

## PENERAPAN METODE NEURAL NETWORK DENGAN STRUKTUR BACKPROPAGATION UNTUK PREDIKSI STOK OBAT DI APOTEK (STUDI KASUS : APOTEK ABC)

Novi Yanti

Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau  
Jl. HR Soebrantas No. 155 Km. 15 Simpang Baru Panam - Kec. Tampan Pekanbaru Riau 28293  
PO Box. 1004 HP. 0812 688 9515  
E-mail: novikaliiye@yahoo.com

### ABSTRAK

Apotek ABC adalah apotek yang menjual obat-obatan. Apotek ini menggunakan cara manual dalam mendata stok obat dan data tersebut kemudian disimpan dalam Microsoft Excel. Untuk mengetahui jumlah stok obat membutuhkan waktu yang lama karena tidak ada gambaran berapa jumlah stok obat yang tersisa. Tujuannya adalah untuk menghindari terjadinya kekosongan stok. Selama ini, untuk mengetahui jumlah stok hanya dengan menghitung sisa stok yang ada, kemudian membandingkan jumlah antara obat yang terjual dengan faktur pembelian. Permasalahan yang terjadi yaitu tidak validnya data stok obat yang ada dan sulitnya melakukan prediksi untuk pemesanan ditahun berikutnya. Pada paper ini mencoba menerapkan metode Neural Network dengan struktur backpropagation untuk melakukan prediksi pendataan stok obat di apotek dengan evaluation pattern menggunakan aplikasi Matlab 6.1. Penerapan metode ini bertujuan untuk mendapatkan hasil prediksi stok obat. Hasil diberikan dalam bentuk angka dan grafik untuk prediksi tahun berikutnya dengan jumlah node input 10, node hidden 3 dan node output 1. Nilai learning rate yang digunakan adalah 0,1, 0,5 dan 0,8. Hasil pengujian yang dilakukan menghasilkan RMSE pada learning rate 0.5. Diharapkan nantinya dengan metode ini dapat memberikan solusi kepada pihak apotek untuk melakukan prediksi stok obat satu tahun kedepan sehingga mempermudah pekerjaan pada bagian inventori dan penjualan di apotek ABC.

Kata Kunci: apotek, backpropagation, neural network, prediksi.

### 1. PENDAHULUAN

Neural Network (NN) merupakan sebuah teknologi komputasi, tidak memberikan suatu keajaiban tetapi jika digunakan secara tepat akan menghasilkan suatu hasil yang luar biasa. Kemampuan NN dalam menyelesaikan masalah yang rumit telah dibuktikan dalam berbagai macam penelitian, seperti analisa data, meteorologi, pengenalan pola, sistem kontrol, deteksi penomena kedokteran, prediksi pasar saham, dan sebagainya (Yani, 2005).

Pendataan stok obat merupakan permasalahan yang sering dihadapi oleh pihak apotek. Masalah ini timbul karena sulitnya menghitung data yang besar dan kurangnya pengecekan terhadap data yang ada. Hal ini mengakibatkan sering terjadi kesulitan dan kesalahan dalam menghitung jumlah stok. Sehingga menimbulkan kekacauan pembukuan dan terjadinya kerugian.

Apotek ABC masih menggunakan sistem manual dalam menghitung stok obat. Pendataan stok dilakukan dengan menghitung sisa stok yang ada, kemudian membandingkan jumlah antara obat yang terjual dengan faktur pembelian. Hal ini selalu memberikan data obat yang tidak pasti. Kemudian hasil perhitungan tersebut disimpan dalam MS Excel.

Karena pendataan stok dilakukan secara manual, sering terjadi kekeliruan dan membutuhkan waktu yang lama. Hal ini disebabkan karena tidak ada gambaran berapa stok obat yang ada (tersisa)

untuk menghindari terjadinya kekosongan stok. Pada saat sekarang solusi permasalahan oleh pihak apotek adalah dengan melakukan pengawasan dengan cara pemantauan sisa obat yang ada. Tujuan utama dari pengawasan ini adalah untuk menjaga agar tidak sampai terjadi kesalahan terhadap penghitungan data dan menghindari terjadinya kekosongan stok obat.

