



tanda tangan yang dapat dilakukan dengan menggunakan komputer, sehingga akan menghemat waktu dan lebih efisien dibandingkan secara manual.

Berdasarkan uraian latar belakang di atas, yang menjadi rumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana merancang sebuah aplikasi pengenalan tanda tangan manual menggunakan jaringan saraf tiruan dengan algoritma *backpropagation* ?
2. Bagaimana penerapan metode *backpropagation* dalam pengenalan tanda tangan.?

Berdasarkan tujuan di atas, maka manfaat bagi penelitian adalah sebagai berikut :

1. Untuk merancang sebuah aplikasi pengenalan tanda tangan yang dapat dikembangkan lebih lanjut dalam memberikan informasi pengenalan tanda tangan.
2. Mengetahui dan mengetahui cara kerja dari *Backpropagation* dalam melakukan pengenalan tanda tangan.

## PENGERTIAN JARINGAN SYARAF TIRUAN (JST)

Jaringan syaraf tiruan (*Artifial Neural Network*) merupakan salah satu sistem pemrosesan informasi yang didesain untuk menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinopsisnya. Menurut Maharani dan Irawan<sup>[1]</sup> Jaringan Syaraf Tiruan (JST) merupakan suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan syaraf biologi (JSB).

## PENGERTIAN *BACKPROPAGATION*

*Backpropagation* adalah algoritma pembelajaran yang terawasi dan biasanya digunakan oleh *percepton* dengan banyak lapisan untuk mengubah *neuron-neuron* yang ada pada lapisan tersembunyi<sup>[2]</sup>.

Metode *backpropagation* adalah metode jaringan syaraf tiruan untuk menangani masalah pengenalan pola-pola yang kompleks. Tahap pelatihan dengan menggunakan metode *backpropagation* terdiri dari tiga fase, yaitu fase propagasi maju, fase propagasi mundur, dan fase perubahan bobot. Ketiga fase tersebut diulang terus menerus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah terasi atau kesalahan atau target error. *Backpropagation* memiliki beberapa unit yang ada dalam satu atau lebih *layer* tersembunyi.

## APLIKASI JARINGAN SYARAF TIRUAN (JST)

Aplikasi jaringan syaraf tiruan merupakan sistem adaptif yang dapat merubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. Adapun beberapa aplikasi jaringan syaraf tiruan adalah sebagai berikut:

1. Pengenalan Pola (*Pattern Recognition*)

Jaringan syaraf tiruan dapat dipakai untuk mengenali pola (misalnya huruf, angka, suara, atau tanda tangan) yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang sudah beberapa waktu tidak dijumpainya (mungkin wajah/bentuk tubuhnya sedikit berubah).

2. Proses Jaringan (*Signal Processing*)  
Jaringan syaraf tiruan (model ADALINE) dapat dipakai untuk menekan noise dalam saluran telepon.
3. Peramalan (*Forecasting*)

Jaringan syaraf tiruan juga dapat dipakai untuk meramalkan apa yang akan terjadi dimasa yang akan datang dengan berdasarkan pola kejadian yang ada dimasa yang lampau. Ini dapat dilakukan mengingat kemampuan jaringan syaraf

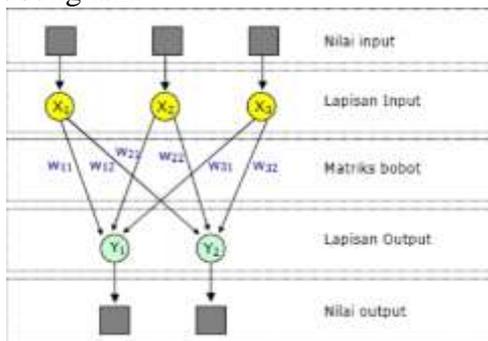
tiruan untuk mengingat dan membuat generalisasi dari apa yang sudah ada sebelumnya.

