PENGENALAN PEMBICARA DENGAN JARINGAN SYARAF TIRUAN *BACKPROPAGATION*

Baskoro Oktianto *, Sugi Guritman †, Ahmad Ridha ‡
* Departemen Ilmu Komputer, Fakultas Matematika dan IPA, Institut Pertanian Bogor
Jl. Pajajaran, Bogor, Indonesia
baszkoro@yahoo.com
† Departemen Ilmu Komputer, Fakultas Matematika dan IPA, Institut Pertanian Bogor
Jl. Pajajaran, Bogor, Indonesia
‡ Departemen Matematika, Fakultas Matematika dan IPA, Institut Pertanian Bogor
Jl. Pajajaran, Bogor, Indonesia

ABSTRAK

Masalah pengenalan pembicara terbagi menjadi dua bagian, yaitu identifikasi pembicara dan verifikasi pembicara. Karena pengenalan pembicara tergolong dalam masalah nonalgorithmic maka digunakan jaringan syaraf tiruan (JST) untuk pencocokan pola. Sebelum diproses dalam JST, data suara terlebih dahulu diproses dengan proses-proses sinyal digital melalui suatu proses feature extraction menggunakan analisis cepstral ditambah dengan proses feature selection menggunakan principal component analysis. Hasil dari JST selanjutnya diolah oleh model pembuatan keputusan. Model pembuatan keputusan dalam sistem identifikasi akan menentukan identitas pembicara dan dalam sistem verifikasi akan menerima atau menolak klaim yang diajukan oleh pembicara. Sistem pengenalan pembicara yang dibangun mampu mengidentifikasi dengan tingkat generalisasi tertinggi sebesar 92,3077% dan melakukan verifikasi dengan nilai equal error rate sebesar 6,5657%.

Kata kunci: Pengenalan pembicara, jaringan syaraf tiruan, analisis cepstral, principal component analysis.

1. PENDAHULUAN

Proses identifikasi atau verifikasi banyak digunakan dalam kehidupan sehari-hari, misalnya dalam penggunaan mesin ATM atau otorisasi seseorang untuk memasuki suatu wilayah tertentu. Proses identifikasi atau verifikasi umumnya dilakukan dengan suatu alat identifikasi seperti kartu ATM atau kartu khusus tertentu. Bila kartu tersebut hilang tentunya akan menjadi masalah bagi pemiliknya.

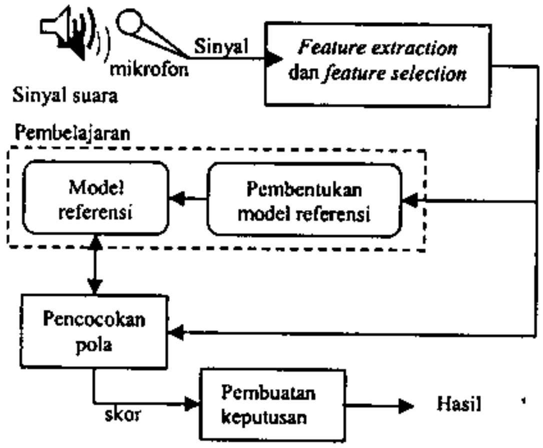
Dengan teknik biometrik proses identifikasi atau verifikasi dapat dilakukan melalui karakteristik fisiologi atau perilaku seseorang [10] dan tidak dibutuhkan alat identifikasi khusus. Beberapa cara untuk melakukan identifikasi atau verifikasi secara biometrik adalah melalui suara, wajah, sidik jari, tanda tangan, retina dan lain-lain. Beberapa hal yang mendorong penggunaan identifikasi atau verifikasi secara biometrik adalah biometrik bersifat universal (terdapat pada setiap orang), unik (tiap orang mempunyai ciri khas tersendiri), dan tidak mudah dipalsukan [10].

Proses identifikasi atau verifikasi dengan suara memiliki keunggulan dibandingkan dengan karakteristik yang lain, yaitu hanya membutuhkan alat tambahan berupa mikrofon dan kartu suara sedangkan karakteristik-karakteristik yang lain misalnya sidik jari atau wajah membutuhkan alat tambahan seperti *scanner*. Hal ini sedikit banyak dapat menekan biaya pengembangan sistem.