Berdasarkan analisa permasalahan diatas, maka dilakukan penerapan metode Neural Network dengan struktur backpropagation untuk prediksi stok obat di apotek satu tahun kedepan. Hasil dari penerapan NN akan di implementasikan dengan menggunakan aplikasi Matlab 6.1, dengan batasan masalah adalah sebagai berikut (Yanti, 2011):

1. Pendataan dilakukan dengan menggunakan data stok obat per 31 Desember 2010.
2. Data inputan terdiri dari: nama obat ( $X_1$ ), jenis obat ( $X_2$ ), dosis obat ( $X_3$ ), satuan obat ( $X_4$ ), kemasan obat ( $X_5$ ), kadaluarsa ( $X_6$ ), stok obat ( $X_7$ ), obat terjual ( $X_8$ ), sisa obat ( $X_9$ ) dan order obat ( $X_{10}$ ).
3. Tidak memperhitungkan batasan tanggal kadaluarsa dan masalah tansaksi penjualan.
4. Tidak membahas faktor eksternal seperti: cuaca, suhu dan wabah penyakit yang sedang berjangkit.

Metode penelitian yang digunakan adalah:

1. Penelitian Pendahuluan dan Studi Pustaka  
Penelitian pendahuluan; melakukan observasi ke apotek untuk melihat dan mengetahui secara

langsung kondisi dan permasalahan yang terjadi. Kemudian studi pustaka untuk mengetahui informasi secara teoritis mengenai pokok permasalahan dan teori pendukung yang digunakan sebagai dasar pemikiran untuk membahas permasalahan yang ada. Membaca buku-buku yang berhubungan dengan metode *NN* dan struktur *BP*.

## 2. Identifikasi Masalah

Masalah yang diidentifikasi adalah bagaimana memprediksi stok obat di apotek untuk satu tahun kedepan.

## 3. Pemilihan Metode

Analisa menerapkan metode *NN* dengan struktur *BP* untuk memprediksi stok obat di apotek sehingga memberikan hasil yang lebih akurat.

## 4. Pengumpulan Data, dengan cara:

a. Wawancara, pada bagian inventori untuk mendapatkan informasi pengadaan dan spesifikasi obat. Kemudian bagian penjualan untuk mendapatkan data sisa stok dan data yang diperoleh dalam bentuk *hard copy*.

b. Studi Literatur, dengan cara mencari dan mengumpulkan referensi tentang metode *NN* dan *BP* serta data-data apotek untuk dijadikan contoh dalam menganalisa prediksi stok obat, baik berupa jurnal, artikel, buku referensi, internet dan sumber lain yang berhubungan.

## 5. Analisa

Mengetahui alur proses penelitian ini, agar hasil sesuai dengan tujuan yang diinginkan.

## 6. Evaluation Pattern

Pengujian dilakukan dengan menggunakan aplikasi *Matlab 6.1*. Tujuannya adalah untuk mendapatkan hasil prediksi dalam bentuk angka dan grafik yang dapat dimengerti oleh pihak apotek.

## 7. Kesimpulan dan Saran

Diharapkan hasil penelitian ini sesuai dengan rumusan masalah dan tujuan yang akan dicapai, serta saran-saran yang diperlukan untuk pengembangan penelitian selanjutnya.

## 2. NEURAL NETWORK DAN ALGORITMA BACKPROPAGATION

### 2.1 Neural Network (NN)

*Neural Network (NN)* adalah suatu metode pembelajaran yang diinspirasi dari jaringan sistem pembelajaran biologis yang terjadi dari jaringan sel syaraf (*neuron*) yang terhubung satu dengan yang lainnya. Struktur *NN* yang digunakan adalah *Backpropagation (BP)* yang merupakan sebuah metode sistematis untuk pelatihan *multiplayer*. Metode ini memiliki dasar matematis yang kuat, objektif dan algoritma ini mendapatkan bentuk persamaan dan nilai koefisien dalam formula dengan meminimalkan jumlah kuadrat galat *error* melalui model yang dikembangkan (*training set*). Menurut Daghli (1994), jenis *backpropagation* lebih fleksibel dan secara umum relatif lebih baik karena paling

banyak penerapannya digunakan khususnya untuk aplikasi dalam dunia industri.

