

IMPLEMENTASI *BACKPROPAGATION NEURAL NETWORK* DALAM PRAKIRAAN CUACA DI DAERAH BALI SELATAN

I Made Dwi Udayana Putra^{1§}, G. K. Gandhiadi², Luh Putu Ida Harini³

¹Jurusan Matematika FMIPA Universitas Udayana [dwiudayana@gmail.com]

²Jurusan Matematika FMIPA Universitas Udayana [gandhiadigk@yahoo.com]

³Jurusan Matematika FMIPA Universitas Udayana [ballidah@gmail.com]

[§]*Corresponding Author*

ABSTRACT

Weather information has an important role in human life in various fields, such as agriculture, marine, and aviation. The accurate weather forecasts are needed in order to improve the performance of various fields. In this study, use artificial neural network method with backpropagation learning algorithm to create a model of weather forecasting in the area of South Bali. The aim of this study is to determine the effect of the number of neurons in the hidden layer and to determine the level of accuracy of the method of artificial neural network with backpropagation learning algorithm in weather forecast models. Weather forecast models in this study use input of the factors that influence the weather, namely air temperature, dew point, wind speed, visibility, and barometric pressure. The results of testing the network with a different number of neurons in the hidden layer of artificial neural network method with backpropagation learning algorithms show that the increase in the number of neurons in the hidden layer is not directly proportional to the value of the accuracy of the weather forecasts, the increase in the number of neurons in the hidden layer does not necessarily increase or decrease value accuracy of weather forecasts we obtain the best accuracy rate of 51.6129% on a network model with three neurons in the hidden layer.

Keywords : Artificial neural network method, Back propagation algorithm, Weather forecasting

1. PENDAHULUAN

Cuaca mempunyai peranan penting bagi kehidupan manusia dalam menjalani aktivitas terutama aktivitas di tempat terbuka. Informasi cuaca sangat bermanfaat bagi berbagai bidang, antara lain bidang pertanian, kelautan, dan penerbangan. Bidang pertanian membutuhkan informasi cuaca agar dapat menentukan pola musim tanam pertanian, informasi cuaca dalam bidang kelautan bermanfaat untuk menentukan waktu berlayar yang tepat bagi nelayan untuk mencari ikan di laut, dan informasi cuaca dalam bidang penerbangan bermanfaat untuk mencegah terjadinya kecelakaan pesawat yang diakibatkan oleh faktor cuaca.

Prakiraan cuaca yang akurat sangat dibutuhkan agar dapat meningkatkan kinerja dari bidang-bidang tersebut. Salah satu metode yang dapat digunakan dalam membuat

prakiraan cuaca adalah jaringan syaraf tiruan karena metode jaringan syaraf tiruan dapat memberikan klasifikasi dari suatu cuaca (Ernawati, 2009).

Jaringan syaraf tiruan merupakan salah satu bagian dari metode kecerdasan buatan, dimana metode ini merepresentasikan cara kerja otak manusia untuk menyelesaikan suatu masalah klasifikasi (Hermawan, 2006). Salah satu algoritma dari jaringan syaraf tiruan yang dapat digunakan dalam menyelesaikan masalah klasifikasi adalah algoritma *back-propagation*.

Backpropagation merupakan algoritma yang menggunakan metode pembelajaran yang terawasi dan termasuk jaringan dengan banyak lapisan. Pada jaringan *backpropagation* terdapat tiga lapisan, yaitu lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Setiap lapisan

dari jaringan memiliki satu atau lebih neuron (Hermawan, 2006).

Lapisan *input* terdiri dari neuron-neuron yang merupakan *input* dari jaringan berupa faktor-faktor yang memengaruhi cuaca yaitu suhu udara, titik embun, kecepatan angin, jarak pandang, dan tekanan udara. Lapisan *output* merupakan solusi dari jaringan di mana neuron dari lapisan *output* berisikan *output* dari jaringan yaitu hasil prakiraan cuaca. Jumlah neuron pada lapisan *input* tergantung pada banyaknya *input* dari jaringan dan jumlah neuron pada lapisan *output* juga tergantung pada banyaknya *output* dari jaringan, akan tetapi untuk jumlah neuron pada lapisan tersembunyi tidak dapat ditentukan secara langsung sehingga perlu dilakukan analisis untuk menentukan jumlah neuron pada lapisan tersembunyi.

