jaringan Syaraf Tiruan Propagasi Balik Resilient", Skripsi Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Pertanian Bogor.

[5] C. Cornaz, U. Hunkeler, 2005, "An Automatic Speaker Recognition System", Ecole Polytechnique, Federale De Lausanne, http://www.ifp.uiuc.edu/~minhdo/teaching/speaker_recognition.

[6] Campbell, Jr JP., 1997, "Speaker Recognition: A Tutorial", Proceeding IEEE. 85:1437-1461.

[7] Todor D. Ganchev, 2005, *"Speaker Recognition"*, Ph.D. Thesis. Wire Communications Laboratory, Department of Computer and Electrical Engineering, University of Patras Greece.

[8] M. Nilsson dan M. Ejnarsson, Maret 2002, "Speech Recognition using Hidden Markov Model : Performance Evaluation in Noisy Environment", Master Thesis, Departement of Telecommunications and Signal Processing, Blekinge Institute of Technology.

[9] Fausett L., 1994, "Fundamentals of Neural Network", New York: Prentice Hall.

[10] Fu LM., 1994, "Neural Networks in Computer Intelligence", Singapore: Mc Graw-Hill.

[11] Mathworks Inc., 1999, "Neural Network for Use With Matlab", Natick: The Mathworks Inc.

Agus Buono, memperoleh gelar Sarjana dan Master bidang statistik di IPB pada tahun 1992 dan 1996. Gelar Master dan Doktor bidang Ilmu Komputer diperoleh diperoleh dari Universitas Indonesia pada tahun 2000 dan 2009. Saat ini sebagai Staf Pengajar Departemen Ilmu Komputer Institut Pertanian Bogor.

Irman Hermadi, memperoleh gelar Sarjana Ilmu Komputer di Jurusan Ilmu Komputer IPB, Master bidang komputer diperoleh dari Arab Saudi, dan sekarang sebagai staf Departemen Ilmu Komputer Ipb sedang tugas belajar pada program Doktor bidang komputer di Australia.

Nurhadi Susanto, memperoleh gelar Sarjana Ilmu Komputer di Jurusan Ilmu Komputer IPB pada tahun 2006.