Riggs (1987) menyatakan bahwa salah satu cara peramalan adalah dengan metode urutan waktu (*time series*) yang menggunakan data histori (data waktu lampau), misalnya data permintaan, untuk membuat ramalan permintaan diwaktu mendatang. Tujuan dari metode ini adalah untuk mengidentifikasi pola data histori dan kemudian mengekstrapolasikan pola ini ke masa datang. Metode *NN* ini dilatih dengan seperangkat data untuk bisa mengenal dan mengidentifikasi pola data atau kurva. Proses pelatihan disebut tahap belajar (*learning process*), yang merupakan bagian penting dalam metode ini. Pemilihan algoritma dan parameter yang bersesuaian dan penentuan berapa banyak perangkat data yang dibutuhkan sangat penting untuk menentukan akurasi dari peramalan yang dihasilkan.

### 2.2 Algoritma Backpropagation (BP)

Algoritma *BP* 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. *Output* lapis terakhir dari *hidden layer* langsung dipakai sebagai *output* dari *neural network* (Hagan, 1996).

Proses pelatihan *BP* memerlukan tiga tahapan, yaitu *feedforward* data *input* untuk pelatihan, *backpropagation* untuk nilai *error* serta penyesuaian nilai bobot tiap *node* masing-masing *layer* pada *NN*. Diawali dengan *feedforward* nilai *input*, tiap *input* unit ke-*i* ( $x_i$ ) menerima sinyal *input* yang selanjutnya akan dipancarkan ke *hidden layer*  $z_1, \dots, z_p$ . Selanjutnya *hidden* unit ke-*j* akan menghitung nilai sinyal ( $z_j$ ), yang akan dipancarkan ke *output layer*, menggunakan fungsi aktivasi *f*.

$$z_{-in_j} = \theta_{1j} + \sum_{i=1}^n x_i v_j \quad (1)$$

dan

$$z_j = f(z_j) \quad (2)$$

dimana  $\theta_{1j}$  = bias *hidden* unit ke-*j*. Nilai bias dan bobot awal dapat diambil secara acak. Tiap unit *output* ke-*k* ( $Y_k$ ).

$$Y_k = \theta_{2k} + \sum_j z_k w_{jk} \quad (3)$$

dan

$$Y_k = f(Y_k) \quad (4)$$

dimana  $\theta_{2k}$  = bias *hidden* unit ke-*k*. Selama proses *training* berlangsung, tiap unit *output* membandingkan nilai target ( $T_m$ ) untuk suatu *input pattern* guna menghitung nilai parameter yang akan

memperbaiki (*update*) bobot nilai tiap unit dalam masing-masing *layer* (Hertz, et all, 1991).

### 2.3 Fungsi Aktivasi

Karakteristik yang harus dimiliki oleh suatu fungsi aktivasi yaitu kontinu, differensiabel dan monoton menurun. Salah satu fungsi yang sering digunakan adalah fungsi sigmoid yang memiliki range (0,1).

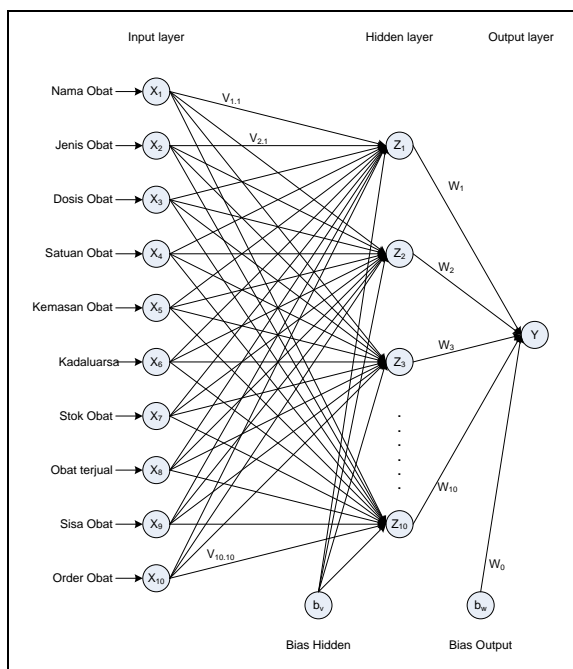
$$f(x) = \frac{1}{1 + \exp(-x)} \quad (5)$$

Node pada *output layer* memiliki nilai antara 0-1.

