

Jurnal Matematika

Universitas Andalas

Volume VII No.2 ISSN : 2303-2910

Mei 2018



2018

Diterbitkan Oleh :
Jurusan Matematika
FMIPA Universitas Andalas

MODEL NON-LINEAR PADA JARINGAN SARAF TIRUAN

MUTIA YOLLANDA, DODI DEVIANTO, HAZMIRA YOZZA

*Jurusan Matematika,
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Andalas,
Kampus UNAND Limau Manis Padang, Indonesia.
email : mutiayollanda@gmail.com*

Abstrak. Jaringan Saraf Tiruan merupakan model yang meniru cara kerja jaringan saraf secara biologi. Algoritma pembelajaran Jaringan Saraf Tiruan digunakan untuk melatih jaringan secara iterasi sehingga bobot antar unit dapat disesuaikan dengan galat yang ditentukan. Metode *Backpropagation* didesain untuk operasi pada jaringan *feedforward* dengan banyak lapisan sehingga memperoleh bobot jaringan dengan galat terkecil. Bobot tersebut digunakan untuk memodelkan data. Fungsi sigmoid digunakan pada jaringan *feedforward* sehingga menghasilkan bobot yang berbentuk tidak linear. Bobot yang berbentuk tidak linear membentuk model non-linear pada Jaringan Saraf Tiruan.

Kata Kunci: Jaringan Saraf Tiruan, Metode *Backpropagation*, Sigmoid, *Feedforward*

1. Pendahuluan

Sebagian besar penelitian ditujukan untuk mengukur hubungan antara variabel yang akan mempengaruhi hasil akhir sehingga terbentuk suatu model. Dalam mengontrol perilaku model, dibutuhkan konstanta yang diperkirakan dalam model yaitu parameter. Pada kasus prediksi, biasanya parameter yang diperkirakan akan membentuk suatu model linear. Kenyataannya, model yang lebih mendekati hasil sebenarnya tidaklah berbentuk linear tetapi membentuk suatu model non-linear. Dalam memperkirakan parameter non-linear biasanya menggunakan prinsip kuadrat terkecil. Salah satu prosedur yang digunakan adalah *gradient descent*. Pada prosedur ini, terjadi perubahan pada perkiraan awal parameter sehingga diperoleh parameter yang akan digunakan pada model non-linear. [8]

Jaringan saraf tiruan diperkenalkan pertama kali pada tahun 1943 oleh seorang ahli saraf Warren McCulloch dan seorang ahli logika Walter Pitts. Analisis ini berawal dari perkembangan pengumpulan data yang terjadi dalam jumlah besar sehingga melibatkan jutaan pengamatan dan ratusan atau ribuan prediktor. Contohnya meliputi data penjualan, data pemakaian kartu kredit, pemantauan proses produksi secara online, pengenalan karakter optik, data penyaringan email internet, data rangkaian microchip dan data rekam medis terkomputerisasi [7].

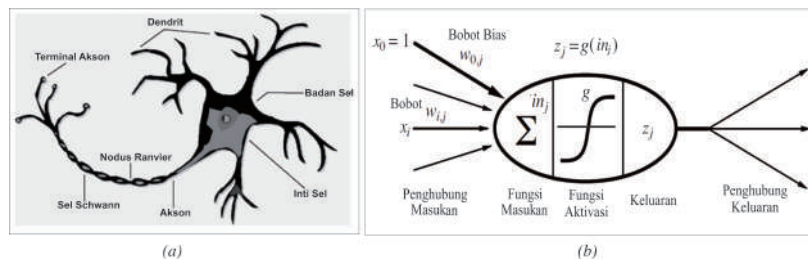
Jaringan saraf tiruan merupakan model yang meniru cara kerja jaringan saraf secara biologi. Neuron tiruan dirancang untuk menirukan karakteristik neuron biologis. Karakteristik jaringan saraf tiruan ditentukan oleh pola hubungan antara neuron-neuronnya yang biasanya disebut arsitektur jaringan, metode penentuan

bobot dan fungsi aktivasinya. Jaringan saraf tiruan memerlukan data dari masalah yang akan diselesaikan dan kemudian menyaring informasi yang diperoleh dari data melalui pelatihan [7].

