

## JARINGAN SARAF TIRUAN DENGAN METODE BACKPROPAGATION UNTUK MENDETEKSI GANGGUAN PSIKOLOGI

**Kiki, Sri Kusumadewi**

*Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia  
Jl. Kaliurang Km. 14 Yogyakarta 55501  
Telp. (0274) 895287 ext. 122, Faks. (0274) 895007ext. 148*

### **ABSTRACT**

*The Artificial Neural Network represents one of the system of information process which designed by imitating the way of activity of brain of human being in finishing a problem by doing process learn through the change of wight sinaps. artificial neural network can do the activity recognition base on the past data. Past data will be studied by artificial neural network, so that have the ability to give the decision to data which have never been studied.*

*Target which wish reached is system to application artificial neural network in predict or detect a disease and or natural trouble at human being such as trouble og result of psychology test, like stress, phobia, obsesif kompulsif and panic, and also design a system that able to do the task in predict the result of psychology test at human being.*

*Process by using 13 variable which is each deputizing indication of clinis of trouble stress, phobia, obsesif kompulsif, and the panic. Target wanted are 0 0 (Stress), 0 1 (Phobia), 1 0 (Obsessive Compulsive), and 1 1 (Panic). Then the veriabels trained and tested by system that can recognize it better. After done by simulation of backward error creeping neural network (backpropagation) got by a best result with the parameter learning constant 0.1 and convergent network, that is at epoch to 17041 with the error 0.0099999293. the accuracy of examination of neural network reach the result 97.5 %.*

*Keywords: Backpropagation, artificial neural network, epoch, error.*

### **1. PENDAHULUAN**

Salah satu cabang dari AI (*Artificial Intelligence*) adalah apa yang dikenal dengan Jaringan Saraf Tiruan (*Artificial Neural Network*). Jaringan saraf tiruan merupakan salah satu sistem pemrosesan informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Jaringan saraf tiruan mampu melakukan pengenalan kegiatan berbasis data masa lalu. Data masa lalu akan dipelajari oleh jaringan saraf tiruan sehingga mempunyai kemampuan untuk memberikan keputusan terhadap data yang belum pernah dipelajari. Dalam analisis ini dicoba untuk dipelajari dan dicoba penerapannya didalam bidang psikologi yaitu mendeteksi test psikologi pada manusia. JST yang

berupa susunan sel-sel saraf tiruan (*neuron*) dibangun berdasarkan prinsip-prinsip organisasi otak manusia.

Usaha manusia dalam mengembangkan suatu sistem yang meniru kemampuan dan perilaku makhluk hidup telah berlangsung selama beberapa decade belakangan ini. Jaringan saraf tiruan (JST), merupakan hasil perkembangan ilmu dan teknologi yang kini sedang berkembang pesat. JST yang berupa susunan sel-sel saraf tiruan (*neuron*) dibangun berdasarkan prinsip-prinsip organisasi otak manusia. Perhatian yang besar pada JST disebabkan adanya keunggulan yang dimilikinya seperti kemampuan untuk belajar, komputasi paralel, kemampuan untuk memodelkan fungsi *nonlinier* dan sifat *fault tolerance*.

Sejak ditemukan pertama kali oleh McCulloch dan Pitts pada tahun 1948, JST telah berkembang pesat dan telah digunakan pada banyak aplikasi. Jaringan saraf tiruan (JST) telah dikembangkan sejak tahun 1940. Belum ada definisi yang baku mengenai JST ini. Teori yang menginspirasi lahirnya sistem jaringan saraf muncul dari bermacam disiplin ilmu: terutama dari neuro science, teknik, dan ilmu komputer, juga dari psikologi, matematika, fisika, dan ilmu bahasa. Ilmu-ilmu ini bekerja bersama untuk satu tujuan yaitu pengembangan sistem kecerdasan (Dayhoff, 1990) (Kusumadewi, 2002).