## 3. ANALISA

### 3.1 Arsitektur Jaringan Yang Akan Dilatih

Pada analisa prediksi stok obat di apotek, arsitektur jaringan yang digunakan adalah arsitektur jaringan lapis banyak (*multilayer*) seperti pada Gambar 1.



Gambar 1. Arsitektur Jaringan Dengan Backpropagation

#### Keterangan:

- $X$  = node input pada lapisan input
- $Z$  = node hidden (lapisan tersembunyi)
- $Y$  = node output pada lapisan output
- $V_{1,1}, \dots, V_{n,n}$  = bobot dari lapisan input ke hidden
- $W_{1,1}, \dots, W_{n,n}$  = bobot dari lapisan hidden ke output
- $b_v$  = bias dari lapisan input ke lapisan hidden
- $b_w$  = bias dari lapisan hidden ke lapisan output

### 3.2 Proses Training

Proses *training* yang dilakukan meliputi data sebagai berikut:

1. *Input-an* parameter jaringan *node*:
  - a. Jumlah *node input layer* = 10 *node*
  - b. Jumlah *node hidden layer* = 3 *node*

- c. Toleransi *error* = 0,01
- d. *Learning Rate* ( $\alpha$ ) = 0,5
- e. Jumlah iterasi = 10000

2. Data yang digunakan adalah data pada Tabel 1.

Tabel 1. Data *Input* Awal Yang Digunakan

$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$	$X_9$	$X_{10}$
1	4	3	5	3	8	308	278	11	297
2	1	2	6	3	6	182	143	17	165
3	5	6	3	1	3	403	326	48	355
4	4	5	2	1	5	199	176	19	180
5	3	3	2	1	2	327	276	42	285
6	3	3	2	1	4	240	221	19	221
7	5	7	4	4	0	443	432	11	432
8	1	3	6	3	6	159	153	6	135
9	6	4	1	2	7	430	345	19	411
10	2	1	6	3	5	135	77	19	116

3. Normalisasi

Data *input* awal pada Tabel 1 dinormalisasikan dengan hasil dapat dilihat pada Gambar 2.

	$f_{10}$	$f_9$	$f_8$	$f_7$	$f_6$	$f_5$	$f_4$	$f_3$	$f_2$	$f_1$
$X_1$	1	0,88888	0,77777	0,66666	0,55555	0,44444	0,33333	0,22222	0,11111	0
$X_2$	0,20000	1	0	0,80000	0,40000	0,40000	0,60000	0,80000	0	0,60000
$X_3$	0	0,50000	0,33333	1	0,33333	0,33333	0,66667	0,83333	0,16667	0,33333
$X_4$	1	0	1	0,60000	0,20000	0,20000	0,20000	0,40000	1	0,80000
$X_5$	0,66667	0,33333	0,66667	1	0	0	0	0	0,66667	0,66667
$X_6$	0,57143	0,85714	0,71429	0	0,42857	0,14286	0,57143	0,28571	0	1
$X_7$	0	0,95779	0,07792	1	0,34091	0,62338	0,20779	0,87013	0,15260	0,56169
$X_8$	0	0,75483	0,21408	1	0,40563	0,56056	0,27881	0,70141	0,18592	0,56620
$X_9$	0,30852	0,30852	0	0,11905	0,30852	0,85174	0,30852	1	0,26190	0,11905
$X_{10}$	0	0,93354	0,06013	1	0,33228	0,53481	0,20253	0,75633	0,15506	0,57278

Gambar 2. *Input* Data Yang Sudah Ter-Normalisasi

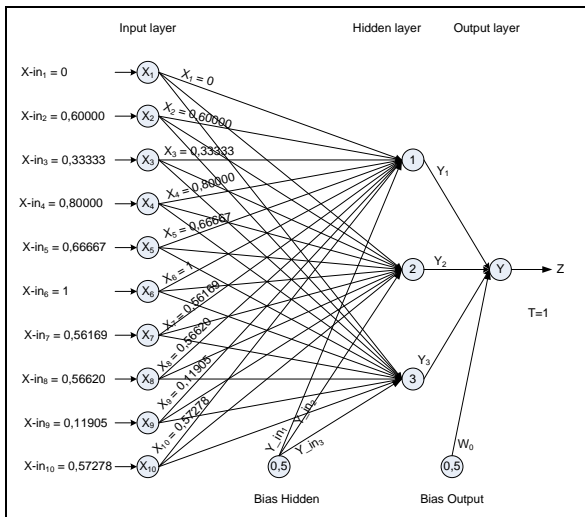
4. Cek Kondisi berhenti.

