

## DIAGNOSA PENYAKIT KULIT PADA ANJING DENGAN ALGORITMA MULTILAYER PERCEPTRON

Rosenty Damanik<sup>1)</sup>, Monalisa Br.Sirait<sup>2)</sup>, Suci Yolanda<sup>3)</sup>,  
Ensunaria Ketaren<sup>4)</sup>, Indra Prianto Sinaga<sup>5)</sup>, Mawaddah Harahap<sup>6)</sup>

<sup>1,2,3,4,5)</sup>Program Studi Teknik Informatika Universitas Prima Indonesia Medan

<sup>6)</sup>Dosen Teknik Informatika Universitas Prima Indonesia Medan

Jl. Sekip Simp. Sei kambing, Medan. Telp 061-4578870

[rosenty29@gmail.com](mailto:rosenty29@gmail.com) , [monalissirait96@gmail.com](mailto:monalissirait96@gmail.com) , [Suciyolanda98@gmail.com](mailto:Suciyolanda98@gmail.com),  
[ensunaria.ketaren@gmail.com](mailto:ensunaria.ketaren@gmail.com) , [indrasinaga456@gmail.com](mailto:indrasinaga456@gmail.com), [mawaddah@unprimdn.ac.id](mailto:mawaddah@unprimdn.ac.id)

### Abstrak

Anjing merupakan hewan peliharaan yang populer. Kurangnya pengetahuan mengenai penyakit kulit pada anjing dan terbatasnya jumlah dokter hewan dapat membahayakan kesehatan hewan dan majikannya sendiri. Pengenalan penyakit kulit anjing dapat dilakukan dengan menggunakan metode jaringan syaraf tiruan Multi Layer Perceptron (MLP). MLP digunakan untuk mempelajari 8 jenis penyakit kulit, dan 20 gejala penyakit. Data training terdiri dari 22 kasus. Pengujian dengan menggunakan 10 layar tersembunyi dan maksimum epoch sebesar 100, menghasilkan nilai akhir error yang sudah cukup rendah, yaitu MSE = 0.01494 dan RMSE = 0.12223 untuk melakukan diagnosa penyakit secara akurat. Semua data pelatihan dan data pengujian berhasil dikenali dengan akurat.

**Kata Kunci** : Sistem Pakar, Anjing, Penyakit Kulit, Multi Layer Perceptron

### 1. PENDAHULUAN

Aktivitas memelihara hewan bukanlah hanya sekedar hobi yang tidak bermanfaat. Banyak manfaat yang didapatkan dari aktivitas tersebut. Berbagai penelitian telah menemukan bahwa hewan peliharaan memiliki manfaat yang beragam, seperti kesehatan fisiologis maupun psikologis. Penelitian menemukan bahwa memelihara anjing secara signifikan meningkatkan kualitas hidup dalam aspek fisik (Nurlayli, dan Hidayati, 2014). Sebuah penelitian menemukan bahwa nenek moyang anjing yang saat itu masih liar menjadi jinak dan berteman dengan manusia ketika mereka mulai suka mengonsumsi makanan manusia. Kebiasaan mengonsumsi makanan manusia ini lambat laun mengubah genetik anjing dan membuat mereka beradaptasi dengan manusia (Mirwa, 2016). Sejak saat itu, anjing menjadi hewan peliharaan yang populer, bahkan anjing sering disebut sebagai sahabat terbaik manusia karena anjing adalah hewan yang jinak, mudah dilatih dan dapat mengerti keinginan majikannya (Firscha, 2018). Majikan yang baik wajib memperhatikan kesehatan hewan peliharaannya. Penyakit kulit sering menyerang anjing yang menyebabkan mereka sering menggaruk dan menjilati bagian tubuh mereka. Kemungkinan penyakit yang diderita sangat banyak, mulai dari alergi hingga parasit (Zhang, 2012). Kurangnya pengetahuan mengenai penyakit kulit pada anjing dan terbatasnya jumlah dokter hewan dapat membahayakan kesehatan hewan dan majikannya sendiri. Permasalahan ini dapat diselesaikan dengan membangun sebuah

aplikasi sistem pakar. Dengan adanya aplikasi sistem pakar, orang awam dapat melakukan diagnosa penyakit yang sebenarnya hanya dapat dilakukan oleh para ahli.

