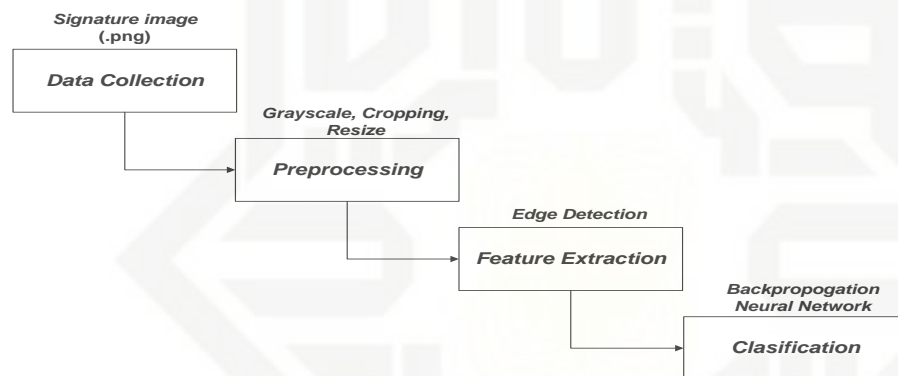


## BAB IV

### ANALISA DAN PERANCANGAN

Pada tahapan analisa dan perancangan hal yang dilakukan adalah membahas, memahami dan mengkaji secara rinci dan spesifik mengenai apa saja yang menjadi pokok permasalahan dalam penelitian agar memperoleh pemahaman yang tepat akan masalah tersebut. Sehingga diperoleh pemahaman yang tepat mengenai masalah, data, proses, hasil dan semua hal yang berkaitan dengan penelitian. Hal tersebut dapat dilihat pada Gambar 4.1 berikut ini :



**Gambar 4.1 Tahapan Identifikasi Tanda Tangan dengan Metode Backpropagation Neural Network (BPNN)**

#### 4.1 Analisis Identifikasi Tanda Tangan

Pada penelitian ini dilakukan analisis tahapan atau proses yang dilakukan agar aplikasi dapat melakukan identifikasi terhadap tanda tangan. Analisis proses yang dilakukan adalah *data collection*, *pre-processing*, *processing* dan *clasification*.

##### 4.1.1 Analisis Data Collection

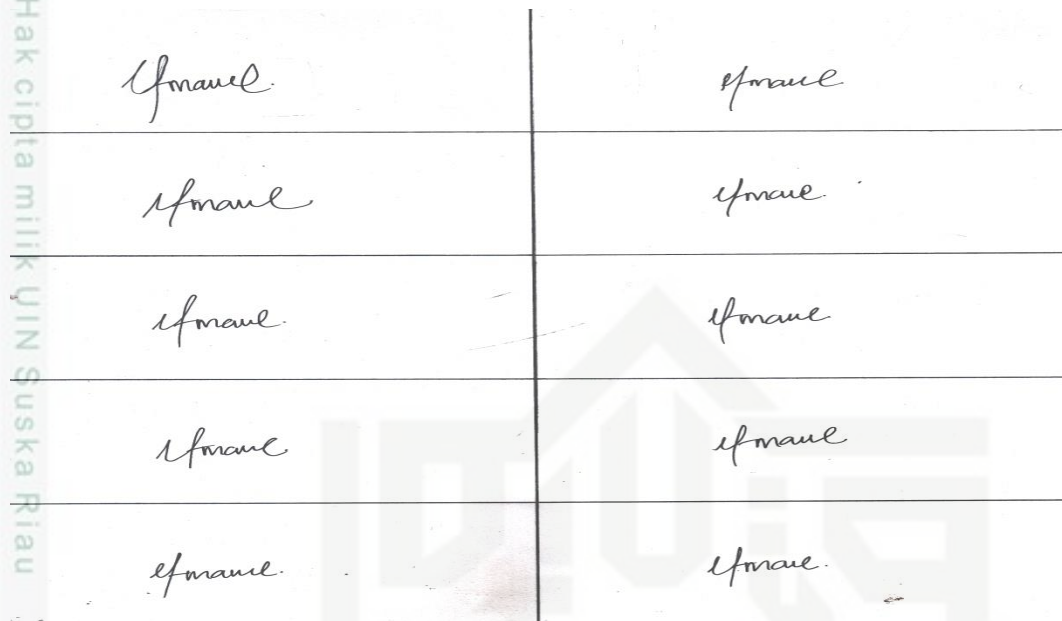
Proses *Data Collection* merupakan proses yang dilakukan untuk memasukkan citra tanda tangan. Citra tanda tangan tersebut diambil dengan menggunakan *scanner* dengan format *.png* . Data citra tanda tangan yang disimpan berjumlah 100 data yang diperoleh dari 10 orang masing-masing dengan 10 pola tanda tangan per orangnya. Gambar 4.2 merupakan 10 contoh citra tanda tangan yang akan diproses :

**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.



**Gambar 4.2 Citra Tanda Tangan Asli Untuk Proses Identifikasi Tanda Tangan**

Kemudian dilakukan pembagian data dari 100 data citra tersebut menjadi :

- 70 data latih dan 30 data uji
- 80 data latih dan 20 data uji
- 90 data latih dan 10 data uji

Pada metode BPNN target atau kelas yang diinginkan sudah ditetapkan terlebih dahulu. Nilai Y0, Y1, Y2, Y3 merupakan nilai output dari arsitektur BPNN. Kelas pada penelitian ini dapat dilihat pada Tabel 4.1

**Tabel 4.1 Kelas Target Citra Tanda Tangan**

No	Nama Pemilik Tanda Tangan	Y0	Y1	Y2	Y3	Kelas
1	Agnes	0	0	0	1	1
2	Dina	0	0	1	0	2
3	Eka	0	0	1	1	3
4	Elri	0	1	0	0	4
5	Ira	0	1	0	1	5
6	Irabiah	0	1	1	0	6

#### Hak Cipta Dilindungi Undang-Undang

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

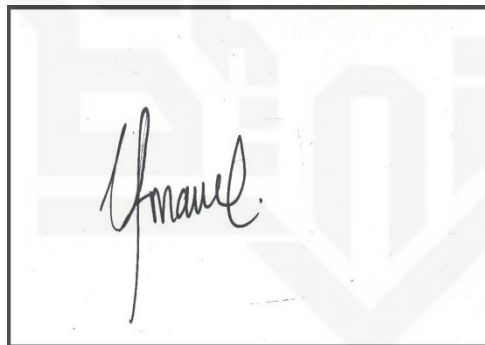
- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

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

7	Irma	0	1	1	1	7
8	Rahmayeni	1	0	0	0	8
9	Soviana	1	0	0	1	9
10	Ratika	1	0	1	0	10

### 4.1.2 Analisis *Pre-processing*


Setelah diperoleh data citra tanda tangan maka tahapan selanjutnya yang dilakukan adalah tahap *pre-processing*. Tahap ini berguna untuk menjadikan suatu citra menjadi memiliki nilai untuk kemudian ke tahap *feature extraction*. Pada tahap ini juga dapat memisahkan objek dari area latar belakang atau *background*. Tahapan yang dilakukan pada *pre-processing* ini adalah dengan melakukan perubahan ciri warna yaitu dengan mengkonversi citra yang bernilai RGB ke *grayscale image*, kemudian dilakukan *cropping*, dan setelah itu gambar akan di *resize* lagi menjadi 100 x 100 pixel. Gambar 4.3 dibawah ini merupakan contoh gambar yang belum melewati tahap *pre-processing* :



**Gambar 4.3 Citra Tanda Tangan Asli Sebelum Tahap *Preprocessing***

#### 4.1.2.1 Konversi RGB ke *Grayscale*

Citra tanda tangan yang diperoleh dari hasil scan memiliki ciri warna RGB. Oleh karena itu citra tanda tangan tersebut harus dikonversi terlebih dahulu ke *grayscale image*. Gambar 4.4 berikut merupakan contoh citra tanda tangan dengan format RGB :



**Gambar 4.4 Citra Tanda Tangan Format RGB**

Nilai dari R, G, B dari masing masing piksel tersebut berbeda-beda. Dapat dilihat pada Gambar 4.5 dibawah ini yang merupakan nilai piksel pada gambar 100 x 100 piksel.

R:193 G:194 B:194	R:179 G:179 B:179	R: 89 G: 91 B: 91	R: 20 G: 22 B: 22	R:163 G:163 B:163	R:255 G:255 B:255	R:255 G:255 B:255	R:235 G:235 B:235	R: 72 G: 73 B: 73	....
R: 22 G: 23 B: 23	R: 6 G: 7 B: 7	R: 69 G: 71 B: 71	R:118 G:119 B:119	R: 45 G: 47 B: 47	R:242 G:242 B:242	R:255 G:255 B:255	R:194 G:195 B:195	R: 80 G: 81 B: 81	....
R:130 G:130 B:130	R: 0 G: 0 B: 0	R: 95 G: 97 B: 97	R:237 G:238 B:238	R: 37 G: 39 B: 39	R:175 G:176 B:176	R:255 G:255 B:255	R:165 G:166 B:166	R:112 G:113 B:113	....
R:196 G:197 B:197	R: 8 G: 9 B: 9	R:109 G:110 B:110	R:255 G:255 B:255	R: 90 G: 91 B: 91	R:114 G:115 B:115	R:255 G:255 B:255	R:142 G:143 B:143	R:140 G:141 B:141	....
R:212 G:213 B:213	R: 20 G: 22 B: 22	R:113 G:114 B:114	R:255 G:255 B:255	R:149 G:149 B:149	R: 74 G: 76 B: 76	R:255 G:255 B:255	R:107 G:108 B:108	R:163 G:165 B:165	....
R:221 G:221 B:221	R: 22 G: 23 B: 23	R:113 G:114 B:114	R:255 G:255 B:255	R:190 G:190 B:190	R: 55 G: 57 B: 57	R:234 G:234 B:234	R: 81 G: 83 B: 83	R:190 G:190 B:190	....
R:233 G:233 B:233	R: 35 G: 37 B: 37	R:110 G:111 B:111	R:255 G:255 B:255	R:214 G:214 B:214	R: 44 G: 45 B: 45	R:196 G:197 B:197	R: 73 G: 75 B: 75	R:213 G:214 B:214	....
R:237 G:238 B:238	R: 43 G: 45 B: 45	R:103 G:104 B:104	R:255 G:255 B:255	R:235 G:236 B:236	R: 51 G: 52 B: 52	R:148 G:149 B:149	R: 68 G: 69 B: 69	R:233 G:234 B:234	....
R:249 G:250 B:250	R: 57 G: 58 B: 58	R: 92 G: 93 B: 93	R:255 G:255 B:255	R:253 G:253 B:253	R: 78 G: 79 B: 79	R: 86 G: 87 B: 87	R: 78 G: 79 B: 79	R:249 G:249 B:249	....
....	....	....	....	....	....	....	....	....	$F(x,y) = f(100,100)$

