



Jaringan Syaraf Tiruan untuk Memprediksi Prestasi Siswa Menggunakan Algoritma Backpropagation

Firdaus

Jurusan Teknik Informatika, STMIK Amik Riau
firdauskoto@stmik-amik-riau.ac.id

Abstrak

Pemberian beasiswa sering tidak tepat di setiap sekolah menengah kejuruan karena waktu pemberian beasiswa tidak menentu dan tidak terjadwal sehingga tidak tepat sasaran pada siswa yang berprestasi. Sistem jaringan syaraf tiruan menggunakan metode algoritma backpropagation yang diterapkan dalam penelitian ini untuk memprediksi prestasi siswa, dan diharapkan dapat membantu memecahkan masalah pemberian beasiswa terhadap siswa yang berprestasi. Dengan menggunakan jaringan syaraf tiruan algoritma backpropagation proses prediksi lebih cepat, akurat, minimalisir kesalahan dan bisa menggunakan teknologi komputer. Serta mudah dalam pengembangannya. Semakin kecil tingkat ketelitian error yang digunakan maka akan semakin kecil penyimpangan hasil jaringan syaraf tiruan dengan target yang di inginkan. Sistem jaringan syaraf tiruan ini diimplementasikan menggunakan matlab sebagai software pendukung.

Kata Kunci : Jaringan Syaraf Tiruan, Prediksi, Prestasi Siswa, Algoritma Backpropagation

1. Pendahuluan

1.1 Latar Belakang Masalah

Perkembangan ilmu pengetahuan dan teknologi sangat pesat dari waktu ke waktu, Teknologi menjadi sebuah kebutuhan dalam segala bidang. Salah satunya di SMKN 3 Pariaman untuk menentukan beasiswa, karena selama ini selalu ada pemberian beasiswa

terhadap siswa yang berprestasi dengan keadaan keluarga yang kurang mampu. Namun pemberian beasiswa sering tidak tepat karena waktu pemberian beasiswa ini tidak menentu, dan sering kali permintaan nama-nama siswa itu mendesak hanya satu atau dua hari harus dikirimkan ke dinas pendidikan atau pun lembaga swasta yang memberikan beasiswa.

Penelitian ini memusatkan kepada pembuatan jaringan syaraf tiruan dalam memprediksi prestasi siswa dan menggunakan algoritma *backpropagation* dalam penyelesaiannya. Jika selama ini penentuan prestasi siswa masih secara manual maka penerapan ini merupakan konsep awal suatu metode untuk memprediksi siswa menggunakan komputer yang didukung oleh jaringan syaraf tiruan dengan sistem pemrosesan informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot dan mampu melakukan pengenalan kegiatan berbasis data masa lalu yang akan dipelajari oleh jaringan syaraf tiruan sehingga mempunyai kemampuan untuk memberikan keputusan yang tepat pada sasaran yang diharapkan.

Dengan jaringan syaraf tiruan menggunakan metode *backpropagation* komputer di fungsikan sebagai alat untuk memprediksi prestasi siswa dengan tidak mempertimbangkan faktor-faktor lain penyebab kemerosotan atau peningkatan prestasi siswa, misalnya faktor lingkungan, fasilitas, motivasi belajar dan guru.

1.2 Rumusan Masalah

Secara garis besar masalah yang harus dipahami pada penelitian ini adalah sebagai berikut:

1. Bagaimana menggunakan jaringan syaraf tiruan untuk memprediksi prestasi siswa dengan metode algoritma backpropagation?
2. Bagaimana output yang dihasilkan oleh sistem menjadi bahan pertimbangan untuk menentukan siswa-siswa yang berprestasi untuk mendapatkan beasiswa?

1.3 Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut:

1. Menentukan prestasi siswa menggunakan pendekatan jaringan saraf tiruan.
2. Jaringan saraf tiruan di implementasikan sehingga menghasilkan sebuah sistem yang siap untuk menyelesaikan masalah prediksi prestasi siswa.

2. Landasan Teori

2.1 Pengertian Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (*artificial neural network*) adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel syaraf biologis di dalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. (Hermawan, Arief. 2006).

Jaringan syaraf biologis merupakan kumpulan dari sel-sel syaraf (*neuron*). *Neuron* mempunyai tugas mengolah informasi. Komponen-komponen utama dari sebuah *neuron* dapat dikelompokkan menjadi 3 bagian (Puspitanigrum, Diyah. 2006), yaitu :

1. Dendrit bertugas untuk menerima informasi
2. Badan Sel (Soma) berfungsi sebagai tempat pengolahan informasi
3. Akson (neurit) mengirimkan impuls-impuls ke sel syaraf lainnya.

