



Perancangan Perangkat Lunak Untuk Verifikasi Telinga Seseorang Menggunakan Metode *Back Propagation Neural Network*

Ratnadewi dan Sionny Putra Gunawan

Jurusan Teknik Elektro, Universitas Kristen Maranatha, Bandung
 Jl. Suria Sumantri 65, Bandung 40164, Indonesia
 ratnadewi@engineer.com, sionnygunawan@yahoo.com

Abstrak: Jaringan Saraf Tiruan merupakan salah satu cabang ilmu dari bidang Kecerdasan Buatan, jaringan saraf tiruan dapat digunakan untuk memecahkan masalah – masalah di bidang yang melibatkan pengelompokan, pengenalan pola, dan peramalan. Dalam makalah ini, dirancang sebuah perangkat lunak untuk verifikasi telinga seseorang menggunakan algoritma Jaringan Propagasi Balik. Citra akan mengalami proses pengolahan citra digital yang meliputi proses pendeteksian tepi dan proses *thinning*. Diperlukan pengalihan bentuk citra menjadi bentuk yang dapat digunakan pada jaringan propagasi balik dengan membentuk token pada citra. Token ini selanjutnya akan menjadi dasar perhitungan jaringan propagasi balik. Berdasarkan hasil pengujian proses mengenali pola pada citra menggunakan metode ini, diperoleh nilai FAR sebesar 0,122 % dan FRR sebesar 8,722 %.

Kata kunci: *back propagation neural network*, verifikasi telinga, *token*

Abstract: *Artificial Neural Network is one of the branches of knowledge from the fields of Artificial Intelligence, artificial neural networks can be used to solve problems in the areas that involve the grouping, pattern recognition, and forecasting. In this paper, the software designed for ear verification using the algorithm of Back propagation Neural Network. The image will undergo a process of digital image processing including edge detection process and thinning process. It is needed the transfer of image into a form that can be used in the back propagation neural network by establishing a token of an image. This token will then be the fundamental of back propagation neural network. Based on the test results to recognize pattern in image using this method, the value of FAR is 0.122% and FRR is 8.722%.*

Keywords: *back propagation neural network, ear verification, token*

I. PENDAHULUAN

Salah satu cabang ilmu dari bidang Kecerdasan Buatan (*Artificial Intelligence*) ialah Jaringan Saraf Tiruan (*Artificial Neural Networks*), jaringan saraf tiruan dapat digunakan untuk memecahkan masalah – masalah di bidang yang melibatkan pengelompokan, pengenalan pola,

dan peramalan. Terdapat berbagai macam algoritma pembelajaran atau pelatihan jaringan saraf tiruan yang digunakan untuk memecahkan masalah – masalah tersebut, salah satunya yaitu algoritma jaringan propagasi balik (*Backpropagation Neural Network*).

Biometrik menggunakan telinga telah dilakukan oleh Burge dan Burger^[1]. Struktur telinga tidak berubah secara radikal terhadap waktu. Laporan literatur medik dari Iannarelli^[2] menjelaskan bahwa pertumbuhan telinga manusia proporsional setelah empat bulan sampai delapan tahun. Akibat gravitasi, walaupun pertumbuhan telinga proporsional telinga akan tertarik dalam arah vertikal. Studi terhadap fitur telinga yang hampir sama secara fisik, menemukan bahwa fitur telinga adalah unik, bahkan untuk kembar identik sekalipun.

Dalam makalah ini, dirancang sebuah perangkat lunak untuk verifikasi telinga seseorang. Citra akan mengalami proses pengolahan citra digital yang meliputi proses pendeteksian tepi dan proses *thinning*. Setelah citra mengalami proses pengolahan citra digital, diperlukan pengalihan bentuk citra menjadi bentuk yang dapat digunakan pada jaringan propagasi balik dengan membentuk token pada citra. Token dibentuk atas dasar perhitungan matematis sudut cosinus dan sinus yang merepresentasikan kriteria dari pengenalan pola dan menjadi dasar perhitungan jaringan propagasi balik. pendeteksian tepi dan proses *thinning*.

Diharapkan setelah melalui proses pelatihan, jaringan propagasi balik mampu mengenali secara sempurna pola yang digunakan selama pelatihan (citra latih) serta mampu memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan (citra uji).

Berdasar latar belakang di atas, masalah utama yang akan diangkat pada makalah ini adalah verifikasi telinga seseorang dengan dasar perhitungan jumlah token menggunakan metode *backpropagation neural network*. Tujuan yang hendak dicapai dalam pengerjaan makalah ini adalah mengimplementasikan perangkat lunak untuk verifikasi telinga seseorang menggunakan metode *backpropagation neural network*.

II. TEORI PENUNJANG

II.1. Citra^{[3],[4]}

Definisi citra menurut kamus Webster adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda. Sebuah citra mengandung informasi tentang objek yang direpresentasikan. Citra digital adalah citra yang didefinisikan sebagai fungsi $f(x,y)$. x menyatakan nomor baris, y menyatakan nomor kolom, dan f menyatakan nilai derajat keabuan dari citra. Dengan demikian (x,y) adalah posisi dari piksel dan f adalah nilai derajat keabuan pada titik (x,y) .

Semakin halus citra digital yang dihasilkan artinya resolusi citra semakin tinggi, komponen yang menentukan tingkat resolusi sebuah citra adalah piksel. Dalam makalah ini digunakan citra berukuran 640 X 480

II.2. Metode *Backpropagation*^{[5]-[7]}

Algoritma *Backpropagation* atau propagasi balik pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai pada JST. Propagasi Balik merupakan suatu teknik pembelajaran *supervised learning*, metode ini merupakan metode yang sangat baik dalam menangani masalah pengenalan pola – pola kompleks dan juga banyak

dipakai pada aplikasi pengaturan karena proses pelatihannya didasarkan pada hubungan yang sederhana, yaitu jika keluaran memberikan hasil yang salah, maka bobot (*weight*) dikoreksi supaya nilai kesalahan dapat diperkecil dan respon jaringan selanjutnya diharapkan akan lebih mendekati harga yang benar.

Dalam Jaringan Saraf Tiruan, fungsi aktivasi digunakan untuk menentukan keluaran suatu neuron. Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya).