Pengenalan penyakit dapat dilakukan dengan menggunakan metode jaringan syaraf tiruan. *Perceptron*. *Perceptron* menempati tempat khusus dalam pengembangan sejarah jaringan saraf, dan *Perceptron* adalah jaringan saraf pertama yang dijelaskan secara algoritmik, ditemukan oleh Rosenblatt, seorang psikolog, yang menginspirasi para insinyur, fisikawan, dan ahli matematika untuk mengabdikan upaya penelitian mereka ke berbagai aspek jaringan saraf di tahun 1960-an dan 1970-an (Anbazhagan dan Ponmuthuramalingam, 2016). *Perceptron* adalah algoritma yang terinspirasi dari jaringan syaraf atau neuron alami untuk melakukan klasifikasi biner (Ravi, 2016). *Perceptron* adalah neuron tunggal yang memproses beberapa sinyal vektor *input*  $x$  dan menghasilkan *output*  $y$  berdasarkan bobot  $w$  (Banda et.al, 2013).

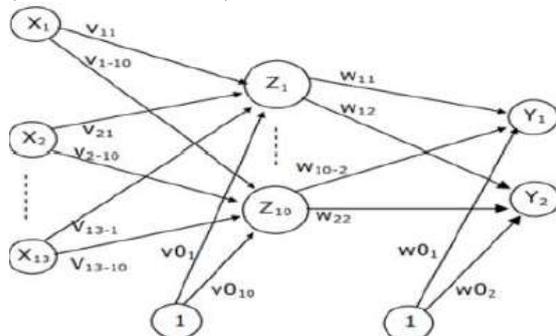
*Multilayer Perceptron (MLP)* adalah varian dari model *Perceptron* asli yang diusulkan oleh Rosenblatt. *MLP* memiliki satu atau lebih lapisan tersembunyi antara lapisan *input* dan lapisan *output*. *Neuron* berada di setiap lapisan, koneksi antar *neuron* berasal dari lapisan bawah ke lapisan atas, neuron di lapisan yang sama tidak saling berhubungan (Ramchoun, 2016). *MLP* merupakan salah satu arsitektur jaringan syaraf tiruan yang memiliki kompleksitas yang rendah dan memiliki kemampuan untuk memberikan hasil yang

memuaskan (Rani, 2017). *MLP* adalah algoritma pembelajaran yang populer karena jaringan syaraf tiruan ini dapat mengenal *output* yang diinginkan dan menyesuaikan nilai bobot sedemikian rupa, sehingga dapat memperoleh *output* yang diinginkan. Algoritma ini adalah algoritma paling populer dan efektif, serta mudah untuk dipelajari sebagai model jaringan syaraf tiruan multi lapisan yang kompleks. (Mia, 2015). *MLP* adalah algoritma jaringan syaraf tiruan penalaran maju (*feed foward*) yang dilatih dengan algoritma standard *Backpropagation* (Sudha, 2014). Penelitian lain terkait dengan *MLP* adalah metode ini digunakan untuk memprediksi jumlah produksi beras tahunan di Tamilnadu (Baskaran and Balaji 2013). *MLP* juga dapat digunakan dalam dunia medis seperti untuk mendeteksi kanker, melakukan klasifikasi dan analisis (Mishra, 2018).

## 2. TINJAUAN PUSTAKA

### 2.1 Multi Layer Perceptron

*Multi Layer Perceptron (MLP)* biasa disebut juga dengan metode *backpropagation* banyak lapisan. menggunakan *error output* dalam mengubah nilai bobot atau disebut dengan *backward*. Untuk mendapatkan nilai *error*, maka langkah awal yang dikerjakan adalah tahap *forward propagation*. Gambar 1 menampilkan arsitektur dari model *MLP* (Guntoro et.al, 2019).