## ARSITEKTUR JARINGAN

Bentuk dari arsitektur pada jaringan ada beberapa jenis diantaranya:

### 1. Jaringan dengan lapisan tunggal (*single layer net*)

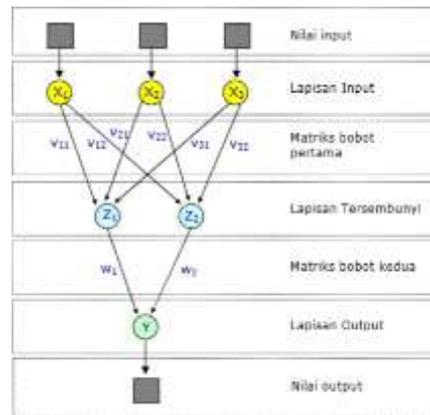
Jaringan ini hanya memiliki satu lapisan dengan bobot. Pada prosesnya hanya melewati *input* menuju *output* tanpa memiliki lapisan tersembunyi. Pada gambar jaringan arsitektur tunggal dibawah terdapat lapisan *input* mempunyai 3 *neuron*, yaitu  $X_1$ ,  $X_2$ , dan  $X_3$ . Pada layer *output* terdapat 2 *neuron* yaitu  $Y_1$  dan  $Y_2$ . *Neuron-neuron* pada lapisan saling berhubungan. Nilai bobot memberikan efek besar terhadap 2 - *neuron* yang saling berhubungan. Nilai *input* dan *ouput* seluruhnya akan dihubungkan.



**Gambar 2.1 Arsitektur Jaringan Syaraf dengan Lapisan Tunggal**

### 2. Jaringan dengan lapisan tunggal (*multilayer layer*)

Jaringan ini hanya memiliki lapisan tersembunyi yang banyak dan terletak Siantar lapisan *input* dan *output*. Namun, pembelajaran yang dihasilkan akan semakin sulit, dan jaringan ini mampu menyelesaikan masalah yang sulit daripada jaringan lainnya.



**Gambar 2.2 Arsitektur Jaringan Syaraf dengan Banyak Lapisan**

## PENERAPAN ALGORITMA BACKPROPAGATION

Algoritma pelatihan *backpropagation* merupakan pelatihan terbimbing yang mempunyai banyak lapisan. Algoritma pelatihan untuk jaringan dengan satu *layer* tersembunyi (dengan fungsi aktivasi *sigmoid biner*) adalah sebagai berikut:

### 1. Algoritma *Backpropagation*

Pada tahap ini nasalisasi bobot (ambil nilai bobot awal dengan random yang cukup kecil) setelah itu kerjakan langkah-langkah selanjutnya sampai kondisi berhenti belum terpenuhi.

### 2. Tahapan Perambatan Maju

Setiap pasangan elemen yang akan dilakukan pembelajaran, maka kerjakan *froward*:

- Tiap unit input ( $x_i, i = 1, 2, 3, \dots, n$ ) mencari sinyal  $x_i$  dan meneruskan sinyal tersebut pada lapisan yang ada diatasnya (lapisan tersembunyi).
- Tiap unit tersembunyi ( $z_j, j = 1, 2, 3, \dots, p$ ) menjumlahkan sinyal-sinyal input terbobot:

$$z\_in_j = v_0 + \sum_{i=1}^n x_i v_{ij} \dots \dots \dots (1)$$

- Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$z_j = f(z\_in_j) \dots \dots \dots (2)$$

- Dan kirimkan sinyal tersebut ke semua unit dilapisan atasnya (unit-unit output).

- c. Tiap unit output ( $y_k, k = 1,2,3,\dots, m$ ) menjumlahkan sinyal-sinyal input terbobot

$$y_{in_k} = w_0k + \sum_{j=1}^p z_j w_{jk} \dots \dots \dots (3)$$

- Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$y_k = f(y_{in_k}) \dots \dots \dots (4)$$

- Dan kirimkan sinyal tersebut ke semua unit dilapisan atasnya (unit-unit output).

### 3. Backpropagation (Perambatan Balik)

- a. Tiap-tiap unit output ( $y_k, k = 1,2,3,\dots, m$ ) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi *error*-nya.

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \dots \dots (5)$$

- Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki  $w_{jk}$ )

$$\Delta w_{jk} = \alpha \delta_k z_{ij} \dots \dots \dots (6)$$

- Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{ok}$ ):

$$\Delta w_{jk} = \alpha \delta_k \dots \dots \dots (7)$$

- Kirimkan  $\sigma_k$  ini ke unit – unit yang ada di lapisan bawahnya.