Penelitian terkait jaringan syaraf tiruan dengan algoritma pembelajaran *backpropagation* dilakukan oleh Priyanta untuk memprakirakan hujan harian di daerah Kuta Selatan Bali. Pada penelitian tersebut, *output* yang dihasilkan hanya dua kejadian, yaitu hujan dan tidak hujan (Priyanta, 2014).

Pada penelitian ini digunakan metode jaringan syaraf tiruan dengan algoritma pembelajaran *backpropagation* untuk memprakirakan cuaca di daerah Bali Selatan dengan *output* 15 kejadian cuaca yaitu cerah, cerah berawan, cerah berawan dengan awan cumulonimbus (CB), berawan, berawan dengan awan cumulonimbus (CB), badai guntur, hujan ringan, hujan ringan dengan awan cumulonimbus (CB), hujan sedang, hujan sedang dengan awan cumulonimbus (CB), hujan lebat, hujan lebat dengan awan cumulonimbus (CB), hujan ringan dengan badai guntur, hujan sedang dengan badai guntur dan hujan lebat dengan badai guntur.

Informasi cuaca dengan awan cumulonimbus (CB) sangat ditekankan pada penelitian ini karena awan cumulonimbus (CB) ini sangat berbahaya dalam suatu penerbangan, mengingat banyak kecelakaan pesawat yang diakibatkan oleh awan cumulonimbus (CB).

2. BACKPROPAGATION

Backpropagation adalah salah satu algoritma pada jaringan syaraf tiruan yang sering digunakan dalam mencari bobot optimal. Pada jaringan *backpropagation* terdapat pola *input* dan pola *output* yang diinginkan. Pada saat jaringan diberikan suatu pola, nilai bobot-bobot diubah agar dapat memperkecil perbedaan antara pola *output* dari jaringan dan pola *output* yang diinginkan. Pelatihan jaringan dilakukan berulang-ulang sampai semua pola *output* dari jaringan dapat mengenali pola *output* yang diinginkan (Hermawan, 2006).

2.1 Inisialisasi Bobot Awal

Inisialisasi bobot awal dari algoritma ini diperoleh dari proses acak atau dengan menggunakan algoritma Nguyen-Widrow. Algoritma Nguyen-Widrow merupakan algoritma optimasi nilai bobot dan bias awal dari jaringan *backpropagation* sehingga proses belajar jaringan untuk mencapai bobot optimal menjadi lebih cepat dibandingkan dengan penginisialisasian secara random/acak.

Algoritma inisialisasi Nguyen-Widrow adalah (Fausett, 1994):

1. Tentukan:

n = jumlah neuron *input*

p = jumlah neuron pada lapisan tersembunyi

β = faktor skala dengan nilai $0,7\sqrt{p}$

2. Inisialisasi bobot-bobot (v_{ij} (lama)) menggunakan nilai acak antara -0,5 dan 0,5.

3. Hitung $\|v_{ij}\| = \sqrt{v_{1j}^2 + v_{2j}^2 + \dots + v_{nj}^2}$

4. Bobot yang dipakai sebagai inisialisasi

$$v_{ij} = \frac{\beta v_{ij}(\text{lama})}{\|v_{ij}\|}$$

Nilai bias yang digunakan, $v_{0,j}$ = bilangan acak antara $-\beta$ dan β .

2.2 Fungsi Aktivasi

Fungsi aktivasi yang digunakan dalam *backpropagation* harus memenuhi tiga syarat, yaitu kontinu, terdiferensial dengan mudah, dan merupakan fungsi tidak turun (Fausett, 1994).

Fungsi aktivasi yang paling sering digunakan dalam jaringan syaraf tiruan adalah fungsi sigmoid biner dan didefinisikan sebagai berikut.

$$f(x) = \frac{1}{1 + e^{-x}}$$

dan turunannya

$$f'(x) = f(x)(1 - f(x))$$

Fungsi sigmoid biner mempunyai nilai maksimum satu. Sehingga untuk pola *input* dan *output* dengan nilai lebih besar dari satu harus ditransformasi agar setiap polanya memiliki *range* nilai yang sama dengan fungsi sigmoid biner.

