



Hak Cipta Dilindungi Undang-Undang  
 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:  
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.  
 b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.  
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

© Hak cipta milik UIN Suska Riau. Ditulis oleh: Maicah n. Peridjo of Sultan Sa'rif Kasim Riau

## BAB II

### LANDASAN TEORI

#### 2.1 Citra digital

Citra adalah suatu representasi (gambaran), kemiripan atau imitasi dari suatu objek. Citra terbagi dua yaitu ada citra bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat kontinu seperti gambar pada monitor televisi, foto sinar-X, hasil dari CT Scan dan lain sebagainya. Sedangkan citra digital adalah citra yang dapat diolah oleh komputer (Sutoyo. T, Mulyanto. Edy, Suhartono. Vincent, Dwi Nurhayati Oky, 2009). Berikut adalah tipe citra digital berdasarkan nilai warna :

##### 2.1.1 Color Image

Pada color image atau RGB (Red, Green, Blue) masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (Red), hijau (Green) dan biru (Blue). Jika masing-masing warna memiliki range 0-255, maka totalnya adalah  $255^3 = 16.581.375$  (16 K) variasi warna berbeda pada gambar (Kusumanto, RD. dan Tompunu, 2011). Dimana variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat dilihat oleh manusia. Penglihatan manusia hanya dapat membedakan hingga 10 juta warna saja. Setiap poin informasi piksel (RGB) disimpan kedalam 1 byte data. 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai hijau pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah (Putra, 2010). Persamaan untuk melakukan konversi warna menjadi tipe RBG dapat dilihat pada Persamaan (2.1) (Santi, 2011).

$$\begin{aligned} R &= \text{warna} \wedge 0x000000 FF; \\ G &= (\text{warna} \wedge 0x0000 FF 00) \gg 8; \\ B &= (\text{warna} \wedge 0x00 FF 0000) \gg 16; \end{aligned} \quad (2.1)$$

Keterangan: R (*Red*), G (*Green*), B (*Blue*).



### 2.1.2 Black and White

Citra digital *black and white (grayscale)* yang hanya memiliki satu nilai kanan pada setiap piksel nya. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatkan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Citra *grayscale* berikut memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan) (Putra, 2010).

Metode yang digunakan dalam konversi citra ini yaitu *luminance*, metode ini banyak dipakai untuk pemrosesan citra karena didesain untuk menyamakan intensitas warna hijau dalam pandangan manusia. Persamaan yang digunakan untuk mendapatkan nilai *grayscale* dapat dilihat pada Persamaan (2.2) (Soleh, 2013):

$$I_{BW}(x,y) = (\text{Merah} \times 0.2126) + (\text{Hijau} \times 0.7152) + (\text{Biru} \times 0.0722) \quad (2.2)$$

Keterangan:

$$I_{BW}(x,y) = \text{nilai piksel } \textit{black and white} \text{ titik } (x,y).$$

### 2.1.3 Binary Image

Citra biner (Binary Image) hanya mempunyai dua nilai derajat keabuan: hitam dan putih. piksel-piksel objek bernilai 1 dan piksel-piksel latar belakang bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 atau 255 adalah hitam (Munir, 2004). Konversi dari citra hitam-putih ke citra biner dilakukan dengan operasi pengambangan (threshold). Operasi pengambangan mengelompokkan nilai derajat keabuan setiap piksel ke dalam 2 kelas, hitam dan putih. Binary image merupakan hasil pengolahan dari black and white image, dengan menggunakan fungsi Persamaan (2.3) atau (2.4) sebagai berikut (Kusumanto, RD. dan Tomponu, 2011):

$$I_{Bin}(x,y) = \begin{cases} 0 & I_{BW}(x,y) < T \\ 255 & I_{BW}(x,y) \geq T \end{cases} \quad (2.3)$$

atau dalam bentuk

$$I_{Bin}(x,y) = \begin{cases} 0 & I_{BW}(x,y) < T \\ 1 & I_{BW}(x,y) \geq T \end{cases} \quad (2.4)$$



Keterangan:

$I_{BW}(x,y)$  = nilai piksel *gray* titik  $(x,y)$ ,

$I_{Bin}(x,y)$  = nilai piksel *Binary* titik  $(x,y)$ ,

T adalah nilai ambang.

## 2.2 Pengolahan Citra Digital

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun Gambar Fergerak (yang berasal dari *webcam*). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra atau gambar dilakukan secara digital menggunakan komputer (Sutoyo. T, Mulyanto. Edy, Suhartono. Vincent, Dwi Nurhayati Oky, 2009). Pengolahan citra merupakan suatu cara sehingga menghasilkan citra yang sesuai dengan keinginan kita atau kualitasnya menjadi lebih baik (Bisri, 2013).