**Gambar 4.5 Piksel dari Citra Tanda Tangan Format RGB**

Dengan menggunakan persamaan(2.1)yaitu rumus konversi RGB ke grayscale, maka akan dilakukan perhitungan sebagai berikut :

$$\begin{aligned}
 \text{Grayscale Piksel (1,1)} &= (0,2989 * 193) + (0,5870 * 194) + (0,1141 * 194) \\
 &= 57,6877 + 113,878 + 22,1354
 \end{aligned}$$



$$= 193,7011$$

$$\text{Grayscale Piksel (1,2)} = (0,2989 * 179) + (0,5870 * 179) + (0,1141 * 179)$$

$$= 53,5031 + 105,073 + 20,4239$$

$$= 179$$

$$\text{Grayscale Piksel (1,3)} = (0,2989 * 89) + (0,5870 * 91) + (0,1141 * 91)$$

$$= 26,6021 + 53,417 + 10,3831$$

$$= 90,4022$$

Demikian seterusnya rumus tersebut dikerjakan sampai Grayscale piksel (100 x 100). Hasil dari perhitungan konversi RGB ke *grayscale* yang telah dilakukan, maka terlihat perbedaan nilai dari masing-masing piksel. Nilai hasil konversi RGB ke *grayscale* tersebut dapat dilihat pada Tabel 4.2 berikut.

**Tabel 4.2 Nilai Hasil Konversi RGB ke Grayscale**

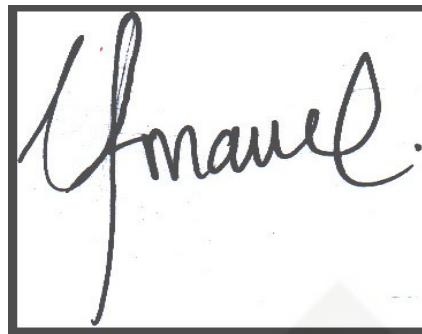
193,7011	179	90,4022	21,4022	163	255	255	235	72,7011	.....
22,7011	6,7011	70,4022	118,7011	46,4022	242	255	194,7011	80,7011	.....
130	0	96,4022	237,7011	38,4022	175,7011	255	165,7011	112,7011	.....
196,7011	8,7011	109,7011	255	90,7011	114,7011	255	142,7011	140,7011	.....
212,7011	21,4022	113,7011	255	149	75,4022	255	107,7011	164,4022	.....
221	22,7011	113,7011	255	190	56,4022	234	82,4022	190	.....
233	36,4022	110,7011	255	214	44,7011	176,3913	74,4022	213,7011	.....
237,7011	44,4022	103,7011	255	235,7011	51,7011	148,7011	68,7011	233,7011	.....
249,7011	57,7011	92,7011	255	253	78,7011	86,7011	78,7011	249	.....
.....	.....	.....	....	.....	....	....	....	.....	F(100,100)

#### 4.1.2.2 Cropping

*Cropping* dilakukan untuk meminimalisir *background* pada citra tanda tangan sehingga didapatkan area yang hanya fokus terhadap tanda tangan saja. Hal ini berguna agar citra tanda tangan dapat diproses dengan lebih baik pada tahap berikutnya. Gambar 4.4 merupakan contoh gambar yang telah di *cropping* sehingga latar belakangnya lebih sedikit dan hanya fokus terhadap tanda tangan saja :

#### Hak Cipta Dilindungi Undang-Undang

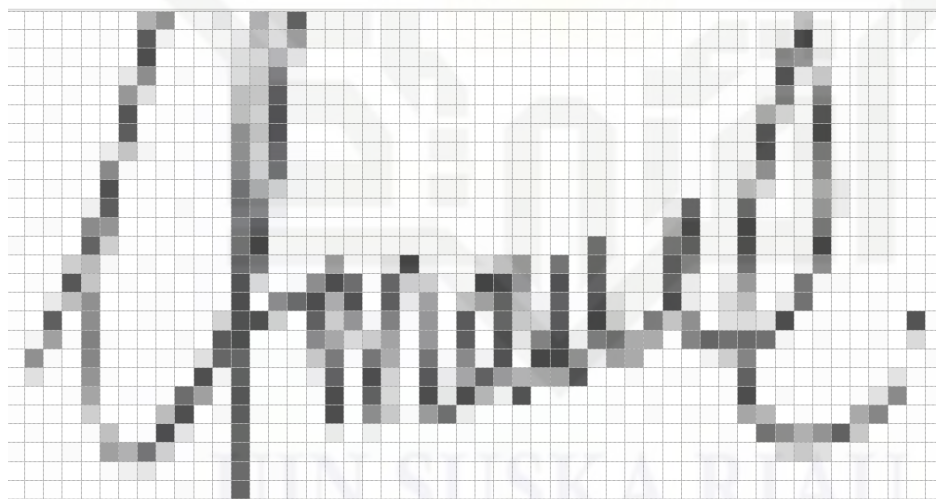
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan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



**Gambar 4.6 Citra Tanda Tangan Setelah di *Cropping***

#### 4.1.2.3 *Resize*

Setelah melalui konversi dan *cropping* maka gambar akan melalui proses *resize*. Dalam proses *resize* ditentukan *aspect ratio* nya menjadi 1 : 1 agar ukuran gambar menjadi sama. Citra tanda tangan di *resize* agar menjadi 100 x 100 piksel. Gambar 4.7 dibawah ini merupakan *sample* gambar tanda tangan yang telah di *resize* :



**Gambar 4.7 Citra Tanda Tangan Setelah di *Resize***

#### 4.1.3 *Analisis Processing (Feature Extraction)*

Dalam analisis *processing* akan dilakukan pengekstraksian ciri suatu citra atau sering disebut juga *feature extraction*. Nilai piksel hasil konversi gambar dari RGB ke *grayscale* akan dihitung dengan menggunakan metode deteksi tepi (*edge*

#### 4.1.3.1 Deteksi Tepi (*Edge Detection*)

Citra tanda tangan yang melalui proses *pre-processing* akan memiliki nilai piksel piksel yang baru. Nilai piksel tersebut kemudian akan dihitung menggunakan metode Deteksi tepi (*Edge Detection*). Nilai piksel yang akan dihitung diambil dari nilai grayscale yang telah dihitung pada Tabel 4.1.

Langkah proses perhitungan nilai piksel yang telah diubah ke *grayscale* dengan menggunakan deteksi tepi (*edge detection*) :

1. Dengan menggunakan persamaan (2.4) yaitu rumus matriks filter pada operator sobel maka akan dicari nilai  $S_x$  nya terlebih dahulu secara vertikal. Dilakukan perhitungan sebagai berikut:

- $193,7011 * (-1) + 22,7011 * (-2) + 130 * (-1) + 90,4022 * 1 + 70,4022 * 2 + 96,4022 * 1$   
 $= -41,4945$
- $22,7011 * (-1) + 130 * (-2) + 196,7011 * (-1) + 70,4022 * 1 + 96,4022 * 2 + 109,7011 * 1$   
 $= -106,495$

Demikian seterusnya dengan menggunakan persamaan yang sama di reduksi seluruh jumlah pikselnya sehingga didapatkan Tabel 4.3 seperti dibawah ini :

**Tabel 4.3 Nilai  $S_x$**

*	*	*	*	*	*	*	*	*
*	-							*
*	41,4945	304,1033	-33,4022	418,1956	725,7934	-124,598	-673,196	*
*	-							*
*	106,495	833,7011	-159	-159	806,0923	-39,2989	-573,196	*
*	-							*
*	306,598	963,8967	-60,7011	-60,7011	651,1956	78,2989	-461,495	*
*	-							*
*	392,299	945,7934	127,8967	127,8967	420,2989	118,5978	-339,495	*
*	-							*
*	435,897	916,7934	291,1956	291,1956	156,3913	114	-141,288	*
*	-							*
*	485,897	880,0923	414,8967	414,8967	-118,217	102,4022	95,6196	*
*	-							*
*	547,299	837,0923	527,5978	527,5978	-377,908	63,7011	369,6087	*
*	...	...	...	...	...	...	F(98,98)	*
*	*	*	*	*	*	*	*	*

2. Setelah mendapatkan nilai  $S_x$  maka kemudian akan dicari nilai  $S_y$  nya secara horizontal. Dilakukan perhitungan sebagai berikut:

- a.  $193,7011 * 1 + 179 * 2 + 90,4022 * 1 + 130 * (-1) + 0 * (-2) + 96,4022 * (-1)$   
 $= 460.3$
- b.  $22,7011 * 1 + 6,7011 * 2 + 70,4022 * 1 + 196,7011 * (-1) + 8,7011 * (-2)$   
 $+ 109,7011 * (-1)$   
 $= -166.1$

Demikian seterusnya dengan menggunakan persamaan yang sama di reduksi seluruh jumlah pikselnya sehingga didapatkan Tabel 4.4 seperti dibawah ini :