Selama kondisi berhenti masih tidak terpenuhi, laksanakan langkah 4 sampai 13. Adapun syarat kondisi berhenti adalah nilai *error* ( $Sse$ )  $\leq 0,01$  atau *epoch*/iterasi  $\geq 1000$ .

5. Inisialisasi nilai bobot.

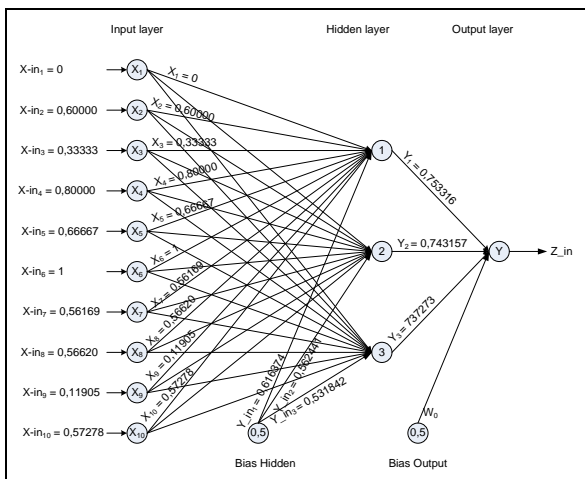
Pada sistem ini, nilai bobot jaringan termasuk nilai bobot untuk bias ditentukan secara acak antara 0-1, dengan nilai bias yang digunakan 0,5.

6. Menentukan *variabel input* dan *output* pada *node input*. Nilai *input* diambil berdasarkan nilai bobot yang sudah ternormalisasi. Dapat dilihat pada Gambar 3.



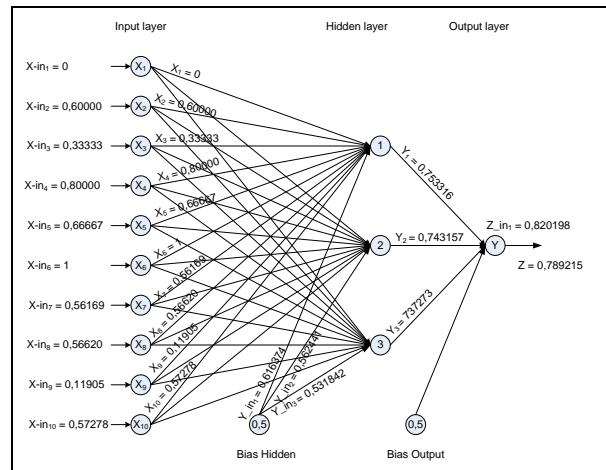
Gambar 3. Nilai Input, Output dan Target

7. Menentukan *input* pada *node hidden*.  
Penjumlahan *signal input* berbobot untuk masing-masing *node* pada *node hidden*.
8. Fungsi aktivasi.  
Setelah memasukkan variabel *input* pada *node hidden*, maka langkah selanjutnya menentukan *variable output* pada *node hidden* dengan melakukan fungsi aktivasi. Sehingga gambar yang terbentuk dapat dilihat pada Gambar 4.



Gambar 4. Nilai Input, Output Pada Hidden Layer

9. Menentukan variabel *input* pada *node output*.  
Setelah melakukan langkah no.7, maka selanjutnya menentukan variabel *input* pada *node output*.
10. Fungsi aktivasi  
Setelah memasukkan *varabel input* pada *node output*, selanjutnya menentukannya *variable output* pada *node output* dengan melakukan fungsi aktivasi, dengan menggunakan fungsi aktivasi *Sigmoid Biner* yang dapat dilihat pada Gambar 5.