Identifikasi atau verifikasi melalui suara termasuk dalam masalah *nonalgorithmic*. Walaupun sirkuit digital (komputer) mempunyai kecepatan yang jauh lebih tinggi daripada otak manusia tetapi dalam memproses masalah-masalah *nonalgorithmic* otak manusia lebih unggul [3]. Suatu teknik yang dibuat dengan memodelkan otak manusia adalah jaringan syaraf tiruan (JST) atau *artificial neural network*. Seperti pada otak manusia, JST terdiri atas neuron-neuron yang saling berhubungan yang dapat bekerja sama satu dengan yang lainnya untuk membentuk suatu sistem. JST dapat belajar untuk mengenali suatu pola melalui pembelajaran dan diharapkan dapat memecahkan masalah-masalah yang bersifat *nonalgorithmic*.

2. SISTEM PENGENALAN PEMBICARA

Pengenalan pembicara terbagi menjadi dua bagian, yaitu identifikasi pembicara (menentukan identitas pembicara) dan verfikasi pembicara (melakukan verfikasi identitas yang diklaim oleh pembicara). Secara umum sistem pengenalan pembicara mempunyai tahapan yang terdiri atas (1) akuisisi data suara digital, (2) *feature extraction* dan *feature selection*, (3) pembentukan model referensi pembicara dan pencocokan pola, dan (4) pembuatan keputusan [1]. Diagram blok tahapan tersebut ditunjukkan dalam Gambar 1.



Gambar 1. Tahapan pengenalan pembicara.

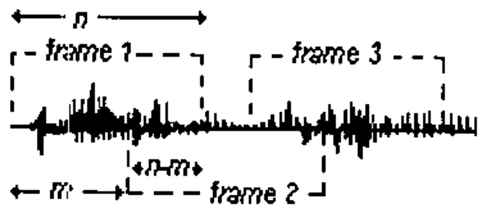
2.1. Akuisisi Data Suara Digital

Suara merupakan gelombang analog yang dapat ditangkap oleh mikrofon. Sinyal analog tersebut dapat diubah menjadi sinyal digital melalui proses *sampling*, yaitu proses untuk memperoleh nilai dari sinyal analog dalam waktu diskret. Proses *sampling* menghasilkan suatu vektor berisi deretan bilangan yang merupakan representasi digital dari sinyal suara atau disebut juga sinyal suara digital. Hal yang perlu diperhatikan dalam melakukan *sampling* adalah frekuensi *sampling (fs)*, yaitu jumlah *sample* dalam 1 detik. Semakin besar *fs* maka semakin besar ukuran data yang diperoleh dengan kualitas suara yang semakin baik, sedangkan semakin kecil *fs* maka ukuran data yang diperoleh akan semakin kecil dengan konsekuensi penurunan kualitas suara. Umumnya *fs* yang digunakan berkisar pada rentang 6-20 kHz [6].

2.2. *Feature Extraction* dan *Feature Selection*

Tujuan dari *feature extraction* adalah untuk mengubah sinyal suara digital menjadi suatu representasi data yang berdimensi lebih kecil untuk diproses lebih lanjut. Manfaat yang diperoleh dari *feature extraction* adalah memudahkan dan mempercepat proses-proses selanjutnya. Hal ini dapat dilakukan karena *feature extraction* dapat mengekstrak informasi yang terdapat dalam sinyal suara digital. Sebelum dilakukan *feature extraction* terlebih dahulu dilakukan langkah-langkah yang terdiri atas (1) *frame blocking*, dan (2) *frame windowing* [10].

1. *Frame blocking*. Dalam analisis sinyal digital terdapat suatu konsep yang dinamakan *short-time analysis* [6]. Asumsi yang digunakan adalah dalam interval waktu yang panjang, pola gelombang suara tidak stasioner, tetapi dalam waktu yang cukup pendek (10-30 milidetik) dapat dikatakan stasioner. Hal ini dikarenakan kecepatan perubahan spektrum suara berkaitan dengan kecepatan perubahan organ-organ penghasil suara pada manusia dan hal ini dibatasi oleh keterbatasan fisiolologi. Berdasarkan pada hal di atas, sinyal suara digital yang telah diakuisisi dapat dibagi-bagi menjadi segmen-segmen dengan durasi 10-30 milidetik yang disebut dengan *frame*. Proses pembentukan *frame-frame* disebut dengan *frame blocking* dan tiap *frame* direpresentasikan dalam sebuah vektor. Dalam pembentukan *frame* umumnya terdapat *overlap* antara *frame-frame* yang bersebelahan. Jika panjang *frame* adalah n, maka pada tiap-tiap *frame* akan terdapat *overlap* sebesar $n - m$ dengan $m < n$. Contoh ilustrasi *frame blocking* dalam bentuk grafik dapat dilihat pada Gambar 2.



Gambar 2. *Frame blocking* pada sinyal suara.