## 2.3 Pengenalan Pola

Pola adalah entitas yang terdefinisi dan dapat diidentifikasi melalui ciri-cirinya (*features*). Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi. Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu objek dengan objek lain (Munir, 2004).

## 2.4 Preprocessing

Preprocessing merupakan proses awal yang dilakukan untuk memperbaiki kualitas citra dengan menggunakan teknik-teknik pengolahan citra sebelum proses *feature extraction* (Munir, 2004). Proses awal yang dilakukan adalah *cropping* citra yang bertujuan untuk mengambil citra FKP saja dengan ukuran 300 x 300 piksel, selanjutnya dilakukan *resize* atau pengurangan ukuran citra menjadi 40 piksel x 40 piksel, kemudian konversi warna citra menjadi RGB kemudian dikonversi lagi menjadi citra *grayscale* lalu dikonversi lagi menjadi citra biner



dengan menggunakan algoritma pengambangan (*threshold*), kemudian yang terakhir adalah mengubah citra menjadi matriks 1 x (baris x kolom).

## 2.5 *Principal Component Analysis (PCA)*

*Feature extraction* atau ekstraksi ciri adalah proses mengambil ciri-ciri yang terdapat pada objek di dalam citra. Lalu menghitung properti-properti objek yang berkaitan sebagai ciri. Beberapa proses ekstraksi ciri mungkin perlu mengubah citra masukan sebagai citra biner, melakukan penipisan pola, dan sebagainya (Munir, 2004). Salah satu algoritma ekstraksi ciri adalah *Principal Component Analysis (PCA)*. PCA telah banyak digunakan dalam berbagai bidang seperti pengolahan citra, pengenalan pola, kompresi data, data mining, *machine learning* dan *computer vision* (Song, F., Guo, Z., dan Mei, 2010).

PCA adalah suatu teknik reduksi dinamis dan ekstraksi fitur yang optimal jika dipandang dari sudut pandang teori informasi (Muntasa, A., Sirajudin, I. A., dan Purnomo, 2011). Dalam tahap ekstraksi ciri menggunakan PCA pada penelitian ini terbagi menjadi dua bagian yaitu ekstraksi ciri citra data latih dan ekstraksi ciri citra data uji. Berikut adalah Langkah-langkah algoritma PCA untuk ekstraksi ciri citra data latih: (Syakhala, 2015):

1. Langkah pertama sebelum melakukan pencarian nilai *eigen* terlebih dahulu perlu melakukan perhitungan nilai rata-rata dari sebuah citra ( $\Psi$ ) dengan Persamaan (2.5).

$$\Psi = \frac{\sum_{i=1}^M \Gamma_i}{M} \quad (2.5)$$

Keterangan:

$\Gamma_i$  adalah nilai pada setiap piksel.

M adalah Jumlah data. Penghitungan nilai rata-rata untuk mengurangi dimensi yang akan di hitung pada proses selanjutnya.

2. Langkah kedua menghitung matriks normalisasi ( $\Phi$ ) dengan Persamaan (2.6) berikut:

$$\Phi_i = \Gamma_i - \Psi \quad (2.6)$$



Keterangan:

Masing-masing nilai pada setiap piksel ( $\Gamma_i$ ) dikurangkan dengan rata-ratanya ( $\Psi$ ).

3. Langkah ketiga adalah proses mencari nilai dari matriks kovarian citra dengan Persamaan (2.7) berikut:

$$C = \Phi \times \Phi^T \quad (2.7)$$

Keterangan:

C adalah matriks kovarian.

$\Phi^T$  adalah matrik *transpos* dari matriks selisih.

$\Phi$  adalah matriks selisih. Tujuan pencarian matriks C ini adalah untuk mempermudah pencarian nilai *eigen* dan vector *eigen*.