## 2. TUJUAN

Pada penelitian ini akan dilakukan studi dan implementasi yang bertujuan untuk mengaplikasikan algoritma jaringan saraf tiruan dengan metode *backpropagation* dan merancang suatu sistem yang dapat melakukan tugas dalam mendeteksi kondisi psikologi berdasarkan gejala-gejala yang sering terjadi pada manusia.

## 3. LANDASAN TEORI

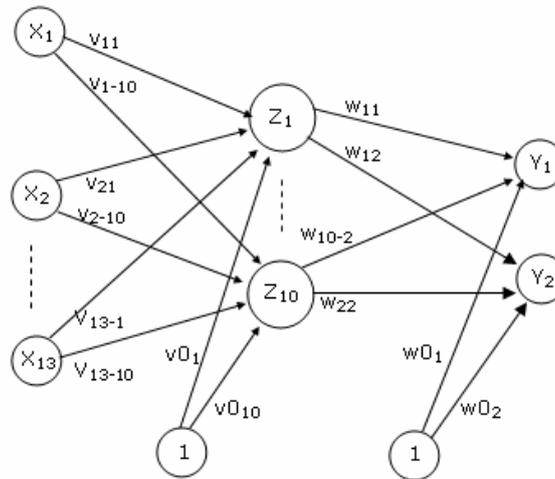
### 3.1 Metode Backpropagation

Perambatan galat mundur (*backpropagation*) adalah sebuah metode sistematis untuk pelatihan *multiplayer* jaringan saraf tiruan. Metode ini memiliki dasar matematis yang kuat, obyektif dan algoritma ini mendapatkan bentuk persamaan dan nilai koefisien dalam formula dengan meminimalkan jumlah kuadrat galat error melalui model yang dikembangkan (*training set*) (Brace, 1997).

- a. Dimulai dengan lapisan masukan, hitung keluaran dari setiap elemen pemroses melalui lapisan luar.
- b. Hitung kesalahan pada lapisan luar yang merupakan selisih antara data aktual dan target.
- c. Transformasikan kesalahan tersebut pada kesalahan yang sesuai di sisi masukan elemen pemroses.
- d. Propagasi balik kesalahan-kesalahan ini pada keluaran setiap elemen pemroses ke kesalahan yang terdapat pada masukan. Ulangi proses ini sampai masukan tercapai.
- e. Ubah seluruh bobot dengan menggunakan kesalahan pada sisi masukan elemen dan luaran elemen pemroses yang terhubung.

### 3.2 Arsitektur Jaringan Metode Backpropagation

Jaringan saraf terdiri dari 3 lapisan, yaitu lapisan masukan/input terdiri atas variabel masukan unit sel saraf, lapisan tersembunyi terdiri atas 10 unit sel saraf, dan lapisan keluaran/output terdiri atas 2 sel saraf. Lapisan masukan digunakan untuk menampung 13 variabel yaitu  $X_1$  sampai dengan  $X_{13}$ , sedangkan 2 lapisan keluaran digunakan untuk mempresentasikan pengelompokan pola, nilai 00 untuk Gangguan Stress, nilai 01 untuk Gangguan Fobia, nilai 10 untuk Gangguan Obsesif kompulsif, dan nilai 11 untuk Gangguan Panik (Brace, 1997).



Keterangan:

- X = Masukan (input)
- J = 1..n (n = 10)
- V = Bobot pada lapisan tersembunyi
- W = Bobot pada lapisan keluaran
- n = Jumlah unit pengolah pada lapisan tersembunyi
- b = Bias pada lapisan tersembunyi dan lapisan keluaran
- k = Jumlah unit pengolah pada lapisan keluaran
- Y = Keluaran hasil

Gambar 1. Arsitektur jaringan *backpropagation*

Tujuan dari perubahan bobot untuk setiap lapisan, bukan merupakan hal yang sangat penting. Perhitungan kesalahan merupakan pengukuran bagaimana jaringan dapat belajar dengan baik. Kesalahan pada keluaran dari jaringan merupakan selisih antara keluaran aktual (*current output*) dan keluaran target (*desired output*).