2. *Frame windowing*. Proses *frame blocking* menyebabkan terjadinya *spectral leakage*, yaitu distorsi frekuensi pada bagian tepi *frame* yang dipengaruhi oleh *frame* di sebelahnya. *Frame windowing* bertujuan untuk meminimalkan diskontinuitas sinyal atau *spectral leakage* pada bagian awal dan akhir pada tiap *frame* [10]. Metode untuk melakukan *frame windowing* adalah dengan memboboti (mengalikan) tiap *frame* dengan suatu *window*. *Window* yang biasa digunakan contohnya adalah *hamming window*. *Hamming window* didefinisikan dalam persamaan berikut:

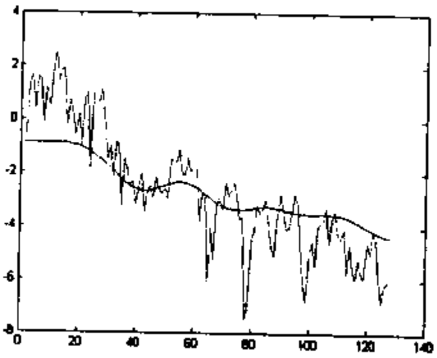
$$w[k] = 0{,}54 - 0{,}64\cos\left(\frac{2\pi k}{N-1}\right),\ 0 \le k \le N-1 \qquad (1)$$

dengan N adalah panjang *window*. *Hamming window* adalah sebuah vektor dengan jumlah elemen sebanyak N. Besarnya N akan disesuaikan dengan banyaknya elemen pada *frame* yang akan diboboti sehingga banyaknya elemen pada *hamming window* akan sama dengan banyaknya elemen pada *frame* yang akan diboboti.

Setelah *frame blocking* dan *frame windowing* selesai dilakukan maka pada tiap-tiap *frame* dilakukan *feature extraction* dengan analisis *cepstral*. Jika sebuah *frame* diberikan oleh vektor **h** maka nilai dari vektor *cepstral* diberikan oleh persamaan berikut:

$$\text{vektor } cepstral = \text{real(ifft(log(abs(fft}(\mathbf{h}))))) \qquad (2)$$

dengan fft adalah transformasi fourier dan ifft adalah invers dari transformasi fourier. Dari vektor *cepstral* dapat diambil hanya 12 koefisien (elemen) pertamanya saja dan yang lainnya dapat dibuat menjadi 0 (dapat diabaikan). Dengan 12 koefisien *cepstral*, spektrum dari sinyal dapat direkonstruksi dan akan menjadi lebih halus [8]. Dalam grafik pada Gambar 3 terdapat dua buah garis dengan garis yang lebih bergelombang adalah sinyal asli, sedangkan garis yang lebih halus adalah hasil dari analisis *cepstral* dengan 12 koefisien pertama.



Gambar 3. Grafik analisis *cepstral* dibandingkan dengan sinyal asli.

Analisis *cepstral* akan menghasilkan sebuah matriks yang tiap kolomnya adalah vektor *cepstral* dari tiap *frame* yang hanya diambil 12 koefisien pertamanya saja.

Data yang telah diproses dengan *feature extraction* selanjutnya akan diolah dengan suatu metode *feature selection*. *Feature selection* bertujuan untuk mengubah dari ruang data ke ruang *feature* yang berdimensi kecil dengan tetap mempertahankan informasi yang penting untuk digunakan dalam aplikasi dan hasil *feature selection* dapat diperbandingkan berdasarkan kemiripan data [10].

Salah satu teknik yang dapat digunakan sebagai *feature selection* adalah analisis komponen utama atau *principal component analysis* (PCA). Hal ini berguna untuk mempersingkat waktu yang diperlukan baik pada saat pembelajaran sistem maupun pada saat digunakan.

Hasil *feature extraction* dari sebuah sinyal suara digital adalah sebuah matriks yang tiap kolomnya adalah koefisien-koefisien *cepstral* masing-masing *frame*. Matriks tersebut akan diubah menjadi sebuah vektor. Jika diberikan matriks hasil *feature extraction* **C** dan v adalah vektor yang merepresentasikan *frame* dengan *b* adalah banyaknya *frame*, maka matriks **C** dapat dibentuk menjadi sebuah vektor sebagai berikut:

$$\mathbf{v}^T = [v_1 \ v_2 \ \cdots \ v_{12}]$$

$$\mathbf{C}^T = [v_{11} \ \cdots \ v_{112} \ v_{21} \ \cdots \ v_{212} \ \cdots \ v_{b1} \ \cdots \ v_{b12}]$$