Gambar 5. Nilai Input, Output Pada Node Output

11. Hitung informasi *error* pada *node output Z*.  
Nilai *output* dari *node Z* dibandingkan dengan nilai *output target* T. T merupakan nilai normalisasi dari Order Obat (X<sub>10</sub>).
12. Menghitung koreksi pada *node hidden*.
  - a. Untuk tiap *node hidden*, dihitung delta *input* yang berasal dari *node* pada *layer*.
  - b. Hitung informasi kesalahan pada *node hidden*
  - c. Hitung koreksi bobot pada *node hidden* yang kemudian digunakan memperbaharui nilai V.
  - d. Hitung koreksi bias pada *node hidden* yang kemudian digunakan memperbaharui nilai V.
13. Memperbaharui nilai bobot dan nilai bias.
  - a. Menentukan bias dan bobot pada *node output*
  - b. Menentukan bias dan bobot pada *node hidden* Sehingga bobot (V) baru yang diperoleh dari *epoch/iterasi* pertama pada *node hidden* dapat dilihat pada Gambar 6 dibawah ini.

		j		
		1	2	3
i	0	0,501629	0,500335	0,501358
	1	0,5	0,5	0,5
	2	0,200977	0,070040	0,080652
	3	0,400543	0,080022	0,090362
	4	0,061303	0,400054	0,200869
	5	0,301086	0,080045	0,300725
	6	0,011629	0,020067	0,021087
	7	0,030915	0,030038	0,070610
	8	0,050922	0,060038	0,010615
	9	0,070194	0,080008	0,050129
	10	0,090933	0,070038	0,040622

Gambar 6. Matrik Bobot Node Input ke Hidden

Nilai bobot (W) baru yang diperoleh dari *epoch/iterasi* pertama pada *node output*, dapat dilihat pada Gambar 7 berikut ini.

		k
j	0	0,517533
	1	0,513208
	2	0,113029
	3	0,412926

Gambar 7. Matrik Bobot Node Hidden ke Output

14. Menguji apakah kondisi berhenti sudah terpenuhi. Kondisi berhenti ini terpenuhi jika nilai kesalahan yang dihasilkan lebih kecil dari nilai kesalahan referensi atau  $epoch/iterasi \geq 1000$ .

#### 4. EVALUATION PATTERN

Pada tahapan *evaluation pattern* menggunakan *Matlab 6.1*.

##### 4.1 Uji Coba

##### 4.1.1 Uji Coba Training dan Testing

Untuk memperoleh *output* pada tahap *training* dan *testing* digunakan data (matriks) 10 record x 10 field = 100 data. Data *input* yang dipakai dalam proses *training* adalah data  $X_1$  sampai dengan  $X_9$  dan data  $X_{10}$  merupakan data nilai Target (*output*) dengan nilai seperti yang terdapat pada Tabel 1. Untuk *testing* dilakukan dengan memasukkan nilai  $\alpha$  dan jumlah *node hidden* 3. *Testing* dilakukan dengan menggunakan beberapa parameter yang berbeda dan mengambil *file* bobot dari proses *training* yang telah dilakukan sebelumnya, untuk mendapatkan arsitektur *NN* yang sesuai untuk memperoleh hasil prediksi optimal.

##### 4.1.2 Skenario Uji Coba

Pada uji coba menggunakan nilai *learning rate* dan *node hidden* 3, 5 dan 8. Tujuan dari uji coba ini adalah untuk menguji nilai rata-rata *error* dan rata-rata persen *error* yang dihasilkan dari tiap-tiap proses yang dilakukan. Dari proses tersebut akan dipilih nilai rata-rata *error* dan rata-rata persen *error* yang minimum sebagai hasil yang terbaik. Berikut skenario uji coba yang akan dilakukan dengan nilai *learning rate* dan *node hidden* seperti pada Tabel 2.

Tabel 2. Data *Learning Rate* dan *Node Hidden*

No	Learning Rate	Node Hidden
1	0,1	3
2	0,5	3
3	0,8	3

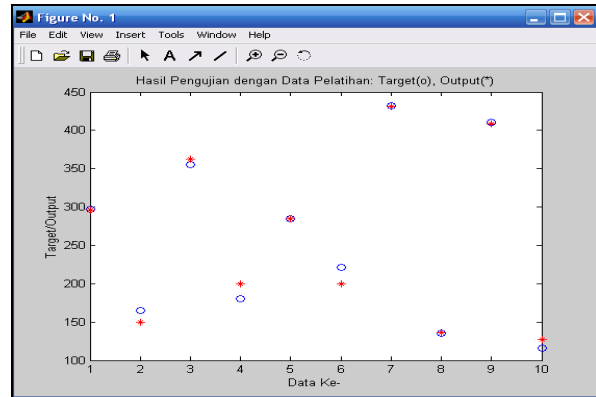
Uji coba untuk *error testing* dengan *learning rate* = 0,1 dan *node hidden* = 3 akan membentuk arsitektur 10-3-1. Hasil *testing* seperti pada Tabel 3.