4. Mencari nilai *eigen* dan vektor *eigen* Untuk mencari nilai *eigen* pada matriks dengan menggunakan Persamaan (2.8).

$$Cx = \lambda x \quad (2.8)$$

Skalar  $\lambda$  disebut nilai *eigen* dari C dan x disebut vektor *eigen* dari C yang bersesuaian dengan  $\lambda$ . Setelah diperoleh nilai *eigen*  $\lambda$ , maka masukan nilai-nilai *eigen* ke dalam Persamaan (2.9).

$$(C - \lambda I)x = 0 \quad (2.9)$$

5. Langkah ke-4 mengurutkan nilai *eigen value* ( $\lambda$ ) dan *eigen vektor* (v) dari besar ke kecil berdasarkan urutan nilai *eigen*. Kemudian hitung nilai matriks *eigenfaces* dengan Persamaan (2.10).

$$Eigenfaces = Eigen\ vector \times \Phi \quad (2.10)$$

6. Langkah kelima hitung *Project Image* dari citra dengan Persamaan (2.11) berikut:

$$Project\ image = \Phi \times eigenfaces^T \quad (2.11)$$

Setelah perhitungan algoritma PCA selesai dan nilai *Project Image* telah diperoleh, maka langkah selanjutnya mereduksi matriks *Project Image* dengan variabel k yang dapat disesuaikan dengan kebutuhan. Maksud dari reduksi dengan variabel k adalah membuang segian kolom sebelah kiri dan hanya mengambil beberapa kolom dari kolom matriks yang paling kanan



dari matriks 2 dimensi tanpa mempengaruhi baris pada matriks. Kemudian setelah matriks *Project Image* direduksi memasukkan nilai *Project Image* yang berupa matriks dua dimensi tersebut ke dalam proses algoritma BPNN untuk dilakukan klasifikasi.

Setelah melakukan tahapan untuk melakukan ekstraksi ciri citra data latih, langkah selanjutnya adalah ekstraksi ciri citra data uji. Berikut adalah Langkah-langkah algoritma PCA untuk ekstraksi ciri citra data uji: (Syakhala, 2015):

#### 1. Menghitung Matriks Normalisasi dari Citra Data Uji

Pada tahap ini membutuhkan nilai mean dari citra data latih dan nilai biner dari citra data uji yang akan digunakan untuk mencari matriks normalisasi atau matriks selisih. Berikut adalah Persamaan (2.12) yang dapat digunakan untuk menghitung matriks normalisasi:

$$\Phi_i = \Gamma_i - \Psi \quad (2.12)$$

Keterangan:

Masing-masing nilai pada setiap piksel ( $\Gamma_i$ ) dari citra data uji kemudian dikurangkan dengan rata-ratanya ( $\Psi$ ) yang merupakan hasil dari proses ekstraksi ciri citra data latih.

#### 2. Menghitung Nilai *Project Image* dari Citra Data Uji

Langkah selanjutnya adalah menghitung nilai *Project Image* dari citra data uji dengan mengalikan nilai *eigenfaces* yang telah didapatkan dari proses ekstraksi citra data latih dengan matriks normalisasi dari citra data uji. Nilai *Project Image* dapat dihitung dengan menggunakan Persamaan (2.13) berikut:

$$Project\ image = \Phi \times eigenfaces^T \quad (2.13)$$

Setelah perhitungan algoritma PCA selesai dan nilai *Project Image* telah diperoleh, maka langkah selanjutnya mereduksi matriks *Project Image* dengan variabel k yang dapat disesuaikan dengan kebutuhan. Kemudian setelah matriks *Project Image* direduksi memasukkan nilai *Project Image* yang berupa matriks dua dimensi tersebut ke dalam proses algoritma BPNN untuk dilakukan identifikasi.



## 2.6 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) atau yang disebut *Artificial Neural Network* merupakan salah satu bagian dari kecerdasan buatan (*Artificial Intelligence*). JST adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi (Siang, 2005). JST ditentukan oleh 3 hal (Siang, 2005):

### 1. Arsitektur Jaringan Syaraf Tiruan

Merupakan pola keterhubungan antara neuron. Hubungan neuron-neuron inilah yang membentuk suatu jaringan. Arsitektur dalam JST terdiri dari beberapa macam jaringan sebagai berikut (Siang, 2005):

#### a. Jaringan Layar Tunggal (*Single Layer Network*)

Dalam jaringan ini, sekumpulan input neuron dihubungkan langsung dengan sekumpulan output. Pada beberapa model (misal perceptron), hanya ada sebuah unit neuron output.

#### b. Jaringan Layar Jamak (*Multi Layer Network*)

Jaringan layar jamak merupakan perluasan dari layar tunggal. Pada jaringan ini, selain unit input dan output, ada unit-unit lain (sering disebut layer tersembunyi).

#### c. Jaringan *Reccurent*

Model jaringan recurrent mirip dengan jaringan layer tunggal ataupun ganda. Hanya saja, ada neuron ouput yang memberikan sinyal pada unit input (sering disebut *feedback loop*).