Pada jaringan saraf tiruan, parameter yang digunakan untuk membentuk model adalah bobot yang merupakan penghubung antar unit lapisan. Pada algoritma pembelajaran jaringan saraf tiruan, prosedur yang digunakan adalah prosedur *gradient descent*. Oleh karena itu, model yang akan terbentuk dari jaringan saraf tiruan adalah model non-linear.

2. Jaringan Saraf Tiruan

Pembuatan struktur jaringan saraf tiruan diilhami oleh struktur jaringan sel saraf. Manusia memiliki kemampuan intelektual untuk mempelajari berbagai masalah yang rumit. Untuk beberapa ahli biologi, psikologi, dan dokter telah mencoba untuk memahami bagaimana fungsi otak [4].



Gambar 1. Sebuah Sel Neuron. (a). Model Biologi [4], (b). Model Tiruan [8].

Jaringan otak manusia tersusun tidak kurang dari 10^{13} buah neuron yang masing-masing terhubung oleh sekitar 10^{15} buah dendrit. Berdasarkan Gambar 1 (a), sebuah sel neuron menerima sinyal dari dendrit. Ketika sebuah neuron menerima sebuah sinyal menyebabkan neuron menjadi aktif. Ketika neuron aktif, sebuah sinyal di teruskan ke akson. Sinyal akan meninggalkan neuron karena menuju ke terminal akson. Sinyal tersebut kemudian diteruskan ke neuron lain. Penjelasan lebih rinci tentang hal ini dapat diperoleh pada ilmu biologi molekular [4].

Pada Gambar 1 (b) disajikan model tiruan dari neuron. Berdasarkan gambar tersebut, unit masukan $x_i, i = 0, 1, \dots, n$ dikalikan dengan masing-masing bobot yang bersesuaian $w_{ij}, j = 1, 2, \dots, m$ dengan $x_0 = 1$ merupakan bias dengan bobot w_{0j} . Kemudian dilakukan penjumlahan dari hasil perkalian tersebut dan menghasilkan keluaran yaitu $in_j, j = 1, 2, \dots, m$ yang dapat dinyatakan sebagai berikut:

$$in_j = \sum_{i=0}^n x_i w_{ij} \quad (2.1)$$

Kemudian fungsi aktivasi g diterapkan ke $in_j, j = 1, 2, \dots, m$ dan hasilnya $z_j = g(in_j) = g(\sum_{i=0}^n x_i w_{ij})$ diteruskan ke neuron lain sebagai masukan baru. Kumpulan dari neuron dibuat menjadi sebuah jaringan yang akan berfungsi sebagai alat komputasi. Setelah memutuskan model matematika untuk sebuah neuron, langkah

selanjutnya adalah menghubungkan neuron-neuron tersebut untuk membentuk suatu jaringan [9].

Dalam menghubungkan neuron-neuron, terdapat dua jenis jaringan yaitu *feedforward* dan *recurrent*. Pada jaringan *feedforward*, hubungan antar lapisan dilakukan satu arah. Jaringan ini dibedakan menjadi dua yaitu jaringan lapisan tunggal dan jaringan lapisan banyak. Pada jaringan lapisan tunggal, masukan neuron secara langsung terhubung pada keluaran. Sedangkan jaringan lapisan banyak, terdapat jenis lapisan lain yang biasa disebut jaringan tersembunyi. Sedangkan pada jaringan *recurrent*, hasil keluarannya kembali ke masukannya sendiri. Jaringan *recurrent* mendukung memori jangka pendek sehingga jaringan ini lebih menarik sebagai model otak, tetapi juga lebih sulit dimengerti [9].

Fungsi aktivasi dapat berupa fungsi linear atau nonlinear. Fungsi aktivasi tertentu dipilih untuk memenuhi beberapa jenis masalah yang berusaha diselesaikan oleh neuron. Fungsi aktivasi lapisan tersembunyi dibutuhkan untuk memperkecil data masukan ke dalam jaringan. Hasil keluaran dari fungsi aktivasi yang diterapkan dilanjutkan ke neuron berikutnya dalam jaringan. Terdapat tiga fungsi aktivasi yang sering digunakan pada jaringan saraf tiruan yaitu linear, sigmoid, dan tangen sigmoid. Fungsi aktivasi yang sering digunakan pada metode *backpropagation* adalah fungsi sigmoid. Hal ini disebabkan fungsi sigmoid lebih mendekati kinerja sinyal pada otak dan sangat mudah digunakan. Selain itu, fungsi sigmoid juga mudah untuk dicari turunannya. Fungsi sigmoid $f(x) = \frac{1}{1+e^{-x}}$ menghasilkan sebuah nilai antara nilai 0 dan 1. Satu hal penting untuk dicatat tentang fungsi aktivasi sigmoid adalah fungsi sigmoid ini hanya menghasilkan nilai positif. Jika dibutuhkan jaringan saraf untuk mengembalikan angka negatif maka fungsi sigmoid tidak tepat untuk digunakan [4].