Jika net $\sum = x_i w_i$ (1)

maka fungsi aktivasinya adalah :

$$f(\text{net}) = f(x_i w_i) \quad (2)$$

Fungsi sigmoid dirumuskan sebagai berikut:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

Fungsi ini sering digunakan karena nilai fungsinya yang sangat mudah untuk didiferensiasikan.

$$f'(x) = f(x)[1 - f(x)] \quad (4)$$

Pada jaringan diberikan sekumpulan contoh pelatihan yang disebut set pelatihan. Set pelatihan ini digambarkan dengan sebuah *feature vector* yang disebut dengan vektor masukan yang diasosiasikan dengan sebuah keluaran yang menjadi target pelatihannya. Dengan kata lain set pelatihan terdiri dari vektor masukan dan juga vektor keluaran target. Keluaran dari jaringan berupa sebuah vektor keluaran aktual. Selanjutnya dilakukan perbandingan antara keluaran aktual yang dihasilkan dengan keluaran target dengan cara melakukan pengurangan diantara kedua keluaran tersebut. Hasil dari pengurangan merupakan *error*. *Error* dijadikan sebagai dasar dalam melakukan perubahan dari setiap bobot yang ada dengan mempropagasikannya kembali.

Setiap perubahan bobot yang terjadi dapat mengurangi *error*. Siklus setiap perubahan bobot (*epoch*) dilakukan pada setiap set pelatihan hingga kondisi berhenti dicapai, yaitu bila mencapai jumlah epoch yang diinginkan atau hingga sebuah nilai ambang yang ditetapkan terlampaui. Algoritma pelatihan jaringan propagasi balik terdiri dari 3 tahap yaitu :

1. Tahap umpan maju (*feedforward*).
2. Tahap umpan mundur (*backpropagation*).
3. Tahap pembaruan bobot dan bias.

Secara rinci algoritma pelatihan jaringan propagasi balik dapat diuraikan sebagai berikut :

Langkah 0 : Inisialisasi bobot – bobot, konstanta laju pelatihan ($\text{learnrate} = \alpha$), toleransi error atau nilai bobot (bila menggunakan nilai bobot sebagai kondisi berhenti) atau set minimal epoch (bila menggunakan banyaknya epoch sebagai kondisi berhenti).

Langkah 1 : Selama kondisi berhenti belum dicapai, maka lakukan langkah ke-2 hingga langkah ke-9.

Langkah 2 : Untuk setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai langkah ke-8.

Tahap I : Umpan Maju (*feedforward*)

Langkah 3 : Setiap unit masukan x_i (dari unit ke-1 hingga unit ke- n pada lapisan masukan) mengirimkan sinyal masukan ke setiap masukan yang berada pada lapisan

tersembunyi.

Langkah 4 : Masing – masing unit di lapisan tersembunyi (dari unit ke-1 hingga unit ke- p) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan dengan biasnya :

$$z_net_j = v_{jo} + \sum_{i=1}^n x_i v_{ji} \quad (5)$$

$$z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}} \quad (6)$$

Langkah 5 : Masing – masing unit keluaran (y_k , $k = 1,2,3, \dots m$) dikalikan dengan bobot dan dijumlahkan serta ditambahkan dengan biasnya.

$$y_net_k = w_{ko} + \sum_{j=1}^p z_j w_{kj} \quad (7)$$

$$y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \quad (8)$$

Tahap II : Umpan Mundur (*backpropagation*)

Langkah 6 : Masing – masing unit keluaran (y_k , $k = 1,2,3,\dots, m$) menerima pola target t_k sesuai dengan pola masukan saat pelatihan dan kemudian informasi kesalahan/error lapisan keluaran (δ_k) dihitung. δ_k dikirim ke lapisan dibawahnya dan digunakan untuk menghitung besarnya koreksi bobot dan bias (Δw_{kj} dan Δw_{ko}) antara lapisan tersembunyi dengan lapisan keluaran :

$$\delta_k = (t_k - y_k) f'(y_net_k) = (t_k - y_k) y_k (1 - y_k) \quad (9)$$

Hitung suku perubahan bobot w_{kj} yang akan digunakan untuk mengubah bobot w_{kj} dengan laju pelatihan α

$$\Delta w_{kj} = \alpha \delta_k z_j ; k = 1,2,3,\dots, m ; j = 0,1,\dots, p \quad (10)$$

Hitung perubahan bias

$$\Delta w_{ko} = \alpha \delta_k \quad (11)$$

Langkah 7 : Pada setiap unit di lapisan tersembunyi (dari unit ke-1 hingga ke- p ; $i = 1 \dots n$; $k = 1 \dots m$) dilakukan perhitungan informasi kesalahan lapisan tersembunyi (δ_j). δ_j kemudian digunakan untuk menghitung besar koreksi bobot dan bias (Δv_{ji} dan Δv_{jo}) antara lapisan masukan dan lapisan tersembunyi.

$$\delta_net_j = \sum_{k=1}^m \delta_k w_{kj} \quad (12)$$

$$\delta_j = \delta_net_j f'(\delta_net_j) = \delta_net_j z_j (1 - z_j) \quad (13)$$

Hitung suku perubahan bobot v_{ji} (yang digunakan untuk perbaikan bobot v_{ji}).

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (14)$$

Hitung perubahan bias (untuk memperbaiki v_{jo}).

$$\Delta v_{jo} = \alpha \delta_j \quad (15)$$

Tahap III : Pembaruan Bobot dan Bias

Langkah 8 : Masing – masing unit keluaran (y_k , $k = 1,2,3,\dots, m$) dilakukan pembaruan bias dan bobotnya ($j = 0,1,2,\dots, p$) sehingga menghasilkan bobot dan bias baru:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (16)$$

Demikian juga untuk setiap unit tersembunyi mulai dari unit ke-1 sampai dengan unit ke- p dilakukan pembaruan bobot dan bias :

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (17)$$

Langkah 9 : Uji kondisi berhenti (akhir iterasi).

Keterangan :

- x_i : unit masukan dari unit ke-1 hingga unit ke- n pada lapisan masukan
- v_{ji} : bobot garis dari unit masukan x_i ke unit lapisan tersembunyi z_j
- v_{jo} : bobot garis yang menghubungkan bias di unit masukan ke unit lapisan tersembunyi z_j
- z_j : keluaran dari unit tersembunyi
- α : laju pelatihan (berupa bilangan acak kecil yaitu : $0.0 < \alpha \leq 1.0$). Jika α terlalu besar akan menyebabkan pelatihan jaringan menjadi lambat.
- w_{kj} : bobot dari unit lapisan tersembunyi z_j ke unit keluaran y_k
- w_{ko} : bobot dari bias di lapisan tersembunyi ke unit keluaran y_k
- y_k : keluaran dari unit keluaran
- t_k : pola target yang ingin dicapai
- δ_k : kesalahan atau error lapisan keluaran
- Δw_{kj} : suku perubahan bobot w_{kj}
- Δw_{ko} : suku perubahan bobot w_{ko}
- δ_j : kesalahan atau error lapisan tersembunyi
- Δv_{ji} : suku perubahan bobot v_{ji}
- Δv_{jo} : suku perubahan bobot v_{jo}

II.3. Momentum^[7]

Penambahan parameter momentum dalam meng-update bobot seringkali bisa mempercepat proses pelatihan karena momentum memaksa proses perubahan bobot terus bergerak. Tinjau kembali persamaan (10) dan (14), jika *error* tidak terjadi maka δ_k dan δ_j bernilai 0. Hal ini akan menyebabkan nilai dari Δw_{kj} dan Δw_{ko} sama dengan 0, atau dengan kata lain pembaruan bobot berlanjut dalam arah yang sama seperti sebelumnya.