### 2. Algoritma Jaringan

Merupakan metode untuk menentukan nilai bobot hubungan. Ada dua metode pada algoritma jaringan saraf tiruan, yaitu metode bagaimana JST tersebut melakukan pelatihan (pembelajaran) dan, metode bagaimana JST tersebut melakukan pengenalan (aplikasi). Beberapa contoh algoritma JST seperti perceptron, BPNN, LVQ, dan lainnya (Puspitaningrum, 2006).

### 3. Fungsi Aktivasi

Merupakan fungsi untuk menentukan nilai keluaran berdasarkan nilai total masukan pada neuron. Fungsi aktivasi suatu algoritma jaringan dapat berbeda dengan fungsi aktivasi algoritma jaringan lain. Beberapa fungsi



aktivasi yang sering dipakai seperti fungsi undak biner, fungsi bipolar, fungsi linier, dan lain-lain (Puspitaningrum, 2006).

## 2.7 Normalisasi

Tahap normalisasi data sangat diperlukan sebelum melakukan proses pelatihan BPNN agar nilai input dan target tersebut berada dalam range 0.1 sampai dengan 0.9. Normalisasi data dapat dihitung dengan menggunakan Persamaan (2.14).

$$x' = (0.8(x - a) / (b - a)) + 0.1 \quad (2.14)$$

Keterangan:

$x'$  : Hasil normalisasi data latih.

$x$  : Nilai data latih yang akan dinormalisasi.

$a$  : Nilai minimal dari semua data latih yang akan dinormalisasi.

$b$  : Nilai maksimal dari semua data latih yang akan dinormalisasi.

## 2.8 Backpropagation Neural Network (BPNN)

Tujuan utama dari algoritma pembelajaran JST adalah melakukan pengaturan terhadap bobot-bobot yang ada pada jaringan syaraf, sehingga diperoleh bobot akhir yang tepat sesuai dengan pola data yang dilatih. Pada dasarnya ada dua metode pembelajaran yaitu metode pembelajaran terawasi dan pembelajaran tidak terawasi. Salah satu dari algoritma pembelajaran terawasi adalah *Backpropagation Neural Network* (BPNN). BPNN adalah salah satu algoritma yang paling banyak digunakan dan suatu teknik yang populer untuk mengoptimalkan umpan maju pelatihan jaringan saraf (Nawi, 2013). BPNN menggunakan pelatihan jenis terkontrol (terawasi) dimana menggunakan pola penyesuaian bobot untuk mencapai nilai kesalahan yang minimum antara keluaran hasil prediksi dengan keluaran yang nyata (Andrijasa.M.F dan Mistianingsih, 2010).



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

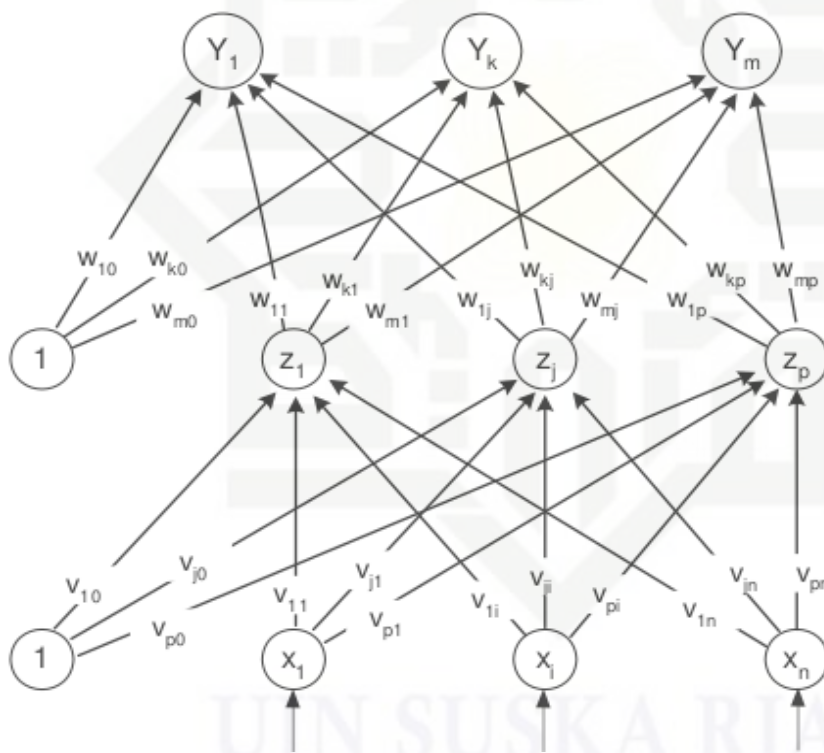
- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