PCA melakukan transformasi terhadap **C** melalui sebuah matriks transformasi **P** dan menghasilkan matriks hasil transformasi **Y** [9] atau dalam representasi notasi akan tampak sebagai berikut:

$$\mathbf{Y} = \mathbf{PC} \quad (3)$$

Jumlah elemen dalam **Y** dapat disesuaikan sehingga lebih kecil dari **C**, sehingga dapat dilakukan reduksi dimensi. **Y** merupakan kombinasi linier dengan vektor-vektor basis dalam matriks **P**. Hal yang penting dalam PCA adalah pembentukan matriks transformasi **P**. Pembentukan **P** hanya dilakukan satu kali sebelum pembelajaran dilakukan. **P** akan dibentuk dari sejumlah sinyal suara digital yang telah diproses dengan *feature extraction*. Selain untuk pembentukan **P** sinyal-sinyal suara digital ini juga akan digunakan untuk pembelajaran. Jika matriks transformasi telah terbentuk maka vektor hasil *feature extraction* dapat langsung dikalikan dengan matriks transformasi sehingga diperoleh vektor baru.

2.3. Pembentukan Model Referensi Pembicara dan Pencocokan Pola

Pembentukan model referensi pembicara akan membentuk suatu model referensi yang akan digunakan untuk pencocokan pola. Salah satu teknik yang dapat digunakan dalam pencocokan pola adalah JST. JST akan melakukan pembelajaran untuk membentuk suatu model referensi, kemudian JST yang telah melakukan pembelajaran tersebut dapat digunakan untuk pencocokan pola.

Sebuah jaringan syaraf tiruan adalah sebuah sistem pemrosesan informasi yang mempunyai karakteristik serupa dengan jaringan syaraf biologi [2]. Sebuah JST direpresentasikan oleh sebuah set *node-node* dan panah-panah penghubung. Sebuah *node* mewakili sebuah neuron dan sebuah panah mewakili hubungan antarneuron dengan arah panah menunjukkan aliran sinyal.

Setiap *node* menerima sebuah set *input* yang akan dikalikan dengan *weight* (bobot) yang dianalogikan sebagai kuat lemahnya *synapsis* dalam sel biologi. Jumlah total dari seluruh *input* yang telah dikalikan bobot akan menentukan level pengaktifan *node* tersebut. Dalam representasi notasi setiap *input* X_i dikalikan bobot W_i sehingga total *input*-nya akan seperti ekspresi berikut:

$$\sum_i X_i W_i$$

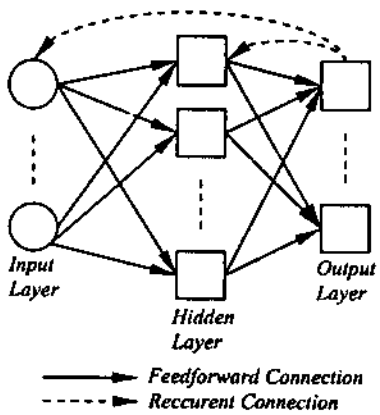
Total *input* tersebut kemudian diproses oleh suatu fungsi pengaktifan dan akan menghasilkan suatu *output*.

Salah satu model JST yang dapat digunakan untuk pencocokan pola adalah JST *backpropagation*. JST *backpropagation* dikembangkan oleh Rumelhart, Hinton dan Williams yang dipopulerkan dalam buku *Parallel Distributed Processing*. JST *backpropagation* menggunakan arsitektur *multi-layer perceptron* dan pembelajaran *backpropagation*. Walaupun JST *backpropagation* membutuhkan waktu yang relatif lama untuk pembelajaran tetapi bila pembelajaran telah selesai dilakukan, JST akan dapat mengenali suatu pola dengan cepat.

Beberapa karakteristik dari JST *backpropagation* adalah sebagai berikut:

- Jaringan *multi-layer*. JST *backpropagation* mempunyai lapisan *input*, lapisan tersembunyi dan lapisan *output* (Gambar 4) dan setiap neuron pada satu lapisan menerima *input* dari semua neuron pada lapisan sebelumnya.
- Fungsi pengaktifan. Fungsi pengaktifan akan menghitung *input* yang diterima oleh suatu neuron, kemudian neuron tersebut meneruskan hasil dari fungsi pengaktifan ke neuron berikutnya, sehingga fungsi pengaktifan berfungsi sebagai penentu kuat lemahnya sinyal yang dikeluarkan oleh suatu neuron. Fungsi yang sering digunakan sebagai fungsi pengaktifan adalah fungsi sigmoid biner dengan fungsi sebagai berikut:

$$f(x) = \frac{1}{1+\exp(-x)} \tag{4}$$



Gambar 4. Model JST *backpropagation*.