2.1.1 Konsep Dasar Jaringan Syaraf Tiruan

Kerangka kerja jaringan syaraf tiruan bisa dilihat dari jumlah lapisan (layer) dan jumlah node pada setiap lapisan. Lapisan-lapisan penyusunan jaringan syaraf tiruan dapat dibagi menjadi 3, yaitu :

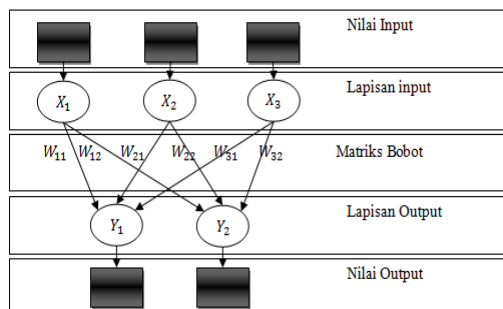
1. Lapisan Input, node-node di dalam lapisan input disebut unit-unit. Unit-unit input menerima input dari dunia luar. Input yang dimasukkan merupakan penggambaran dari suatu masalah.
2. Lapisan Tersembunyi, node-node dalam lapisan ini disebut unit-unit tersembunyi. Output dari lapisan ini secara langsung dapat diamati.

Lapisan Output, node-node pada lapisan output disebut unit-unit output. Keluaran atau output dari lapisan ini merupakan output jaringan syaraf tiruan terhadap suatu permasalahan.

2.1.2 Arsitektur Jaringan Syaraf Tiruan

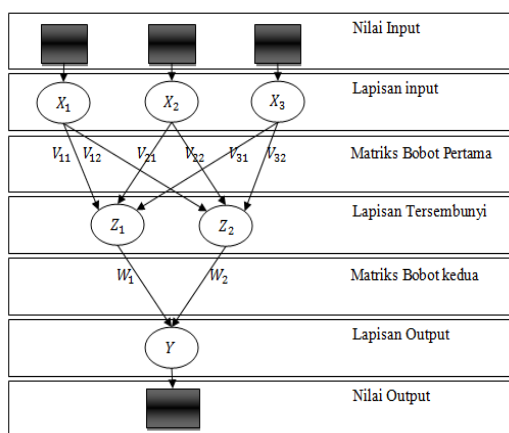
Arsitektur jaringan syaraf tiruan ada 3 yaitu :

1. Jaringan lapis tunggal (*single layer net*), jaringan yang hanya memiliki satu buah lapisan dengan bobot terhubung.



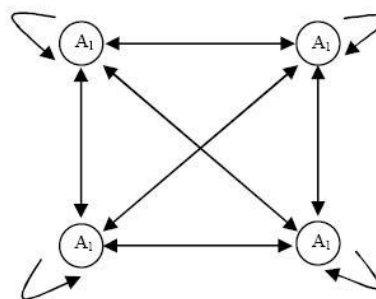
Gambar 1. Jaringan syaraf tiruan lapisan tunggal

2. Jaringan dengan banyak lapisan (*multilayer net*), merupakan jaringan dengan satu atau lebih lapisan tersembunyi



Gambar 2. Jaringan Syaraf Tiruan Dengan Banyak Lapisan

3. Jaringan dengan lapisan kompetitif (*competitive layer net*), pada jaringan ini sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif.



Gambar 3. Jaringan Syaraf Tiruan Dengan Kompetitif

2.1.3 Algoritma Umum Jaringan Syaraf Tiruan

Algoritma pembelajaran/pelatihan jaringan syaraf tiruan (Puspitanigrum, Diyah. 2006), dimasukkan contoh pelatihan ke dalam jaringan syaraf tiruan. Lakukan :

1. Inisialisasi bobot-bobot jaringan , set $i = 1$.
2. Masukkan contoh ke- i (dari sekumpulan contoh pembelajaran yang terdapat dalam set pelatihan) ke dalam jaringan pada lapisan input.
3. Cari tingkat aktivasi unit-unit output menggunakan algoritma aplikasi
If kinerja jaringan memenuhi standar yang ditentukan sebelumnya (memenuhi syarat bersih).
Then exit.
4. Update bobot-bobot dengan menggunakan aturan pembelajaran jaringan
5. If $i = n$, then reset $i = 1$,
Else $i = i - 1$,
Ke langkah 2.

2.2 Jaringan Syaraf Tiruan Algoritma Backpropagation

Jaringan syaraf tiruan algoritma *backpropagation* adalah jaringan syaraf tiruan dengan topologi multi lapis (*multilayer*) dengan satu lapis masukan (lapis X), satu atau lebih lapis *hidden* atau tersembunyi (lapis Z) dan satu lapis keluaran (lapis Y)(Aribowo, 2010). Setiap lapis memiliki *neuron-neuron* (unit-unit). Diantara *neuron* pada satu lapis berikutnya dihubungkan dengan model koneksi yang memiliki bobot (*weight*). Lapis tersembunyi dapat memiliki bias yang memiliki bobot sama.

2.2.1 Arsitektur Backpropagation

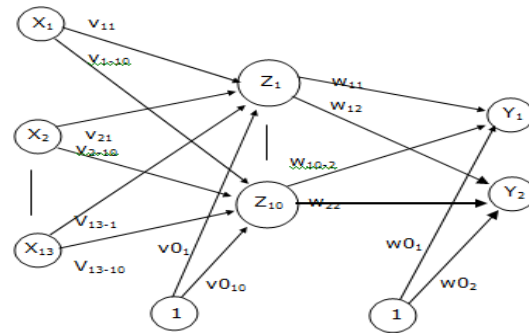
Backpropagation merupakan suatu teknik pembelajaran/pelatihan *supervised learning* yang paling banyak digunakan. Metode ini merupakan salah satu metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks(Effendy & Faisal, 2008).