- b. Tiap-tiap unit tersembunyi ( $z_j, j = 1,2,3, \dots, p$ ) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan atasnya):

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \dots \dots \dots (8)$$

- Kalikan nilai ini dengan turunan dari fungsi aktivitasnya untuk menghitung informasi *error*.

$$\delta_j = \delta_{in_j} f'$$

- Kemudian hitung koreksi bobot ( yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ ):

$$\Delta v_{jk} = \alpha \delta_j x_i \dots \dots \dots (10)$$

- Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{oj}$ ):

$$\Delta v_{oj} = \alpha \delta_j \dots \dots \dots (11)$$

- Tiap unit output ( $y_k, k = 1,2,3,\dots, m$ ) memperbaiki bias dan bobotnya ( $j= 0,1,2,3,\dots, p$ )

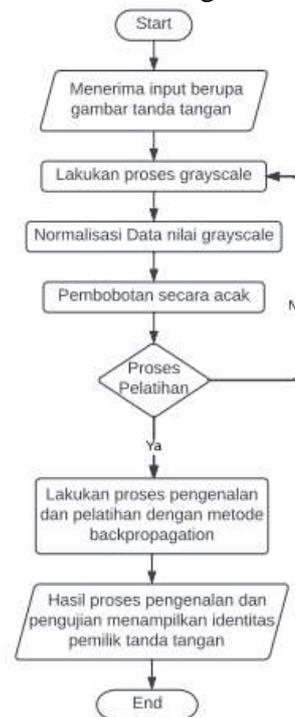
$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots \dots \dots (12)$$

## 2. METODE PENELITIAN

Pada penelitian ini mengangkat jenis pengembangan pada perangkat lunak yang akan digunakan pada skripsi ini, maka pada tahapan pertama yang dilakukan adalah analisis terhadap kebutuhan yang diperlukan. Pada analisis kebutuhan merupakan tahap awal untuk menentukan aplikasi seperti apa yang akan dibuat. Dalam perancangan aplikasi pengenalan tanda tangan banyak informasi mengenai pengenalan tanda tangan seperti teknik perhitungan yang akan digunakan untuk merancang sebuah aplikasi pengenalan tanda tangan.

### Perancangan Flowchart

Flowchart merupakan alur dari berjalannya sebuah sistem yang akan dibuat. Berikut ini adalah flowchart dari pengenalan tanda tangan:

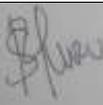
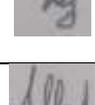


Gambar 3.4 Flowchart

## Data Pendukung Penelitian

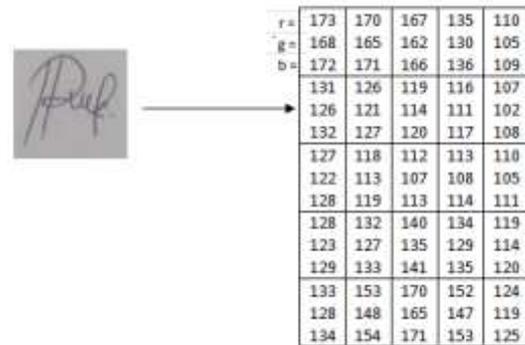
Penelitian ini menggunakan data berupa citra tanda tangan. Data tanda tangan terdiri dari citra tanda tangan dari 10 responden berbeda. Setiap responden memberikan 3 citra tanda tangan sehingga banyaknya data adalah 30 citra tanda tangan. Tipe citra yang digunakan adalah citra berekstensi .jpg dan melalui *preprocessing* untuk mendapatkan data *input*.

**Tabel 3.1 Citra Tanda Tangan**

No	Nama	Citra Tanda Tangan
1.	Raoudha Puspita	
2.	M. Ihsan	
3.	Siti Rianti Rizki Utami	
4.	Tengku Haliza Nur	
5.	Divya Angreny	
6.	Marisa	
7.	Benny Nugraha	
8.	Annisa Dwi Lestari	
9.	Desy Nastuty	
10.	Rizwan Fauzan	

## Penerapan Metode

Penerapan metode sangatlah dibutuhkan dalam memecahkan suatu permasalahan yang sangat sulit untuk ditentukan dalam proses penilaiannya. Berikut merupakan langkah awal dalam data tanda tangan diubah ke data citra dan dijadikan *grayscale*.