Gambar 1. Arsitektur *MLP*

Fungsi aktivasi mempunyai beberapa karakteristik yang harus dipenuhi yaitu kontinu, diferensial dan fungsi yang tidak turun. Fungsi aktivasi yang sering digunakan adalah fungsi sigmoid yang memiliki *range* (0, 1) seperti yang terlihat pada persamaan 1 (Guntoro et.al, 2019).

$$f(x) = \frac{1}{1+e^{-x}}, \quad \text{dengan turunan}$$

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

dimana  $e$  = konstanta matematika yang bernilai 2.71828182845905

Algoritma pelatihan *backpropagation* adalah sebagai berikut (Guntoro et.al, 2019):

0. Inisialisasi bobot (ambil bobot awal dengan nilai random yang kecil). Tentukan pula nilai angka pembelajaran ( $\alpha$ ), nilai toleransi error (bila menggunakan nilai ambang sebagai

kondisi berhenti) dan set maksimum epoch (bila menggunakan banyaknya epoch sebagai kondisi berhenti).

1. While kondisi berhenti tidak terpenuhi, maka untuk setiap pasangan elemen yang akan dilatih, lakukan langkah ke-2 sampai ke-8.

#### Fase 1 : *Feedforward*

2. Setiap input  $x_i$  (dari unit ke-1 sampai unit ke- $n$  pada lapisan input) mengirimkan sinyal input ke semua unit yang ada di lapisan atasnya (ke lapisan tersembunyi):  $x_i$ .
3. Pada setiap unit di lapisan tersembunyi ( $z_j$ ;  $i = 1, 2, \dots, p$ ) menjumlahkan sinyal-sinyal input terbobot:

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

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

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

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

4. Tiap-tiap unit output ( $Y_k$ ,  $k=1, 2, 3, \dots, m$ ) menjumlahkan sinyal-sinyal input terbobot.

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

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

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

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

#### Fase 2 : *Backpropagation*

5. Tiap-tiap unit output ( $Y_k$ ,  $k=1, 2, 3, \dots, m$ ) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi errornya:

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

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{jk}$ ):

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

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{0k}$ ):

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

kirimkan  $\delta_k$  ini ke unit-unit yang ada di lapisan bawahnya.

6. Tiap-tiap unit tersembunyi ( $z_j$ ;  $j = 1, 2, \dots, p$ ) menjumlahkan delta inputnya (dari unit-unit yang berada di lapisan atasnya):

$$\delta\_in_j = \sum_{k=1}^m \delta_k w_{jk}$$

kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error:

$$\delta_j = \delta\_in_j f'(z\_in_j)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ ):

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

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{0j}$ ):

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

**Fase 3 : Tahap Update Bobot dan Bias**

7. Tiap-tiap unit output ( $Y_k, k=1,2,3,...m$ ) memperbaiki bias dan bobotnya ( $j=0,1,2,...p$ ):

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

Tiap-tiap unit tersembunyi ( $z_j; j = 1,2,...p$ ) memperbaiki bias dan bobotnya ( $i=0,1,2,...,n$ ):

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

8. Tes kondisi berhenti, bila mencapai maksimum *epoch* atau kuadrat *error* < target *error*.

Algoritma pengenalan hanya perlu menjalankan tahap *feedforward*. Pasangan  $y_k$  adalah *output* dari metode *MLP* (Guntoro et.al, 2019).

**2.2 Proses Pengujian**

Pengujian dilakukan pada pengaruh jumlah layer tersembunyi, *epoch* dan nilai parameter  $a$  terhadap nilai *error* dalam proses pelatihan data gejala penyakit. Nilai *error* diuji dengan menggunakan pengukuran *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)*. *MSE* dan *RMSE* dihitung dengan menggunakan persamaan berikut (Rani, 2017).

$$MSE = \frac{\sum t_i - o_i^2}{n} \quad RMSE = \sqrt{MSE}$$

dimana  $t$  merupakan *output* yang diharapkan,  $o$  adalah *output* sebenarnya dari jaringan *MLP* dan  $n$  adalah jumlah sampel di dalam *dataset*.