Di dalam jaringan propagasi balik, setiap unit yang berada di lapisan input terhubung dengan setiap unit yang ada di lapisan tersembunyi. Setiap unit yang ada di lapisan tersembunyi terhubung dengan setiap unit yang ada di lapisan output. Jaringan ini terdiri dari banyak lapisan (*multilayer network*). Ketika jaringan diberikan pola masukan sebagai pola pelatihan, maka pola tersebut menuju unit-unit lapisan tersembunyi untuk selanjutnya diteruskan pada unit-unit di lapisan keluaran. Kemudian unit-unit lapisan keluaran akan memberikan respon

sebagai keluaran jaringan syaraf tiruan. Saat hasil keluaran tidak sesuai dengan yang diharapkan, maka keluaran akan disebarkan mundur (*backward*) pada lapisan tersembunyi kemudian dari lapisan tersembunyi menuju lapisan masukandibawah gambar (rata tengah). Keterangan tabel posisinya diatas tabel (rata tengah).

2.2.2 Arsitektur Backpropagation

Model propagasi balik memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi. Gambar 4 adalah arsitektur model propagasi balik dengan n buah masukan (ditambah satu bias), sebuah layer tersembunyi yang terdiri dari p unit (ditambah sebuah bias) serta n buah unit keluaran(Mastur & Hadi, 2005).



Gambar 4. Arsitektur *backpropagation*

Di mana :

- X = Masukan (input).
- $J = 1$ s/d 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

2.2.3 Algoritma Backpropagation

Algoritma pelatihan jaringan syaraf tiruan propagasi balik terbagi 2 langkah, yaitu :

1. Langkah maju (*Feed Forward*), dimana perhitungan bobot-bobot neuron hanya didasarkan pada vector masukan dan tidak mempunyai hubungan berulang (*loop*) didalamnya. Contoh : perceptron banyak banyak lapisan, *radial basis function*, *delta neural network*.
2. Propagasi balik (*Back Propagation*), dimana bobot diperhalus dengan memperhitungkan nilai target alau keluaran dan mempunyai hubungan berulang (*loop*) didalamnya yang menghubungkan antar syarafnya. Contoh : *elman neural network*, *Jordan network*, *times delay neural network*.

2.3 Peramalan dengan Jaringan Syaraf Tiruan Backpropagation

Algoritma propagasi balik untuk *neural network* umumnya diterapkan pada *perceptron* berlapis banyak (*multilayer perceptrons*). *Perceptron* paling tidak mempunyai bagian input, bagian output dan beberapa lapis yang berada diantara input dan output. Lapis ditengah ini, yang juga dikenal dengan lapis tersembunyi (*hidden layers*), bisa satu, dua, tiga dan seterusnya. Dalam praktek, banyaknya *hidden layer* paling banyak adalah tiga lapis. Dengan tiga lapis ini hampir semua permasalahan dalam dunia industri telah bisa diselesaikan.

Untuk penerapan peramalan, banyaknya lapis yang diperlukan cukup dua lapis. Hanya dengan dua layer, *perceptron neural network* hampir dapat mengepaskan semua bentuk kurva data runtun waktu yang ada. Banyaknya *neuron* yang diperlukan bisa divariasi.

3. Hasil dan Pembahasan

3.1 Pengujian Hasil dan Pengolahan Data Prediksi Prestasi Siswa

Untuk melakukan pengujian hasil pengolahan data prediksi prestasi siswa, maka data akan dilakukan pengujian dengan menggunakan *software* Matlab 6.1. Matlab merupakan perangkat lunak yang cocok dipakai sebagai alat komputasi yang melibatkan penggunaan matriks dan vektor. Matlab adalah singkatan dari *Matrix Laboratory*, merupakan bahasa pemrograman yang dikembangkan oleh *The MathworkInc*. Matlab dibuat untuk memudahkan perhitungan persamaan linier, program linier, sistem yang kompleks seperti pengolahan citra dan lain sebagainya.

Matlab menyediakan fungsi-fungsi khusus untuk menyelesaikan model jaringan syaraf tiruan, pemakai hanya tinggal memasukan vektor masukan, target, model dan parameter yang diinginkan.

3.2. Pemrograman Jaringan Syaraf Tiruan (Backpropagation)

Jaringan Syaraf Tiruan (*Backpropagation*) dibentuk dengan membuat generalisasi aturan pelatihan dalam model Widrow-Hooff dengan cara menambahkan lapisan tersembunyi (*hidden layer*). Standar Metode *backpropagation* menggunakan algoritma penurunan gradien (*gradient descent*).