Dengan penambahan parameter momentum, bobot baru pada waktu ke $(t+1)$ didasarkan atas bobot pada waktu t dan $(t-1)$. Oleh karena itu harus ditambahkan 2 variabel baru yang mencatat besarnya momentum untuk 2 iterasi terakhir. Jika μ adalah konstanta ($0 \leq \mu \leq 1$) yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan persamaan:

$$w_{kj}(t+1) = w_{kj}(t) + \alpha \delta_k z_j + \mu(w_{kj}(t) - w_{kj}(t-1)) \quad (18)$$

dan

$$v_{ji}(t+1) = v_{ji}(t) + \alpha \delta_j x_i + \mu(v_{ji}(t) - v_{ji}(t-1)) \quad (19)$$

II.4. Pendeteksian Tepi Citra^{[3],[6]}

Pertemuan antara bagian objek dan bagian latar belakang dari objek disebut tepi objek.

Dalam pengolahan citra, tepi objek ditandai oleh titik yang nilai keabuannya memiliki perbedaan yang cukup besar dengan titik yang ada di sebelahnya (titik tetangga).

Deteksi tepi adalah proses untuk menemukan perubahan intensitas yang berbeda nyata dalam sebuah bidang citra. Sebuah operator deteksi tepi merupakan operasi bertetangga, yaitu sebuah operasi yang memodifikasi nilai keabuan sebuah titik berdasarkan nilai – nilai keabuan dari titik – titik yang ada di sekitarnya (tetangganya) yang masing – masing mempunyai bobot tersendiri. Bobot – bobot tersebut nilainya tergantung pada operasi yang akan dilakukan.

Operator yang digunakan untuk mendeteksi tepi pada makalah ini yaitu operator Prewitt. Proses penggunaan operator berbasis gradien untuk mendeteksi tepi dapat dilakukan dengan langkah – langkah :

1. Penentuan gradien citra untuk mengetahui intensitas variasi lokal dengan melakukan konvolusi dengan matriks konvolusi G_x dan G_y . Matriks konvolusi G_x dan G_y diperoleh dari pendekatan diskret derivatif parsial fungsi $f(x,y)$. Penentuan matriks konvolusi ditunjukkan dalam hubungan – hubungan dari persamaan-persamaan berikut :

$$\nabla f(x, y) = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} \quad (20)$$

Gradien G_x diperoleh dari pendekatan diferensial horisontal atau derivatif parsial terhadap x pada fungsi $f(x,y)$:

$$\frac{\partial f(x, y)}{\partial x} = f(x+1, y) - f(x, y) \quad (21)$$

Gradien G_y diperoleh dari pendekatan diferensial vertikal atau derivatif parsial terhadap y pada fungsi $f(x,y)$:

$$\frac{\partial f(x, y)}{\partial y} = f(x, y+1) - f(x, y) \quad (22)$$

2. Penentuan magnitude citra $G[f(x,y)]$ sebagai tepi:

$$\text{magnitude}(\nabla f) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} = \sqrt{G_x^2 + G_y^2} \quad (23)$$

Penentuan besar sudut atau arah untuk mengetahui kecenderungan arah tepi lokal:

$$\begin{aligned} \text{arah}(\nabla f) &= \tan^{-1}(G_y / G_x) \\ \theta(x, y) &= \arctan\left(\frac{\partial f / \partial y}{\partial f / \partial x}\right) \end{aligned} \quad (24)$$

Hasil pendeteksian tepi adalah citra tepi $g(x,y)$ yang nilai setiap pixelnya menyatakan kekuatan tepi $g(x,y) = G[f(x,y)]$. Keputusan apakah suatu piksel merupakan tepi atau bukan tepi dinyatakan dengan operasi pengambangan sebagai berikut:

$$g(x,y) = \begin{cases} 1, & G[f(x,y)] \geq T \\ 0, & \text{lainnya} \end{cases} \tag{25}$$

T adalah suatu nilai ambang yang digunakan pada saat melakukan proses pendeteksian tepi. Misalkan susunan piksel – piksel di sekitar piksel (x,y) adalah :

a_0	a_1	a_2
a_7	(x,y)	a_3
a_6	a_5	a_4

berdasarkan susunan piksel di atas, besar gradien yang dihitung menggunakan operator Prewitt adalah:

$$M = \sqrt{G_x^2 + G_y^2} \tag{26}$$

dan turunan parsial dapat dihitung menggunakan persamaan berikut:

$$\begin{aligned} G_x &= (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \\ G_y &= (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \end{aligned} \tag{27}$$

dengan c adalah konstanta yang bernilai 1, maka akan didapatkan operator Prewitt yaitu:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

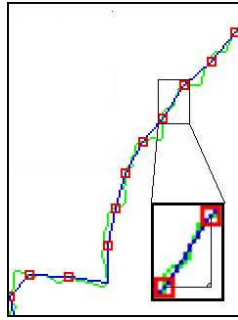
II.5. Proses Thinning^[3]

Secara umum, proses *thinning* berguna untuk mengurangi nilai ambang batas pada citra keluaran yang dihasilkan dari proses pendeteksian tepi menjadi garis dengan ukuran ketebalan satu piksel saja.

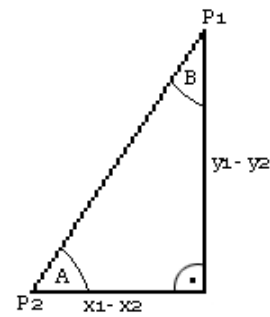
Proses *thinning* dilakukan terhadap citra biner (citra dengan nilai intensitas 0 dan 1) dengan cara mem-passing sebuah structuring element terhadap sebuah citra dengan cara yang hampir sama dengan konvolusi. Structuring element dapat berukuran sembarang dan juga memiliki titik poros (disebut juga titik origin/ titik asal/titik acuan).