Langkah berikutnya adalah menghitung nilai SSE (*Sum Square Error*) yang merupakan hasil penjumlahan nilai kuadrat error neuron1 dan neuron2 pada lapisan output tiap data, dimana hasil penjumlahan keseluruhan nilai SSE akan digunakan untuk menghitung nilai RMSE (*Root Mean Square Error*) tiap iterasi (Kusumadewi, 2002).

**Sum Square Error (SSE).** SEE dihitung sebagai berikut:

- a. Hitung lapisan prediksi atau keluaran model untuk masukan pertama.

- b. Hitung selisih antara nilai luar prediksi dan nilai target atau sinyal latihan untuk setiap keluaran.
- c. Kuadratkan setiap keluaran kemudian hitung seluruhnya. Ini merupakan kuadrat kesalahan untuk contoh lain.

$$SSE = \sum_{i=1}^N (D_{ij} - f_j(X_i)) \quad (1)$$

**Root Mean Square Error (RMS Error).** Dihitung sebagai berikut:

- a. Hitung SSE.
- b. Hasilnya dibagi dengan perkalian antara banyaknya data pada latihan dan banyaknya luaran, kemudian diakarkan.

$$RMSE = \sqrt{\frac{SSE}{N * K}} \quad (2)$$

dimana

RMSE = Root Mean Square Error  
 SSE = Sum Square Error  
 N = Banyak data pada latihan  
 K = Banyak luaran.

### 3.3 Langkah-langkah Algoritma Perambatan Galat Mundur

#### a. Algoritma Pelatihan

Pelatihan suatu jaringan dengan algoritma *backpropagation* meliputi dua tahap : perambatan maju dan perambatan mundur.

Selama perambatan maju, tiap unit masukan ( $x_i$ ) menerima sebuah masukan sinyal ini ke tiap-tiap lapisan tersembunyi  $z_1, \dots, z_p$ . Tiap unit tersembunyi ini kemudian menghitung aktivasinya dan mengirimkan sinyalnya ( $z_j$ ) ke tiap unit keluaran. Tiap unit keluaran ( $y_k$ ) menghitung aktivasinya ( $y_k$ ) untuk membentuk respon pada jaringan untuk memberikan pola masukan.

Selama pelatihan, tiap unit keluaran membandingkan perhitungan aktivasinya  $y_k$  dengan nilai targetnya  $t_k$  untuk menentukan kesalahan pola tersebut dengan unit itu. Berdasarkan kesalahan ini, faktor  $\delta_k$  ( $k = 1, \dots, m$ ) dihitung.  $\delta_k$  digunakan untuk menyebarkan kesalahan pada unit keluaran  $y_k$  kembali ke semua unit pada lapisan sebelumnya (unit-unit tersembunyi yang dihubungkan ke  $y_k$ ). Juga digunakan (nantinya) untuk mengupdate bobot-bobot antara keluaran dan lapisan tersembunyi. Dengan cara yang sama, faktor ( $j = 1, \dots, p$ ) dihitung untuk tiap unit tersembunyi  $z_j$ . Tidak perlu untuk menyebarkan kesalahan kembali ke lapisan masukan, tetapi  $\delta_j$  digunakan untuk mengupdate bobot-bobot antara lapisan tersembunyi dan lapisan masukan.