Variasi terhadap model standar dilakukan dengan mengganti algoritmanya dengan algoritma lain. Hasil percobaan menunjukkan bahwa Jaringan Syaraf Tiruan (*backpropagation*) yang sudah dilatih dengan baik akan memberikan keluaran yang masuk akal jika diberi

masukan yang serupa dengan pola yang dipakai untuk pelatihan. Sifat generalisasi ini membuat pelatihan lebih efisien karena tidak perlu dilakukan pada semua data.

3.2. Langkah-langkah Pengolahan Data Manual dengan Matlab

Adapun langkah-langkah pengolahan data menggunakan Matlab ditetapkan sebagai berikut:

1. Menetapkan Data Input dan Target untuk Pelatihan serta Data Input dan Target Pengujian.

Dari 60 data dibagi menjadi dibagi menjadi 40 data untuk pelatihan dan 20 data untuk pengujian dengan menggunakan pola arsitektur 4-2-1. Pembagian tersebut dapat dilihat pada table 1. dan tabel 2 berikut:

Tabel 1. Data input dan target jaringan untuk data pelatihan

NO	DATA INPUT PELATIHAN				TARGET
	X1	X2	X3	X4	
1	0.54	0.58	0.65	0.61	0.57
2	0.81	0.90	0.80	0.61	0.90
3	0.37	0.54	0.80	0.39	0.49
4	0.59	0.64	0.75	0.39	0.60
5	0.81	0.71	0.50	0.75	0.72
6	0.63	0.64	0.80	0.90	0.78
7	0.50	0.64	0.65	0.54	0.57
8	0.68	0.84	0.85	0.83	0.90
9	0.72	0.80	0.70	0.61	0.78
10	0.81	0.80	0.75	0.83	0.89
20	0.63	0.84	0.65	0.83	0.80
30	0.59	0.22	0.65	0.68	0.43
39	0.59	0.10	0.30	0.25	0.10
40	0.68	0.77	0.50	0.25	0.56

Tabel 2. Data input dan target jaringan untuk data pengujian

NO	DATA INPUT PELATIHAN				TARGET
	X1	X2	X3	X4	
1	0.60	0.60	0.56	0.20	0.48
2	0.90	0.50	0.67	0.40	0.64
3	0.85	0.43	0.79	0.30	0.60
4	0.80	0.87	0.61	0.80	0.90
5	0.70	0.47	0.27	0.30	0.33
6	0.80	0.73	0.10	0.40	0.49
7	0.10	0.73	0.84	0.70	0.61
8	0.50	0.37	0.50	0.70	0.40

9	0.70	0.67	0.56	0.30	0.58
10	0.70	0.83	0.56	0.50	0.74
11	0.60	0.83	0.73	0.90	0.87
12	0.55	0.10	0.44	0.20	0.10
13	0.60	0.47	0.67	0.40	0.50
14	0.45	0.30	0.27	0.20	0.11
15	0.80	0.67	0.56	0.10	0.57
16	0.60	0.67	0.79	0.30	0.65
17	0.50	0.90	0.50	0.60	0.70
18	0.80	0.63	0.56	0.60	0.68
19	0.55	0.40	0.90	0.90	0.67
20	0.60	0.50	0.39	0.40	0.39

2. Menetapkan Bobot dan Bias

Bobot awal lapisan *input* ke lapisan tersembunyi pertama adalah pada tabel 3 berikut:

Tabel 3. Tabel bobot awal dan bobot awal bias *input* ke lapisan tersembunyi.

	Z_1	Z_2
X_1	0.58	-0.34
X_2	0.14	-0.02
X_3	0.46	0.30
X_4	-0.06	-0.70
Bias	-0.93	-1.96

Bobot awal dan bobot bias awal lapisan tersembunyi ke lapisan *output* seperti tabel 4. berikut:

Tabel 4. Tabel bobot awal dan bias lapisan tersembunyi ke *output*.

	Y
Z_1	0.64
Z_2	-0.11
Bias	0.23

3. Menetapkan Parameter-parameter yang digunakan

Sebelum data diuji dalam proses pelatihan, maka perlu ditetapkan terlebih dahulu parameter-parameter yang diperlukan dalam proses pelatihan. Adapun parameter-parameter yang diperlukan adalah sebagai berikut:

```
>> net.trainParam.epochs=5000;
```

Parameter ini digunakan untuk menentukan jumlah *epochs* maksimum pelatihan.

```
>> net.trainParam.goal=0.01;
```

Parameter ini digunakan untuk menentukan batas nilai MSE agar iterasi dihentikan. Iterasi akan berhenti jika $MSE < \text{batas}$ yang ditentukan dalam `net.trainParam.goal` atau jumlah *epochs* yang telah ditentukan dalam `net.trainParam.epochs`.

```
>> net.trainParam.lr=0.1;
```

Parameter ini digunakan untuk menentukan laju pemahaman ($\alpha = \text{learning rate}$). *Default*=0.01, semakin besar nilai α , maka semakin cepat pula proses pelatihan. Akan tetapi jika nilai α terlalu besar, maka algoritma menjadi tidak stabil dan mencapai titik *minimum local*.

```
>> net.trainParam.show=200;
```