Proses pembentukan model ada dua yaitu proses pelatihan dan proses validasi. Pada proses pelatihan, bobot-bobot antar unit dihubungkan antar lapisan. Terdapat dua jenis metode dalam proses pelatihan, yaitu metode pelatihan dengan pengawasan dan metode pelatihan dengan tanpa pengawasan. Pada metode pelatihan dengan pengawasan, data keluaran dari data pengamatan dimasukkan ke dalam proses pelatihan. Sedangkan pada metode pelatihan tanpa pengawasan, data keluaran pengamatan tidak diperlukan untuk proses pelatihan. Metode pelatihan tanpa pengawasan biasanya digunakan untuk mengklasifikasi suatu data masukan. Sedangkan pada proses validasi, jaringan saraf sangat penting karena menentukan apakah pelatihan tambahan diperlukan. Bobot disesuaikan berdasarkan seberapa baik jaringan saraf pada proses pelatihan. Proses ini diulang hingga kesalahan validasi berada dalam batas yang dapat diterima [4].

Galat merupakan perbedaan antara keluaran yang diperoleh dari jaringan dengan keluaran dari data pengamatan. *Gradient descent* merupakan prosedur untuk menghitung kemiringan dari fungsi $Err(W)$ yang berubah-ubah sampai nilai minimum galat tercapai. Perubahan bobot dinotasikan sebagai ΔW dan kemiringan dari fungsi $Err(W)$ dinotasikan sebagai $\nabla Err(W)$. Terdapat hubungan antara kemiringan tunggal dari fungsi $Err(W)$ untuk setiap bobot tunggal w_{jk} sehingga

dapat dinyatakan sebagai berikut:

$$\Delta w_{jk} = -\eta \frac{\partial Err(W)}{\partial w_{jk}} \quad (2.2)$$

di mana w_{jk} yaitu bobot yang berada di antara lapisan tersembunyi pada unit j dan lapisan keluaran pada unit k [6].

Selama proses pelatihan, jaringan mencoba untuk meminimalkan galat. Galat untuk setiap data pelatihan ke- k didefinisikan sebagai berikut:

$$Err_k(W) = \frac{1}{2} \sum_{j=1}^p (t_{kj} - y_{kj})^2 \quad (2.3)$$

di mana t_{kj} merupakan keluaran yang ingin dicapai dari data pengamatan ke j untuk dari data ke- k atau target dan y_{kj} adalah nilai keluaran dari lapisan keluaran ke j untuk data ke- k [1].

Sebuah vektor masukan yaitu $\mathbf{x}_k = (x_{k1}, x_{k2}, \dots, x_{kN})^T$ berada di lapisan masukan. Nilai-nilai unit masukan yang didistribusikan ke setiap unit tersembunyi dapat dinyatakan sebagai berikut:

$$in_{kp}^h = \sum_{i=1}^N w_{pi}^h x_{ki} + w_{p0}^h \quad (2.4)$$

di mana w_{pi}^h adalah bobot penghubung unit masukan i ke unit tersembunyi p dan w_{p0}^h adalah bias. Asumsikan f_p^h adalah fungsi aktivasi ke lapisan tersembunyi [3].

$$i_{kp} = f_p^h(in_{kp}^h) \quad (2.5)$$

Persamaan pada unit keluaran dan fungsi aktivasinya dapat dinyatakan sebagai berikut:

$$in_{kj}^o = \sum_{p=1}^L w_{jp}^o i_{kp} + w_{j0}^o \quad (2.6)$$

$$y_{kj} = f_j^o(in_{kj}^o) \quad (2.7)$$

di mana "h" dan "o" menunjukkan lapisan tersembunyi (*hidden layer*) dan lapisan keluaran (*output layer*) [3].