Setelah seluruh faktor  $\delta$  ditentukan, bobot untuk semua lapisan diatur secara serentak. Pengaturan bobot  $w_{jk}$  (dari unit tersembunyi  $z_j$  ke unit keluaran  $y_k$ ) didasarkan pada faktor  $\delta_k$  dan aktivasi  $z_j$  dari unit tersembunyi  $z_j$ . didasarkan pada faktor  $\delta_j$  dan dan aktivasi  $x_i$  unit masukan. Untuk langkah selengkapnya adalah (Fausset, 1994):

#### b. Prosedur Pelatihan

Langkah 0: Inisialisasi bobot. (sebaiknya diatur pada nilai acak yang kecil),

Langkah 1: Jika kondisi tidak tercapai, lakukan langkah 2-9,

Langkah 2: Untuk setiap pasangan pelatihan, lakukan langkah 3-8,

### Perambatan Maju

Langkah 3: Tiap unit masukan ( $x_i, i = 1, \dots, n$ ) menerima sinyal  $x_i$  dan menghantarkan sinyal ini ke semua unit lapisan di atasnya (unit tersembunyi),

Langkah 4 : Setiap unit tersembunyi ( $x_i, i = 1, \dots, p$ ) jumlahkan bobot sinyal masukannya,

$$z\_in_j = v_{oj} + \sum_{i=1}^n x_i v_{ij} \quad (3)$$

$v_{oj}$  = bias pada unit tersembunyi  $j$  aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya,  $z_j = f(z\_in_j)$ , dan kirimkan sinyal ini keseluruh unit pada lapisan di atasnya (unit keluaran).

Langkah 5: Tiap unit keluaran ( $y_k, k = 1, \dots, m$ ) jumlahkan bobot sinyal masukannya,

$$y\_in_k = w_{ok} + \sum_{j=1}^n z_j w_{jk} \quad (4)$$

$w_{ok}$  = bias pada unit keluaran  $k$  dan aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya,  $y_k = f(y\_in_k)$ .

### Perambatan Mundur

Langkah 6: Tiap unit keluaran ( $y_k, k = 1, \dots, m$ ) menerima pola target yang saling berhubungan pada masukan pola pelatihan, hitung kesalahan informasinya,

$$\delta_k = (t_k - y_k) f'(y\_in_k) \quad (5)$$

hitung koreksi bobotnya (digunakan untuk memperbaharui wjk nantinya),

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (6)$$

hitung koreksi biasnya (digunakan untuk memperbaharui  $w_{ok}$  nantinya), dan kirimkan  $\delta_k$  ke unit-unit pada lapisan dibawahnya,

Langkah 7: Setiap unit lapisan tersembunyi ( $z_j, j = 1, \dots, p$ ) jumlahkan hasil perubahan masukannya (dari unit-unit lapisan di atasnya),

$$\Delta\_in_j = \sum_{k=1}^m \delta_k w_{jk} \quad (7)$$

kalikan dengan turunan fungsi aktivasinya untuk menghitung informasi kesalahannya,

$$\delta_j = \Delta\_in_j f'(z\_in_j) \quad (8)$$

hitung koreksi bobotnya (digunakan untuk memperbaharui  $v_{oj}$  nanti),

Langkah 8: Tiap unit keluaran ( $y_k, k = 1.. m$ ) update bias dan bobotnya ( $j = 0, \dots, p$ ):

$$w_{jk} \text{ (baru)} = w_{jk} \text{ (lama)} + \Delta w_{jk} \quad (9)$$

Tiap unit lapisan tersembunyi ( $z_j, j = 1, \dots, p$ ) update bias dan bobotnya

( $i = 0, \dots, n$ ) :

$$v_{ij} \text{ (baru)} = v_{ij} \text{ (lama)} + \Delta v_{ij} \quad (10)$$

Langkah 9: Test kondisi berhenti.

### c. Prosedure Pengujian

Setelah pelatihan, jaringan saraf *backpropagation* diaplikasikan dengan hanya menggunakan tahap perambatan maju dari algoritma pelatihan. Prosedur aplikasinya adalah sebagai berikut:

Langkah 0: Inisialisasi bobot (dari algoritma pelatihan).

Langkah 1: Untuk tiap vektor masukan, lakukan langkah 2-4.

Langkah 2: for  $i = 1, \dots, n$  : atur aktivasi unit masukan  $x_i$ .