Parameter ini digunakan untuk menampilkan frekuensi perubahan MSE (*default*:setiap 25 *epochs*).

```
>> net.trainParam.mc=0.8;
```

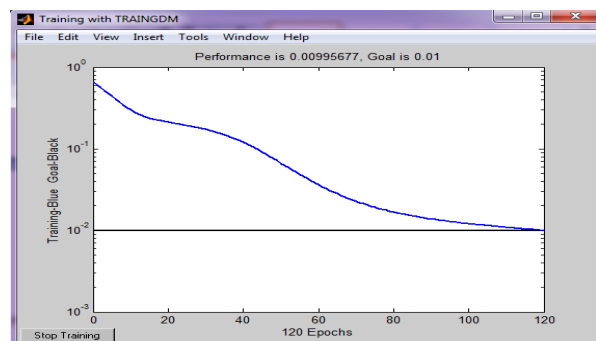
Parameter ini digunakan untuk menentukan besarnya faktor momentum dengan memberikan nilai 0-1 (*default*-nya : 0.9).

4. Melakukan Pelatihan dan Pembelajaran

Proses pembelajaran atau pelatihan dapat dilakukan dengan menggunakan perintah sebagai berikut:

```
>>net=train(net,rn,tn);
```

Dengan perintah di atas akan menghasilkan keluaran seperti gambar 5.



Gambar 5. Hasil pembelajaran atau pelatihan sampai 5000 *epochs*

TRAININGDM, Epoch 0/5000, MSE 0.656699/0.01, Gradient 0.69411/1e-010

TRAININGDM, Epoch 120/5000, MSE 0.0095677/0.01, Gradient 0.0286413/1e010

TRAININGDM, Performance goal met.

Pelatihan (*training*) akan dihentikan pada *epochs* yang ke 5000, goal yang diinginkan adalah 0.01 ($MSE = 0.0095677$).

5. Melakukan Pengujian terhadap data pelatihan

Proses pengujian terhadap 40 data pelatihan dilakukan dengan cara simulasi menggunakan perintah sebagai berikut sehingga diperoleh nilai target, output, dan *error* seperti pada tabel 5.5:

```
>> %Melakukan Simulasi
```

```
>> an=sim(net,rn);
```

```
>> a=poststd(an,meant,stdt);
```

```
>> H=[(1:size(P,2))'T'a'(T'-a)'];
```

Tabel 5. Pengujian data pelatihan

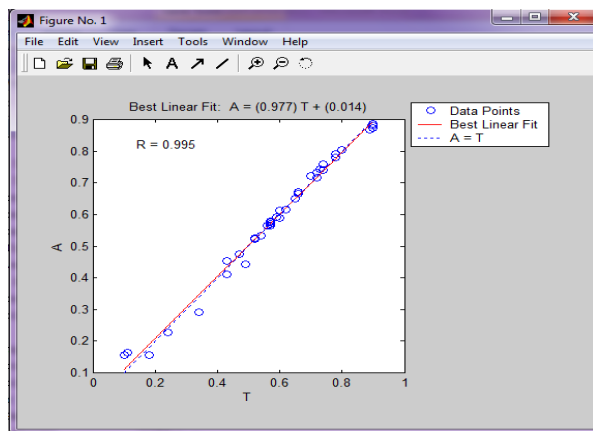
Data Ke	Target	Hasil	Error
1	0.57	0.58	-0.01
2	0.90	0.89	0.01
3	0.49	0.44	0.05
4	0.60	0.61	-0.01
5	0.72	0.72	0.00
6	0.78	0.78	0.00
7	0.57	0.58	-0.01
8	0.90	0.88	0.02
9	0.78	0.79	-0.01
10	0.89	0.87	0.02
20	0.80	0.80	-0.00
30	0.43	0.45	-0.02
40	0.56	0.57	-0.01

6. Evaluasi Output Jaringan (Data Pelatihan dan Target)

Output jaringan data pelatihan dan target dianalisis dengan regresi linear menggunakan perintah sebagai berikut:

```
>>[m1,a1,r1]=postrega(a,T)
```

Hasilnya akan terlihat pada gambar 6 yang menggambarkan hubungan regresi linear terhadap data pelatihan dan target.

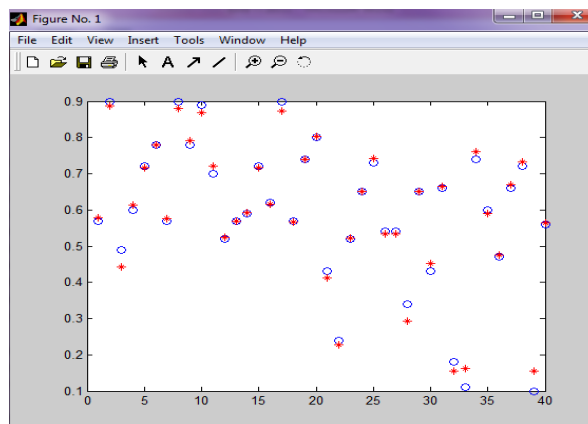


Gambar 6. Perbandingan data pelatihan dengan target

Selanjutnya untuk melihat hasil pengujian dengan data pelatihan dan target, maka gunakan perintah sebagai berikut:

```
>> plot([1:size(R,2)],T,'bo',[1:size(R,2)],a,'r*');
>> title('Hasil Pengujian dengan Data Pelatihan; Target(o), Output(*)');
>> xlabel('Data Ke-');ylabel('Target/Output');
```

Hasil dari perintah di atas terlihat pada gambar 7 yang menggambarkan bahwa data hasil pelatihan hampir sesuai dengan target.