II.6. Token pada Citra^[7]

Setelah citra mengalami proses pengolahan citra digital, diperlukan pengalihan bentuk citra menjadi bentuk yang dapat digunakan pada jaringan saraf tiruan. Digunakan dasar matematis yaitu sudut cosinus dan sinus yang merepresentasikan kriteria dari pengenalan pola.



Gambar 1. Token pada Citra



Gambar 2. Segitiga Siku – Siku

Gambar 1 merupakan contoh gambar yang telah mengalami proses pengolahan citra (pendeteksiian tepi dan *thinning*). Pada Gambar 1 tersebut terdapat 3 bagian, yaitu:

1. Garis Hijau
Merupakan garis yang terbentuk setelah citra tersebut mengalami pendeteksiian tepi dan *thinning*.
2. Persegi Merah
Persegi ini merepresentasikan sebuah titik pada citra hasil pengolahan. Persegi ini akan digunakan untuk membentuk garis jika dihubungkan dengan persegi lainnya.
3. Garis Biru
Merupakan garis yang terbentuk saat dua persegi merah saling dihubungkan, yang akan digunakan untuk menghitung sudut cosinus dan sinus.

Pada Gambar 2 terdapat segitiga siku – siku yang diperbesar ukuran gambarnya, sudut – sudut pada segitiga siku – siku ini dan rangkuman seluruh segitiga siku – siku lainnya pada citra adalah representasi token – token pada citra. Setiap token berisi informasi berupa nilai cosinus dan sinus dan setelah token pada citra diketahui dapat dimulai perhitungan pada jaringan propagasi balik.

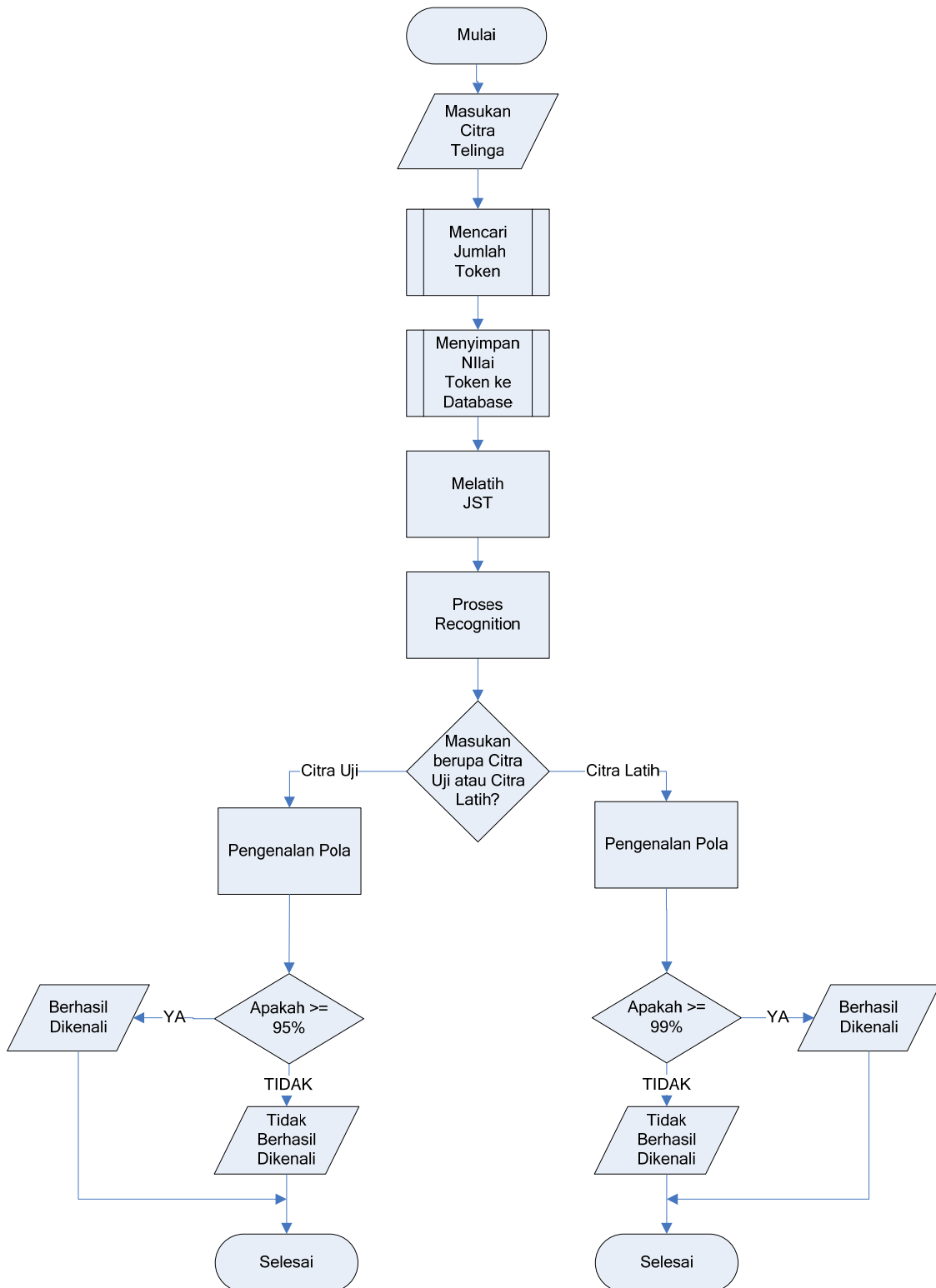
Pada Gambar 2 dapat lebih jelas dilihat bahwa sudut A dan B adalah 2 bagian yang diperlukan yang akan masuk ke dalam jaringan propagasi balik. Dengan dua sudut tersebut, dapat digambarkan sisi miring dari P1 ke P2 yang diperlukan untuk representasi citra.

$$\sin A = \frac{y1 - y2}{\sqrt{(y1 - y2)^2 + (x1 - x2)^2}} \quad (28)$$

$$\sin B = \frac{x1 - x2}{\sqrt{(y1 - y2)^2 + (x1 - x2)^2}} \quad (29)$$

III. PERANCANGAN DAN REALISASI

Pada bab ini dijelaskan tentang perancangan perangkat lunak yang meliputi proses pengolahan citra, menentukan nilai dari parameter – parameter untuk pelatihan JST, dan proses untuk pengenalan pola pada citra. Gambar 3. adalah diagram alir perangkat lunak secara umum.