**Tabel 4.4 Nilai Sy**

*	*	*	*	*	*	*	*	*
*	415,7011	-49,2989	-314	112,1956	283,1956	148,5978	98,5978	*
*	-217,299	-216,897	-356,196	-97,5978	210,2989	179,2989	44	*
*	-142,804	-73,2989	-162,495	-138,196	90	158,2989	64,2989	*
*	-56,2989	-22	-103,299	-140,299	38,2989	160,5978	92,2989	*
*	-47,2989	-9	-62	-99,2989	75,0109	221,2174	95,9076	*
*	-50,1033	-1,7011	-35,7011	-86,7011	49	189	69	*
*	-41,2989	14,7011	-21	-112	-17,3098	141,0815	45,7935	*
*	...	...	...	...	...	...	F(98,98)	*
*	*	*	*	*	*	*	*	*

3. Dengan menggunakan persamaan (2.5) yaitu dengan menggunakan persamaan magnitude. Sehingga didapat hasil perhitungan pada Tabel 4.5 dibawah ini :

**Tabel 4.5 Nilai Magnitude**

417,7669	308,0734	315,7716	432,9843	779,0865	193,9225	680,3777	...
241,9915	861,4533	390,0722	186,5645	833,0729	183,5552	574,8819	...
338,2238	966,6797	173,4621	150,9392	657,3855	176,6048	465,9523	...
396,318	946,0492	164,4026	189,8456	422,0402	199,6424	351,8176	...
438,4554	916,8376	297,7228	307,6608	173,4499	248,8637	170,7647	...
488,4731	880,0939	416,4299	423,8589	127,9701	214,9586	117,9157	...
548,8549	837,2214	528,0156	539,3547	378,3038	154,7961	372,4347	...
...	...	...	...	...	...	...	100, 100

4. Lakukan *Tresholding*, misalkan nilai *threshod* = 199
- Jika  $M \geq 199$  Maka  $G = 1$  (Putih)
- Jika  $M < 199$  Maka  $G = 0$  (Hitam)



5. Nilai yang didapatkan dari proses *thresholding* tersebut menjadi nilai yang digunakan sebagai pola karakter yang akan dilatih dan diuji. Nilai biner pola tersebut dapat dilihat pada Tabel 4.6 dibawah ini :

**Tabel 4.6 Hasil *Edge Detection***

1	1	1	1	1	0	1	1
1	1	1	0	1	0	1	1
1	1	0	0	1	0	1	1
1	1	0	0	1	0	1	1
1	1	1	1	0	1	0	1
1	1	1	1	0	1	0	1
1	1	1	1	1	0	1	1
1	1	1	1	1	0	1	1

Sehingga representasi nilai *biner* dari pola karakter juga dapat dilihat sebagai berikut :

111110111110101111001011110010111110101111101011111011111011111011

#### 4.1.4 Analisis *Clasification* dengan *Backpropagation Neural Network*

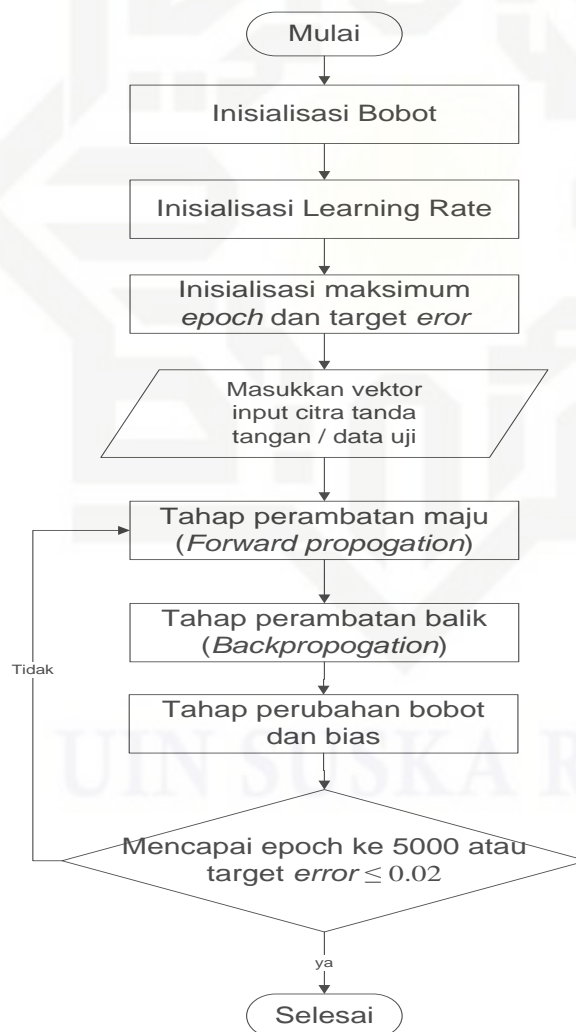
Setelah melalui beberapa tahapan yaitu data *collection* – *preprocessing* – *processing*, maka tahapan terakhir untuk mendapatkan hasil dari identifikasi tanda tangan adalah tahap *clasification*. Tahap *clasification* ini mengklasifikasikan nilai biner yang telah diperoleh sebelumnya pada tahap *feature extraction* dan kemudian akan dijadikan data latih (*training*) yang selanjutnya akan diuji dengan data uji (*testing*) dengan menerapkan jaringan syaraf tiruan dengan menggunakan metode BPNN.

##### 4.1.4.1 *Flowchart Backpropagation Neural Network*

*Flowchart* merupakan diagram alir yang menggambarkan alur/proses yang berjalan. Pada Gambar 4.8 dibawah ini merupakan *flowchart* dari tahap *training* pada proses identifikasi tanda tangan dengan menggunakan metode *backpropagation*. Langkah pertama yang dilakukan adalah inisialisasi bobot awal input ke *hidden layer*, bobot awal bias ke *hidden layer*, bobot awal *hidden layer* ke *output layer*, dan bobot awal bias ke *output layer*. Setelah itu tentukan *learning rate*, maksimum *epoch* dan target *error*. Langkah berikutnya adalah inisialisasi nilai biner yang telah diperoleh dari proses *preprocessing* dan *processing*,

kemudian dilakukan inialisasi target citra tanda tangan. Setelah itu, dilakukan tahap perambatan maju (*forward propagation*), tahap perambatan balik (*backpropagation*), dan tahap perubahan bobot dan bias. Untuk data kedua dilakukan operasi yang sama dengan data pertama, akan tetapi nilai-nilai bobot dan bias awal yang digunakan adalah nilai-nilai bobot dan bias baru dari hasil perhitungan dari data pertama. Demikian seterusnya sampai data terakhir (1 *epoch*). Proses ini diteruskan hingga maksimum *epoch* ke 5.000 atau akan berhenti jika kuadrat *error* (*target error*)  $\leq 0.02$ .

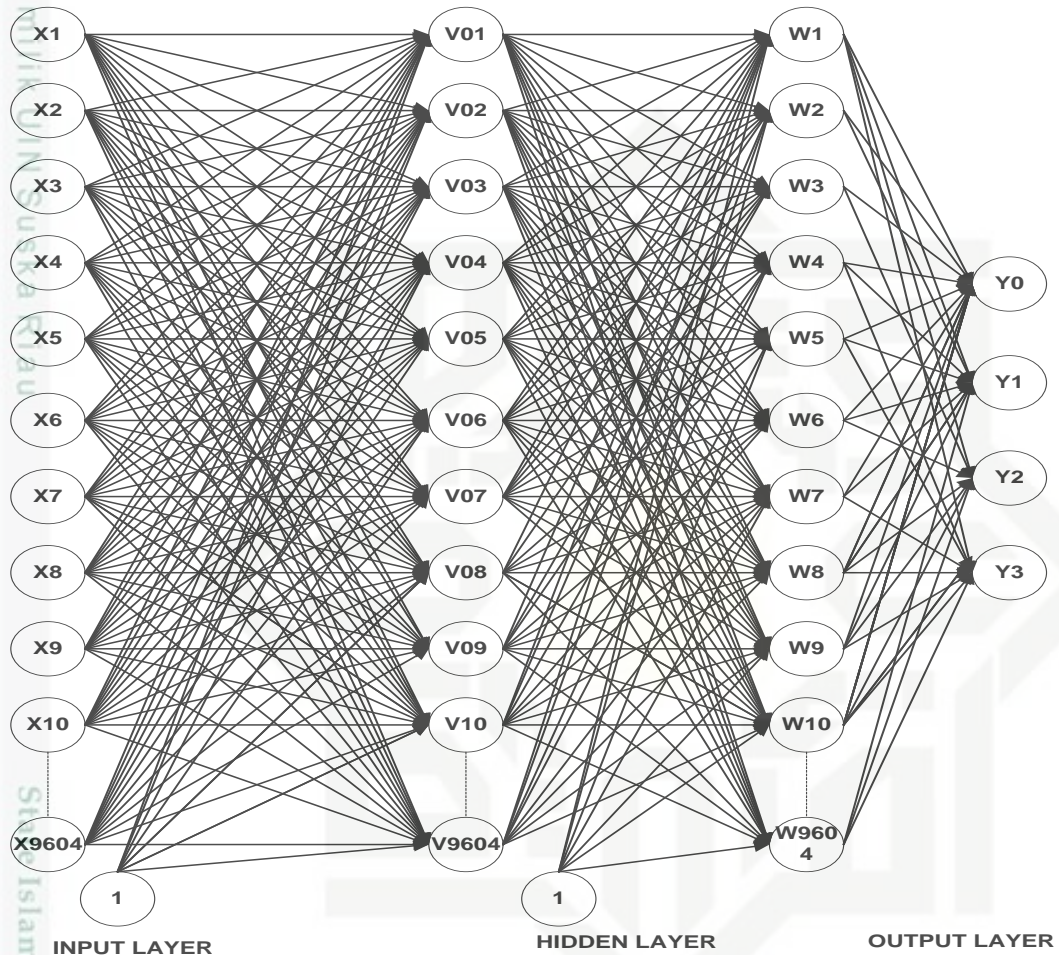
*Flowchart* pengenalan karakter huruf dengan metode BPNN dapat dilihat pada Gambar 4.8 berikut:



**Gambar 4.8 Flowchart BPNN Identifikasi Tanda Tangan**

#### 4.1.4.2 Arsitektur *Backpropagation Neural Network*

Arsitektur jaringan syaraf tiruan dengan metode BPNN pada proses identifikasi tanda tangan dilihat pada Gambar 4.9 berikut:



**Gambar 4.9 Arsitektur Jaringan Syaraf Tiruan BPNN untuk Identifikasi Tanda Tangan**

Keterangan Gambar 4.9:

1. Jumlah *neuron input* pada *input layer* (disimbolkan dengan X) bergantung pada ukuran matriks yang digunakan. Pada penelitian ini jumlah *neuron input* adalah sebanyak 9605 karena menggunakan matriks berukuran 100 x 100 piksel. Kemudian setelah mengalami *feature extraction* maka matriks nya menjadi 98 x 98 yaitu 9604 dan ditambah kan dengan 1 bias menjadi 9605 neuron input.