Gambar 7 Hasil pengujian dengan data pelatihan dan target

Pada gambar 7 dapat dilihat data hasil pengujian dengan data pelatihan dan target seperti pada tabel 3.6 di bawah ini dengan menggunakan perintah sebagai berikut:

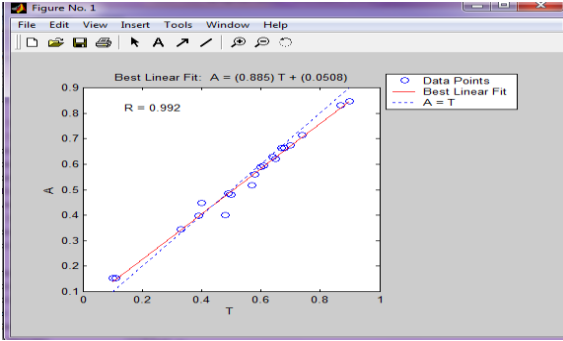
```
>> sprintf('%2d %9.2f %7.2f %5.2f\n',H')
```

Tabel 6. Pengujian data pengujian

Data Ke	Target	Hasil	Error
1	0.48	0.40	0.08
2	0.64	0.63	0.01
3	0.60	0.59	0.01
4	0.90	0.85	0.05
5	0.33	0.34	-0.01
6	0.49	0.49	0.00
7	0.61	0.59	0.02
8	0.40	0.45	-0.05
9	0.58	0.56	0.02
10	0.74	0.72	0.02
11	0.87	0.83	0.04
12	0.10	0.15	-0.05
13	0.50	0.48	0.02
14	0.11	0.15	-0.04
15	0.57	0.52	0.05
16	0.65	0.62	0.03
17	0.70	0.67	0.03
18	0.68	0.66	0.02
19	0.67	0.66	0.01
20	0.39	0.40	-0.01

Untuk melihat hasil pengujian dengan data pengujian target, maka gunakan perintah sebagai berikut sehingga hasilnya akan terlihat pada gambar 8

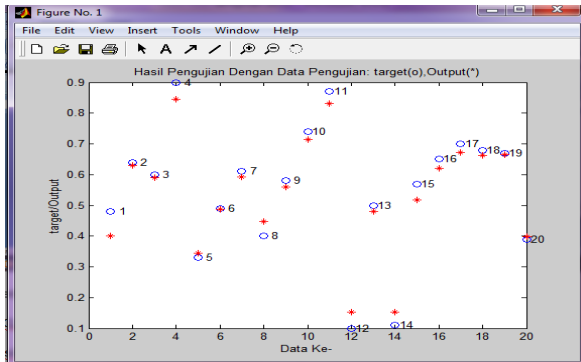
```
>>[m2,b1,r2]=postreg(b,TQ)
```

Gambar 8. Perbandingan data pengujian dengan target

Selanjutnya untuk melihat hasil pengujian dengan data pengujian dan target dapat dilihat pada gambar 9 yang menggambarkan hubungan data pengujian terhadap data pengujian dengan menggunakan perintah:

```
>> k=[1:size(Q,2)];
>> plot(k,TQ,'bo','k,b','r*');
>> title('Hasil Pengujian dengan Data Pengujian: Target(o), Output(*)');
>> xlabel('Data Ke-');ylabel('Target/Output');
>> text(k+0.2*ones(length(k),1),TQ,int2str(k));
```



Gambar 9. Hasil pengujian dengan data pengujian dan target

Setelah melakukan pengolahan data menggunakan pola arsitektur 4-2-1 dengan 40 data pelatihan dan 20 data pengujian menggunakan program Matlab, maka didapatkan hasil perbandingan antara target, hasil dan error yang didapatkan.

3.3 Hasil Pengolahan dengan Matlab

Pengolahan data dengan Matlab menggunakan beberapa pola arsitektur untuk mendapatkan hasil yang lebih baik. Dengan menggunakan beberapa pola arsitektur yang dilakukan pada Matlab sehingga mendapatkan hasil persentase kevalidan seperti terlihat pada table 7 dan 8 di bawah ini:

Tabel 7. Hasil pelatihan pola arsitektur jaringan

No	Data		Pola Arsitektur	% Kevalidan		Keterangan
	Pelatihan	Pengujian		Pelatihan	Pengujian	
1	30	30	4-2-1	95%	23%	
2			4-4-1	93%	7%	
3			4-6-1	90%	5%	
4			4-8-1	87%	47%	
5			4-10-1	83%	25%	
6	40	20	4-2-1	97%	80%	Terbaik
7			4-4-1	90%	80%	
8			4-6-1	85%	45%	
9			4-8-1	87%	55%	
10			4-10-1	88%	55%	