Gambar 3.1 Nilai Pixel pada Input Gambar

Selanjutnya lakukan perhitungan dari nilai citra asli ke citra grayscale.

$$GRAY = (R + G + B)/3$$

Piksel -1 (x =1, y =1)

$$RGB = ( 173, 168, 172)$$

## Perancangan JST *Backpropagation*

Proses pelatihan jaringan dilakukan dengan menggunakan data *input* dan perhitungan pelatihan dengan menggunakan metode *backpropagation*:

1. *Learning Rate* ( $\alpha$ ) = 0,2
2. *Target Error* = 0,01
3. *Maksimum epoch* = 1000
4. Bobot awal *input* ke *hidden layer* ( $V_{ij}$ ):

Hidden Layer ( $V_{ij}$ )					
$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	
0,2	0,4	0,3	0,2	0,1	
0,4	0,2	0,2	0,1	0,3	
0,1	0,3	0,1	0,4	0,2	
0,1	0,2	0,4	0,3	0,1	
0,2	0,3	0,1	0,2	0,4	

5. Bobot awal bias ke *hidden layer* ( $V_{0j}$ ):

Hidden Layer ( $V_{0j}$ )					
$V_{01}$	$V_{02}$	$V_{03}$	$V_{04}$	$V_{05}$	

0.2	0.3	0.1	0.4	0.2
-----	-----	-----	-----	-----

6. Bobot awal *hidden layer* ke *output layer* ( $W_{jk}$ ):

Output Layer ( $W_{ij}$ )				
$W_{11}$	$W_{21}$	$W_{31}$	$W_{41}$	$W_{51}$
0.1	0.3	0.2	0.4	0.2

7. Bobot awal bias ke *output layer* ( $W_{0j}$ ):

$W_{01}$	0,1
----------	-----

### Tahap Perambatan Maju (*forward propagation*)

$$\begin{aligned} Z_{in1} &= V_{01} + X_1 * V_{11} + X_2 * V_{21} + X_3 * \\ & \quad V_{31} + X_4 * V_{41} + X_5 * V_{51} \\ &= 0,2 + (1 * 0,2) + (0,086 * 0,4) + (0,1 * 0) \\ & \quad + (0,021 * 0,1) + (0,130 * 0,2) \\ &= 0,46 \end{aligned}$$

$$\begin{aligned} Z_{in2} &= V_{02} + X_1 * V_{12} + X_2 * V_{22} + X_3 * V_{32} \\ & \quad + X_4 * V_{42} + X_5 * V_{52} \\ &= 0,3 + (1 * 0,4) + (0,086 * 0,2) + (0,3 * 0) \\ & \quad + (0,021 * 0,2) + (0,130 * 0,3) \\ &= 0,76 \end{aligned}$$

$$\begin{aligned} Z_{in3} &= V_{03} + X_1 * V_{13} + X_2 * V_{23} + X_3 * V_{33} \\ & \quad + X_4 * V_{43} + X_5 * V_{53} \\ &= 0,1 + (1 * 0,3) + (0,086 * 0,2) + (0,1 * 0) \\ & \quad + (0,021 * 0,4) + (0,130 * 0,1) \\ &= 0,43 \end{aligned}$$

$$\begin{aligned} Z_{in4} &= V_{04} + X_1 * V_{14} + X_2 * V_{24} + X_3 * V_{34} \\ & \quad + X_4 * V_{44} + X_5 * V_{54} \\ &= 0,4 + (1 * 0,2) + (0,086 * 0,1) + (0,4 * 0) \\ & \quad + (0,021 * 0,3) + (0,130 * 0,2) \\ &= 0,71 \end{aligned}$$

$$\begin{aligned} Z_{in5} &= V_{05} + X_1 * V_{15} + X_2 * V_{25} + X_3 * V_{35} \\ & \quad + X_4 * V_{45} + X_5 * V_{55} \\ &= 0,2 + (1 * 0,1) + (0,086 * 0,3) + (0,2 * 0) \\ & \quad + (0,021 * 0,1) + (0,130 * 0,4) \\ &= 0,37 \end{aligned}$$