2.3 Algoritma Pelatihan *Backpropagation*

Berikut adalah algoritma pelatihan *backpropagation* (Hermawan, 2006):

- (1) Inisialisasi bobot awal dengan bilangan acak kecil.
- (2) Tentukan maksimum iterasi, Target *error* yang diinginkan, dan *Learning rate* (α).
- (3) Bila kondisi berhenti bernilai FALSE, kerjakan langkah-langkah berikut:
 1. Untuk setiap pasangan elemen pelatihan, kerjakan:

Feedforward:

- a. Setiap neuron *input* x_i dengan $i = 1, 2, 3, \dots, n$, mendapat sinyal x_i dan diteruskan ke semua neuron pada lapisan tersembunyi.
- b. Setiap neuron pada suatu lapisan tersembunyi z_j dengan $j = 1, 2, 3, \dots, p$, menjumlahkan sinyal-sinyal *input* berbobot:

$$z_{net_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

dengan menerapkan fungsi aktivasi, hitung sinyal *output* pada lapisan tersembunyi:

$$z_j = f(z_{net_j})$$

selanjutnya kirimkan sinyal tersebut ke semua neuron di lapisan *output*.

- c. Setiap neuron *output* y_k dengan $k = 1, 2, 3, \dots, m$, menjumlahkan sinyal-sinyal *input* berbobot:

$$y_{net_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

dengan menerapkan fungsi aktivasi, hitung sinyal *output* pada lapisan *output*:

$$y_k = f(y_{net_k})$$

Catatan: langkah (b) dilakukan sebanyak jumlah lapisan tersembunyi.

Backpropagation:

- d. Setiap neuron *output* y_k dengan $k = 1, 2, 3, \dots, m$, menerima target pola *output* yang berkaitan dengan pola *input* pelatihan. Hitung *error* pada lapisan *output*:

$$\delta_k = (t_k - y_k) f'(y_{net_k})$$

t_k adalah target *output* yang diharapkan.

Selanjutnya hitung perubahan bobot (digunakan untuk memperbaiki nilai bobot w_{jk}):

$$\Delta w_{jk} = \alpha \delta_k z_j$$

Hitung juga perubahan bobot bias (digunakan untuk memperbaiki nilai bobot w_{0k}):

$$\Delta w_{0k} = \alpha \delta_k$$

Kirimkan δ_k ini ke neuron pada lapisan tersembunyi.

- e. Setiap neuron pada lapisan tersembunyi z_j dengan $j = 1, 2, 3, \dots, p$, menjumlahkan faktor delta pada lapisan tersembunyi:

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{jk}$$

nilai ini dikalikan dengan turunan dari fungsi aktivasinya dan digunakan untuk menghitung informasi *error* pada lapisan tersembunyi:

$$\delta_j = \delta_{net_j} f'(z_{net_j})$$

selanjutnya hitung perubahan bobot (digunakan untuk memperbaiki nilai bobot v_{ij})

$$\Delta v_{ij} = \alpha \delta_j x_i$$

Selanjutnya hitung perubahan bobot bias (digunakan untuk memperbaiki nilai bobot v_{0j}):

$$\Delta v_{0j} = \alpha \delta_j$$

Perubahan bobot

- f. Setiap neuron *output* y_k dengan $k = 1, 2, 3, \dots, m$, memperbaharui bobot dan biasnya ($j = 0, 1, 2, \dots, p$):

$$w_{jk}^* = w_{jk} + \Delta w_{jk}$$

setiap neuron pada lapisan tersembunyi z_j dengan, memperbaharui bobot dan biasnya ($i = 0, 1, 2, \dots, n$):

$$v_{ij}^* = v_{ij} + \Delta v_{ij}$$

2. Tes kondisi berhenti

Kondisi penghentian yang digunakan dalam algoritma ini adalah jumlah maksimum iterasi dan target *error*. Pelatihan akan dihentikan apabila jumlah iterasi pelatihan melebihi jumlah iterasi maksimum atau jika *error* yang diperoleh dalam pelatihan lebih kecil dari target *error*.

Pada tahap perubahan bobot dapat dimodifikasi dengan menambahkan momentum (μ) agar lebih cepat mencapai kekonvergenan.

Kecepatan kovergensi juga dapat ditingkatkan dengan memodifikasi *learning rate* menjadi *adaptive* dimana *learning rate* akan berubah-ubah selama proses pelatihan. Jika selisih antara *error* yang diperoleh dan *error* sebelumnya melebihi nilai maksimum kenaikan kinerja maka nilai bobot-bobot yang baru diabaikan, dan nilai *learning rate* diturunkan dengan cara mengalikan dengan nilai penurunan *learning rate*. Sebaliknya jika selisih *error* yang diperoleh dan *error* sebelumnya kurang dari maksimum kenaikan kinerja, semua nilai bobot yang baru akan disimpan, dan nilai *learning rate* ditingkatkan dengan cara mengalikan dengan nilai kenaikan *learning rate* (Hermawan, 2006).