© Hak cipta milik UIN Suska Riau  
State Islamic University of Sultan Syarif Kasim Riau

### 2.8.1 Arsitektur Jaringan BPNN

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi. Gambar 2.4 pada halaman berikutnya adalah arsitektur backpropagation dengan  $n$  buah masukan (ditambah sebuah bias), sebuah layer tersembunyi yang terdiri dari  $p$  unit (ditambah sebuah bias), serta  $m$  buah unit keluaran.  $V_{ji}$  merupakan bobot garis dari unit masukan  $X_i$  ke unit layar tersembunyi  $Z_j$  ( $V_{j0}$  merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi  $Z_j$ ).  $W_{kj}$  merupakan bobot dari unit layar tersembunyi  $Z_j$  ke unit keluaran  $Y_k$  ( $W_{k0}$  merupakan bobot dari bias di layar tersembunyi ke unit keluaran  $Z_k$ ).



Gambar 2.1 Arsitektur Jaringan Backpropagation (Siang, 2005)



### 2.8.2 Algoritma BPNN

Pada penelitian ini Algoritma BPNN memiliki dua peranan yaitu melatih citra data latih dan melakukan identifikasi citra data uji. Pelatihan Backpropagation meliputi 3 fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi. Algoritma pelatihan untuk jaringan dengan satu layar tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut:

1. Normalisasi Data latih dan Inisialisasi semua bobot dengan bilangan acak kecil dalam range -0.5 sampai 0.5 dan inisialisasi maksimum epoch, target error, learning rate dan MSE. Normalisasi data latih dengan menggunakan Persamaan (2.14).
2. Jika kondisi penghentian belum terpenuhi, lakukan langkah 3-10.
3. Untuk setiap pasang data pelatihan, lakukan langkah 4-9.

#### Fase I : Propagasi maju

4. Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.
5. Hitung semua keluaran di unit tersembunyi  $Z_j$  ( $j = 1, 2, \dots, p$ ) dengan menggunakan Persamaan (2.15) dan (2.16).

$$Z_{net_j} = V_{j0} + \sum_{i=1}^n X_i V_{ji} \quad (2.15)$$

Keterangan:

$Z_{net_j}$  : Nilai operasi pada *hidden layer*.

$V_{j0}$  : Nilai bobot bias dari *input layer*.

$X_j$  : Nilai neuron pada *input layer*.

$V_{ji}$  : Nilai bobot.



Hak Cipta Dilindungi Undang-Undang  
 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:  
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.  
 b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.  
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

© Hak cipta milik UIN Suska Riau

State Islamic University of Sultan Syarif Kasim Riau

$$Z_j = f(Z_{net_j}) = \frac{1}{1 + e^{-Z_{net_j}}} \quad (2.16)$$

Keterangan:

$Z_{net_j}$  : Nilai operasi pada *hidden layer*.

$Z_j$  : Nilai keluaran dari *hidden layer*.

$f$  : Fungsi aktivasi.

6. Hitung semua keluaran jaringan di unit  $y_k$  ( $k = 1, 2, \dots, m$ ) dengan menggunakan Persamaan (2.17) dan (2.18).

$$y_{net_k} = W_{k0} + \sum_{i=1}^p Z_j W_{kj} \quad (2.17)$$

Keterangan:

$y_{net_k}$  : Nilai operasi pada *output layer*.

$W_{k0}$  : Nilai bobot bias dari *hidden layer*.

$Z_j$  : Nilai neuron dari *hidden layer*.

$W_{kj}$  : Nilai bobot dari *hidden layer*.

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (2.18)$$

Keterangan:

$y_{net_k}$  : Nilai operasi pada *output layer*.

$y_k$  : Nilai keluaran dari *output layer*.

$f$  : Fungsi aktivasi dari *hidden layer*.

### Fase II : Propagasi mundur

7. Hitung faktor  $\delta$  unit keluaran berdasarkan kesalahan di setiap unit keluaran  $y_k$  ( $k = 1, 2, \dots, m$ )

Hitung faktor  $\delta$  unit keluaran dengan menggunakan Persamaan (2.19) berikut:

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (2.19)$$

$\delta_k$  merupakan unit kesalahan yang akan digunakan dalam perubahan bobot layar di bawahnya (langkah 8).