### Fungsi aktivasi *sigmoid* pada *hidden layer*:

$$Z_1 = \frac{1}{1 + e^{-Z_{in1}}} = \frac{1}{1 + e^{-0,46}} = 0,61$$

$$Z_2 = \frac{1}{1 + e^{-Z_{in2}}} = \frac{1}{1 + e^{-0,76}} = 0,68$$

$$Z_3 = \frac{1}{1 + e^{-Z_{in3}}} = \frac{1}{1 + e^{-0,43}} = 0,60$$

$$Z_4 = \frac{1}{1 + e^{-Z_{in4}}} = \frac{1}{1 + e^{-0,71}} = 0,67$$

$$Z_5 = \frac{1}{1 + e^{-Z_{in5}}} = \frac{1}{1 + e^{-0,37}} = 0,59$$

### Operasi pada *output layer*:

$$\begin{aligned} y_{in1} &= w_{k1} \sum_{i=1}^5 Z_j w_{kj} \\ Y_{in} &= W_{01} + Z_1 * W_{11} + Z_2 * W_{21} + Z_3 * \\ & \quad W_{31} + Z_4 * W_{41} + Z_5 * W_{51} \\ &= 0,2 + (0,61 * 0,2) + (0,68 * 0,3) + (0,60 * \\ & \quad 0,1) + (0,67 * 0,4) + (0,59 * 0,2) \\ &= 0,95 \end{aligned}$$

### Fungsi aktivasi *sigmoid biner* pada *output layer*:

$$y_1 = \frac{1}{1 + e^{-y_{in1}}} = \frac{1}{1 + e^{-0,95}} = 0,72$$

### Tahap Perambatan Balik (*Backpropagation*):

$$\delta_k = (T_k - Y_k) f'(Y_{nk})$$

$$\delta_1 = (T_1 - Y_1) f'(Y_{in}(1 - Y_{in}))$$

$$\delta_1 = (1 - 0,72) * 0,72(1 - 0,72)$$

$$\delta_1 = 0,056$$

### Menghitung perbaikan bobot $W_{jk}$ :

$$\Delta w_{jk} = \alpha \cdot \delta_j \cdot Z_j$$

$$\Delta w_{11} = \alpha \cdot \delta_1 \cdot Z_1 = 0,2 * 0,056 * 0,46 = 0,005$$

$$\Delta w_{12} = \alpha \cdot \delta_2 \cdot Z_2 = 0,2 * 0,056 * 0,76 = 0,008$$

$$\Delta w_{13} = \alpha \cdot \delta_3 \cdot Z_3 = 0,2 * 0,056 * 0,43 = 0,004$$

$$\Delta w_{14} = \alpha \cdot \delta_4 \cdot Z_4 = 0,2 * 0,056 * 0,71 = 0,007$$

$$\Delta w_{15} = \alpha \cdot \delta_5 \cdot Z_5 = 0,2 * 0,056 * 0,37 = 0,004$$

### Menghitung perbaikan bias :

$$w_{01} = \alpha \cdot \delta_1 = 0,2 * 0,056 = 0,011$$

Unit tersembunyi menjumlahkan

*deltainput* :

$$\delta_{in1} = \delta_1 * w_{11} = 0,1 * 0,056 = 0,005$$

$$\delta_{in2} = \delta_1 * w_{21} = 0,3 * 0,056 = 0,016$$

$$\delta_{in3} = \delta_1 * w_{31} = 0,2 * 0,056 = 0,011$$

$$\delta_{in4} = \delta_1 * w_{41} = 0,4 * 0,056 = 0,022$$

$$\delta_{in5} = \delta_1 * w_{51} = 0,2 * 0,056 = 0,011$$

### Hitung informasi *output*:

$$\delta_j = \delta_{inj} \cdot Z_j (1 - Z_j)$$

$$\delta_1 = \delta_{in1} \cdot Z_1 (1 - Z_1)$$

$$= 0,005 * 0,46 (1 - 0,46) = 0,0012$$

$$\begin{aligned}\delta_2 &= \delta_{in2} \cdot Z_2 (1 - Z_2) \\ &= 0,016 * 0,76 (1 - 0,76) = 0,0029 \\ \delta_3 &= \delta_{in3} \cdot Z_3 (1 - Z_3) \\ &= 0,011 * 0,43 (1 - 0,43) = 0,0026 \\ \delta_4 &= \delta_{in4} \cdot Z_4 (1 - Z_4) \\ &= 0,022 * 0,71 (1 - 0,71) = 0,0045 \\ \delta_5 &= \delta_{in5} \cdot Z_5 (1 - Z_5) \\ &= 0,011 * 0,37 (1 - 0,37) = 0,0025\end{aligned}$$