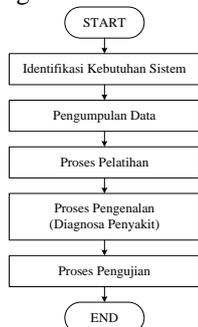
**3. METODE PENELITIAN**

**3.1 Jenis Penelitian**

Jenis penelitian yang dilakukan adalah penelitian kualitatif, dimana penelitian ini merupakan penelitian tentang riset yang melakukan analisis terhadap cara kerja metode *Multilayer Perceptron (MLP)* dalam melakukan diagnosa awal terhadap penyakit kulit yang dialami oleh anjing.

**3.2 Prosedur Kerja**

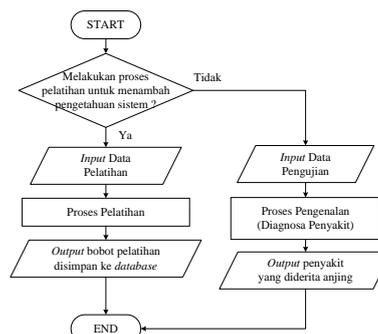
Langkah-langkah dalam menyelesaikan penelitian ditunjukkan pada gambar 2.



**Gambar 2.** Langkah-Langkah Penelitian

**3.3 Tahapan Metode MLP**

Tahapan di dalam penelitian terdiri dari dua proses, yaitu proses pelatihan dan proses pengenalan (diagnosa) penyakit kulit pada anjing. Tahapan proses penelitian dapat dilihat pada gambar 3.



**Gambar 3.** Tahapan Proses Penelitian

**4. HASIL DAN PEMBAHASAN**

**4.1 Hasil**

Tabel 1 menampilkan gejala-gejala penyakit yang mungkin dialami oleh anjing.

**Tabel 1.** Gejala Penyakit

Kode	Nama Gejala
G01	Memiliki kulit bercak kemerah-merahan
G02	Terus menerus menggaruk atau gatal berlebihan
G03	Memiliki kulit pecah
G04	Mengalami iritasi, gatal dan kulit wama berubah
G05	Mengalami gatal pada tapak kaki atau telinga
G06	Terus menerus mengganuk di area telinga atau menjilat kaki-kakinya
G07	Memiliki luka infeksi, benjolan dan kudis atau koreng pada kulit
G08	Memiliki bulu kusam dan rontok
G09	Memiliki kulit bersisik
G10	Memiliki kulit melepuh dan berisi nanah yang bisa pecah
G11	Memiliki lepuhan pada area perut yang tidak berbulu
G12	Memiliki bulu berminyak
G13	Mengalami ketombe atau kerak pada bulu
G14	Memiliki luka pada kepala, tapak, telinga dan kaki depan yang berbentuk lingkaran
G15	Memiliki luka radang yang sakit
G16	Mengalami luka gatal memerah pada telinga, wajah dan kaki
G17	Mengalami botak, kudis atau koreng dan sakit pada daerah luka
G18	Mengalami kurang darah atau anemia
G19	Memiliki benjolan keras pada kulit
G20	Mengalami penebalan kulit

Tabel 2 menampilkan hipotesis atau jenis penyakit kulit yang mungkin dialami oleh anjing.

**Tabel 2.** Hipotesis (Penyakit Kulit Anjing)

Kode Hipotesis	Nama Hipotesis	Nomor Urut
H0	Alergi Dematitis (Alergi Kulit)	000
H1	Infeksi Jamur	001
H2	Infeksi Bakteri	010
H3	Impetigo (Nanah)	011
H4	Seborrhea	100
H5	Ringworm	101
H6	Parasit (Kutu atau Tungau)	110
H7	Tumor Kulit	111

Tabel 3 menampilkan pengetahuan atau kasus penyakit kulit yang dialami oleh anjing (dihimpun dari penelitian) dan disertai dengan gejala penyakit.