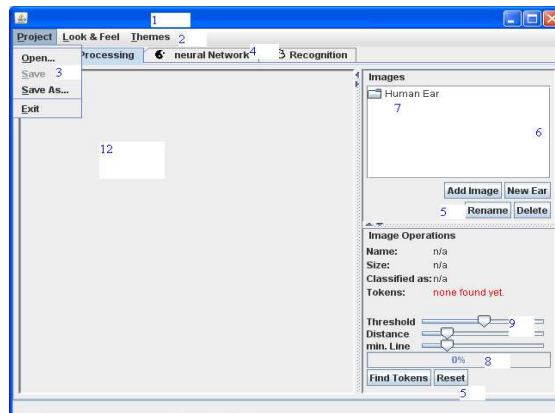


Gambar 3. Diagram Alir umum

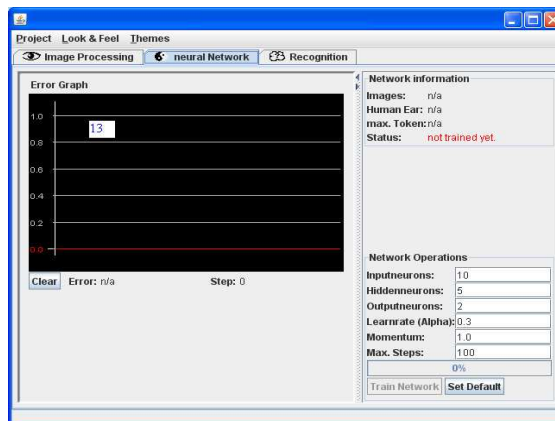
III.1. Perancangan User Interface

Pada bagian ini akan membahas perancangan user interface menggunakan Java. User interface yang dirancang memiliki 3 tab page utama yaitu tab page Image Processing, Neural

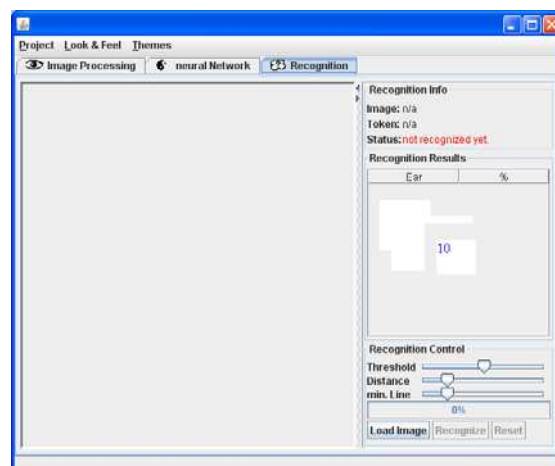
Network, dan Recognition. Masing – masing tab page akan menampilkan visualisasi yang berbeda bagi user. Gambar 4(a), (b), dan (c) menampilkan visualisasi untuk masing – masing tab page.



Gambar 4 (a) Tab Page Image Processing



Gambar 4 (b) Tab Page Neural Network



Gambar 4 (c) Tab Page Recognition

III.2. Proses Pengambilan Citra

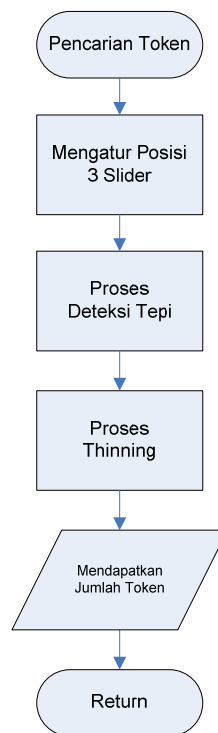
Program akan menampilkan dan mengambil citra yang terdapat pada media

penyimpanan di komputer. Format citra yang digunakan pada Makalah ini adalah jpg dengan ukuran 640 X 480.

III.3. Pencarian Jumlah Token

Proses untuk mencari jumlah token pada citra telinga dipengaruhi oleh resolusi citra yang digunakan serta pengaturan posisi pada slider Threshold, Distance, dan min. Line.

Pengaturan posisi ketiga slider tersebut akan berpengaruh terhadap proses pengolahan citra digital yang meliputi proses pendeteksian tepi citra dan proses *thinning*. Gambar 5 menunjukkan diagram alir proses pencarian jumlah token.



Gambar 5. Diagram Alir Pencarian Jumlah Token

III.4. Pengaturan Posisi Slider

Pada tab page Image Processing user dapat menggunakan ketiga slider sebagai konfigurasi untuk menentukan nilai ambang batas untuk proses deteksi tepi, menentukan jarak dari mark points (tanda – tanda berbentuk persegi berwarna merah yang terletak pada garis yang menyusun tepi citra), dan menentukan banyaknya garis – garis pemeriksa. Untuk mendapatkan hasil yang baik untuk setiap citra telinga yang diolah maka diperlukan proses uji coba dalam menentukan posisi ketiga slider ini.

III.5. Pendeteksian Tepi

Pengaturan posisi slider dengan proses deteksi tepi merupakan hal yang saling berkaitan. Pengaturan slider Threshold digunakan untuk memberikan suatu nilai ambang T yang digunakan pada saat melakukan proses pendeteksian tepi, keputusan apakah suatu piksel merupakan tepi atau bukan tepi dinyatakan dengan operasi pengambangan pada persamaan (23).

Jika magnitudo tepi atau kekuatan tepi adalah $G[f(x,y)]$ maka tepi pada citra akan dibentuk jika kekuatan tepi $G[f(x,y)]$ mempunyai nilai yang lebih besar atau sama dengan nilai ambang T sedangkan jika kekuatan tepi $G[f(x,y)]$ lebih kecil dari nilai ambang T maka tidak akan terbentuk tepi pada citra. Nilai inisial untuk nilai ambang T pada slider Threshold saat menjalankan program pertama kali adalah 50.

III.6. Proses Thinning

Proses *thinning* digunakan untuk mengurangi nilai ambang batas pada citra keluaran yang dihasilkan dari proses pendeteksian tepi menjadi garis dengan ukuran ketebalan satu piksel saja. Berikut ini adalah listing program dari proses *thinning* :

Citra keluaran proses *thinning* akan menghasilkan garis yang tidak kontinu seluruhnya, slider Check lines digunakan untuk memeriksa garis-garis pada citra hasil proses *thinning*. Garis-garis yang panjangnya melebihi atau sama dengan nilai slider Check lines akan dipertahankan sedangkan yang lebih pendek akan dibuang. Nilai inisial pada slider min. Line saat menjalankan program pertama kali adalah 20.