**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. Pada penelitian ini jumlah lapisan *hidden layer* akan diuji mulai dari 1 lapisan *hidden layer* sampai 3 lapisan *hidden layer* untuk melihat akurasi yang terbaik. Neuron *hidden layer* terdiri dari 9605 buah *neuron*.
3. Jumlah tanda tangan yang digunakan sebanyak 100 citra tanda tangan yang diperoleh dari 10 orang dengan masing-masing 10 pola. Untuk membedakan target antara citra tanda tangan yang satu dan yang lainnya maka dengan menggunakan beberapa digit *biner*. karena pada penelitian ini menggunakan fungsi aktivasi *sigmoid biner*. Karena kelas nya ada 10, maka digit biner yang digunakan adalah 0001,0010,0011,0100,0101,0110, 0111, 1000, 1001, 1010.
4. *Neuron* pada *output layer* terdiri dari 4 buah *neuron output*. *Neuron output* disimbolkan dengan huruf Y0, Y1, Y2, Y3.

#### 4.1.4.3 Contoh Perhitungan *Backpropagation Neural Network*

Berikut merupakan contoh perhitungan BPNN menggunakan data citra dengan *sample3x3* piksel yang telah dilakukan tahap *preprocessing* dan *processing*. Data latih yang digunakan merujuk pada Tabel 4.6.

Data Latih: [1 1 1 1 1 0 1 1] Target = 7

##### Langkah 0:

##### Inisialisasi bobot (ambil nilai *random* yang cukup kecil)

Bobot yang akan di inisialisasi diusahakan cukup kecil yaitu dibawah 1. Nilai bobot tersebut dapat dilihat pada Tabel 4.7 di bawah ini :

**Tabel 4.7 Bobot Awal *Input* ke *Hidden Layer***

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0,2	0,9	0,6	0,4	0,3	0,7	0,4	0,4	0,5	0,9
0,2	0,1	0,7	0,4	0,2	0,6	0,1	0,2	0,3	0,1
0,1	0,8	0,8	0,1	0,4	0,9	0,1	0,3	0,1	0,3
0,9	0,7	0,2	0,3	0,1	0,2	0,4	0,4	0,3	0,2
0,7	0,1	0,8	0,4	0,2	0,1	0,4	0,3	0,2	0,4
0,5	0,9	0,2	0,9	0,1	0,3	0,6	0,3	0,4	0,5
0,3	0,2	0,4	0,8	0,2	0,1	0,1	0,2	0,2	0,7
0,3	0,2	0,4	0,1	0,5	0,1	0,4	0,3	0,5	0,1
0,1	0,1	0,4	0,1	0,9	0,1	0,1	0,2	0,2	0,8



Kemudian inisialisasi bobot awal bias ke *hidden layer*. Nilai bobot tersebut dapat di lihat pada Tabel 4.8 dibawah ini:

**Tabel 4.8 Bobot Awal Bias ke *Hidden Layer***

0,1	0,3	0,4	0,8	0,8	0,1	0,3	0,3	0,9	0,4
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Selanjutnya dilakukan inisialisasi bobot awal *hidden layer* ke *output layer* dan nilai bobot nya dapat dilihat pada Tabel 4.9 dibawah ini:

**Tabel 4.9 Bobot Awal *Hidden Layer* ke *Output Layer***

bobot z-y	y0	y1	y2	y3
w0	0,4	0,2	0,1	0,4
w1	0,3	0,4	0,1	0,3
w2	0,2	0,3	0,3	0,3
w3	0,4	0,2	0,2	0,4
w4	0,4	0,1	0,1	0,1
w5	0,2	0,4	0,1	0,1
w6	0,4	0,2	0,4	0,2
w7	0,4	0,2	0,2	0,3
w8	0,3	0,1	0,2	0,4
w9	0,3	0,2	0,3	0,2

**Learning rate ( $\alpha$ ) = 0.1**

**Maksimum *epoch* = 10000**

**Target *error* = 0.02**

**a) Tahap Perambatan Maju (*Forward Propagation*)**

Operasi pada *hiddenlayer*:

Jumlahkan semua sinyal yang masuk kedalam persamaan (2.10)

$$\begin{aligned}
 Z_{in_1} &= (0,1 + (1 * 0,2) + (1 * 0,2) + (1 * 0,1) + (1 * 0,9) + \\
 &\quad (1 * 0,7)(1 * 0,5)(0 * 0,3)(1 * 0,3)(1 * 0,1) \\
 &= 3,1
 \end{aligned}$$

Demikian seterusnya dilakukan perhitungan dengan persamaan (2.10) sampai diperoleh  $Z_{in_{10}}$ . Hasil Perhitungan tersebut dapat dilihat pada Tabel 4.10 dibawah ini :

**Tabel 4.10 Operasi Pada *Hidden Layer***

z_in1	z_in2	z_in3	z_in4	z_in5	z_in6	z_in7	z_in8	z_in9	z_in10
3,1	4,1	4,5	3,5	3,5	3,1	2,8	2,7	3,4	3,7

Selanjutnya Hitung fungsi aktivasi pada *hidden layer* dengan menggunakan persamaan (2.11):

$$z_1 = \frac{1}{1+e^{-3,1}} = 0,9568$$

Demikian seterusnya dilakukan perhitungan dengan persamaan (2.11) sampai dengan diperoleh  $Z_{10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.11 dibawah ini :

**Tabel 4.11 Fungsi Aktivasi pada *Hidden Layer***

Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
0,9568	0,9836	0,9890	0,9706	0,9706	0,9568	0,9426	0,9370	0,9677	0,9758

Setelah itu dilakukan perhitungan operasi pada *output layer* menggunakan persamaan (2.12):

$$\begin{aligned} y_{-in} &= (0,4 + (0,9568 * 0,3) + (0,9836 * 0,2) + (0,9890 * 0,4) + \\ &\quad (0,9706 * 0,2) + (0,9706 * 0,4) + (0,9568 * 0,4) + \\ &\quad (0,9426 * 0,2) + (0,9370 * 0,4) + (0,9677 * 0,4) + (0,9758 * \\ &\quad 0,2) \\ &= 3,388246585 \end{aligned}$$

Demikian seterusnya dilakukan perhitungan dengan persamaan (2.12) sampai dengan diperoleh  $y_{-in3}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.12 di bawah ini :

**Tabel 4.12 Perhitungan Operasi Pada *Output Layer***

y_in0	y_in1	y_in2	y_in3
3,388246585	2,325757427	2,02933494	2,907214605

Kemudian dilakukan perhitungan untuk mencari nilai fungsi aktivasi pada *output layer* dengan menggunakan persamaan (2.13):

$$\begin{aligned} y_0 &= \frac{1}{1+e^{-3,180938238}} \\ &= 0,960110614 \end{aligned}$$

Demikian seterusnya dilakukan perhitungan dengan persamaan (2.13) sampai dengan diperoleh  $y_3$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.13 di bawah ini :

**Tabel 4.13 Fungsi Aktivasi pada *Output Layer***

$y_0$	$Y_1$	$y_2$	$y_3$
0,967335186	0,91098791	0,883842817	0,94820193

**b) Tahap Perambatan Balik (*Backpropagation*) ya**

**Untuk  $T_0$ :**

Hitung *Error* dengan persamaan (2.14):

$$\begin{aligned}\delta_k &= (t_k - y_k) * y_k * (1 - y_k) \\ \delta &= ((0 - 0,967335186) * 0,967335186 * (0,967335186)) \\ &= -0,030565687\end{aligned}$$

Koreksi bias dengan menggunakan persamaan (2.15)

$$\begin{aligned}\Delta w_0 &= 0,1 * -0,030565687 \\ &= -0,0030565687\end{aligned}$$

Koreksi Bobot dengan menggunakan persamaan (2.16):

$$\Delta w_1 = 0,1 * (-0,030565687) * 0,956892745 = -0,003056569$$

Demikian seterusnya dilakukan perhitungan menggunakan persamaan (2.16) sampai dengan didapat  $\Delta w_{10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.14 di bawah ini :

**Tabel 4.14 Koreksi Bobot dan Bias**

$\Delta w_0$	$\Delta w_1$	$\Delta w_2$	$\Delta w_3$	$\Delta w_4$	$\Delta w_5$	$\Delta w_6$	$\Delta w_7$	$\Delta w_8$	$\Delta w_9$	$\Delta w_{10}$
-0,003056569	0,002924808	0,003006739	0,003022986	0,002966974	0,002966974	0,002924808	0,002881353	0,002864086	0,002957855	0,002982823

Setelah itu setiap unit tersembunyi ( $z_j$ ) menjumlahkan delta *input*-nya dengan menggunakan persamaan (2.17):

$$\delta_{in_1} = (-0,030565687 * 0,3) = -0,011031159$$

Demikian seterusnya dilakukan perhitungan dengan menggunakan persamaan (2.17) sampai dengan diperoleh  $\delta_{in_{10}}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.15 di bawah ini :

**Tabel 4.15 Nilai Delta Input Lapisan Tersembunyi**

$\delta_{in1}$	$\delta_{in2}$	$\delta_{in3}$	$\delta_{in4}$	$\delta_{in5}$	$\delta_{in6}$	$\delta_{in7}$	$\delta_{in8}$	$\delta_{in9}$	$\delta_{in10}$
0,00916 9706	- 0,00611 3137	- 0,01222 6275	- 0,01222 6275	- 0,00611 3137	- 0,01222 6275	- 0,01222 6275	- 0,00916 9706	- 0,00916 9706	- 0,00611 3137