**Tabel 3.** Pengetahuan atau Kasus yang Terjadi

Kode Pengetahuan	Ciri Gejala (x <sub>1</sub> x <sub>2</sub> x <sub>3</sub> x <sub>4</sub> x <sub>5</sub> ... x <sub>20</sub> )	Kode Hipotesis	Nomor Urut (y <sub>1</sub> y <sub>2</sub> y <sub>3</sub> )
P001	11100000000000000000	H0	0 0 0
P002	11110000000000000000	H0	0 0 0
P003	11101000000000000000	H0	0 0 0
P004	00011100000000000000	H1	0 0 1
P005	01001100000000000000	H1	0 0 1
P006	01011100000000000000	H1	0 0 1
P007	00000011100000000000	H2	0 1 0
P008	00000011100001000000	H2	0 1 0
P009	00000011100000100000	H2	0 1 0
P010	00010000011000100000	H3	0 1 1
P011	00010000011000000000	H3	0 1 1
P012	00000000110001000000	H3	0 1 1
P013	01010000000110000000	H4	1 0 0
P014	01000000000110000000	H4	1 0 0
P015	00010000000110000000	H4	1 0 0
P016	00000001100001100000	H5	1 0 1
P017	00000001100000100000	H5	1 0 1
P018	00000001100001000000	H5	1 0 1
P019	11100001000000011000	H6	1 1 0
P020	11100011000000111100	H6	1 1 0
P021	11000011000000111100	H6	1 1 0
P022	00000000000000000011	H7	1 1 1

Proses perulangan akan berulang sebanyak maksimum *epoch*, yang telah ditentukan di awal. Dalam setiap perulangan *epoch* pada proses pelatihan akan terjadi hal berikut:

1. tahap *feedforward*: update semua nilai neuron, z dan kemudian nilai y.
2. tahap *backpropagation*: hitung nilai *error* w dan v.
3. tahap *update* bobot: ubah nilai bobot garis w dan v.

Berikut merupakan proses diagnosis penyakit kulit pada anjing yang dilakukan di dalam aplikasi: Sebagai contoh, gejala yang dialami seperti terlihat pada tabel 4.

**Tabel 4.** Gejala yang dialami

Kode Gejala	Mengalami (Tidak = 0, Ya = 1)
G01	1
G02	1
G03	1
G04	0
G05	0
G06	0
G07	0
G08	1
G09	0
G10	0
G11	0
G12	0
G13	0
G14	0
G15	0
G16	1
G17	1
G18	0
G19	0
G20	0

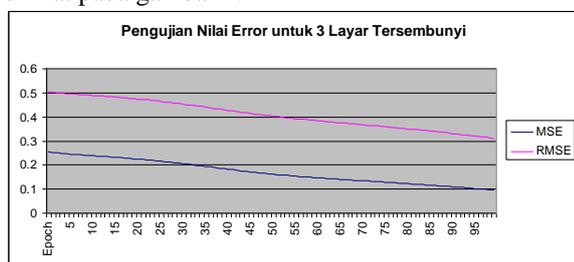
Seperti terlihat pada tabel 6, *input* ciri gejala yang dialami secara berurutan (x<sub>1</sub> x<sub>2</sub> x<sub>3</sub> ... x<sub>20</sub>) = 11100001000000011000, maka proses pengenalan pola penyakit kulit anjing dengan *Perceptron* adalah:

- *Output* bit-1 (k = 1)
  - $y_{in}(1) = w(0,1) + (z(1) * w(1,1)) + (z(2) * w(2,1)) + (z(3) * w(3,1)) + (z(4) * w(4,1)) + (z(5) * w(5,1)) + (z(6) * w(6,1)) + (z(7) * w(7,1)) + (z(8) * w(8,1)) + (z(9) * w(9,1)) + (z(10) * w(10,1))$
  - $y_{in}(1) = -0.261598 + (0.729606 * 0.572348) + (0.350762 * -0.556721) + (0.921523 * 0.069701) + (0.228771 * -2.15047) + (0.865118 * 1.013162) + (0.666601 * -2.084711) + (0.053637 * 1.159417) + (0.883979 * 2.535756) + (0.309745 * -1.965847) + (0.655039 * 2.016754)$
  - $y_{in}(1) = -0.261598 + 2.297296 = 2.035698$
  - $y(1) = f(y_{in}(1)) = 0.884494$
  - $y(1) = 0.884494 \geq 0.5 \rightarrow$  dibulatkan menjadi 1
- *Output* bit-2 (k = 2)
  - $y_{in}(2) = w(0,2) + (z(1) * w(1,2)) + (z(2) * w(2,2)) + (z(3) * w(3,2)) + (z(4) * w(4,2)) + (z(5) * w(5,2)) + (z(6) * w(6,2)) + (z(7) * w(7,2)) + (z(8) * w(8,2)) + (z(9) * w(9,2)) + (z(10) * w(10,2))$
  - $y_{in}(2) = 0.082795 + (0.729606 * 1.155243) + (0.350762 * -1.928306) + (0.921523 * -0.968859) + (0.228771 * -1.736737) + (0.865118 * 0.178018) + (0.666601 * 1.370579) + (0.053637 * -2.667362) + (0.883979 * -0.048671) + (0.309745 * 2.680747) + (0.655039 * 1.271328)$
  - $y_{in}(2) = 0.082795 + 1.421015 = 1.50381$
  - $y(2) = f(y_{in}(2)) = 0.818142$
  - $y(2) = 0.818142 \geq 0.5 \rightarrow$  dibulatkan menjadi 1
- *Output* bit-3 (k = 3)
  - $y_{in}(3) = w(0,3) + (z(1) * w(1,3)) + (z(2) * w(2,3)) + (z(3) * w(3,3)) + (z(4) * w(4,3)) + (z(5) * w(5,3)) + (z(6) * w(6,3)) + (z(7) * w(7,3)) + (z(8) * w(8,3)) + (z(9) * w(9,3)) + (z(10) * w(10,3))$
  - $y_{in}(3) = 0.315194 + (0.729606 * -0.2948) + (0.350762 * -1.028699) + (0.921523 * -1.442094) + (0.228771 * 1.075159) + (0.865118 * -1.433933) + (0.666601 * -0.652496) + (0.053637 * 3.163312) + (0.883979 * -2.173846) + (0.309745 * 1.263196) + (0.655039 * 2.216657)$
  - $y_{in}(3) = 0.315194 + -3.243048 = -2.927854$
  - $y(3) = f(y_{in}(3)) = 0.050794$
  - $y(3) = 0.050794 < 0.5 \rightarrow$  dibulatkan menjadi 0
- Hasil nilai y (*output*) = 1 1 0

Hipotesis dengan nomor urut = 6 [ 1 1 0 ] adalah H6 atau penyakit Parasit (Kutu atau Tungau). Dari hasil pengujian, *Perceptron* akan selalu mencari pola penyakit dengan ciri gejala paling mirip dengan ciri gejala dan hipotesis yang telah dipelajari sebelumnya pada pengetahuan (tabel 2).

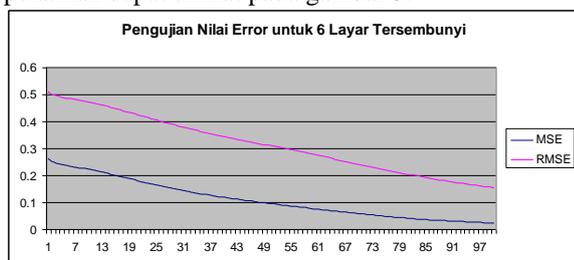
#### 4.2 Pembahasan

Pembahasan membahas hasil pengujian terhadap pengaruh jumlah layar tersembunyi, *epoch* dan nilai parameter  $\alpha$  terhadap nilai *error* dalam proses pelatihan data. Nilai *error* diukur dengan menggunakan *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)*. Pengaruh jumlah *hidden layer* sebanyak 3 lapisan terhadap penurunan nilai *error* pada proses pelatihan dapat dilihat pada gambar 4.



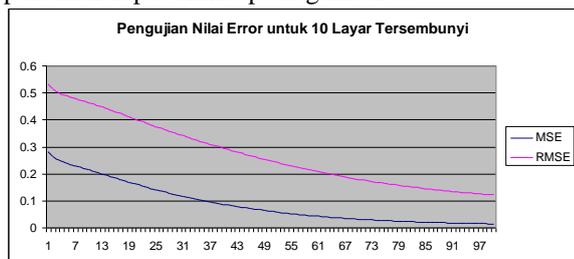
Gambar 4. Pengaruh 3 Lapisan *Hidden Layer* Terhadap Nilai *Error*

Pengaruh jumlah *hidden layer* sebanyak 6 lapisan terhadap penurunan nilai *error* pada proses pelatihan dapat dilihat pada gambar 5.