Setelah pelatihan, dilakukan pengujian jaringan *backpropagation* dengan hanya menggunakan tahap *feedforward* dengan menggunakan bobot yang diperoleh dari proses pelatihan.

2.4 Pengukuran Ketepatan Model Prakiraan

Mengukur ketepatan model prakiraan dapat dihitung dengan persamaan berikut (Whidhiasih, Guritman, & Suprio, 2012):

$$\text{akurasi} = \frac{\text{total prakiraan yang benar}}{\text{total prakiraan}} \times 100\%$$

3. METODE PENELITIAN

Penelitian ini menggunakan data sekunder, yang didapat dari website ogimet.com. Data yang digunakan dalam penelitian ini yaitu kecepatan angin (knot), jarak pandang (meter), suhu udara ($^{\circ}\text{C}$), titik embun ($^{\circ}\text{C}$), tekanan udara (milibar) dan cuaca di daerah Bali Selatan.

Data pada penelitian ini dibagi menjadi dua, yaitu data pelatihan yang ditetapkan dari 1 Desember 2013 sampai 31 Januari 2014 dan data pengujian yang ditetapkan dari 1 Desember 2014 sampai 31 Januari 2015. Pemilihan periode data pada bulan-bulan tersebut dikarenakan pada bulan-bulan tersebut cuaca yang terjadi cukup bervariasi.

Pada penelitian ini teknik analisis data yang digunakan adalah sebagai berikut:

1. Menentukan banyaknya neuron pada lapisan *input*. Banyaknya neuron yang digunakan pada lapisan *input* adalah lima buah neuron karena jumlah *input* yang digunakan sebanyak lima buah *input* berupa faktor yang memengaruhi cuaca yaitu suhu udara, titik embun, kecepatan angin, tekanan udara, dan jarak pandang.
2. Menganalisis banyaknya neuron pada lapisan tersembunyi. Jumlah neuron pada lapisan tersembunyi harus kurang dari dua kali jumlah neuron pada lapisan *input* (Yohannes & Mahmudy, 2015). Jumlah neuron pada lapisan *input* adalah lima buah, sehingga dua kali jumlah neuron pada lapisan *input* adalah 10. Oleh karena itu jumlah neuron pada lapisan tersembunyi harus kurang dari 10 buah neuron. Pada

penelitian ini jumlah neuron pada lapisan tersembunyi akan dianalisis dari satu buah neuron hingga 10 buah neuron.

3. Menentukan jumlah neuron pada lapisan *output*. *Output* yang dihasilkan adalah 15 kejadian cuaca yaitu cuaca cerah, cerah berawan, cerah berawan dengan awan cumulonimbus (CB), berawan, berawan dengan awan cumulonimbus (CB), badai guntur, hujan ringan, hujan ringan dengan awan cumulonimbus (CB), hujan sedang, hujan sedang dengan awan cumulonimbus (CB), hujan lebat, hujan lebat dengan awan cumulonimbus (CB), hujan ringan dengan badai guntur, hujan sedang dengan badai guntur dan hujan lebat dengan badai guntur. *Output* akan dibuat dalam empat buah neuron. Empat buah neuron pada lapisan *output* masing-masing akan berisikan satu digit bilangan biner sehingga akan ada empat digit bilangan biner, dimana kombinasi empat digit bilangan biner akan menghasilkan 16 kombinasi bilangan biner. Oleh karena itu, 15 kejadian cuaca akan dibuat ke dalam empat buah neuron.
4. Inisialisasi bobot dan bias awal menggunakan algoritma Nguyen-Widrow.
5. Melakukan pelatihan jaringan terhadap data-data pelatihan.
6. Melakukan pengujian jaringan dengan menggunakan data-data pengujian. Proses pengujian jaringan hanya menggunakan tahap *feed-forward* dan menggunakan bobot-bobot yang diperoleh dari proses pelatihan.
7. Validasi model prakiraan cuaca. Tahap ini dilakukan untuk mengetahui persentase ketepatan model prakiraan cuaca.

4. HASIL DAN PEMBAHASAN

4.1 Prakiraan Cuaca dengan Metode Jaringan Syaraf Tiruan

Penerapan metode jaringan syaraf tiruan dengan algoritma pembelajaran *backpropagation* pada data cuaca di daerah Bali Selatan menggunakan fungsi aktivasi sigmoid biner.