Kemudian dicari nilai informasi *error* dengan menggunakan persamaan (2.18):

$$\begin{aligned}\delta_1 &= -0,009169706 * 0,956892745 * (1 - (0,956892745)) \\ &= -0,000378241\end{aligned}$$

Demikian seterusnya dilakukan perhitungan dengan menggunakan persamaan (2.18) sampai dengan nilai  $\delta_{10}$ . Hasil perhitungannya dapat dilihat pada Tabel 4.16 di bawah ini

**Tabel 4.16 Nilai informasi *error***

$\delta_1$	$\delta_2$	$\delta_3$	$\delta_4$	$\delta_5$	$\delta_6$	$\delta_7$	$\delta_8$	$\delta_9$	$\delta_{10}$
- 0,0003 78241	- 0,0000 980347	- 0,0001 32854	- 0,0003 47874	- 0,0001 73937	- 0,0005 04322	- 0,0006 60685	- 0,0005 41083	- 0,0002 86576	- 0,0001 43933

Setelah itu dilakukan Koreksi Bobot dengan menggunakan persamaan (2.19):

$$\Delta v_{11} = (0,1 * (-0,000378241) * 1) = 0$$

Demikian seterusnya dengan menghitung menggunakan persamaan (2.19) hingga diperoleh nilai  $\Delta V_{10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.16:

**Tabel 4.17 Koreksi Bobot**

$\Delta V_1$	$\Delta V_2$	$\Delta V_3$	$\Delta V_4$	$\Delta V_5$	$\Delta V_6$	$\Delta V_7$	$\Delta V_8$	$\Delta V_9$	$\Delta V_{10}$
- 0,0000 37824 1	- 0,0000 09803 5	- 0,0000 13285 4	- 0,0000 34787 4	- 0,0000 17393 7	- 0,0000 50432 2	- 0,0000 66068 5	- 0,0000 54108 3	- 0,0000 28657 6	- 0,0000 14393 3
- 0,0000 37824 1	- 0,0000 09803 5	- 0,0000 13285 4	- 0,0000 34787 4	- 0,0000 17393 7	- 0,0000 50432 2	- 0,0000 66068 5	- 0,0000 54108 3	- 0,0000 28657 6	- 0,0000 14393 3
- 0,0000	- 0,0000	- 0,0000	- 0,0000	- 0,0000	- 0,0000	- 0,0000	- 0,0000	- 0,0000	- 0,0000



378241	098035	132854	347874	173937	504322	660685	541083	286576	143933
-	-	-	-	-	-	-	-	-	-
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
378241	098035	132854	347874	173937	504322	660685	541083	286576	143933
-	-	-	-	-	-	-	-	-	-
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
378241	098035	132854	347874	173937	504322	660685	541083	286576	143933
-	-	-	-	-	-	-	-	-	-
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
378241	098035	132854	347874	173937	504322	660685	541083	286576	143933
0	0	0	0	0	0	0	0	0	0
-	-	-	-	-	-	-	-	-	-
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
378241	098035	132854	347874	173937	504322	660685	541083	286576	143933
-	-	-	-	-	-	-	-	-	-
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
378241	098035	132854	347874	173937	504322	660685	541083	286576	143933

Setelah itu dihitung koreksi bias dengan menggunakan persamaan (2.20):

$$\Delta v_{01} = (0,1 * (-0,000378241)) = -0,0000378241$$

Demikian seterusnya dilakukan perhitungan menggunakan persamaan (2.20) , dan hasil perhitungannya dapat dilihat pada Tabel 4.17 di bawah ini :

**Tabel 4.18 Koreksi Bias**

$\Delta V_0$
-0,0000378241
-0,0000098035
-0,0000132854
-0,0000347874
-0,0000173937
-0,0000504322
-0,0000660685
-0,0000541083
-0,0000286576
-0,0000143933

**Untuk  $T_1$ :**

Hitung *Error* dengan persamaan (2.14):

$$\delta_k = (t_k - y_k) * y_k * (1 - y_k)$$

$$\delta = ((1 - 0,91098791) * 0,91098791 * (0,91098791))$$

$$= 0,007217896$$

Koreksi bias dengan menggunakan persamaan (2.15)

$$\begin{aligned}\Delta w_0 &= 0,1 * 0,007217896 \\ &= 0,0007217896\end{aligned}$$

Koreksi Bobot dengan menggunakan persamaan (2.16):

$$\Delta w_1 = 0.1 * (0,007217896) * 0,956892745 = 0,000690675$$

Demikian seterusnya dilakukan perhitungan menggunakan persamaan (2.16) sampai dengan didapat  $\Delta w_{10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.19 di bawah ini :

**Tabel 4.19 Koreksi Bobot dan Bias**

$\Delta w_0$	$\Delta w_1$	$\Delta w_2$	$\Delta w_3$	$\Delta w_4$	$\Delta w_5$	$\Delta w_6$	$\Delta w_7$	$\Delta w_8$	$\Delta w_9$	$\Delta w_{10}$
0,00072179	0,000690675	0,000710023	0,000713859	0,000700632	0,000700632	0,000690675	0,000680414	0,000676336	0,000698479	0,000704375

Setelah itu setiap unit tersembunyi ( $z_j$ ) menjumlahkan delta *input*-nya dengan menggunakan persamaan (2.17):

$$\delta_{in_1} = (0,007217896 * 0.4) = 0,002887158$$

Demikian seterusnya dilakukan perhitungan dengan menggunakan persamaan (2.17) sampai dengan diperoleh  $\delta_{in_{10}}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.15 di bawah ini :

**Tabel 4.20 Nilai Delta Input Lapisan Tersembunyi**

$\delta_{in1}$	$\delta_{in2}$	$\delta_{in3}$	$\delta_{in4}$	$\delta_{in5}$	$\delta_{in6}$	$\delta_{in7}$	$\delta_{in8}$	$\delta_{in9}$	$\delta_{in10}$
0,002887158	0,002165369	0,001443579	0,00072179	0,002887158	0,001443579	0,001443579	0,00072179	0,001443579	0,00072179

Kemudian dicari nilai informasi *error* dengan menggunakan persamaan (2.18):

$$\begin{aligned}\delta_1 &= 0,002887158 * 0,956892745 * (1 - (0,956892745)) \\ &= 0,000119092\end{aligned}$$

Demikian seterusnya dilakukan perhitungan dengan menggunakan persamaan (2.18) sampai dengan nilai  $\delta_{10}$ . Hasil perhitungannya dapat dilihat pada Tabel 4.21 di bawah ini:

**Tabel 4.21 Nilai informasi error**

$\delta 1$	$\delta 2$	$\delta 3$	$\delta 4$	$\delta 5$	$\delta 6$	$\delta 7$	$\delta 8$	$\delta 9$	$\delta 10$
0,000 11909 2	0,0000 34725 4	0,0000 156863	0,0000 205371	0,0000 821484	0,0000 595462	0,0000 780083	0,0000 425912	0,0000 451154	0,0000 169945

Setelah itu dilakukan Koreksi Bobot dengan menggunakan persamaan (2.19):

$$\Delta v_{11} = (0,1 * (0,00011909) * 1) = 0,0000119092$$

Demikian seterusnya dengan menghitung menggunakan persamaan (2.19) hingga diperoleh nilai  $\Delta V10$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.22:

**Tabel 4.22 Koreksi Bobot**

$\Delta V1$	$\Delta V2$	$\Delta V3$	$\Delta V4$	$\Delta V5$	$\Delta V6$	$\Delta V7$	$\Delta V8$	$\Delta V9$	$\Delta V10$
0,0000 11909 2	0,0000 03472 5	0,0000 01568 6	0,0000 02053 7	0,0000 08214 8	0,0000 05954 6	0,0000 07800 8	0,0000 04259 1	0,0000 04511 5	0,0000 01699 4
0,0000 11909 2	0,0000 03472 5	0,0000 01568 6	0,0000 02053 7	0,0000 08214 8	0,0000 05954 6	0,0000 07800 8	0,0000 04259 1	0,0000 04511 5	0,0000 01699 4
0,0000 11909 2	0,0000 03472 5	0,0000 01568 6	0,0000 02053 7	0,0000 08214 8	0,0000 05954 6	0,0000 07800 8	0,0000 04259 1	0,0000 04511 5	0,0000 01699 4
0,0000 11909 2	0,0000 03472 5	0,0000 01568 6	0,0000 02053 7	0,0000 08214 8	0,0000 05954 6	0,0000 07800 8	0,0000 04259 1	0,0000 04511 5	0,0000 01699 4
0,0000 11909 2	0,0000 03472 5	0,0000 01568 6	0,0000 02053 7	0,0000 08214 8	0,0000 05954 6	0,0000 07800 8	0,0000 04259 1	0,0000 04511 5	0,0000 01699 4
0,0000 11909 2	0,0000 03472 5	0,0000 01568 6	0,0000 02053 7	0,0000 08214 8	0,0000 05954 6	0,0000 07800 8	0,0000 04259 1	0,0000 04511 5	0,0000 01699 4
0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
0,0000 11909 2	0,0000 03472 5	0,0000 01568 6	0,0000 02053 7	0,0000 08214 8	0,0000 05954 6	0,0000 07800 8	0,0000 04259 1	0,0000 04511 5	0,0000 01699 4
0,0000 11909 2	0,0000 03472 5	0,0000 01568 6	0,0000 02053 7	0,0000 08214 8	0,0000 05954 6	0,0000 07800 8	0,0000 04259 1	0,0000 04511 5	0,0000 01699 4

Setelah itu dihitung koreksi bias dengan menggunakan persamaan (2.20):

$$\Delta v_{01} = (0,1 * (0,00011909)) = 0,000011909$$

**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 ilmiah.
- 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.

Demikian seterusnya dilakukan perhitungan menggunakan persamaan (2.20) , dan hasil perhitungannya dapat dilihat pada Tabel 4.23 di bawah ini :

**Tabel 4.23 Koreksi Bias**

$\Delta V_0$
0,0000119092
0,0000034725
0,0000015686
0,0000020537
0,0000082148
0,0000059546
0,0000078008
0,0000042591
0,0000045115
0,0000016994