#### Menghitung perbaikan bobot :

$$\begin{aligned}\Delta v_{jk} &= \alpha * \delta_j * X_i \\ \Delta v_{11} &= \alpha * \delta_1 * X_1 = 0,2 * 0,0012 * 1 \\ &= 0,0002 \\ \Delta v_{12} &= \alpha * \delta_2 * X_2 = 0,2 * 0,0029 * 1 \\ &= 0,0005 \\ \Delta v_{13} &= \alpha * \delta_3 * X_3 = 0,2 * 0,0026 * 1 \\ &= 0,0005 \\ \Delta v_{14} &= \alpha * \delta_4 * X_4 = 0,2 * 0,0045 * 1 \\ &= 0,0009 \\ \Delta v_{15} &= \alpha * \delta_5 * X_5 = 0,2 * 0,0025 * 1 \\ &= 0,0005\end{aligned}$$

**Tabel 3.4 Perubahan masing-masing bobot**

V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	V <sub>4</sub>	V <sub>5</sub>
0,0002	2,06	0	5,04	3,12
0,0005	4,98	0	1,21	7,54
0,0005	4,47	0	1,09	6,67
0,0009	7,74	0	1,89	0,0001
0,0005	4,3	0	1,05	6,5

#### Menghitung perbaikan bobot bias:

$$\Delta v_{0j} = \alpha \cdot \delta_j$$

**Tabel 3.5 Perubahan masing-masing bobot bias**

	V <sub>0j</sub>
V <sub>01</sub>	0,0002
V <sub>02</sub>	0,0005
V <sub>03</sub>	0,0005
V <sub>04</sub>	0,0009
V <sub>05</sub>	0,0005

#### Update perubahan bobot dan bias:

$$\begin{aligned}W_{jk}(\text{baru}) &= W_{jk}(\text{lama}) + \Delta W_{jk} \\ W_{11}(\text{baru}) &= 0,1 + 0,005 = 0,105 \\ W_{21}(\text{baru}) &= 0,3 + 0,008 = 0,308 \\ W_{31}(\text{baru}) &= 0,2 + 0,004 = 0,204 \\ W_{41}(\text{baru}) &= 0,4 + 0,007 = 0,407 \\ W_{51}(\text{baru}) &= 0,2 + 0,004 = 0,204 \\ V_{ij}(\text{baru}) &= V_{ij}(\text{lama}) + \Delta V_{ij}\end{aligned}$$

$$\begin{aligned}V_{11}(\text{baru}) &= 0,2 + 0,0002 = 0,2002 \\ V_{21}(\text{baru}) &= 0,4 + 0,0005 = 0,4005 \\ V_{31}(\text{baru}) &= 0,1 + 0,0005 = 0,1005 \\ V_{41}(\text{baru}) &= 0,1 + 0,0009 = 0,1009 \\ V_{51}(\text{baru}) &= 0,2 + 0,0005 = 0,2005\end{aligned}$$

### 3. KESIMPULAN

Adanya perubahan bobot pada tahapan pembobotan awal secara acak dimana perubahan tersebut terjadi sesuai dengan perhitungan dalam penerapan metode yang akan berubah kembali jika dilakukan tahap iterasi selanjutnya. Dan perubahan bobot akan berhenti ketika hasil dari iterasi tersebut sudah mencapai batas maksimal. Adapun pada perhitungan kearusasian yang telah diuji pada salah satu contoh tanda tangan mendapatkan hasil 80%. Dimana hasil tersebut dilakukan dengan data pengujian dan data benar sehingga menghasilkan kemungkinan bahwa tanda tangan tersebut asli.

### DAFTAR PUSTAKA

- Suyanto, 2011, *Artificial Intelligence*, Bandung : Informatika.
- D. Putra, *Pengolahan Citra Digital*, Yogyakarta: Penerbit ANDI, 2010.
- A D. Dongare, R. R. Kharde, Ana A. D. Kachare, "Introduction doArtificial Neural Network," *In. J. Eng. Innov. Technol.*, vol. 2, no. 1, Pp. 189-194, 2012.
- Sri Kusuma Dewi, Sri Hartati. (2016), *Neuro-Fuzzy*, Yogyakarta, Andi Offset.
- Puspitanigrum, D., 2006, *Pengantar Jaringan Syaraf Tiruan*, Yogyakarta : Andi.