Algoritme pembelajaran JST *backpropagation* bersifat iteratif dan didesain untuk meminimalkan *mean square error* (MSE) antara *output* yang dihasilkan dengan *output* yang diinginkan. Langkah-langkah algoritme pembelajaran JST *backpropagation* yang diformulasikan oleh Rumelhart, Hinton, dan Williams secara singkat adalah sebagai berikut:

- Inisialisasi bobot. Inisialisasi dapat dilakukan secara acak atau melalui metode Nguyen-Widrow.
- Perhitungan nilai pengaktifan. Tiap neuron menghitung nilai pengaktifan dari *input* yang diterimanya. Pada lapisan *input* nilai pengaktifan adalah fungsi identitas. Pada lapisan tersembunyi dan *output* nilai pengaktifan dihitung melalui fungsi pengaktifan.
- Penyesuaian bobot. Penyesuaian bobot dipengaruhi oleh besarnya nilai kesalahan (*error*) antara target *output* dan nilai *output* jaringan saat ini.
- Iterasi akan terus dilakukan sampai kriteria *error* tertentu dipenuhi.

JST *backpropagation* dikenal sebagai JST yang dapat memberikan respon yang cukup baik untuk pola-pola yang serupa tetapi tidak identik dengan pola pembelajaran [2]. Pengujian JST untuk pengenalan pola dapat dilakukan dengan generalisasi, yaitu jumlah (dalam %) pola yang berhasil diklasifikasi dengan benar oleh JST. Generalisasi diberikan oleh persamaan berikut [4]:

$$\text{Generalisasi} = \frac{\text{Jumlah pola yang dikenali}}{\text{Jumlah seluruh pola}} \times 100\% \tag{5}$$

2.4. Pembuatan Keputusan

Sistem pengenalan pembicara mempunyai dua buah model pembuatan keputusan, yaitu untuk sistem identifikasi dan untuk sistem verifikasi. Proses pembuatan keputusan terkait erat dengan teknik pencocokan pola yang digunakan. Pembuatan keputusan identifikasi dapat dianalogikan sebagai masalah klasifikasi pola dengan tiap kelas merepresentasikan tiap pembicara. Pada masalah pengenalan pola, JST akan memberikan skor bagi pola yang masuk untuk semua kelas yang ada. Metode nilai maksimum melakukan pembuatan keputusan dengan melihat kelas yang mempunyai skor maksimum [7], dan pola yang masuk akan diklasifikasi ke dalam kelas (pembicara) tersebut.

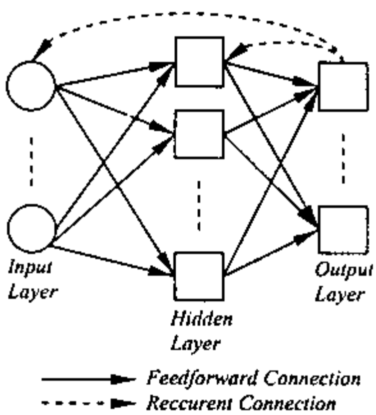
Pembuatan keputusan untuk verifikasi pada pengenalan pembicara akan menentukan diterima atau tidaknya data suara yang masuk. Salah satu metode yang dapat digunakan untuk melakukan verifikasi adalah metode *threshold* [5]. Pembuatan keputusan dilakukan melalui perbandingan skor hasil pencocokan pola dengan besaran *threshold*. Jika skor lebih besar atau sama dengan *threshold* maka verifikasi diterima dan jika lebih kecil maka verifikasi ditolak. *Threshold* yang digunakan dapat berlaku untuk seluruh pembicara atau dapat juga tiap pembicara memiliki *threshold* yang berbeda-beda. *Threshold* yang berbeda-beda untuk tiap pembicara menawarkan fleksibilitas karena besarnya nilai *threshold* dapat diatur sesuai kebutuhan dan perubahan *threshold* pada satu pembicara tidak akan mempengaruhi *threshold* pembicara lainnya. Salah satu metode penentuan *threshold* untuk sistem verifikasi dan telah disesuaikan untuk JST *backpropagation* adalah sebagai berikut [5]:

- Bentuk sebuah matriks **O** menggunakan sistem identifikasi yang tiap kolomnya berisi hasil *output* JST untuk tiap data pengujian. Data yang tidak berhasil diidentifikasi dengan benar tidak dimasukkan ke dalam **O**. Karena data dalam matriks **O** dapat diidentifikasi dengan benar, nilai maksimum dari tiap kolom merepresentasikan identitas pembicara.
- Untuk tiap set *sample* pembicara bentuk sebuah vektor **m** yang berisi nilai maksimum dari tiap kolom pada