Tabel 3. *Learning Rate*=0,1 dan *Node Hidden* = 3

No	Nama Obat	Nilai Target	Hasil Prediksi	Error	RMSE
1	Amobiotic Drops	297	296,44	0,56	0,05600
2	Angioten	165	149,75	15,25	1,52500
3	Betadin	355	362,11	-7,11	0,71100
4	Cefixim Generik Syrup	180	200,15	-20,15	2,01500
5	Infusan Ring-As	285	285,11	-0,11	0,01100
6	Kaen 4 A	221	199,88	21,12	2,11200
7	Kassa Gulong Panjang	432	430,70	1,30	0,13000
8	Lesifit	135	136,46	-1,46	0,14600
9	Sapol Injeksi	411	408,37	2,63	0,26300

10	Vomceran	116	127,50	-11,50	1,15000
----	----------	-----	--------	--------	---------

Dari Tabel 3, nilai *error* minimum adalah pada data ke-5 dengan nilai *RMSE* 0,01100. Grafik yang terbentuk seperti yang terlihat pada Gambar 8.



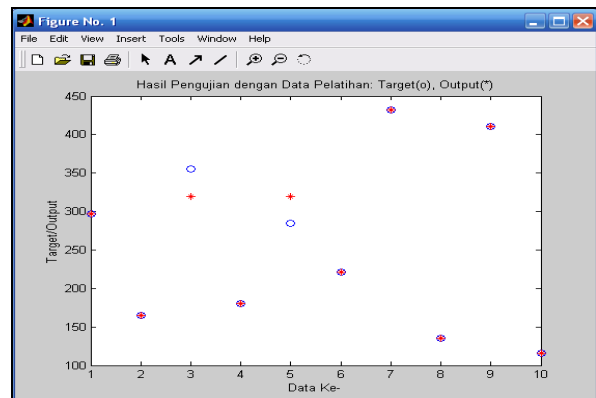
Gambar 8. Grafik Hasil Pengujian  $LR=0,1$

Uji coba untuk *error testing* dengan *learning rate* = 0,5 dan *node hidden* = 3 akan membentuk arsitektur 10-3-1. Hasil *testing* seperti pada Tabel 4.

Tabel 4. *Learning Rate*=0,5 dan *Node Hidden* = 3

No	Nama Obat	Nilai Target	Hasil Prediksi	Error	RMSE
1	Amobiotic Drops	297	297,00	-0,00	0,00000
2	Angioten	165	165,00	-0,00	0,00000
3	Betadin	355	320,00	35,00	3,50000
4	Cefixim Generik Syrup	180	180,00	-0,00	0,00000
5	Infusan Ring-As	285	320,00	-35,00	3,50000
6	Kaen 4 A	221	221,00	0,00	0,00000
7	Kassa Gulong Panjang	432	432,00	-0,00	0,00000
8	Lesifit	135	135,00	0,00	0,00000
9	Sapol Injeksi	411	411,00	-0,00	0,00000
10	Vomceran	116	116,00	0,00	0,00000

Dari Tabel 4, nilai *error* minimum adalah pada data ke-1,2,4,6,7,8,9 dan 10 dengan nilai *RMSE* 0,00000. Grafik yang terbentuk seperti yang terlihat pada Gambar 9.



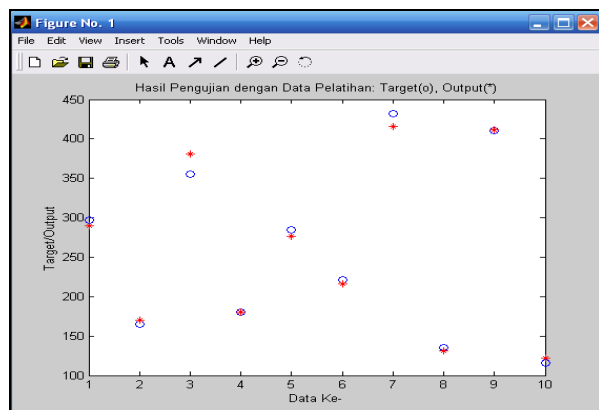
Gambar 9. Grafik Hasil Pengujian  $LR=0,5$

Uji coba untuk *error testing* dengan *learning rate* = 0,8 dan *node hidden* = 3 akan membentuk arsitektur 10-3-1. Hasil *testing* seperti pada Tabel 5.