Hitung suku perubahan bobot  $W_{kj}$  (yang nanti akan digunakan untuk merubah bobot  $W_{kj}$ ) dengan laju percepatan atau *learning rate*( $\alpha$ ) dengan menggunakan Persamaan (2.20).

$$\Delta w_{kj} = \alpha \delta_k z_j; k=1,2,\dots,m; j=0,1,\dots,p \quad (2.20)$$

8. Hitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi  $Z_j$  ( $j = 1, 2, \dots, p$ )

Hitung faktor  $\delta$  unit tersembunyi dengan menggunakan Persamaan (2.21) berikut:

$$\delta_{net_j} = \sum_{k=1}^m \delta_k W_{kj} \quad (2.21)$$

Hitung fungsi aktivasi dari Faktor  $\delta$  unit tersembunyi dengan menggunakan Persamaan (2.22) berikut:

$$\delta_j = \delta_{net_j} f'(Z_{net_j}) = \delta_{net_j} Z_j(1 - Z_j) \quad (2.22)$$

Hitung suku perubahan bobot  $V_{ji}$  (yang akan digunakan untuk merubah bobot  $V_{ji}$ ) dengan menggunakan Persamaan (2.23) berikut:

$$\Delta v_{ji} = \alpha \delta_j x_i; j=1,2,\dots,p; i=0,1,\dots,n \quad (2.23)$$

### Fase III : Perubahan Bobot

9. Hitung semua perubahan bobot

Hitung semua perubahan bobot garis yang menuju ke unit keluaran dengan Persamaan (2.24) berikut:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (k=1,2,\dots,m; j=0,1,\dots,p) \quad (2.24)$$

Perubahan bobot garis yang menuju ke unit tersembunyi dengan menggunakan Persamaan (2.25) berikut:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (j=1,2,\dots,p; i=0,1,\dots,n) \quad (2.25)$$

10. Tes kondisi berhenti apabila error ditemukan jika kondisi *stop* telah terpenuhi, maka pelatihan jaringan dapat dihentikan. Untuk memeriksa kondisi *stop*, biasanya dengan menghitung kriteria MSE (*Mean Square Error*) dengan menggunakan Persamaan (2.26) berikut ini:

$$MSE = 1/n \times ((t_{k1} - y_{k1})^2 + \dots + (t_{km} - y_{km})^2) \quad (2.26)$$



Hak Cipta Dilindungi Undang-Undang  
 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:  
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.  
 b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.  
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

© Hakipta milik UIN Suska Riau  
 State Islamic University of Sultan Syarif Kasim Riau

Pengujian BPNN dilakukan melalui *feed forward* langkah-langkahnya sebagai berikut:

1. Inisialisasikan bobot (dari hasil pelatihan) dan normalisasi data uji.
2. Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.
3. Hitung semua keluaran di unit tersembunyi  $Z_j$  ( $j = 1, 2, \dots, p$ ). dengan menggunakan Persamaan (2.27) dan (2.28).

$$Z_{net_j} = V_{j0} + \sum_{i=1}^n X_i V_{ji} \tag{2.27}$$

$$Z_j = f(Z_{net_j}) = \frac{1}{1 + e^{-Z_{net_j}}} \tag{2.28}$$

4. Hitung semua keluaran jaringan di unit  $y_k$  ( $k = 1, 2, \dots, m$ ) dengan menggunakan Persamaan (2.29) dan (2.30).

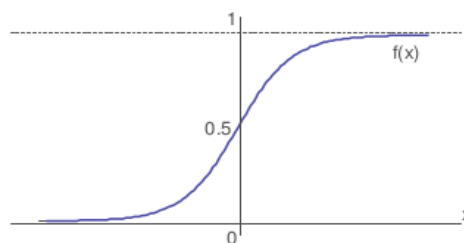
$$y_{net_k} = W_{k0} + \sum_{j=1}^p Z_j W_{kj} \tag{2.29}$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \tag{2.30}$$

### 2.8.3 Fungsi Aktivasi

Fungsi aktivasi yang digunakan dalam BPNN harus memenuhi beberapa syarat yaitu kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering digunakan adalah fungsi *sigmoid* biner yang memiliki range (0,1) yang dituliskan dalam Persamaan (2.31) dan dapat dilihat pada Gambar 2.2 berikut:

$$f(x) = \frac{1}{1 + e^{-x}} \text{ dengan turunan } f'(x) = f(x)(1 - f(x)) \tag{2.31}$$



Gambar 2.2 Sigmoid Biner (Siang, 2005)