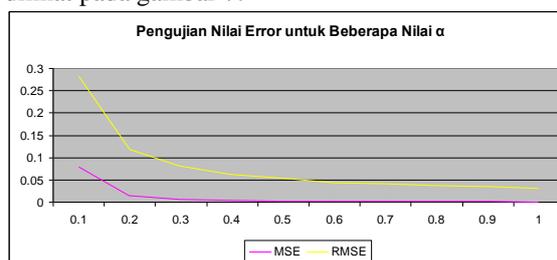
Gambar 5. Pengaruh 6 Lapisan *Hidden Layer* Terhadap Nilai *Error*

Pengaruh jumlah *hidden layer* sebanyak 10 lapisan terhadap penurunan nilai *error* pada proses pelatihan dapat dilihat pada gambar 6.



Gambar 6. Pengaruh 10 Lapisan *Hidden Layer* Terhadap Nilai *Error*

Hasil pengujian pada grafik garis pada gambar 4, gambar 5 dan gambar 6 memperlihatkan bahwa semakin tinggi *epoch* (semakin banyak perulangan), maka nilai *error* semakin menurun. Hal lain yang dapat disimpulkan adalah penurunan nilai *error* lebih cepat pada penggunaan jumlah layar tersembunyi sebanyak 10 lapisan, daripada penurunan *error* pada jumlah *hidden layer* 3 dan 6 lapisan. Pengujian dengan menggunakan 10 layar tersembunyi dan maksimum *epoch* sebesar 100, menghasilkan nilai akhir *error* yang sudah cukup rendah, yaitu  $MSE = 0.01494$  dan  $RMSE = 0.12223$  untuk melakukan diagnosa penyakit secara akurat. Hasil pengujian dengan nilai  $\alpha$  yang berbeda dapat dilihat pada gambar 7.



Gambar 7. Pengaruh Nilai  $\alpha$  Terhadap Nilai *Error*

Hasil pengujian pada grafik garis pada gambar 7 memperlihatkan bahwa semakin besar nilai  $\alpha$ , maka penurunan *error* akan semakin cepat dan proses pelatihan akan semakin cepat selesai bila menggunakan kriteria pengecekan *limit error*. Akan tetapi berdasarkan hasil pengujian, nilai  $\alpha$  yang terlalu tinggi mengakibatkan ketelitian yang rendah dan beberapa hasil pengenalan kurang akurat. Penelitian menggunakan nilai  $\alpha = 0.2$  dengan nilai  $MSE = 0.01387$  dan  $RMSE = 0.11777$  yang dianggap sudah cukup rendah untuk melakukan semua pengenalan terhadap semua data latih secara akurat.

#### 5. KESIMPULAN

Setelah menyelesaikan penelitian mengenai penerapan jaringan syaraf tiruan *Multi Layer Perceptron* dalam sistem pakar diagnosis penyakit kulit pada anjing, terdapat beberapa kesimpulan yang dapat ditarik sebagai berikut:

- Jaringan syaraf tiruan *Multi Layer Perceptron* dapat digunakan untuk melakukan proses diagnosa awal terhadap penyakit kulit pada anjing, sehingga pemilik anjing dapat mengetahui penyakit yang diderita dan melakukan tindakan pengobatan lebih dini terhadap hewan peliharaannya.
- Metode *Multi Layer Perceptron* di dalam penelitian ini digunakan untuk mempelajari 22 kasus (pengetahuan) dengan 20 gejala dan 8 penyakit kulit yang mungkin diderita anjing (hipotesis).
- Hasil diagnosa (pengenalan pola penyakit kulit anjing) adalah pola penyakit dengan ciri gejala

paling mirip dengan ciri gejala dan hipotesis yang telah dipelajari sebelumnya pada proses *training*.