Tabel 5. *Learning Rate*=0,8 dan *Node Hidden* = 3

No	Nama Obat	Nilai Target	Hasil Prediksi	Error	RMSE
1	Amobiotic Drops	297	290,16	6,84	0,68400
2	Angioten	165	170,27	-5,27	0,52700
3	Betadin	355	380,48	-25,48	2,54800
4	Cefixim Generik Syrup	180	179,87	0,13	<b>0,01300</b>
5	Infusan Ring-As	285	276,14	8,86	0,88600
6	Kaen 4 A	221	215,76	5,24	0,52400
7	Kassa Gulung Panjang	432	415,96	16,04	1,60400
8	Lesifit	135	131,61	3,39	0,33900
9	Sapol Injeksi	411	411,74	-0,74	0,07400
10	Vomceran	116	121,55	-5,55	0,55500

Dari Tabel 5, nilai *error* minimum adalah pada data ke-4 dengan nilai *RMSE* 0,01300. Grafik yang terbentuk seperti yang terlihat pada Gambar 10.



Gambar 10. Grafik Hasil Pengujian *LR*=0,8

#### 4.1.3 Kesimpulan Pengujian

Berdasarkan pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa hasil prediksi yang dilakukan dengan nilai *learning rate* 0,5 dan nilai *hidden* 3 memberikan nilai *Error* 0,000000 dengan nilai hasil pengujian dapat dilihat pada Tabel 4.

## 5. PENUTUP

### 5.1 Kesimpulan

Beberapa kesimpulan yang didapat antara lain:

- Hasil prediksi dengan *NN* memberikan informasi dalam bentuk angka dan grafik serta memberikan nilai persentase *error* dan nilai *error* rata-rata.
- Pemilihan parameter dan nilai bobot untuk prediksi berdasarkan nilai *error testing* minimum
- Keberhasilan dalam memprediksi dipengaruhi oleh jumlah *node hidden* dan nilai *learning rate* yang digunakan.

- Tingkat toleransi kesalahan dipengaruhi oleh proses *training*, jika nilai toleransi *error* semakin kecil maka proses *training* akan semakin akurat.
- Nilai akurasi *NN* lebih tinggi sehingga dapat mengurangi terjadinya *redundancy*.

### 5.2 Saran

Saran untuk pengembangan berikutnya adalah:

- Untuk menghasilkan prediksi/permalan yang lebih baik diperlukan data *input*-an dengan jumlah yang lebih besar sehingga analisa menjadi semakin optimal dan akurat.
- Pada proses *training* dapat menggunakan parameter yang lebih bervariasi, sehingga akan menghasilkan hasil prediksi yang lebih baik.
- Diharapkan pengembangan hasil analisis yang dilakukan ini dapat diimplementasikan dengan membangun sebuah sistem/aplikasi dengan menerapkan metode *NN* untuk melakukan prediksi/permalan stok obat di apotek.

### PUSTAKA

Dagli, Cihan H., 1994, *Artificial Neural Networks for Intelligent Manufacturing*, Chapman & Hall, United Kingdom.

Hagan, Martin T., 1996, *Neural Network Design*, PWS Publishing Co., USA

J. Hertz, A. Krogh and R. G. Palmer, 1991, *Intriduction to The Theory of Neural Computation*, addison Wesley.

Riggs, James L., 1987, *Production Systems-Planning, Analysis and Control*, edisi ke-4, John Wiley & Sons, Singapore.

Yani, Eli, 2005, *Pengantar Jaringan Syaraf Tiruan*, [http://trirezqariantoro.files.wordpress.com/2007/05/jaringan\\_syaraf\\_tiruan.pdf](http://trirezqariantoro.files.wordpress.com/2007/05/jaringan_syaraf_tiruan.pdf). Akses 05 Maret 2011.

Yanti, Novi, 2011, *Analisa Perbandingan Metode Rough Set Dan Neural Network Untuk Prediksi Pendataan Obat Di Apotek*, Tesis Magister Ilmu Komputer Universitas Putra Indonesia "YPTK", Padang.