Pembaharuan bobot dapat dilakukan pada lapisan tersembunyi dan lapisan keluaran. Pertama, pembaharuan bobot lapisan keluaran ke lapisan tersembunyi dapat dilakukan dengan melakukan turunan parsial fungsi galat pada Persamaan (2.3) terhadap bobot dari lapisan tersembunyi ke lapisan keluaran sehingga diperoleh persamaan sebagai berikut:

$$\frac{\partial Err_k(W)}{\partial w_{jp}^o} = -(t_{kj} - y_{kj}) f_j^{o'}(in_{kj}^o) i_{kp} \quad (2.8)$$

Kedua, pembaharuan bobot lapisan tersembunyi ke lapisan masukan dapat dinyatakan sebagai berikut:

$$\frac{\partial Err_k(W)}{\partial w_{pi}^h} = - \sum_{j=1}^Q (t_{kj} - y_{kj}) f_j^{o'}(in_{kj}^o) w_{jp}^o f_p^{h'}(in_{kp}^h) x_{ki} \quad (2.9)$$

di mana $Err_k(W)$ adalah fungsi galat yang memetakan nilai dari bobot W ke galat yang diperoleh dari hasil keluaran jaringan. Bobot yang telah diperbaharui dapat digunakan kembali untuk melakukan pembaharuan bobot apabila kriteria pemberhentian tidak terpenuhi [3].

Metode *backpropagation* termasuk pada pelatihan dengan pengawasan dan didesain untuk operasi pada jaringan *feedforward* dengan banyak lapisan. Secara umum, algoritma pembelajaran jaringan digunakan untuk melatih jaringan secara iterasi sehingga bobot antar unit dapat disesuaikan dan galat antara keluaran oleh jaringan dan keluaran yang ingin dicapai menjadi kurang dari yang ditentukan sebelumnya. Berikut, algoritma pembelajaran jaringan saraf tiruan. Pertama, Inisialisasi jaringan. Nilai awal dari bobot perlu ditentukan agar dapat melakukan proses pelatihan. Jaringan saraf tiruan umumnya diinisialisasi dengan bobot acak. Kedua, *Feedforward*. Informasi diteruskan ke depan melalui jaringan dari lapisan masukan ke lapisan tersembunyi dan lapisan keluaran melalui fungsi aktivasi data masukan dan bobot. Ketiga, Penilaian galat. Keluaran yang ingin dicapai dibandingkan terhadap keluaran oleh jaringan. Jika galat diperoleh kurang dari batas yang ditentukan sebelumnya maka proses pelatihan dan algoritma pembelajaran dihentikan. Keempat, *Backpropagation*. Galat pada lapisan keluaran digunakan untuk memodifikasi ulang bobot. Langkah ini membagikan galat ke lapisan sebelumnya melalui jaringan dan menghitung perubahan galat yang berhubungan dengan perubahan nilai bobot. Kelima, Penyesuaian bobot. Penyesuaian pada bobot dapat menggunakan *gradien descent* dengan tujuan mengurangi galat pada bobot dan bias. Setiap siklus melalui proses ini disebut iterasi [5].

3. Model Non-Linear pada Jaringan Saraf Tiruan

Berikut langkah-langkah dalam menentukan model non-linear jaringan saraf tiruan menggunakan metode *backpropagation*.

Pertama, pendefinisian data masukan dan data keluaran yang ingin dicapai. Data yang digunakan sebagai masukan adalah data x_{k1} dan x_{k2} . Sedangkan data keluaran yang ingin dicapai (*target*) adalah t_{k1} . Data yang akan digunakan dalam proses pelatihan jaringan saraf tiruan adalah sebagai berikut:

Tabel 1. Contoh Data

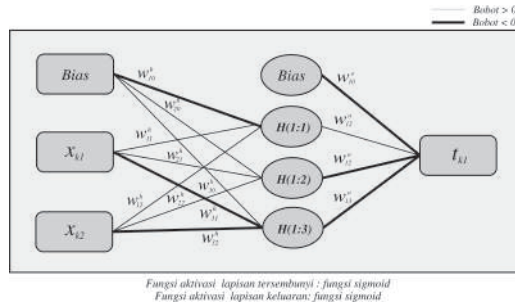
k	x_{k1}	x_{k2}	t_{k1}
1	1	1	0
2	1	0	1
3	0	1	1
4	0	0	0

Kedua, normalisasi data. Pada Tabel 1 disajikan data bernilai 0 dan 1. Karena fungsi aktivasi yang digunakan adalah fungsi sigmoid maka tidak memerlukan proses normalisasi.