Langkah 3: for  $j = 1, \dots, p$  :

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (11)$$

$$z_j = f(z\_in_j) \quad (12)$$

Langkah 4: for  $k = 1, \dots, m$  :

$$y\_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (13)$$

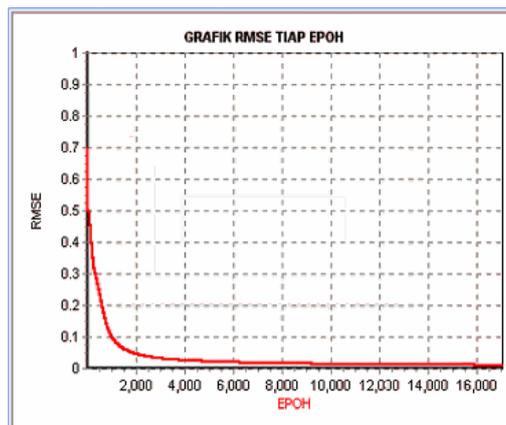
$$y_k = f(y\_in_k) \quad (14)$$

Langkah 5: Jika  $y_k \geq 0,5$  maka  $y_k = 1$ , else  $y_k = 0$ .

## 4. IMPLEMENTASI DAN ANALISIS

### 4.1 Implementasi Hasil

Pelatihan dilakukan terhadap terhadap data-data pelatihan dengan target error 0,01; learning rate 0,1; dan maksimum epoch 50000; menghasilkan RMSE 0,0099 pada epoch ke 17041 (Gambar 2).



Gambar 2 . Hasil pelatihan

Bobot antara lapisan input dan lapisan tersembunyi akhir (V) dan bias antara lapisan input dan lapisan tersembunyi akhir setelah iterasi ke-17041 seperti terlihat pada Tabel 1 dan Tabel 2.

Tabel 1. Bobot akhir lapisan input ke lapisan tersembunyi

i \ j	$v_{ij}$									
	1	2	3	4	5	6	7	8	9	10
1	-2.8701	6.7849	0.6292	0.6982	0.4418	0.4664	0.7383	1.8366	1.013	1.401
2	4.0473	2.5613	0.5111	-1.938	0.2676	-0.3507	0.3829	2.5675	1.5531	-0.5839
3	4.2825	5.2062	-0.2435	3.1664	0.7431	0.0059	0.5706	2.6897	0.5338	-0.0926
4	-0.3247	1.1968	0.3717	0.5196	-0.0295	0.6197	0.7639	-0.9933	0.1727	1.1283
5	-5.7136	-0.2981	1.4896	-1.1177	1.1668	-0.3022	0.4901	6.237	3.083	3.1617
6	-0.9437	2.4767	0.7083	-8.4186	0.3936	0.2923	0.9557	-1.3401	-0.3916	0.2966
7	-1.8774	-0.4856	1.0271	1.2319	0.7074	-0.01	0.3117	1.4324	1.6391	2.0804
8	2.7029	3.4916	0.3689	2.71	0.6514	0.7891	0.7953	-0.7518	0.2109	1.5125
9	-1.317	-2.6859	0.6333	0.7197	0.749	0.7153	0.5377	-2.5414	-0.8222	-0.8309
10	2.0315	-2.566	0.6573	2.1861	0.4587	0.6341	0.205	1.2248	0.7449	0.9816
11	3.346	-3.6697	1.1956	5.6243	0.0652	-0.1762	0.9849	1.7048	1.0011	-5.3371
12	2.4777	-3.3917	-0.3299	-0.292	0.7671	0.0384	0.5633	0.9817	-0.2008	0.1743
13	-2.0597	2.1161	-0.256	1.5764	0.9008	0.6957	0.211	-0.0587	-0.3585	0.1139

Tabel 2. Bias akhir lapisan input ke lapisan tersembunyi

$v_{0j}$									
1	2	3	4	5	6	7	8	9	10
-0,4845	-2.7926	0.4219	0.8961	0.9161	-0.1227	0.4678	-0.0062	-0.6224	-1.1519

Bobot akhir lapisan tersembunyi ke lapisan output (w), dan bias akhir lapisan tersembunyi ke lapisan output (w<sub>0</sub>), terlihat pada Tabel 3 dan 4.