Tabel 8. Hasil dan error data input pelatihan 4-2-1 (4x40)

NO	Data Input Pelatihan (4x40)				Target	JST 4-2-1		Keputusan
	x1	x2	x3	x4		Hasil	Error	
1	0.54	0.58	0.65	0.61	0.57	0.58	-0.01	Valid
2	0.81	0.90	0.80	0.61	0.90	0.89	0.01	Invalid
3	0.37	0.54	0.80	0.39	0.49	0.44	0.05	Valid
4	0.59	0.64	0.75	0.39	0.60	0.61	-0.01	Valid
5	0.81	0.71	0.50	0.75	0.72	0.72	0.00	Valid
6	0.63	0.64	0.80	0.90	0.78	0.78	0.00	Valid
7	0.50	0.64	0.65	0.54	0.57	0.58	-0.01	Valid
8	0.68	0.84	0.85	0.83	0.90	0.88	0.02	Valid
9	0.72	0.80	0.70	0.61	0.78	0.79	-0.01	Valid
10	0.81	0.80	0.75	0.83	0.89	0.87	0.02	Valid
36	0.63	0.51	0.50	0.46	0.47	0.48	-0.01	Valid
37	0.50	0.74	0.65	0.68	0.66	0.67	-0.01	Valid
38	0.90	0.61	0.65	0.61	0.72	0.73	-0.01	Valid
39	0.59	0.10	0.30	0.25	0.10	0.16	-0.06	Valid
40	0.68	0.77	0.50	0.25	0.56	0.57	-0.01	Valid
Jumlah Keputusan JST yang Valid								39
Persentase								97%

Tabel 9. Hasil dan error data input pengujian 4-2-1 (4x20)

NO	Data Input Pengujian (5x20)				Target	JST 5-2-1		Keputusan
	x1	x2	x3	x4		Hasil	Error	
1	0.60	0.60	0.56	0.20	0.48	0.40	0.08	Invalid
2	0.90	0.50	0.67	0.40	0.64	0.63	0.01	Valid
3	0.85	0.43	0.79	0.30	0.60	0.59	0.01	Valid
4	0.80	0.87	0.61	0.80	0.90	0.85	0.05	Valid
5	0.70	0.47	0.27	0.30	0.33	0.34	-0.01	Valid
6	0.80	0.73	0.10	0.40	0.49	0.49	0.00	Valid
7	0.10	0.73	0.84	0.70	0.61	0.59	0.02	Valid
8	0.50	0.37	0.50	0.70	0.40	0.45	-0.05	Invalid

9	0.70	0.67	0.56	0.30	0.58	0.56	0.02	Valid
10	0.70	0.83	0.56	0.50	0.74	0.72	0.02	Valid
11	0.60	0.83	0.73	0.90	0.87	0.83	0.04	Valid
12	0.55	0.10	0.44	0.20	0.10	0.15	-0.05	Invalid
13	0.60	0.47	0.67	0.40	0.50	0.48	0.02	Valid
14	0.45	0.30	0.27	0.20	0.11	0.15	-0.04	Valid
15	0.80	0.67	0.56	0.10	0.57	0.52	0.05	Valid
16	0.60	0.67	0.79	0.30	0.65	0.62	0.03	Valid
17	0.50	0.90	0.50	0.60	0.70	0.67	0.03	Valid
18	0.80	0.63	0.56	0.60	0.68	0.66	0.02	Invalid
19	0.55	0.40	0.90	0.90	0.67	0.66	0.01	Valid
20	0.60	0.50	0.39	0.40	0.39	0.40	-0.01	Valid
Jumlah Keputusan JST yang Valid								16
Persentase								80 %

Pada tabel 8 dapat dilihat data Keputusan Jaringan Syaraf Tiruan (KJST) yang bernilai valid ada 39 data dan bernilai tidak valid ada 1 data, maka dari data pada tabel 8 didapatkan persentase kevalidan data pelatihan adalah 97%.

Pada tabel 9 juga dapat dilihat data Keputusan Jaringan Syaraf Tiruan (KJST) yang bernilai valid ada 16 data dan bernilai tidak valid ada 4 data, maka dari data pada tabel 9 didapatkan persentase kevalidan data pengujian adalah 80%.