Ketiga, perancangan arsitektur jaringan saraf tiruan. Berdasarkan Tabel 1, ar-

sitektur jaringan yang akan digunakan adalah 2-3-1 yang berarti jaringan memuat dua unit masukan, tiga unit tersembunyi, dan satu unit keluaran.

Berikut akan ditampilkan arsitektur jaringan saraf tiruan.



Gambar 2. Arsitektur Jaringan Saraf Tiruan

Keempat, proses pelatihan jaringan saraf tiruan. Algoritma pembelajaran jaringan saraf tiruan yang akan dilakukan adalah sebagai berikut:

- (1) Inisialisasi jaringan.

Berdasarkan Gambar 2, inisialisasi bobot awal untuk Tabel 1 adalah $w_{10}^h = -0,5$, $w_{20}^h = 0,5$, $w_{30}^h = 0,5$, $w_{11}^h = 0,5$, $w_{21}^h = 0,5$, $w_{31}^h = -0,5$, $w_{12}^h = 0,5$, $w_{22}^h = 0,5$, $w_{32}^h = -0,5$, $w_{10}^o = -0,5$, $w_{11}^o = 0,5$, $w_{12}^o = -0,5$, dan $w_{10}^o = -0,5$.

Dalam pelatihan jaringan, data yang digunakan adalah 4 data, tingkat pemahaman (η) adalah 0,2, fungsi aktivasi antar lapisan menggunakan fungsi sigmoid dan penilaian galat menggunakan batas kurang dari 0,5.

- (2) *Feedforward*.

Setiap unit input ($x_{ki}, i = 1, 2$) menerima data masukan x_{ki} dan mendistribusikan data x_{ki} kepada semua unit pada lapisan tersembunyi (*hidden unit*). Berikut langkah-langkah pada jaringan *Feedforward*. Pada lapisan tersembunyi, hitung $in_{kp}^h = w_{p0}^h + \sum_{i=1}^2 x_{ki}w_{pi}^h$ dan $i_{kp} = f_p^h(in_{kp}^h)$, untuk $k = 1, j = 1, p = 1, 2$, dan f_p^h merupakan fungsi aktivasi. Sedangkan pada lapisan keluaran, hitung $in_{kj}^o = w_{j0}^o + \sum_{p=1}^2 i_{kp}w_{jp}^o$ dan $y_{kj} = f_j^o(in_{kj}^o)$, untuk $k = 1, j = 1$, dan $p = 1, 2$.

- (3) Penilaian galat.

Apabila $|(t_{k1} - o_{k1})| > 0,5$ pada data ke- k maka kriteria pemberhentian belum terpenuhi sehingga pelatihan jaringan belum bisa dihentikan.

- (4) Metode *backpropagation*.

Berikut langkah-langkah metode *backpropagation*. Untuk mengetahui informasi galat pada unit keluaran, hitung $\delta_{kj}^h = (t_{kj} - y_{kj})f_j^o'(in_{kj}^o)$ untuk $k, j = 1, p = 1, 2$. Selanjutnya untuk mengetahui galat pada setiap unit tersembunyi, hitung $\delta_{kp}^h = f_p^h'(in_{kp}^h)(\delta_{kj}^o)w_{jp}^o$ untuk $k, j = 1, p = 1, 2$.

Pada lapisan tersembunyi, hitung $\Delta w_{jp}^o = \eta \delta_{kj}^o i_{kp}$ untuk $j = 1, k = 1$ dan $p = 1, 2$ untuk mengetahui nilai koreksi bobot dan bias. Sedangkan pada

lapisan masukan, hitung $\Delta w_{pi}^h = \eta \delta_{kp}^h x_{ki}$ untuk $k = 1, i = 1, 2$ dan $x_{k0} = 1$ untuk mengetahui nilai koreksi bobot dan bias.