Tabel 3. Bobot akhir lapisan tersembunyi ke lapisan output

i \ j	$w_{ij}$	
	1	2
1	-13.1121	-1.2601
2	2.0026	-10.9415
3	0.823	2.5154
4	-5.4238	-11.7636
5	0.7728	1.565
6	0.7715	0.5964
7	0.2805	1.779
8	1.332	9.9133
9	1.8441	3.1371
10	4.7243	-4.935

Tabel 4. Bias akhir lapisan tersembunyi ke lapisan output

$w_{0j}$	
1	2
0.595	2.2867

Setelah dilakukan pembelajaran, dan pengujian pola-pola yang dilatih, diperoleh hasil bahwa 97,5 % test terhadap pola-pola tersebut benar (akurat). Gambar 3, 4, 5, dan 6 adalah contoh implementasi *pengujian* yang dilakukan terhadap pola nilai yang ikut dilatih.

latihan  
Pelatihan Cek Prediksi Simulasi Prediksi

**Aplikasi JST Pada Prediksi Hasil Test Psikologi**

Kode Pasien : 8 Nama Pasien : h

Cemas	0.5	Kurang Beraktivitas	1
Gangguan Tidur	1	Berkeringat	0.5
Kelelahan	0.5	Pelupa	0.5
Nyeri Kepala	0	Depresi	0.5
ketegangan Hidup	1	Kurang Istirahat	1
jantung Berdebar	0.5	Marah-Marah	0.5
Merasa Takut	0.25		

Target Gangguan :

Kode Gangguan 00 Hasil Gangguan : STRESS (0 0)

Nama Gangguan STRESS (0 0)

Data Pasien Testing Label Hasil Selesai

Gambar 3 . Hasil testing terhadap nilai untuk data kasus 00 (stress)

latihan  
Pelatihan Cek Prediksi Simulasi Prediksi

**Aplikasi JST Pada Prediksi Hasil Test Psikologi**

Kode Pasien : 14 Nama Pasien : n

Cemas	0.25	Kurang Beraktivitas	1
Gangguan Tidur	0	Berkeringat	1
Kelelahan	0.5	Pelupa	1
Nyeri Kepala	1	Depresi	0.75
ketegangan Hidup	0.5	Kurang Istirahat	0
jantung Berdebar	0	Marah-Marah	0.25
Merasa Takut	0.5		

Target Gangguan :

Kode Gangguan 01 Hasil Gangguan : FOBIA (0 1)

Nama Gangguan FOBIA (0 1)

Data Pasien Testing Label Hasil Selesai

Gambar 4. Hasil testing terhadap nilai untuk data kasus 01 (fobia)

latihan  
Pelatihan | Cek Prediksi | Simulasi Prediksi

### Aplikasi JST Pada Prediksi Hasil Test Psikologi

Kode Pasien :       Nama Pasien :

Cemas	<input type="text" value="0.5"/>	Kurang Beraktivitas	<input type="text" value="0.25"/>
Gangguan Tidur	<input type="text" value="0"/>	Berkeringat	<input type="text" value="0.5"/>
Kelelahan	<input type="text" value="0.5"/>	Pelupa	<input type="text" value="0.5"/>
Nyeri Kepala	<input type="text" value="1"/>	Depresi	<input type="text" value="0.5"/>
ketegangan Hidup	<input type="text" value="0.5"/>	Kurang Istirahat	<input type="text" value="0"/>
jantung Berdebar	<input type="text" value="1"/>	Marah-Marah	<input type="text" value="0.5"/>
Merasa Takut	<input type="text" value="0.5"/>		

Target Gangguan :