**Untuk  $T_2$ :**

Hitung *Error* dengan persamaan (2.14):

$$\begin{aligned}\delta_k &= (t_k - y_k) * y_k * (1 - y_k) \\ \delta &= ((1 - 0,883842817) * 0,883842817 * (0,883842817)) \\ &= 0,011925241\end{aligned}$$

Koreksi bias dengan menggunakan persamaan (2.15)

$$\begin{aligned}\Delta w_0 &= 0,1 * 0,011925241 \\ &= 0,0011925241\end{aligned}$$

Koreksi Bobot dengan menggunakan persamaan (2.16):

$$\Delta w_1 = 0.1 * (0,011925241) * 0,956892745 = 0,001141118$$

Demikian seterusnya dilakukan perhitungan menggunakan persamaan (2.16) sampai dengan didapat  $\Delta w_{10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.24 di bawah ini :

**Tabel 4.24 Koreksi Bobot dan Bias**

$\Delta w_0$	$\Delta w_1$	$\Delta w_2$	$\Delta w_3$	$\Delta w_4$	$\Delta w_5$	$\Delta w_6$	$\Delta w_7$	$\Delta w_8$	$\Delta w_9$	$\Delta w_{10}$
0,001192524	0,001141118	0,001173083	0,001179422	0,001157569	0,001157569	0,001141118	0,001124164	0,001117427	0,001154011	0,001163752



Setelah itu setiap unit tersembunyi ( $z_j$ ) menjumlahkan delta *input*-nya dengan menggunakan persamaan (2.17):

$$\delta_{in1} = (0,011925241 * 0.1) = 0,0011925241$$

Demikian seterusnya dilakukan perhitungan dengan menggunakan persamaan (2.17) sampai dengan diperoleh  $\delta_{in10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.25 di bawah ini :

**Tabel 4.25 Nilai Delta Input Lapisan Tersembunyi**

$\delta_{in1}$	$\delta_{in2}$	$\delta_{in3}$	$\delta_{in4}$	$\delta_{in5}$	$\delta_{in6}$	$\delta_{in7}$	$\delta_{in8}$	$\delta_{in9}$	$\delta_{in10}$
0,00119 2524	0,00357 7572	0,00238 5048	0,00119 2524	0,00119 2524	0,00477 0097	0,00238 5048	0,00238 5048	0,00357 7572	0,00119 2524

Kemudian dicari nilai informasi *error* dengan menggunakan persamaan (2.18):

$$\begin{aligned}\delta_1 &= 0,001192524 * 0,956892745 * (1 - (0,956892745)) \\ &= 0,0000491905\end{aligned}$$

Demikian seterusnya dilakukan perhitungan dengan menggunakan persamaan (2.18) sampai dengan nilai  $\delta_{10}$ . Hasil perhitungannya dapat dilihat pada Tabel 4.26 di bawah ini

**Tabel 4.26 Nilai informasi *error***

$\delta_1$	$\delta_2$	$\delta_3$	$\delta_4$	$\delta_5$	$\delta_6$	$\delta_7$	$\delta_8$	$\delta_9$	$\delta_{10}$
0,0000 49190 5	0,0000 57372 6	0,0000 25916 5	0,0000 33930 9	0,0000 33930 9	0,0001 96761 8	0,0001 28883 5	0,0001 40736 2	0,0001 11808 0	0,0000 28077 9

Setelah itu dilakukan Koreksi Bobot dengan menggunakan persamaan (2.19):

$$\Delta w_{11} = (0,1 * (0,0000491905) * 1) = 0,00000491905$$

Demikian seterusnya dengan menghitung menggunakan persamaan (2.19) hingga diperoleh nilai  $\Delta V_{10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.27:

**Tabel 4.27 Koreksi Bobot**

$\Delta V_1$	$\Delta V_2$	$\Delta V_3$	$\Delta V_4$	$\Delta V_5$	$\Delta V_6$	$\Delta V_7$	$\Delta V_8$	$\Delta V_9$	$\Delta V_{10}$
0,0000 04919 0	0,0000 05737 3	0,0000 02591 6	0,0000 03393 1	0,0000 03393 1	0,0000 19676 2	0,0000 12888 4	0,0000 14073 6	0,0000 11180 8	0,0000 02807 8

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mengemukakan dan menyebutkan sumber.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
04919	05737	02591	03393	03393	19676	12888	14073	11180	02807
0	3	6	1	1	2	4	6	8	8
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
04919	05737	02591	03393	03393	19676	12888	14073	11180	02807
0	3	6	1	1	2	4	6	8	8
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
04919	05737	02591	03393	03393	19676	12888	14073	11180	02807
0	3	6	1	1	2	4	6	8	8
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
04919	05737	02591	03393	03393	19676	12888	14073	11180	02807
0	3	6	1	1	2	4	6	8	8
0	0	0	0	0	0	0	0	0	0
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
04919	05737	02591	03393	03393	19676	12888	14073	11180	02807
0	3	6	1	1	2	4	6	8	8
0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
04919	05737	02591	03393	03393	19676	12888	14073	11180	02807
0	3	6	1	1	2	4	6	8	8

Setelah itu dihitung koreksi bias dengan menggunakan persamaan (2.20):

$$\Delta v_{01} = (0,1 * (0,0000491905)) = 0,00000491905$$

Demikian seterusnya dilakukan perhitungan menggunakan persamaan (2.20) , dan hasil perhitungannya dapat dilihat pada Tabel 4.28 di bawah ini :

**Tabel 4.28 Koreksi Bias**

$\Delta v_0$
0,0000049190
0,0000057373
0,0000025916
0,0000033931
0,0000033931
0,0000196762
0,0000128884
0,0000140736
0,0000111808
0,0000028078

### Untuk $T_3$ :

Hitung *Error* dengan persamaan (2.14):

$$\begin{aligned}\delta_k &= (t_k - y_k) * y_k * (1 - y_k) \\ \delta &= ((1 - 0,94820193) * 0,94820193 * (0,94820193)) \\ &= 0,002544064\end{aligned}$$

Koreksi bias dengan menggunakan persamaan (2.15)

$$\begin{aligned}\Delta w_0 &= 0,1 * 0,002544064 \\ &= 0,0002544064\end{aligned}$$

Koreksi Bobot dengan menggunakan persamaan (2.16):

$$\Delta w_1 = 0.1 * (0,002544064) * 0,956892745 = 0,00024344$$

Demikian seterusnya dilakukan perhitungan menggunakan persamaan (2.16) sampai dengan didapat  $\Delta w_{10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.29 di bawah ini :

**Tabel 4.29 Koreksi Bobot dan Bias**

$\Delta w_0$	$\Delta w_1$	$\Delta w_2$	$\Delta w_3$	$\Delta w_4$	$\Delta w_5$	$\Delta w_6$	$\Delta w_7$	$\Delta w_8$	$\Delta w_9$	$\Delta w_{10}$
0,000254406	0,00024344	0,000250259	0,000251611	0,000246949	0,000246949	0,00024344	0,000239823	0,000238386	0,00024619	0,000248268

Setelah itu setiap unit tersembunyi ( $z_j$ ) menjumlahkan delta *input*-nya dengan menggunakan persamaan (2.17):

$$\delta_{in_1} = (0,002544064 * 0.3) = 0,000763219$$

Demikian seterusnya dilakukan perhitungan dengan menggunakan persamaan (2.17) sampai dengan diperoleh  $\delta_{in_{10}}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.30 di bawah ini :

**Tabel 4.30 Nilai Delta Input Lapisan Tersembunyi**

$\delta_{in1}$	$\delta_{in2}$	$\delta_{in3}$	$\delta_{in4}$	$\delta_{in5}$	$\delta_{in6}$	$\delta_{in7}$	$\delta_{in8}$	$\delta_{in9}$	$\delta_{in10}$
0,000763219	0,000763219	0,001017625	0,000254406	0,000254406	0,000508813	0,000763219	0,001017625	0,000508813	0,000763219

Kemudian dicari nilai informasi *error* dengan menggunakan persamaan (2.18):

$$\begin{aligned}\delta_1 &= 0,000763219 * 0,956892745 * (1 - (0,956892745)) \\ &= 0,000031482\end{aligned}$$

Demikian seterusnya dilakukan perhitungan dengan menggunakan persamaan (2.18) sampai dengan nilai  $\delta_{10}$ . Hasil perhitungannya dapat dilihat pada Tabel 4.31 di bawah ini

**Tabel 4.31 Nilai informasi error**

$\delta_1$	$\delta_2$	$\delta_3$	$\delta_4$	$\delta_5$	$\delta_6$	$\delta_7$	$\delta_8$	$\delta_9$	$\delta_{10}$
0,000031482	0,0000122395	0,0000110578	0,0000072386	0,0000072386	0,000020988	0,0000412429	0,0000600478	0,0000159017	0,0000179699

Setelah itu dilakukan Koreksi Bobot dengan menggunakan persamaan (2.19):

$$\Delta v_{11} = (0,1 * (0,000031482) * 1) = 0,0000031482$$

Demikian seterusnya dengan menghitung menggunakan persamaan (2.19) hingga diperoleh nilai  $\Delta V_{10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.32:

**Tabel 4.32 Koreksi Bobot**

$\Delta V_1$	$\Delta V_2$	$\Delta V_3$	$\Delta V_4$	$\Delta V_5$	$\Delta V_6$	$\Delta V_7$	$\Delta V_8$	$\Delta V_9$	$\Delta V_{10}$
0,0000031482	0,0000012240	0,0000011058	0,0000007239	0,0000007239	0,0000020988	0,0000041243	0,0000060048	0,0000015902	0,0000017970
0,0000031482	0,0000012240	0,0000011058	0,0000007239	0,0000007239	0,0000020988	0,0000041243	0,0000060048	0,0000015902	0,0000017970
0,0000031482	0,0000012240	0,0000011058	0,0000007239	0,0000007239	0,0000020988	0,0000041243	0,0000060048	0,0000015902	0,0000017970
0,0000031482	0,0000012240	0,0000011058	0,0000007239	0,0000007239	0,0000020988	0,0000041243	0,0000060048	0,0000015902	0,0000017970
0,0000031482	0,0000012240	0,0000011058	0,0000007239	0,0000007239	0,0000020988	0,0000041243	0,0000060048	0,0000015902	0,0000017970
0,0000031482	0,0000012240	0,0000011058	0,0000007239	0,0000007239	0,0000020988	0,0000041243	0,0000060048	0,0000015902	0,0000017970
0	0	0	0	0	0	0	0	0	0
0,0000031482	0,0000012240	0,0000011058	0,0000007239	0,0000007239	0,0000020988	0,0000041243	0,0000060048	0,0000015902	0,0000017970
0,0000031482	0,0000012240	0,0000011058	0,0000007239	0,0000007239	0,0000020988	0,0000041243	0,0000060048	0,0000015902	0,0000017970



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.