(5) Penyesuaian bobot.

Sebagai langkah terakhir dari algoritma pembelajaran jaringan saraf tiruan, berikut langkah-langkah penyesuaian bobot. Pada lapisan tersembunyi, hitung $w_{jp}^o(t+1) = w_{jp}^o(t) + \Delta w_{jp}^o(t)$, $j = 1, p = 1, 2$. Sedangkan pada lapisan masukan, hitung $w_{pi}^h(t+1) = w_{pi}^h(t) + \Delta w_{pi}^h(t)$, $j = 1, p = 1, 2$, di mana $w_{pi}^h(t+1)$ dan $w_{jp}^o(t+1)$ adalah bobot baru yang akan digunakan pada proses selanjutnya.

Sebelum ke langkah selanjutnya, ulangi langkah (b) sampai (d) hingga penilaian galat terpenuhi. Tabel 2 adalah data pelatihan iterasi ke-1984.

Tabel 2. Data pelatihan pada iterasi ke-1984

k	1	2	3	4	k	1	2	3	4
i_{11}	0,9994	0,9595	0,9622	0,2660	w_{12}^o	-2,9403	-2,9334	-2,9252	-2,9269
i_{12}	0,7496	0,4598	0,5063	0,2166	w_{13}^o	0,4300	0,4339	0,4385	0,4347
i_{13}	0,1329	0,2603	0,2818	0,4752	w_{10}^h	-1,0198	-1,0175	-1,0151	-1,0214
y_{11}	0,4995	0,6642	0,6453	0,2280	w_{11}^h	4,1847	4,1870	4,1870	4,1870
Galat	< 0,5	< 0,5	< 0,5	< 0,5	w_{12}^h	4,2532	4,2532	4,2556	4,2556
δ_{11}^o	-0,1249	0,0749	0,0812	-0,0401	w_{20}^h	-1,2626	-1,2736	-1,2855	-1,2815
δ_{11}^h	-0,0003	0,0116	0,0118	-0,0314	w_{21}^h	1,1013	1,0904	1,0904	1,0904
δ_{12}^h	0,0685	-0,0547	-0,0595	0,0199	w_{22}^h	1,2989	1,2989	1,2870	1,2870
δ_{13}^h	-0,0062	0,0062	0,0071	-0,0044	w_{30}^h	-0,1018	-0,1006	-0,0992	-0,1000
w_{10}^o	-1,8905	-1,8756	-1,8593	-1,8673	w_{31}^h	-0,9426	-0,9414	-0,9414	-0,9414
w_{11}^o	3,9734	3,9878	4,0034	4,0013	w_{32}^h	-0,8350	-0,8350	-0,8336	-0,8336

Kelima, proses validasi model jaringan saraf tiruan. Berdasarkan Tabel 1, proses validasi tidak dibutuhkan karena semua data pada Tabel 1 telah digunakan pada proses pelatihan.

Keenam, pemodelan non-linear pada jaringan saraf tiruan. Setelah melakukan algoritma pembelajaran jaringan, diperoleh bobot-bobot baru yang akan digunakan untuk memodelkan data pada Tabel 1. Berikut langkah-langkah pemodelan data Tabel 1. Pertama, hitung $in_{kp}^h = w_{p0}^h + \sum_{i=1}^2 x_{ki} w_{pi}^h$ dan $H(1 : p) = i_{kp} = f_p^h(in_{kp}^h)$, untuk $k = 1, j = 1, p = 1, 2$, dan f_p^h merupakan fungsi aktivasi pada lapisan tersembunyi. Kedua, pada lapisan keluaran, hitung $in_{kj}^o = w_{j0}^o + \sum_{p=1}^2 i_{kp} w_{jp}^o$ dan $y_{kj} = f_j^o(in_{kj}^o)$, untuk $k = 1, j = 1, p = 1, 2$, dan f_j^o merupakan fungsi aktivasi pada lapisan keluaran.