Beberapa karakteristik dari JST *backpropagation* adalah sebagai berikut:

- Jaringan *multi-layer*. JST *backpropagation* mempunyai lapisan *input*, lapisan tersembunyi dan lapisan *output* (Gambar 4) dan setiap neuron pada satu lapisan menerima *input* dari semua neuron pada lapisan sebelumnya.
- Fungsi pengaktifan. Fungsi pengaktifan akan menghitung *input* yang diterima oleh suatu neuron, kemudian neuron tersebut meneruskan hasil dari fungsi pengaktifan ke neuron berikutnya, sehingga fungsi pengaktifan berfungsi sebagai penentu kuat lemahnya sinyal yang dikeluarkan oleh suatu neuron. Fungsi yang sering digunakan sebagai fungsi pengaktifan adalah fungsi sigmoid biner dengan fungsi sebagai berikut:

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



Gambar 4. Model JST *backpropagation*.

Algoritme pembelajaran JST *backpropagation* bersifat iteratif dan didesain untuk meminimalkan *mean square error* (MSE) antara *output* yang dihasilkan dengan *output* yang diinginkan. Langkah-langkah algoritme pembelajaran JST *backpropagation* yang diformulasikan oleh Rumelhart, Hinton, dan Williams secara singkat adalah sebagai berikut:

- Inisialisasi bobot. Inisialisasi dapat dilakukan secara acak atau melalui metode Nguyen-Widrow.
- Perhitungan nilai pengaktifan. Tiap neuron menghitung nilai pengaktifan dari *input* yang diterimanya. Pada lapisan *input* nilai pengaktifan adalah fungsi identitas. Pada lapisan tersembunyi dan *output* nilai pengaktifan dihitung melalui fungsi pengaktifan.
- Penyesuaian bobot. Penyesuaian bobot dipengaruhi oleh besarnya nilai kesalahan (*error*) antara target *output* dan nilai *output* jaringan saat ini.
- Iterasi akan terus dilakukan sampai kriteria *error* tertentu dipenuhi.

JST *backpropagation* dikenal sebagai JST yang dapat memberikan respon yang cukup baik untuk pola-pola yang serupa tetapi tidak identik dengan pola pembelajaran [2]. Pengujian JST untuk pengenalan pola dapat dilakukan dengan generalisasi, yaitu jumlah (dalam %) pola yang berhasil diklasifikasi dengan benar oleh JST. Generalisasi diberikan oleh persamaan berikut [4]:

$$\text{Generalisasi} = \frac{\text{Jumlah pola yang dikenali}}{\text{Jumlah seluruh pola}} \times 100\% \qquad (5)$$

2.4. Pembuatan Keputusan

Sistem pengenalan pembicara mempunyai dua buah model pembuatan keputusan, yaitu untuk sistem identifikasi dan untuk sistem verifikasi. Proses pembuatan keputusan terkait erat dengan teknik pencocokan pola yang digunakan. Pembuatan keputusan identifikasi dapat dianalogikan sebagai masalah klasifikasi pola dengan tiap kelas merepresentasikan tiap pembicara. Pada masalah pengenalan pola, JST akan memberikan skor bagi pola yang masuk untuk semua kelas yang ada. Metode nilai maksimum melakukan pembuatan keputusan dengan melihat kelas yang mempunyai skor maksimum [7], dan pola yang masuk akan diklasifikasi ke dalam kelas (pembicara) tersebut.

Pembuatan keputusan untuk verifikasi pada pengenalan pembicara akan menentukan diterima atau tidaknya data suara yang masuk. Salah satu metode yang dapat digunakan untuk melakukan verifikasi adalah metode *threshold* [5]. Pembuatan keputusan dilakukan melalui perbandingan skor hasil pencocokan pola dengan besaran *threshold*. Jika skor lebih besar atau sama dengan *threshold* maka verifikasi diterima dan jika lebih kecil maka verifikasi ditolak. *Threshold* yang digunakan dapat berlaku untuk seluruh pembicara atau dapat juga tiap pembicara memiliki *threshold* yang berbeda-beda. *Threshold* yang berbeda-beda untuk tiap pembicara menawarkan fleksibilitas karena besarnya nilai *threshold* dapat diatur sesuai kebutuhan dan perubahan *threshold* pada satu pembicara tidak akan mempengaruhi *threshold* pembicara lainnya. Salah satu metode penentuan *threshold* untuk sistem verifikasi dan telah disesuaikan untuk JST *backpropagation* adalah sebagai berikut [5]:

- Bentuk sebuah matriks **O** menggunakan sistem identifikasi yang tiap kolomnya berisi hasil *output* JST untuk tiap data pengujian. Data yang tidak berhasil diidentifikasi dengan benar tidak dimasukkan ke dalam **O**. Karena data dalam matriks **O** dapat diidentifikasi dengan benar, nilai maksimum dari tiap kolom merepresentasikan identitas pembicara.
- Untuk tiap set *sample* pembicara bentuk sebuah vektor **m** yang berisi nilai maksimum dari tiap kolom pada

matriks **O**. Misal dalam **O** terdapat 5 *sample* pembicara 1, maka vektor **m** untuk pembicara 1 akan mempunyai elemen sebanyak 5 elemen. Vektor **m** akan berjumlah sama dengan jumlah pembicara.

- Setelah itu elemen-elemen vektor **m** diurutkan. Untuk tiap pembicara dipilih satu nilai dari vektor **m** yang bersesuaian sebagai *threshold* bagi pembicara tersebut.

3. HASIL EKSPERIMEN

Pengujian dilakukan dengan 208 data yang terdiri atas 13 orang pembicara dan 2 orang *impostor* untuk pengujian verifikasi. Sebanyak 13 orang pembicara masing-masing mengucapkan kata-kata dengan 10 kali pengulangan dan 5 data akan digunakan untuk pembentukan model referensi sedangkan sisanya digunakan untuk pengujian identifikasi, penentuan *threshold* verifikasi, dan pengujian verifikasi. Impostor sebanyak 2 orang masing-masing mengucapkan 3 pengulangan untuk seluruh 13 pembicara.

Dalam Tabel 1 disajikan data untuk proses sinyal digital dan Tabel 2 untuk data JST yang digunakan.

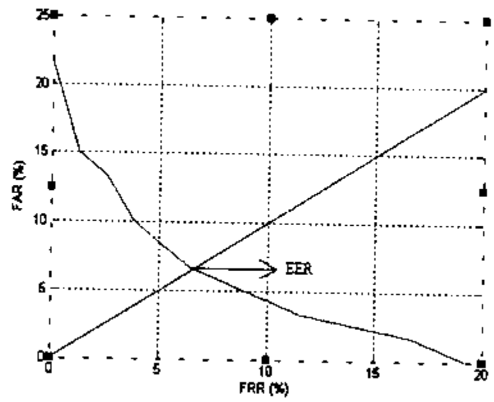
Tabel 1. Data proses sinyal digital

Karakteristik	Spesifikasi
Frekuensi *sampling*	11.025 Hz
Durasi perekaman	3 detik
Panjang *frame* (*n*)	256 *sample*
Overlap (*n* - *m*)	(256 - 100) *sample*
Frame windowing	*Hamming window*
Koefisien *cepstral*	12 koefisien

Tabel 2. Data JST

Karakteristik	Spesifikasi
Arsitektur	1 lapisan tersembunyi
Neuron *input*	Hasil *feature extraction & feature selection*
Neuron tersembunyi	Dari pengujian (dimulai dari 10)
Neuron *output*	Jumlah pembicara
Inisialisasi bobot	Nguyen-Widrow
Fungsi pengaktifan	Sigmoid biner
Toleransi galat	0,01; 0,001 dan 0,0001
Laju pembelajaran	0,1; 0,2 dan 0,3
Epoch maksimum	5.000 *epoch*
Sample pembelajaran tiap pembicara	5 *sample*
Sample pengujian tiap pembicara	5 *sample*

Untuk identifikasi sistem yang dikembangkan mampu mengidentifikasi dengan benar dengan tingkat generalisasi 92,3077%. Untuk sistem verifikasi pengujian dilakukan dengan memperhatikan nilai *equal error rate* (EER), yaitu suatu nilai sedemikian sehingga jumlah kesalahan penerimaan atau *false acceptance rate* (FAR) hampir sama dengan kesalahan penolakan atau *false rejection rate* (FRR). Sistem yang memberikan EER yang semakin kecil akan semakin baik dan dari sistem didapat nilai EER sebesar 6,5657%. Dalam sistem verifikasi terjadi *tradeoff* antara FAR dan FRR dan *tradeoff* dari sistem yang dikembangkan digambarkan dalam grafik *detection error tradeoff* pada Gambar 5.



Gambar 5. Grafik *detection error tradeoff* (DET).