Setelah itu dihitung koreksi bias dengan menggunakan persamaan (2.20):

$$\Delta v_{01} = (0,1 * (0,000031482)) = 0,0000031482$$

Demikian seterusnya dilakukan perhitungan menggunakan persamaan (2.20) , dan hasil perhitungannya dapat dilihat pada Tabel 4.33 di bawah ini :

**Tabel 4.33 Koreksi Bias**

$\Delta V_0$
0,0000031482
0,0000012240
0,0000011058
0,0000007239
0,0000007239
0,0000020988
0,0000041243
0,0000060048
0,0000015902
0,0000017970

### c) Tahap perubahan bobot dan bias

Perubahan bobot input menuju *hidden layer* dengan menggunakan persamaan (2.21) sehingga hasil perubahan bobot dapat dilihat pada Tabel 4.34 di bawah ini:

**Tabel 4.34 Perubahan Bobot**

v1baru	v2baru	v3baru	v4baru	v5baru	v6baru	v7baru	v8baru	v9baru	v10baru
0,1999 79004	0,8999 99406	0,5999 94021	0,3999 70659	0,2999 4214	0,69 9975 199	0,3999 54621	0,3999 64224	0,4999 87035	0,899 99011 4
0,1999 79004	0,0999 99406	0,6999 94021	0,3999 70659	0,1999 4214	0,59 9975 199	0,0999 54621	0,1999 64224	0,2999 87035	0,099 99011 4
0,0999 79004	0,7999 99406	0,7999 94021	0,0999 70659	0,3999 4214	0,89 9975 199	0,0999 54621	0,2999 64224	0,0999 87035	0,299 99011 4
0,8999 79004	0,6999 99406	0,1999 94021	0,2999 70659	0,0999 4214	0,19 9975 199	0,3999 54621	0,3999 64224	0,2999 87035	0,199 99011 4
0,6999 79004	0,0999 99406	0,7999 94021	0,3999 70659	0,1999 4214	0,09 9975	0,3999 54621	0,2999 64224	0,1999 87035	0,399 99011

0,4999 79004	0,8999 99406	0,1999 94021	0,8999 70659	0,09999 4214	199 0,29 9975 199	0,5999 54621	0,2999 64224	0,3999 87035	0,499 99011 4
0,3	0,2	0,4	0,8	0,2	0,1	0,1	0,2	0,2	0,7
0,2999 79004	0,1999 99406	0,3999 94021	0,0999 70659	0,49999 4214	0,09 9975 199	0,3999 54621	0,2999 64224	0,4999 87035	0,099 99011 4

Perubahan bias ke *hiddenlayer* dapat dilihat pada Tabel 4.35 dibawah ini

**Tabel 4.35 Perubahan Bias:**

v0(baru)
0,0999790042
0,2999994063
0,3999908749
0,7999706594
0,7999942142
0,0999751986
0,2999546207
0,2999642244
0,8999870347

Perubahan bobot ke *outputlayer* dengan menggunakan persamaan (2.22) sehingga hasilnya dapat dilihat pada Tabel 4.36 di bawah ini:

**Tabel 4.36 Perubahan Bobot ke Output Layer**

bobot z-y	y0	y1	y2	y3
w0	0,399112151	0,199112151	0,099112151	0,399112151
w1	0,299150424	0,399150424	0,099150424	0,299150424
w2	0,199126626	0,299126626	0,299126626	0,299126626
w3	0,399121906	0,199121906	0,199121906	0,399121906
w4	0,399138176	0,099138176	0,099138176	0,099138176
w5	0,199138176	0,399138176	0,099138176	0,099138176
w6	0,399150424	0,199150424	0,399150424	0,199150424
w7	0,399163047	0,199163047	0,199163047	0,299163047
w8	0,299168062	0,099168062	0,199168062	0,399168062
w9	0,299140825	0,199140825	0,299140825	0,199140825
W10	0,1991335725	0,099133573	0,099133573	0,299133573

Perubahan bias ke *outputlayer* dapat dilihat pada Tabel 4.37di bawah ini:

**Tabel 4.37 Perubahan Bias ke Output Layer**

bobot z-y	y0	y1	y2	y3
w0	0,396943431	0,20072179	0,101192524	0,300254406
w1	0,297075192	0,400690675	0,101141118	0,300254406
w2	0,196993261	0,300710023	0,301173083	0,400254406
w3	0,396977014	0,200713859	0,201179422	0,100254406
w4	0,397033026	0,100700632	0,101157569	0,100254406
w5	0,197033026	0,400700632	0,101157569	0,200254406
w6	0,397075192	0,200690675	0,401141118	0,300254406
w7	0,397118647	0,200680414	0,201124164	0,400254406
w8	0,297135914	0,100676336	0,201117427	0,200254406
w9	0,297042145	0,200698479	0,301154011	0,300254406
W10	0,197017177	0,100704375	0,101163752	0,000254406

Untuk data kedua, dilakukan operasi yang sama namun nilai-nilai bobot dan bias awal yang digunakan adalah nilai-nilai bobot dan bias baru dari hasil perhitungan pertama. Operasi terus dilakukan hingga maksimum *epoch* ke 5000 atau kuadrat *error* (target *error*)  $\leq 0.02$ .

Untuk pengujian, digunakan bobot terakhir yang digunakan dan dilakukan operasi seperti tahap perambatan maju (*forward propagation*) hingga mendapatkan nilai fungsi aktivasi pada *outputlayer* dengan batas ambang  $\emptyset = 0,9$

#### Contoh Pengujian:

Data Latih: [0 1 1 1 0 1 0 0 1] Target = 1

#### Tahap Perambatan Maju (*Forward Propagation*)

Jumlahkan semua sinyal yang masuk kedalam persamaan (2.10)

$$\square_{\square\square}_I = (0,1 + (0 * 0,2) + (1 * 0,2) + (1 * 0,1) + (1 * 0,9) + (0 * 0,7)(1 * 0,5)(0 * 0,3)(0 * 0,3)(1 * 0,1) = 2,2$$

Demikian seterusnya sampai diperoleh  $\square_{\square\square}_{10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.38

**Tabel 4.38 Data Uji Operasi Pada *Hidden Layer***

z_in1	z_in2	z_in3	z_in4	z_in5	z_in6	z_in7	z_in8	z_in9	z_in10
1,9	2,9	2,7	2,6	2,5	2,2	1,6	1,7	2,2	2,3

Hitung Fungsi aktivasi pada *hidden layer* dengan menggunakan persamaan (2.11):

$$z_1 = \frac{1}{1+e^{-1,9}} = 0,869891526$$

Demikian seterusnya sampai diperoleh  $\square_{10}$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.39

**Tabel 4.39 Fungsi Aktivasi Pada *Hidden Layer***

Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
0,8698	0,9478	0,9370	0,9308	0,9241	0,9002	0,8320	0,8455	0,9002	0,9088

Jumlahkan semua sinyal yang masuk ke unit K dengan persamaan (2.12):

$$\begin{aligned} \square_{-} &= (0,4 + (0,8698 * 0,3) + (0,9478 * 0,2) + (0,9370 * 0,4) + \\ & (0,9308 * 0,2) + (0,9241 * 0,4) + (0,9002 * 0,4) + \\ & (0,8320 * 0,2) + (0,8455 * 0,4) + (0,9002 * 0,4) + (0,9088 * 0,2) \\ &= 3,180938238 \end{aligned}$$

Demikian seterusnya sampai diperoleh  $\square_3$ . Hasil perhitungan tersebut dapat dilihat pada Tabel 4.40

**Tabel 4.40 Data Uji Perhitungan Operasi Pada *Output Layer***

y0	y1	y2	y3
3,180938238	2,184403415	1,900821738	2,726214712

Hitung keluaran dengan fungsi aktivasi seperti pada Tabel 4.41 dibawah ini

**Tabel 4.41 Fungsi Aktivasi Pada *Output Layer***

y0	Y1	y2	y3
0,960110614	0,898840163	0,869984502	0,938555907

Nilai  $\emptyset$  yang dipakai adalah 0,9. Maka prediksi keluaran y0, y1, y2 dan y3 adalah 1 0 0 1 yang termasuk pada kelas ke 9. Pengujian ini menggunakan 1 *epoch*.

## 4.2 Perancangan

Tahap perancangan adalah tahap yang dilakukan setelah tahap analisa, Perancangan dilakukan untuk dijadikan pedoman dalam membuat sebuah aplikasi



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.

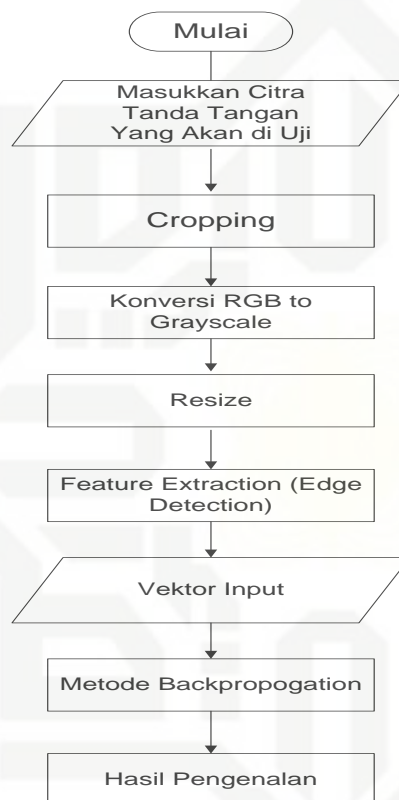
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

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

identifikasi tanda tangan. Perancangan yang dilakukan adalah *flowchart*, dan perancangan tampilan (*interface*) untuk aplikasi yang akan dibuat.