4.2 Transformasi Data

Fungsi aktivasi sigmoid biner memiliki nilai maksimum satu, pada penelitian ini menggunakan data dengan nilai lebih dari satu, maka data harus ditransformasi ke selang $[0,1]$ terlebih dahulu agar data tersebut memiliki *range* yang sama dengan *range* nilai dari fungsi sigmoid biner.

4.3 Pola Output

Output atau target yang diinginkan pada penelitian ini berupa cuaca di daerah Bali Selatan dan terbagi dalam 15 pola. lima belas pola *output* di buat dalam empat buah neuron pada lapisan *output*. Empat buah neuron pada lapisan *output* berisikan empat digit bilangan biner dimana masing-masing neuron berisikan satu digit bilangan biner. Pola dari *output* atau target yang diinginkan dapat dilihat pada Tabel 1.

Tabel 1 Pola *output* atau target

No	Cuaca	Pola Output
1	Cerah	0000
2	Cerah Berawan	0001
3	Berawan	0010
4	Cerah Berawan dg Awan CB	0011
5	Berawan dg Awan CB	0100
6	Badai Guntur	0101
7	Hujan Ringan	0110
8	Hujan Sedang	0111
9	Hujan Ringan dg Awan CB	1000
10	Hujan Sedang dg Awan CB	1001
11	Hujan Ringan dg Badai Guntur	1010
12	Hujan Sedang dg Badai Guntur	1011
13	Hujan Lebat	1100
14	Hujan Lebat dg Awan CB	1101
15	Hujan Lebat dg Badai Guntur	1110

4.4 Pelatihan Jaringan

Tahap pelatihan jaringan merupakan proses pembelajaran terhadap pola data yang akan dikenali. Tahap pelatihan jaringan dilakukan dalam beberapa tahap, antara lain dengan inisialisasi bobot dan bias awal menggunakan algoritma Nguyen-Widrow, kemudian dilatih dengan algoritma pembelajaran *backpropagation*.

Tahap pelatihan jaringan menggunakan algoritma pembelajaran *backpropagation* dengan momentum dan *adaptive learning rate*. Pelatihan jaringan dilakukan menggunakan Matlab 7.8.0 (R2009a).

Proses pelatihan dilakukan menggunakan data-data pelatihan dengan target *error* ditentukan sebesar 0,01; untuk nilai *learning rate* ditentukan sebesar 0,1; untuk nilai momentum ditentukan sebesar 0,9 dan maksimum iterasi ditentukan 1.000.000 iterasi. Pada proses pelatihan, nilai *learning rate* bersifat *adaptive* yang berubah-ubah selama proses pelatihan. Jika selisih antara *error* yang diperoleh dan *error* sebelumnya melebihi nilai maksimum kenaikan kinerja maka nilai bobot-bobot yang baru diabaikan, dan nilai *learning rate* diturunkan dengan cara mengalikan dengan nilai penurunan *learning rate* dimana nilai penurunan *learning rate* yang ditentukan sebesar 0,7. Sebaliknya jika selisih *error* yang diperoleh dan *error* sebelumnya kurang dari maksimum kenaikan kinerja, semua nilai bobot yang baru akan disimpan, dan nilai *learning rate* ditingkatkan dengan cara mengalikan dengan nilai kenaikan *learning rate* dimana nilai kenaikan *learning rate* yang ditentukan sebesar 1,05. Nilai maksimum kenaikan kinerja ditentukan sebesar 1,04.

Proses pelatihan jaringan akan berhenti apabila jumlah iterasi telah melebihi batas maksimum yang ditentukan yaitu 1.000.000 iterasi, atau pelatihan akan dihentikan jika target *error* yang ditentukan telah tercapai.

Tahap pelatihan jaringan akan menghasilkan nilai bobot dan bias yang akan digunakan pada tahap pengujian jaringan.

4.5 Pengujian Jaringan

Tahap pengujian jaringan digunakan untuk menguji validasi data yang telah dilakukan pada tahap pelatihan dengan memasukkan data baru yang belum pernah dilatih sebelumnya untuk mengetahui nilai akurasi yang dihasilkan. Data yang digunakan pada tahap pengujian jaringan ini adalah data pengujian. Tahap pengujian jaringan dilakukan dengan menggunakan nilai

bobot dan bias yang diperoleh dari tahap pelatihan jaringan. Pada tahap pengujian jaringan ini hanya menggunakan tahap *feedforward* pada algoritma pembelajaran *backpropagation*. Hasil yang didapat pada tahap ini adalah *output* berupa prakiraan cuaca.