Kode Gangguan  Hasil Gangguan :

Nama Gangguan

Gambar 5. Hasil testing terhadap nilai untuk data kasus 10 (obsesif kompulsif)

latihan  
Pelatihan | Cek Prediksi | Simulasi Prediksi

### Aplikasi JST Pada Prediksi Hasil Test Psikologi

Kode Pasien :       Nama Pasien :

Cemas	<input type="text" value="0.25"/>	Kurang Beraktivitas	<input type="text" value="0.5"/>
Gangguan Tidur	<input type="text" value="1"/>	Berkeringat	<input type="text" value="1"/>
Kelelahan	<input type="text" value="0.5"/>	Pelupa	<input type="text" value="0.5"/>
Nyeri Kepala	<input type="text" value="1"/>	Depresi	<input type="text" value="0"/>
ketegangan Hidup	<input type="text" value="1"/>	Kurang Istirahat	<input type="text" value="0"/>
jantung Berdebar	<input type="text" value="0.5"/>	Marah-Marah	<input type="text" value="0.25"/>
Merasa Takut	<input type="text" value="0.25"/>		

Target Gangguan :

Kode Gangguan  Hasil Gangguan :

Nama Gangguan

Gambar 6. Hasil testing terhadap nilai untuk data kasus 11 (panik)

Setelah dilakukan pelatihan dengan berbagai macam kombinasi Pelatihan (pembelajaran), nilai Error (*The Last RMSE*) tidak tergantung pada besarnya nilai *Learning Rate*, tetapi jumlah iterasi (*epoch*) semakin kecil. Dan ternyata hasil yang paling baik adalah dengan menggunakan kombinasi pembelajaran 0.1 dengan target error 0.01, karena dengan kombinasi pembelajaran tersebut hasil target yang akan diinginkan dapat tercapai.

Dari hasil pengujian sistem aplikasi model jaringan saraf tiruan yang telah dibuat dengan konstanta belajar 0,1, keberhasilan dalam memprediksi hasil test psikologi sebesar 97,5 %. Tabel 5 menunjukkan hasil pengujian dengan beberapa learning rate.

Tabel 5 . Hasil pelatihan dengan kombinasi tingkat pembelajaran

<i>Learning Rate</i>	<i>The Last RMSE</i>	<i>Sum Epoh ( Iterasi )</i>	<i>Hasil Persentase</i>
0.1	0.00999992930	17041	97,5 %
0.2	0.00999997321	9326	95 %
0.3	0.00999931386	5971	95 %
0.4	0.00999901269	4335	92,5 %
0.5	0.00999936690	3248	95 %
0.6	0.00999858438	2494	92,5 %
0.7	0.0099987996	2148	92,5 %
0.8	0.00999740942	1855	90 %
0.9	0.00999689805	1549	95 %

*Keterangan : Target Error = 0.01, The Last RMSE ≤ Target Error.*

## 5. KESIMPULAN

Dari hasil penelitian, dapat disimpulkan bahwa metode *Backpropagation* dapat digunakan untuk melakukan pendeteksian suatu jenis penyakit, gangguan, maupun kasus yang memiliki data masa lalu, dan dengan menggunakan metode *Backpropagation*, target output yang diinginkan lebih mendekati ketepatan dalam melakukan pengujian, karena terjadi penyesuaian nilai bobot dan bias yang semakin baik pada proses pelatihan.

## PUSTAKA

- Dayhoff, J. E. *Neural Network Architectures*. Van Nostrand Reinhold, New York, 1990.
- Fausset, L. (1994). *Fundamental of Neural Network:: Architecture, Algorithm, and Application*. New Jersey: Prentice-Hall.
- Brace, H. (1997). *Pengantar Psikologi*. Edisi 11, Jilid 2. Jakarta.
- Kusumadewi, S. (2002) *Buku Ajar Kecerdasan Buatan*. Yogyakarta: Jurusan Teknik Informatika, Universitas Islam Indonesia.