4. KESIMPULAN

JST *backpropagation* dapat melakukan pembelajaran dan pengenalan terhadap suatu pola dengan tingkat generalisasi yang cukup tinggi. Sistem identifikasi menghasilkan tingkat generalisasi sebesar 92,3077%. Sistem verifikasi menghasilkan nilai EER sebesar 6,5657% yaitu nilai yang memberikan keseimbangan antara FAR dan FRR (FAR ≈FRR).

Proses *feature extraction* dan *feature selection* mengekstrak data suara yang berdimensi besar menjadi data baru yang berdimensi kecil yang merepresentasikan data asli. Proses-proses sinyal digital seperti *frame blocking*, *frame windowing* dan analisis *cepstral* dapat mengekstrak pola suara yang dihasilkan. Hasil ekstraksi berupa vektor *feature* dapat diolah lebih lanjut dengan teknik *feature selection* seperti PCA. Dengan PCA dapat dibentuk data yang tereduksi tanpa harus kehilangan banyak informasi.

Kombinasi teknik-teknik tertentu dibutuhkan dalam membangun suatu sistem pengenalan pembicara. Dalam hal ini proses-proses sinyal digital (*frame blocking*, *frame windowing* dan analisis *cepstral*), teknik *feature selection* dengan PCA dan *pattern matching* dengan JST *backpropagation* dapat melakukan hal tersebut.

Penelitian ini masih dapat dikembangkan lebih jauh dan lebih dalam lagi yang nantinya diharapkan dapat terbentuk suatu sistem yang lebih baik. Saran–saran bagi penelitian ini lebih lanjut antara lain:

- Penggunaan teknik pemrosesan sinyal digital yang lain seperti *vector quantization* dan *linear predictive analysis* untuk kemudian diperbandingkan hasilnya sehingga dapat ditentukan teknik yang paling optimal.
- Penggunaan teknik *feature selection* selain PCA, seperti *linear discriminant analysis* dan *independent component analysis* untuk kemudian juga diperbandingkan sehingga diperoleh teknik yang paling optimal.
- Penggunaan JST yang bersifat *incremental learning* sehingga JST dapat mengenali pola baru dengan lebih cepat.
- Penggunaan teknik *pattern matching* selain JST atau mengkombinasikannya dengan JST bila memungkinkan (membentuk suatu teknik hibrida yang diharapkan memberikan hasil yang lebih baik).
- Penggunaan teknik *filtering*, *noise reduction*, dan *end point detection* sehingga sinyal suara digital yang dihasilkan akan lebih baik dari segi kualitas maupun dalam jumlah besarnya data.
- Melakukan penambahan jumlah pembicara untuk melihat kinerja sistem dengan jumlah data yang besar.
- Tersedianya fasilitas yang memungkinkan penambahan pembicara secara otomatis.
- Penggunaan alat-alat audio (mikrofon dan kartu suara) yang lebih baik sehingga data audio yang diperoleh akan lebih baik kualitasnya.

REFERENSI

[1] **Campbell, J.P, JR.** 1997. Speaker Recognition: A Tutorial. Proc. IEEE, vol. 85, no. 9, pp. 1437-1462, 1997.

[2] **Fausett, L.** 1994. Fundamentals of Neural Network. Prentice Hall, Englewood Cliffs, NJ.

[3] **Fu, L.** 1994. Neural Network in Computer Intelligence. McGraw-Hill, Singapore.

[4] **Herryadie, F.D.** 1999. Penggunaan Analisis Komponen Utama dan Jaringan Syaraf Propagasi Balik untuk Pengenalan Wajah. Skripsi. Jurusan Ilmu Komputer, IPB.

[5] **Ho, C.E.** 1998. Speaker Recognition System. Project Report. California Institute of Technology.

[6] **Owens, F.J.** 1993. *Signal* Processing of Speech. Macmillan, London.

[7] **Riadi.** 2001. Jaringan Syaraf Tiruan untuk Pengenalan Tanda Tangan. Skripsi. Jurusan Ilmu Komputer, IPB.

[8] **Roweis, S.** 1998. Speech Processing Background. http://www.dna.caltech.edu/courses/cns187/references/roweis_spblet.ps. [16 Maret 2004].

[9] **Shlens, J.** 2003. A Tutorial on Principal Component Analysis. http://www.snl.salk.edu/~shlens/pub/notes/-pca.pdf. [16 Maret 2004].

[10] **Xafopoulos, A.** 2001. Speaker Verification (an overview). TUT – TICSP presentation. TICSP (Tampere International Center for Signal Processing), TUT (Tampere Univ. of Technology), Tampere, Finland.