Pengaturan slider Distance digunakan untuk menentukan jarak antar mark points (persegi berwarna merah), jika menggeser posisi slider ke arah kanan akan menyebabkan jarak antar mark points semakin jauh sedangkan jika menggeser posisi slide ke arah kiri akan menyebabkan jarak antar mark points semakin dekat. Saat dua buah mark points saling dihubungkan maka dapat dihitung nilai sudut cosinus dan sinus sesuai dengan persamaan (28) dan (29). Nilai inisial pada slider Distance saat menjalankan program pertama kali adalah 20.

III.7. Mendapatkan Jumlah Token

Setelah citra telinga diambil dari media penyimpanan dan user mengatur posisi ketiga slider, dengan menekan button Find Tokens maka proses untuk pencarian jumlah token dari suatu citra dapat dilakukan. Setelah progress bar menunjukkan angka 100 %, maka proses tersebut selesai dan akan ditampilkan citra telinga yang sudah didapatkan jumlah tokennya.

III.8. Proses Melatih JST

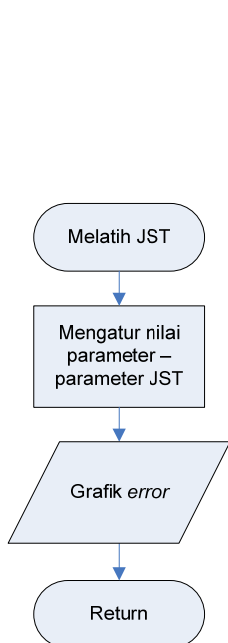
Jika jumlah token telah didapatkan, maka proses selanjutnya yaitu melatih JST, untuk melakukan proses ini user berinteraksi pada tab page Neural Network. Untuk melatih JST, user perlu memberikan nilai – nilai masukan pada parameter yang diperlukan untuk proses pelatihan yaitu parameter Inputneurons, Hiddenneurons, Outputneurons, Learnrate (Alpha), Momentum, dan Max. Step.

Parameter – parameter ini berada pada bagian Network Operations pada tab page Neural Network. Parameter – parameter tersebut mempunyai nilai inisial jika user menekan tombol Set Default yaitu sebagai berikut :

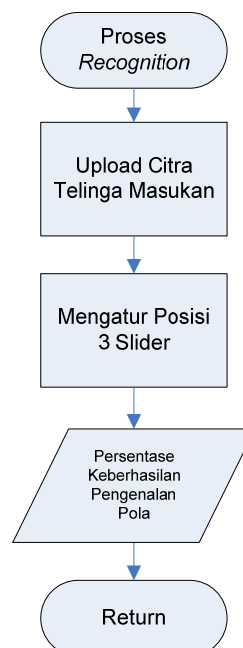
1. Inputneurons : 2 x Jumlah maksimal token yang sudah didapatkan
2. Hiddenneurons : 20
3. Outputneurons : Jumlah dari node anak yang berisi citra berformat jpg
4. Learnrate (Alpha) : 0.3
5. Momentum : 1.0
6. Max. Step : 500

Nilai – nilai inisial yang diberikan tersebut tidak bisa dipakai sebagai acuan saat proses

pelatihan jaringan, untuk mendapatkan hasil pelatihan yang baik perlu dilakukan proses uji coba pengaturan nilai – nilai parameter tersebut sehingga dapat dihasilkan keluaran yang diinginkan.



Gambar 6. Diagram Alir Pelatihan JST



Gambar 7. Diagram Alir Proses Recognition

III.9. Proses Recognition

Pada saat melakukan proses pengenalan pola citra masukan (proses recognition), hanya terjadi proses umpan maju saja pada jaringan propagasi balik tanpa adanya perubahan nilai bobot – bobot pada jaringan sebab nilai bobot – bobot yang digunakan adalah nilai bobot terakhir yang diperoleh dari proses pelatihan jaringan propagasi balik.









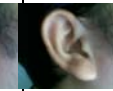





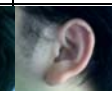
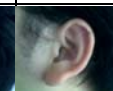

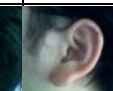


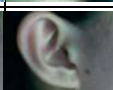































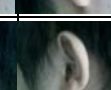


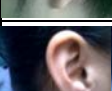


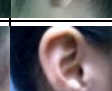
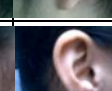
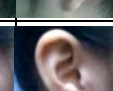
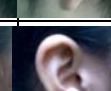





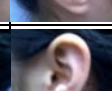
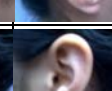


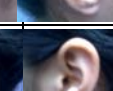


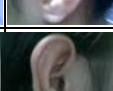

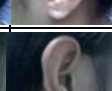
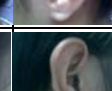
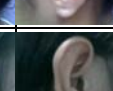
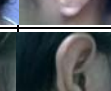










Citra telinga masukan yang akan diuji dapat diambil pada media penyimpanan di komputer, kemudian citra masukan ini mengalami proses pencarian jumlah token. Citra masukan akan mengalami proses pendeteksian tepi dan proses *thinning*, kedua proses ini dipengaruhi oleh pengaturan posisi dari ketiga slider seperti diperlihatkan pada Gambar 7.

Untuk proses pengenalan pola pada citra, user berinteraksi pada tab page Recognition. Setelah citra telinga masukan diambil dari media penyimpanan dan posisi ketiga slider telah diatur, maka dengan menekan button Recognize proses untuk mengenali pola pada citra dapat dilakukan. Setelah progress bar menunjukkan angka 100 %, maka proses tersebut selesai dan akan ditampilkan citra masukan yang jumlah tokennya sudah diketahui serta ditampilkan nilai persentase keberhasilan dari pengenalan pola tersebut dalam tabel pada bagian Recognition Results.

IV. DATA PENGAMATAN

Pada makalah ini digunakan citra telinga kiri dan telinga kanan dari 5 orang yang berbeda yaitu Andre, Ana, Sionny, Marnia, dan Juan. Masing-masing orang diambil 9 citra telinga kanan dan 9 citra telinga kiri, sehingga total citra 90 buah. Citra telinga yang digunakan sebagai data pengamatan ditampilkan pada Tabel 1.

TABEL 1. CITRA TELINGA YANG DIGUNAKAN SEBAGAI DATA PENGAMATAN

Nama	1	2	3	4	5	6	7	8	9
Telinga kanan Andre									
Telinga kiri Andre									
Telinga kanan Ana									
Telinga kiri Ana									
Telinga kanan Sionny									
Telinga kiri Sionny									
Telinga kanan Marnia									
Telinga kiri Marnia									
Telinga kanan Juan									
Telinga kiri Juan									

IV.1. Pencarian Token Citra Latih

Citra telinga pada proses ini akan dicari jumlah tokennya. Proses mencari jumlah token pada citra telinga ini dipengaruhi oleh perubahan posisi pada slider threshold, distance, dan min. line. Nilai slider threshold yang digunakan 10, 20, 30, 40, 50, 60, 70. Nilai distance, yang digunakan 10, dan nilai min. line yang digunakan 10, 20, 30. Proses pencarian token pada masing – masing citra dilakukan sebanyak 10 kali percobaan.