Berdasarkan Tabel 2 pada iterasi ke-1984 dan dengan memilih galat yang paling kecil yaitu data k=4, diperoleh model non-linear pada jaringan saraf tiruan sebagai berikut:

$$y_{k1} = \frac{1}{1 + e^{-(-1,86734 + 4,00128H(1:1) - 2,92693H(1:2) + 0,434696H(1:3))}} \quad (3.1)$$

Berdasarkan model jaringan saraf tiruan yang diperoleh, hasil peramalan untuk data pada Tabel 1 ditampilkan sebagai berikut:

Pada data Tabel 1 disajikan data yang bernilai 0 dan 1. Berdasarkan Tabel

Tabel 3. Hasil Peramalan Model Jaringan Saraf Tiruan

k	x_{k1}	x_{k2}	t_{k1}	$H(1 : 1)$	$H(1 : 2)$	$H(1 : 3)$	y_{k1}
1	1	1	0	0,9994	0,7495	0,1330	0,4989
2	1	0	1	0,9595	0,4524	0,2609	0,6816
3	0	1	1	0,9621	0,5014	0,2822	0,6542
4	0	0	0	0,2648	0,2173	0,4750	0,2249

3 dan melakukan pembulatan data keluaran oleh jaringan, disimpulkan bahwa model jaringan saraf tiruan pada Persamaan (3.1) dapat digunakan untuk memprediksi nilai t_{k1} pada data Tabel 1.

4. Kesimpulan

Jaringan saraf tiruan menggunakan metode *backpropagation* merupakan suatu metode penyelesaian masalah terkait dengan sekumpulan data yang didasarkan pada cara kerja otak manusia dengan melakukan perubahan bobot-bobot yang diperoleh secara acak sehingga galat kurang dari batas yang telah ditentukan.

Model jaringan saraf tiruan untuk data Tabel 1 dapat dinyatakan sebagai berikut:

$$y_{k1} = \frac{1}{1 + e^{-(-1,86734+4,00128H(1:1)-2,92693H(1:2)+0,434696H(1:3))}} \quad (4.1)$$

dengan

$$H(1 : 1) = \frac{1}{1 + e^{-(-1,02137+4,187007x_{k1}+4,255588x_{k2})}}$$

$$H(1 : 2) = \frac{1}{1 + e^{-(-1,28147+1,090377x_{k1}+1,287024x_{k2})}}$$

$$H(1 : 3) = \frac{1}{1 + e^{-(-0,10004-0,94135x_{k1}-0,83359x_{k2})}}$$

di mana x_{k1}, x_{k2} adalah unit pada lapisan masukan, $H(1 : j)$ adalah unit pada lapisan tersembunyi untuk $j = 1, 2, 3$, dan y_{k1} adalah unit pada lapisan keluaran. Pada unit lapisan tersembunyi, model yang dihasilkan berbentuk tidak linear sehingga bobot yang digunakan pada unit lapisan keluaran juga tidak linear. Oleh karena itu, Persamaan (4.1) merupakan model non-linear pada jaringan saraf tiruan.

Jika pada proses pelatihan menggunakan batas galat yang sangat kecil maka iterasi yang diperoleh dari proses pelatihan menjadi semakin besar.

Daftar Pustaka

- [1] Braspenning, P.J., F. Thuijsman and A.J.M.M. Weijters. 1986. *Artificial Neural Network: An Introduction to ANN Theory and Practice*. Springer, New York.
- [2] Ertel, W. 2011. *Introduction of Artificial Intelligence*. Springer, New York.
- [3] Freeman, J.A. and D.M. Skapura. 2004. *Neural Network : Algorithms, Applications, and Programming Techniques*. Addison-Wisley Publishing Company, Inc, Amerika.

- [4] Heaton, J. 2008. *Introduction to Neural Networks for C*. Heaton Research, Inc, Amerika.
- [5] Lewis, N.D. 2015. *Build Your Own Neural Network Today !: With Step by Step Instructions Showing You How to Build Them Faster than You Imagined Possible Using R*. Create Space Independent Publishing Platform. ISBN : 978-1519101235.
- [6] McClelland, J.L. and D.E. Rumelhart. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge.
- [7] Neter, J, W. Wasserman, and M.H. Kutner. 2004. *Applied Linear Statistical Models*. The McGraw-Hill Companies, Inc., Singapore.
- [8] Rawlings, J.O, S.G Pantula and D.A Dickey . 1932. *Applied Regression Analysis*. Springer, USA.
- [9] Russel, S. and P. Norvig. 2010. *Artificial Intelligence : A Modern Approach*. Pearson Education Inc., New Jersey.