#### 4.2.1 Flowchart

Alur proses aplikasi identifikasi tanda tangan dengan menggunakan metode *Backpropagation* dapat dilihat pada Gambar 4.10 *flowchart* di bawah ini.



**Gambar 4.10 Flowchart Aplikasi Identifikasi Tanda Tangan**

Alur proses aplikasi yang dibuat dimulai dengan menginputkan citra tanda tangan berformat .png ke dalam aplikasi. Setelah itu citra tanda tangan akan *cropping* untuk meminimalkan *background* pada objek gambar. Selanjutnya, gambar yang telah dilakukan *cropping* akan dikonversikan ke *grayscale* dan kemudian akan di *resize* sehingga berukuran 100 x 100 piksel. Setelah itu akan di proses menggunakan metode *edge detection* sehingga menghasilkan vektor input berupa angka biner. Kemudian vektor *input* dari matriks tersebut akan dilakukan proses klasifikasi menggunakan metode jaringan syaraf tiruan *backpropagation* sehingga gambar yang di-*input*-kan dapat di identifikasi.

## 4.2.2 Perancangan Tampilan Aplikasi (*Interface*)

Perancangan tampilan (*interface*) penting dilakukan dalam membuat sebuah sistem/aplikasi. *Interface* menjadi sebuah sarana pengembangan yang digunakan untuk membuat komunikasi yang baik antara sistem/aplikasi dan pemakainya. Sebuah perancangan *interface* yang baik meliputi tampilan yang baik, konsisten dalam sisi bahasa, dan mudah dipahami.

Perancangan tampilan pada proses identifikasi tanda tangan akan dibuat meliputi perancangan tampilan halaman utama, perancangan tampilan identifikasi tanda tangan, perancangan tampilan *help* yang berisi cara penggunaan aplikasi dan perancangan tampilan *about creator*.

### 4.2.2.1 Perancangan Tampilan Halaman Utama

Halaman utama adalah halaman yang pertama muncul saat aplikasi dijalankan. Perancangan tampilan halaman utama meliputi *button* **Data Latih** untuk melatih seluruh citra tanda tangan yang akan diidentifikasi, *button* **Pengenalan** digunakan untuk mengidentifikasi tanda tangan, dan *button* **Keluar** untuk keluar dari aplikasi. Tampilan halaman utama dapat dilihat pada Gambar 4.11 berikut.

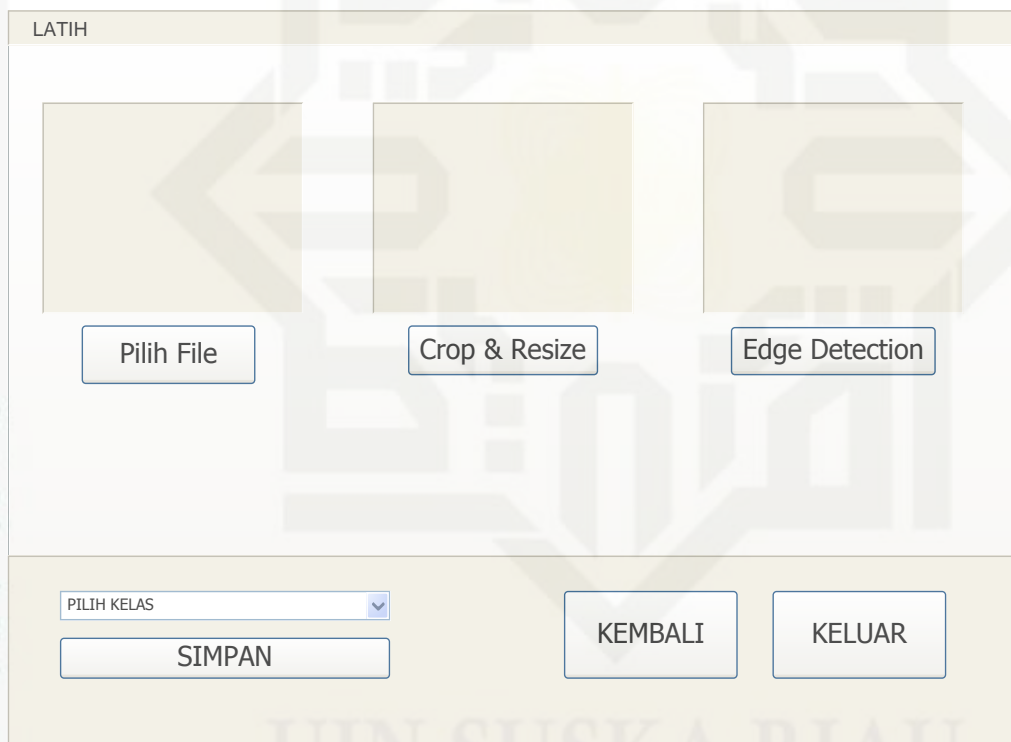


**Gambar 4.11 Perancangan Tampilan Halaman Utama**

#### 4.2.2.2 Perancangan Tampilan Data Latih

Pada perancangan tampilan identifikasi tanda tangan ini terdiri dari *button* **Pilih File** yang digunakan untuk memilih citra tanda tangan yang akan di latih, *button* **crop& Resize** digunakan untuk menyamakan ukuran seluruh gambar agar hasil yang diperoleh maksimal, dan *button* **edge detection** untuk memproses tanda tangan dengan metode deteksi tepi. Kemudian *radio button* **Pilih Kelas** untuk memilih kelas pemilik tanda tangan. *Button* **simpan** berguna untuk menyimpan data yang telah di latih. Kemudian ada *button* **kembali** untuk kembali ke halaman awal aplikasi dan *button* **keluar** untuk keluar dari aplikasi.

Perancangan Tampilan Data Latih dapat dilihat pada Gambar 4.12 berikut:



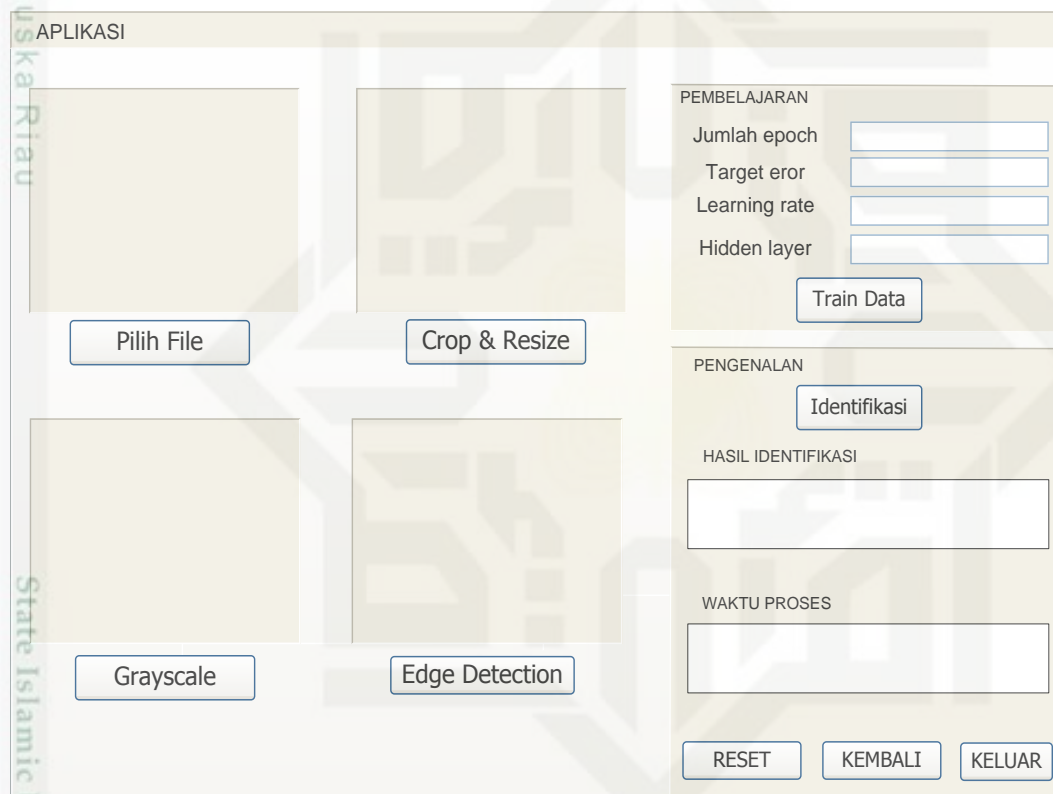
Gambar 4.12 Perancangan Tampilan Data Latih

#### 4.2.2.3 Perancangan Tampilan Pengenalan

Tampilan Pengenalanterdiri dari *button* **pilih file** untuk memilih citra tanda tangan yang akan diuji atau di identifikasi. *Button* **Crop & Resize** berfungsi untuk menyamakan ukuran gambar. Kemudian *button* **grayscale** digunakan untuk mengubah gambar RGB menjadi Grayscale image. Setelah itu terdapat *button* **Edge Detection** yang berfungsi untuk mengolah gambar dengan metode deteksi

- 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.

tepi. Di dalam gambar 4.14 dapat juga dilihat pembelajaran dengan menginputkan jumlah *epoch*, target *error*, *learning rate* dan *hidden layer*. Setelah itu dalam pengenalan terdapat **button identifikasi** dan kemudian hasil identifikasi yang berupa nama pemilik tanda tangan dan waktu proses akan muncul. **Button reset** digunakan untuk mengosongkan semua agar dapat dimulai dari awal. **Button kembali** digunakan untuk kembali kehalaman awal, dan **button keluar** digunakan untuk keluar dari aplikasi.



**Gambar 4.13 Perancangan Tampilan Pengenalan**