Tabel 2 adalah hasil uji coba pengaturan nilai slider pada telinga kanan Andre01. Terlihat bahwa tidak semua kombinasi nilai slider threshold, distance, dan min. line akan menghasilkan jumlah token yang diperlukan untuk memperoleh citra yang baik. Terlalu besar jumlah token maka citra yang dihasilkan akan terlalu detail sehingga noise akan masuk, terlalu sedikit jumlah token maka fitur citra telinga menjadi tidak terwakili. Pemilihan nilai slider threshold, distance, min. line dan jumlah token harus dapat merepresentasikan fitur citra telinga dengan baik. Berdasarkan hasil percobaan, citra telinga yang baik diperoleh dengan jumlah token antara 500-629.

Tabel 3 adalah rangkuman hasil uji coba pencarian token dan hasil pengolahan citra telinga kanan citra Andre. Proses yang sama dilakukan pada citra yang lain.

TABEL 2. HASIL UJI COBA PENGATURAN NILAI SLIDER PADA TELINGA KANAN ANDRE01

Nama	Nilai slider Threshold	Nilai slider Distance	Nilai slider min. Line	Jumlah Token
Andre01	10	10	10	Proses gagal
Andre01	20	10	10	Proses gagal
Andre01	30	10	10	4535
Andre01	40	10	10	1427
Andre01	50	10	10	723
Andre01	60	10	10	629
Andre01	70	10	10	490
Andre01	60	10	20	593
Andre01	60	10	30	586
Andre01	70	10	20	479

TABEL 3. NILAI KETIGA SLIDER YANG DIPILIH PADA TELINGA KANAN ANDRE

Nama	Nilai slider Threshold	Nilai slider Distance	Nilai slider min. Line	Jumlah Token	Hasil
Andre01	60	10	10	629	
Andre02	70	10	10	609	
Andre03	60	10	10	545	
Andre04	60	10	20	603	
Andre05	60	10	10	525	
Andre06	60	10	10	500	
Andre07	60	10	20	609	
Andre08	60	10	10	612	
Andre09	70	10	10	608	

IV.2. Pelatihan dan Pengujian

Untuk melatih JST diperlukan pengaturan pada parameter yang diperlukan untuk proses pelatihan yaitu parameter Inputneurons, Hiddenneurons, Outputneurons, Learnrate (Alpha), Momentum, dan Max. Step. Pengaturan pada parameter ini diperlukan agar saat proses pelatihan jaringan propagasi balik menghasilkan nilai error yang kecil, sebab nilai error yang kecil menunjukkan keluaran yang dihasilkan jaringan (v_k) sudah mendekati target yang ingin dicapai (t_k).

Target yang ingin dicapai atau keluaran yang diharapkan untuk setiap proses pengujian pengenalan pola bernilai 100 %. Jaringan propagasi balik dikatakan berhasil mengenali pola

pada citra latih jika persentase keluaran jaringan mendekati nilai target yang ingin dicapai serta memberikan respon yang baik untuk mengenali pola pada citra uji.

Setelah melakukan beberapa kali uji coba untuk memberikan nilai masukan pada parameter pelatihan, digunakan nilai – nilai sebagai berikut :

Inputneurons	: 700
Hiddenneurons	: 55
Outputneurons	: 100
Learnrate (Alpha)	: 0.1
Momentum	: 1.0
Max. Step.	: 10000
Error	: 0.18185939263730885

Data pengamatan untuk hasil pengujian pada pelatihan ini ditampilkan pada Tabel 4. Tabel 4 adalah tabel persentase kesamaan pengenalan pola citra Andre kanan terhadap citra Andre kanan pada database. Tabel 5 adalah tabel persentase kesamaan pengenalan pola citra Andre kanan terhadap citra Andre kiri pada database. Tabel 6 adalah tabel persentase kesamaan pengenalan pola citra Andre terhadap semua citra Ana pada database.

Pada setiap tabel yang ditampilkan, kolom paling kiri berisi nama citra – citra masukan yang akan diuji (terdiri dari 9 citra latih dan 1 citra uji) sedangkan nama citra – citra pada database berada pada baris paling atas. Pada setiap proses pengujian untuk mengenali pola pada citra masukan ini, dihasilkan nilai persentase keberhasilan jaringan propagasi balik dalam mengenali pola pada citra latih dan citra uji.

TABEL 4. PERSENTASE KESAMAAN PENGENALAN POLA CITRA ANDRE KANAN DENGAN CITRA ANDRE KANAN PADA DATABASE

	Andre Kanan01	Andre Kanan02	Andre Kanan03	Andre Kanan04	Andre Kanan05	Andre Kanan06	Andre Kanan07	Andre Kanan08	Andre Kanan09
Andre Kanan01	99.878	98.915	97.925	96.940	95.956	94.968	93.983	93.002	92.014
Andre Kanan02	98.948	99.917	98.946	97.971	96.981	95.992	95.006	94.008	93.010
Andre Kanan03	97.795	98.791	99.782	98.805	97.824	96.840	95.848	94.859	93.875
Andre Kanan04	96.910	97.904	98.881	99.858	98.874	97.891	96.918	95.935	94.938
Andre Kanan05	95.871	96.870	97.870	98.858	99.853	98.893	97.901	96.916	95.927
Andre Kanan06	94.777	95.776	96.776	97.775	98.775	99.764	98.805	97.840	96.878
Andre Kanan07	93.958	94.950	95.939	96.930	97.924	98.916	99.891	98.917	97.939
Andre Kanan08	92.935	93.931	94.926	95.921	96.900	97.876	98.857	99.832	98.854
Andre Kanan09	91.961	92.959	93.952	94.946	95.944	96.938	97.926	98.901	99.874
Andre Kanan10	97.040	97.550	96.654	96.564	95.669	94.689	94.647	94.003	93.210

Berdasarkan data pengamatan pada Tabel 4 dapat dilihat bahwa untuk proses pengujian yang dilakukan pada masing – masing citra latih Andre Kanan01 sampai Andre Kanan09, nilai

persentase kesamaan saat dibandingkan dengan citra pada database mencapai $\geq 99\%$ dan dapat dikenali dengan benar sebagai citra telinga kanan Andre. Sedangkan saat proses pengujian citra uji Andre Kanan10, jaringan propagasi balik memberikan respon yang baik dalam mengenali citra uji ini dengan nilai persentase kesamaan tertinggi sebesar 97.550% (Andre Kanan02).