- Nilai *error* akan turun seiring dengan bertambahnya *epoch*. Jumlah lapisan tersembunyi berbanding lurus dengan penurunan nilai *error*. Semakin banyak lapisan tersembunyi yang digunakan, maka semakin cepat *error* turun. Pengujian dengan menggunakan 10 layer tersembunyi dan maksimum *epoch* sebesar 100, menghasilkan nilai akhir *error* yang sudah cukup rendah, yaitu MSE = 0.01494 dan RMSE = 0.12223 untuk melakukan diagnosa penyakit secara akurat.
- Semakin besar nilai  $\alpha$ , maka penurunan *error* akan semakin cepat dan proses pelatihan akan semakin cepat selesai bila menggunakan kriteria pengecekan *limit error*. Akan tetapi berdasarkan hasil pengujian, nilai  $\alpha$  yang terlalu tinggi mengakibatkan ketelitian yang rendah dan beberapa hasil pengenalan kurang akurat. Penelitian menggunakan nilai  $\alpha = 0.2$  dengan nilai MSE = 0.01387 dan RMSE = 0.11777 yang dianggap sudah cukup rendah untuk melakukan semua pengenalan terhadap semua data latih secara akurat.

#### DAFTAR PUSTAKA

- [1] Nurlayli, R.K., dan Hidayati, D.S. 2014, *Kesepian Pemilik Hewan Peliharaan yang Tinggal Terpisah dari Keluarga*, Universitas Muhammadiyah, Malang.
- [2] Mirwa, T. 2016, *Kesetiaan: Anjing Sebagai Subjek dalam Penciptaan Karya Seni Patung*, ISI, Yogyakarta.
- [3] Firscha, P. 2018. *Inilah 6 Penjelasan Ilmiah Mengenai Sifat Setia Anjing kepada Manusia*, IDNTimes, Jakarta.
- [4] Zhang, W. 2012. *Jenis-Jenis Sakit Kulit Pada Anjing*. Anjingkita.com.
- [5] Anbazhagan, R. dan Ponmuthuramalingam P. 2016, *Neural Networks Based Image Retrieval System Using Rosenblatt's Perceptron Algorithm*, GAC, India.
- [6] Ravi, D. et.al, 2016. *Deep Learning for Health Informatics*, University College, London.
- [7] Banda, et.al. 2013, *Online Learning in a Chemical Perceptron*, University of New Mexico, Mexico.
- [8] Ramchoun, H., et.al. 2016, *Multilayer Perceptron: Architecture Optimization and Training*, *International Journal of Interactive Multimedia and Artificial Intelligence*, Morocco.
- [9] Rani, K. H. 2017, *Advancements in Multi-Layer Perceptron Training to Improve Classification Accuracy*, *International Journal on Recent and Innovation Trends in Computing and Communication*, India.
- [10] Mia, et.al. 2015, *An Algorithm For Training Multilayer Perceptron (MLP) For Image Reconstruction Using Neural Network Without Overfitting*, University of Information Technology & Science, Bangladesh.
- [11] Sudha, G.M. 2014, *Software Defect Prediction System using Multilayer Perceptron Neural Network with Data Mining*, Al-Ameen Engineering College, India.
- [12] Baskaran, K. and Balaji, S.A. 2013, *Design and Development of Artificial Neural Networking (ANN) System Using Sigmoid Activation Function to Predict Annual Rice Production in Tamilnadu*, Karpagam University, India.
- [13] Mishra, et.al. 2018, *Comprehensive and Comparative Analysis of Neural Network*, Banasthali Vidyapith, India.
- [14] Witari, R.W. dan Gandhiadi, I.G. 2013, *Sistem Pakar Pendiagnosa Penyakit Menular pada Anjing*, Universitas Udayana, Bali.
- [15] Himawan, dkk. 2018, *Sistem Diagnosis Penyakit Hewan Pada Anjing Dengan Menggunakan Metode Naive Bayes*, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Malang.
- [16] Ramadhoni, et.al. 2019, *Sistem Pakar*

Diagnosa Penyakit Kulit Pada Anjing  
Menggunakan Metode Bayes, SAINTEKS,  
Padang.

- [17] Guntoro, et.al. 2019, Prediksi Jumlah  
Kendaraan di Provinsi Riau Menggunakan  
Metode *Backpropagation*, Jurnal Ilmiah  
Ilmu Komputer, Pekanbaru.