3.4. Perbandingan Hasil Analisa Manual dengan Matlab

Tabel 3.10 Hasil transformasi data hasil dan error

Manual		Matlab		Manual		Matlab		Nilai
Hasil	Error	Hasil	Error	Hasil	Error	Hasil	Error	
0.58	-0.01	0.58	-0.01	0	0	1	0	Sama
0.85	0.05	0.89	0.01	0	0	1	0	Sama
0.49	0.48	0.44	0.05	1	0	0	0	Sama
0.62	-0.02	0.61	-0.01	0	0	1	0	Sama
0.74	-0.02	0.72	0.00	0	0	1	0	Sama
0.80	-0.02	0.78	0.00	0	0	1	0	Sama
0.58	-0.01	0.58	-0.01	0	0	1	0	Sama
0.88	0.02	0.88	0.02	0	0	1	0	Sama
0.78	0.00	0.79	-0.01	0	0	1	0	Sama
0.88	0.01	0.87	0.02	0	0	1	0	Sama
-0.03	0.73	0.72	-0.02	1	0	1	0	Sama
0.52	0.00	0.53	-0.01	0	0	1	0	Sama
0.57	0.00	0.57	0.00	0	0	1	0	Sama
0.58	0.01	0.59	0.00	0	0	1	0	Sama
0.74	-0.02	0.72	0.00	0	0	1	0	Sama
0.61	0.01	0.62	0.00	0	0	1	0	Sama
0.88	0.02	0.87	0.03	0	0	1	0	Sama
0.57	0.00	0.57	0.00	0	0	1	0	Sama
0.75	-0.01	0.74	0.00	0	0	1	0	Sama
0.81	-0.01	0.80	-0.00	0	0	1	0	Sama
0.40	0.03	0.41	0.02	1	0	0	0	Sama
0.23	0.01	0.23	0.01	1	0	0	0	Sama
0.54	-0.02	0.52	-0.00	0	0	1	0	Sama

Tabel 3.11 Hasil transformasi data hasil dan error

Manual		Matlab		Manual		Matlab		Nilai
Hasil	Error	Hasil	Error	Hasil	Error	Hasil	Error	
0.52	0.02	0.53	0.01	0	0	1	0	Sama
0.52	0.02	0.53	0.01	0	0	1	0	Sama
0.34	-0.00	0.29	0.05	1	0	0	0	Sama
0.65	-0.00	0.65	-0.00	0	0	1	0	Sama
0.43	0.00	0.45	-0.02	1	0	0	0	Sama
0.66	-0.00	0.67	-0.01	0	0	1	0	Sama
0.17	0.01	0.15	0.03	1	0	0	0	Sama
0.15	0.04	0.16	-0.05	1	0	0	0	Sama
0.73	0.01	0.76	-0.02	0	0	1	0	Sama
0.57	0.03	0.59	0.01	0	0	1	0	Sama
0.44	0.03	0.48	-0.01	1	0	0	0	Sama
0.69	0.03	0.67	-0.01	0	0	1	0	Sama
0.75	0.03	0.73	-0.01	0	0	1	0	Sama
0.13	0.03	0.16	-0.06	1	0	0	0	Sama
0.53	0.03	0.57	-0.01	0	0	1	0	Sama

Dari sebanyak 40 data pelatihan yang dilakukan semuanya sama dengan hasil output. Dari hasil data tersebut maka dapat disimpulkan bahwa tingkat kepercayaan dari perbandingan data yang diolah dengan menggunakan Matlab dibandingkan dengan cara manual adalah 100% sama.

4. Simpulan

Setelah melakukan pelatihan dan pengujian serta implementasi dengan menggunakan software Matlab 6.1 terhadap data nilai siswa SMKN 3 Pariaman, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Arsitektur jaringan yang paling tepat digunakan untuk prediksi prestasi siswa adalah dengan jaringan syaraf tiruan menggunakan algoritma *backpropagation* dengan pola arsitektur 4-2-1 dengan membagi data menjadi dua bagian yaitu 40 data pelatihan dengan persentase kebenaran data 97 % dan 20 data pengujian dengan persentase 80 %.
2. Algoritma *backpropagation* merupakan salah satu algoritma yang mudah diterapkan untuk menyelesaikan beberapa permasalahan terutama yang berkaitan dengan masalah prediksi.
3. Dengan menggunakan jaringan saraf tiruan dengan algoritma *backpropagation* proses prediksi lebih cepat, akurat, meminimalisir kesalahan dan bisa menggunakan teknologi komputer. Serta mudah dalam pengembangannya.
4. Semakin kecil tingkat ketelitian error yang digunakan maka akan semakin kecil penyimpangan hasil jaringan saraf tiruan dengan target yang diinginkan.

Referensi

- Hermawan, Arief. 2006. Jaringan Saraf Tiruan Teori dan Aplikasi. Andi Offset. Yogyakarta.
- Puspitanigrum, Diyah. 2006. Pengantar Jaringan Saraf Tiruan. Andi Offset. Yogyakarta.
- Aribowo, W. (2010). Stabilisator sistem tenaga berbasis jaringan syaraf tiruan berulang untuk sistem mesin tunggal. *Telkomnika*, Vol. 8, No, 65–72.
- Effendy, N., & Faisal, A. (2008). Prediksi Penyakit Jantung Koroner (Pjk) Berdasarkan Faktor Risiko Menggunakan Jaringan Syaraf Tiruan Backpropagation. *SNATI*, 2008(Snati).
- Mastur, I., & Hadi, L. (2005). Implementasi Jaringan Syaraf Tiruan Untuk. *ISSN 0853-8697*, 10(3), 197–208.