TABEL 5. PERSENTASE KESAMAAN PENGENALAN POLA CITRA ANDRE KANAN DENGAN CITRA ANDRE KIRI PADA DATABASE

	Andre Kiri01	Andre Kiri02	Andre Kiri03	Andre Kiri04	Andre Kiri05	Andre Kiri06	Andre Kiri07	Andre Kiri08	Andre Kiri09
Andre Kanan01	91.022	90.024	89.030	88.038	87.038	86.038	85.038	84.039	83.039
Andre Kanan02	92.014	91.014	90.014	89.015	88.015	87.015	86.015	85.015	84.015
Andre Kanan03	92.881	91.897	90.926	89.952	88.963	87.978	86.990	85.994	84.996
Andre Kanan04	93.951	92.952	91.958	90.961	89.962	88.963	87.977	86.998	86.016
Andre Kanan05	94.934	93.937	92.937	91.940	90.946	89.949	88.954	87.959	86.964
Andre Kanan06	95.911	94.938	93.954	92.978	91.989	90.996	90.002	89.004	88.007
Andre Kanan07	96.948	95.965	94.978	93.983	92.990	91.991	90.991	89.992	88.993
Andre Kanan08	97.869	96.886	95.903	94.918	93.940	92.948	91.953	90.958	89.973
Andre Kanan09	98.890	97.905	96.923	95.941	94.943	93.950	92.959	91.963	90.976
Andre Kanan10	92.479	91.496	90.724	89.754	88.763	87.773	86.776	86.001	85.824

TABEL 6. PERSENTASE KESAMAAN PENGENALAN POLA CITRA ANDRE KANAN DENGAN CITRA ANA KANAN PADA DATABASE

	Ana Kanan01	Ana Kanan02	Ana Kanan03	Ana Kanan04	Ana Kanan05	Ana Kanan06	Ana Kanan07	Ana Kanan08	Ana Kanan09
Andre Kanan01	82.041	81.054	80.054	79.055	78.058	77.059	76.061	75.061	74.062
Andre Kanan02	83.017	82.021	81.021	80.022	79.022	78.022	77.022	76.022	75.022
Andre Kanan03	84.000	83.003	82.006	81.010	80.010	79.010	78.012	77.019	76.021
Andre Kanan04	85.022	84.028	83.028	82.030	81.031	80.032	79.032	78.032	77.032
Andre Kanan05	85.983	84.990	84.002	83.025	82.055	81.074	80.078	79.081	78.093
Andre Kanan06	87.013	86.020	85.024	84.031	83.036	82.055	81.060	80.071	79.087
Andre Kanan07	88.006	87.009	86.010	85.019	84.020	83.023	82.023	81.024	80.024
Andre Kanan08	88.978	87.988	87.005	86.014	85.027	84.035	83.051	82.053	81.054
Andre Kanan09	89.991	89.000	88.001	87.006	86.008	85.011	84.013	83.016	82.016
Andre Kanan10	84.835	83.838	82.838	81.839	80.840	79.840	78.842	77.849	76.849

Tabel 5 adalah proses pengujian yang dilakukan terhadap orang yang sama tetapi telinga yang berbeda. Tabel 6 adalah proses pengujian yang dilakukan terhadap orang yang berbeda. Pada Tabel 5 dan 6 dapat dilihat bahwa tidak ada persentase kesamaan yang $\geq 99\%$, sehingga citra didefinisikan berbeda. Pada Tabel 5 persentase kesamaan paling tinggi 98.890 % dan pada Tabel 6 persentase kesamaan paling tinggi 89.991 %.

IV.3. Analisa Nilai FAR dan FRR

Derajat kesamaan antara dua fitur biometrik dinilai dengan False Acceptance Rate (FAR) dan False Rejection Rate (FRR). FAR menyatakan tingkat kesalahan penerimaan yang muncul akibat dari sistem menganggap sama, pola masukan dengan pola yang berbeda dalam database. FRR menyatakan tingkat kesalahan penolakan yang muncul akibat dari sistem menganggap tidak sama, pola masukan dengan pola yang sama dalam database^[8].

Dengan menggunakan nilai ambang yang diambil dari hasil proses pengujian yaitu $\geq 99\%$ untuk persentase kesamaan pengenalan pola pada citra latih yang dibandingkan dengan citra pada database dan $\geq 95\%$ untuk citra uji serta dengan jumlah data keseluruhan adalah 9000, maka persentase nilai FAR dan FRR dapat dihitung sebagai berikut :

$$FAR = \frac{11}{9000} \times 100\% = 0.122\%$$

$$FRR = \frac{785}{9000} \times 100\% = 8.722\%$$

V. KESIMPULAN

Kesimpulan yang diperoleh berdasarkan hasil simulasi perangkat lunak pada makalah ini yaitu berdasarkan hasil pengujian untuk verifikasi telinga seseorang menggunakan metode Backpropagation Neural Network, nilai persentase keberhasilan pengenalan seseorang pada pengujian menggunakan citra latih mencapai $\geq 99\%$ sedangkan pada pengujian citra uji mencapai $\geq 95\%$ serta diperoleh nilai FAR sebesar 0.122 % dan FRR sebesar 8.722 %.

DAFTAR PUSTAKA

- [1] M. Burge and W. Burger, "Ear biometrics for machine vision", *In 21st Workshop of the Austrian Association for Pattern Recognition*, Hallstatt. ÖAGM, R. Oldenbourg Verlag, 1997
- [2] A. Iannarelli, *Ear Identification*, Forensic Identification Series, Paramount Publishing Company, Fremont, California, 1989.
- [3] T. Sutoyo, E. Mulyanto, V. Suhartono, O. D. Nurhayati, dan Wijanarto. 2009. *Teori Pengolahan Citra Digital*, Yogyakarta: Penerbit Andi, 2009
- [4] <http://www.webster-dictionary.org/definition/image>, diakses tanggal 10 Januari 2010.
- [5] D. Puspitaningrum, *Pengantar Jaringan Saraf Tiruan*, Yogyakarta: Penerbit Andi, 2006.
- [6] D. Putra, *Sistem Biometrika*, Yogyakarta: Penerbit Andi, 2009.
- [7] J. J. Siang, *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*, Yogyakarta: Penerbit Andi, 2005.
- [8] P. Gregory and M. A. Simon, *Biometrics for Dummies*, Wiley Publishing, 2008