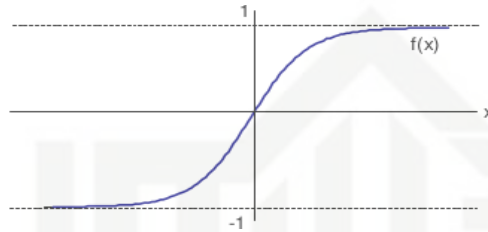


Hak Cipta Dilindungi Undang-Undang  
 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:  
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.  
 b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.  
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

© Hak Cipta milik UIN Suska Riau  
 Scientific University of Sultan Syarif Kasim Riau

Fungsi lain yang sering digunakan adalah fungsi sigmoid bipolar yang bentuk fungsinya mirip dengan fungsi sigmoid biner, tapi dengan range (-1, 1) seperti terlihat pada Persamaan (2.32) dan Gambar 2.3.

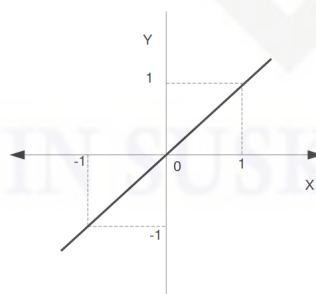
$$f(x) = \frac{2}{1 + e^{-x}} - 1 \text{ dengan turunan } f'(x) = \frac{(1+f(x))(1-f(x))}{2} \tag{2.32}$$



**Gambar 2.3 Sigmoid Bipolar (siang, 2005)**

Fungsi sigmoid memiliki nilai maksimum = 1. Maka untuk pola yang targetnya > 1, pola masukan dan keluaran harus terlebih dahulu ditransformasi sehingga semua polanya memiliki range yang sama seperti fungsi sigmoid yang digunakan. Alternatif lain adalah menggunakan fungsi aktivasi sigmoid hanya pada layar yang bukan layar keluaran. Pada layar keluaran, fungsi aktivasi yang digunakan adalah fungsi identitas :  $f(x) = x$ .

Pada beberapa kondisi fungsi aktivasi lain yang dapat digunakan adalah fungsi Identitas atau linear seperti terlihat pada Gambar 2.4. Digunakan jika keluaran yang dihasilkan oleh jaringan syaraf tiruan merupakan sembarang bilangan riil (bukan hanya pada range [0,1] atau [1, - 1])  $Y = X$ .



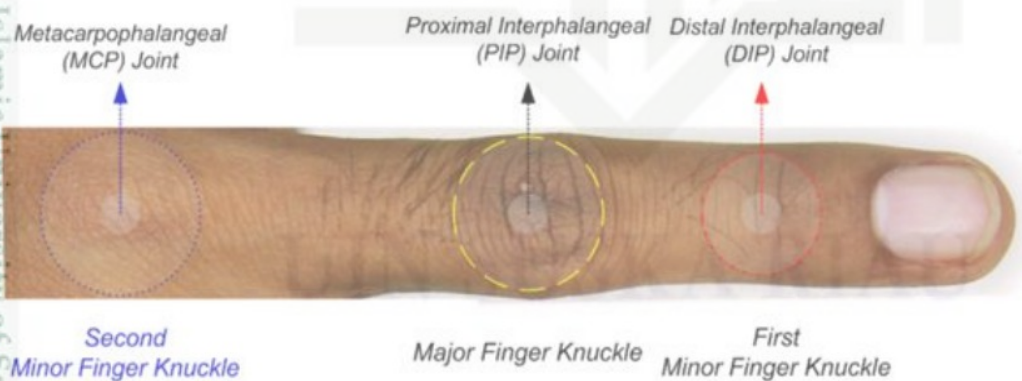
**Gambar 2.4 Fungsi Aktivasi Identitas (Agustin dan Prahasto, 2012)**

## Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

## 2.9 Finger Knuckle Print (FKP)

*Finger knuckle Print* merupakan biometrik yang masih relatif baru diteliti dan digunakan untuk sistem pengenalan diri. Sebenarnya, fitur garis permukaan luar dari sendi jari lebih jelas dari permukaan telapak tangan, sementara permukaan luar dari sendi jari memiliki wilayah jauh lebih kecil dari pada permukaan telapak tangan (Kadir, A., & Susanto, 2013). Hasil penelitian (Kadir, A., & Susanto, 2013) menunjukkan bahwa sistem otentikasi FKP diusulkan dapat memverifikasi identitas pribadi secara real time dengan tingkat akurasi tinggi. Dibandingkan dengan sidik jari, FKP sulit untuk terkelupas karena orang memegang barang dengan bagian dalam tangan (Smith, 2002). Posisi FKP yang sedikit menekuk ketika sedang dicitrakan dalam sistem yang diusulkan. hal ini dapat membuat pola FKP ditangkap dengan jelas dan fitur unik dari FKP dapat lebih baik dimanfaatkan (Hana, 2013). Tingkat kesalahan atau *equal error rate* dari pencocokan *Minor Finger knuckle* adalah 1,04% dan pencocokan *Major Finger knuckle* adalah 0,22% (Kumar, 2012). Berdasarkan tingkat kesalahan dari pencocokan beberapa bagian FKP, maka dalam penelitian ini menggunakan bagian *Major Finger knuckle* dengan tingkat kesalahan yang lebih rendah. *knuckle* dapat terlihat jelas seperti pada Gambar 2.5.



Gambar 2.5 *Finger Knuckle* (Kaur, Supreet dan Singla, 2016)



## 2.10 Confusion Matrix

*Confusion matrix* adalah sebuah tabel yang menyatakan jumlah data uji yang benar diidentifikasi dan jumlah data uji yang salah diidentifikasi (Indriani, 2014). Contoh *confusion matrix* klasifikasi biner ditunjukkan pada Tabel 2.1.

**Tabel 2.1 Confusion Matrix (Indriani, 2014)**

		Kelas Prediksi	
		True	False
Kelas Sebenarnya	True	TP	FN
	False	FP	TN

Perhitungan akurasi dinyatakan dalam Persamaan (2.29):

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \times 100\% \quad (2.33)$$

**Keterangan dari Tabel 2.1 dinyatakan sebagai berikut (Indriani, 2014):**

- *True Positive* (TP), yaitu jumlah dokumen dari kelas *true* yang benar dan diklasifikasikan sebagai kelas *true*.
- *True Negative* (TN), yaitu jumlah dokumen dari kelas *false* yang benar diklasifikasikan sebagai kelas *false*.
- *False Positive* (FP), yaitu jumlah dokumen dari kelas *false* yang salah diklasifikasikan sebagai kelas *true*.
- *False Negative* (FN), yaitu jumlah dokumen dari kelas *true* yang salah diklasifikasikan sebagai kelas *false*.

## 2.11 Penelitian Terkait

Penelitian yang pernah dilakukan tentang identifikasi *Finger knuckle* dengan menggunakan atau menerapkan PCA maupun BPNN. Penelitian terkait dapat dilihat pada Tabel 2.2.





**Tabel 2.2 Penelitian Terkait**

Peneliti	Judul	Metode	Tahun	Akurasi
(Devi, S. Suganthi dan Suhasini, 2016)	<i>Implimentation of FKP Based Biometric Identification System Using PCA with Neuro Fuzzy Neural Network</i>	<i>Principal Component Analysis, Neuro Fuzzy Based Neural Network</i>	2016	98,5%
(Gajbhiye dan Arora, 2016)	<i>ANN Classifier for Finger knuckle Print Recognition Using Gabor Feature</i>	<i>Artificial Neural Network (ANN) Gabor Feature</i>	2016	95%
(Hana, 2013)	Sistem Identifikasi Biometrik FKP Menggunakan Histogram Equalization dan Principal Component Analysis (PCA)	<i>Histogram Equalization dan Principal Component Analysis (PCA)</i>	2013	94%
(Abulkashem, M., Akhter, M., Ahmed, 2011)	<i>Face Recognition System Based on Principal Component Analysis (PCA) with Backpropagation</i>	<i>Principal Component Analysis (PCA), Backpropagation Neural Networks</i>	2011	96,5%
(Lestari, 2013)	Sistem Deteksi Pola Sepuluh Sidik Jari Seseorang dengan Menggunakan Pengolahan Citra dan Jaringan Syaraf Tiruan	<i>Mathematical Morphology, Dilasion, Erosion, Back Propagation Algorithm, Neural Network</i>	2013	93,33%
(Arifin, 2011)	Pengenalan Pola Sidik Jari Menggunakan JST dengan Metode Pembelajaran Backpropagation	<i>Backpropagation</i>	2011	93,12%

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

(Indah, 2009)	Identifikasi Sidik Jari Menggunakan Pendekatan Jaringan Syaraf Buatan dengan Metode Propagasi Balik	<i>Backpropagation Neural Networks</i>	2009	99,99%
---------------	---	--	------	--------

© Hak cipta milik UIN Suska Riau

State Islamic University of Sultan Syarif Kasim Riau

**Hak Cipta Dilindungi Undang-Undang**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