Setelah diperoleh hasil prakiraan cuaca, kemudian dihitung akurasi kete-patan kejadian cuaca yang sebenarnya dengan hasil prakiraan cuaca menggunakan jaringan syaraf tiruan dengan algoritma pembelajaran *backpropagation*. Akurasi hasil prakiraan cuaca dapat dilihat pada Tabel 2.

Tabel 2. Akurasi hasil prakiraan cuaca

No	Jumlah Neuron pada Lapisan Input	Jumlah Neuron pada Lapisan Tersembunyi	Jumlah Neuron pada Lapisan Output	Akurasi (%)
1	5	1	4	45,1613
2	5	2	4	39,5161
3	5	3	4	51,6129
4	5	4	4	45,9677
5	5	5	4	44,3548
6	5	6	4	34,6774
7	5	7	4	41,9355
8	5	8	4	40,7258
9	5	9	4	40,3226
10	5	10	4	46,7742

Penambahan jumlah neuron pada lapisan tersembunyi tidak berbanding lurus dengan nilai akurasi hasil prakiraan cuaca. Nilai akurasi tertinggi terjadi pada model jaringan dengan tiga buah neuron pada lapisan tersembunyi dengan nilai akurasi sebesar 51,6129% sedangkan nilai akurasi terendah diperoleh pada model jaringan dengan enam buah neuron pada lapisan tersembunyi dengan nilai akurasi sebesar 34,6774%.

Model jaringan dengan satu buah neuron pada lapisan tersembunyi mendapatkan nilai akurasi sebesar 45,1613%, namun terjadi penurunan nilai akurasi pada model jaringan dengan dua buah neuron pada lapisan tersembunyi yang mendapatkan nilai akurasi sebesar 39,5161%. Pada model jaringan dengan tiga buah neuron mendapatkan nilai akurasi sebesar 51,6129% dimana terjadi peningkatan

nilai akurasi dari nilai akurasi model jaringan dengan dua buah neuron pada lapisan tersembunyi. Hal ini menandakan bertambahnya jumlah neuron pada lapisan tersembunyi tidak serta merta meningkatkan atau menurunkan nilai akurasi prakiraan cuaca yang diperoleh dengan metode jaringan syaraf tiruan dengan algoritma pembelajaran *backpropagation*.

5. KESIMPULAN

Berdasarkan hasil dan pembahasan, dapat disimpulkan sebagai berikut:

1. Penambahan jumlah neuron pada lapisan tersembunyi tidak berbanding lurus dengan nilai akurasi hasil prakiraan cuaca. Bertambahnya jumlah neuron pada lapisan tersembunyi tidak serta merta meningkatkan atau menurunkan nilai akurasi prakiraan cuaca yang diperoleh dengan metode jaringan syaraf tiruan dengan algoritma pembelajaran *backpropagation*.
2. Hasil prakiraan cuaca di daerah Bali Selatan menggunakan metode ja-ringan syaraf tiruan dengan algoritma pembelajaran *backpropagation* diperoleh tingkat akurasi prakiraan cuaca terbaik sebesar 51,6129% pada model jaringan dengan tiga buah neuron pada lapisan tersembunyi.

DAFTAR PUSTAKA

- Ernawati, S. (2009). Aplikasi Hopfield Neural Network untuk Prakiraan Cuaca. *Meteorologi dan Geofisika*, 151-175.
- Fausett, L. (1994). *Fundamentals of Neural Network*. New York: Prentice Hall.
- Hermawan, A. (2006). *Jaringan Syaraf Tiruan Teori dan Aplikasi*. Yogyakarta: ANDI.
- Priyanta, I. B. (2014). Penerapan Algoritma Genetika pada Jaringan Syaraf Tiruan dalam Prakiraan Hujan Harian di Daerah Kuta Selatan Provinsi Bali. *Skripsi*. Program Studi Teknik Informatika Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana. Bukit Jimbaran.
- Whidhiasih, R. N., Guritman, S., & Suprio, P. T. (2012). Klasifikasi Kematangan Buah Manggis Ekspor dan Lokal Berdasarkan

Warna dan Tekstur Menggunakan Fuzzy Neural Network. *JIKA*, 71-77.

- Yohannes, E., & Mahmudy, W. F. (2015). Penentuan Upah Minimum Kota Berdasarkan Tingkat Inflasi Menggunakan Backpropagation Neural Network (BPNN). *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 33-40.