

ANALISIS VARIASI PARAMETER *BACKPROPAGATION* *ARTIFICIAL NEURAL NETWORK* TERHADAP PENGENALAN POLA DATA IRIS

Ikhwannul Kholis, ST. MT.

Universitas 17 Agustus 1945 Jakarta
ikkholis27@gmail.com

Abstrak

Pengenalan pola data iris dapat dilakukan dengan menggunakan metode *Backpropagation Artificial Neural Network* (ANN). ANN dibuat menyerupai sistem syaraf manusia, disebut juga Jaringan Syaraf Tiruan (JST). Dengan beberapa parameter pada *Backpropagation*, dapat diketahui karakteristik *Backpropagation* sehingga dapat memperkecil *error* dan *epoch*, serta memperbesar *Recognition Rate*. Hasil percobaan menunjukkan hubungan antara parameter *alpha*, koefisien momentum, dan jumlah *neuron hidden layer* terhadap *error*, *epoch*, dan *Recognition Rate* yang diperoleh.

Kata Kunci: ANN, *backpropagation*, *epoch*, *error*, JST, *recognition rate*.

Abstract

Pattern recognition of Iris data can be done by using Backpropagation Artificial Neural Network (ANN). ANN is made to resemble the human neural system. By varying some parameters on backpropagation, backpropagation characteristics is known to minimize errors and epoch and enlarge Recognition Rate. The experimental results show the relationship between the parameters alpha, coefficient of momentum, and the number of hidden layer neurons against error, epoch, and the recognition rate obtained.

Keywords: ANN, *Backpropagation*, *epoch*, *error*, JST, *Recognition Rate*.

Tanggal Terima Naskah :
Tanggal Persetujuan Naskah : 13 Februari 2015

1 PENDAHULUAN

Salah satu cabang dari *Artificial Intelligence* adalah apa yang dikenal dengan Jaringan Saraf Tiruan (*Artificial Neural Network*). Jaringan syaraf tiruan telah dikembangkan sejak tahun 1940. Pada tahun 1943, Mc. Culloch dan W.H. Pitts memperkenalkan pemodelan matematis *neuron*. Tahun 1949, Hebb mencoba mengkaji proses belajar yang dilakukan oleh *neuron*. Teori ini dikenal sebagai *Hebbian Law*. Tahun 1958, Rosenblatt memperkenalkan konsep *perceptron* suatu jaringan, yang terdiri dari beberapa lapisan yang saling berhubungan melalui umpan maju (*feed forward*). Konsep ini dimaksudkan untuk memberikan ilustrasi tentang dasar-dasar kecerdasan

secara umum. Hasil kerja Rosenblatt yang sangat penting adalah *perceptron convergence theorem* (tahun 1962), yang membuktikan bahwa bila setiap *perceptron* dapat memilah-milah dua buah pola yang berbeda, maka siklus pelatihannya dapat dilakukan dalam jumlah yang terbatas.

Fisher telah melakukan penelitian terhadap spesies bunga Iris. Iris *dataset* diperkenalkan sebagai contoh untuk analisis diskriminan. Data melaporkan empat karakteristik, yaitu *sepal width*, *sepal length*, *petal width*, dan *petal length* dari tiga spesies bunga Iris. Spesies bunga Iris tersebut adalah *Iris Setosa*, *Iris Versicolor*, dan *Iris Virginica* [1].

Pengenalan pola bunga Iris dapat dilakukan dengan menggunakan ANN *Backpropagation*. *Dataset* dari bunga Iris dapat diklasifikasikan terhadap setiap spesies bunga Iris tersebut. Dengan demikian, klasifikasi spesies, dalam hal ini bunga Iris, dapat dilakukan secara lebih mudah dengan menggunakan metode ANN *Backpropagation*.

2 KONSEP DASAR

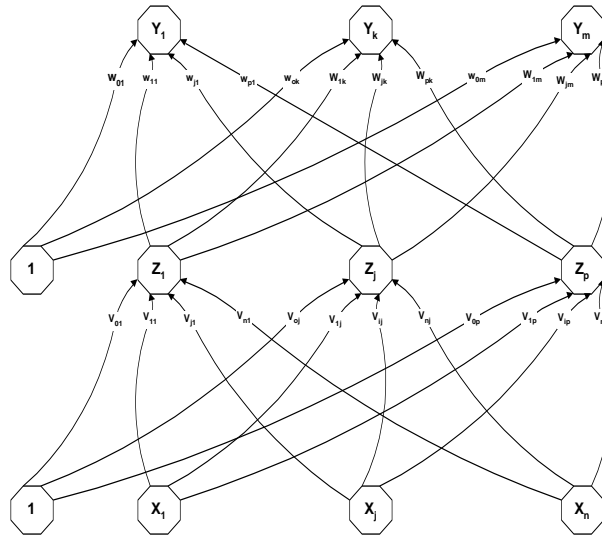
2.1 Pengertian *Backpropagation*

Jaringan Syaraf Tiruan (JST) merupakan salah satu sistem pemrosesan informasi atau data yang didesain dengan meniru cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Salah satu metode yang digunakan dalam JST adalah *Backpropagation*. *Backpropagation* adalah algoritma pembelajaran untuk memperkecil tingkat *error* dengan cara menyesuaikan bobotnya berdasarkan perbedaan *output* dan target yang diinginkan. *Backpropagation* juga merupakan sebuah metode sistematis untuk pelatihan *multilayer JST*. *Backpropagation* memiliki tiga *layer* dalam proses pelatihannya, yaitu *input layer*, *hidden layer*, dan *output layer*, dimana *backpropagation* ini merupakan perkembangan dari *single layer network* (Jaringan Lapisan Tunggal) yang memiliki dua *layer*, yaitu *input layer* dan *output layer* [2].

Dengan adanya *hidden layer* pada *backpropagation* dapat menyebabkan tingkat *error* pada *backpropagation* lebih kecil dibanding tingkat *error* pada *single layer network*. Hal ini dikarenakan *hidden layer* pada *backpropagation* berfungsi sebagai tempat untuk meng-*update* dan menyesuaikan bobot, sehingga didapatkan nilai bobot yang baru yang bisa diarahkan mendekati dengan target *output* yang diinginkan.

2.2 Arsitektur *Backpropagation*

Arsitektur algoritma *backpropagation* terdiri dari tiga *layer*, yaitu *input layer*, *hidden layer*, dan *output layer*. Pada *input layer* tidak terjadi proses komputasi, hanya terjadi pengiriman sinyal *input* ke *hidden layer* [3]. Pada *hidden* dan *output layer* terjadi proses komputasi terhadap bobot dan bias, serta dihitung pula besarnya *output* dari *hidden* dan *output layer* tersebut berdasarkan fungsi aktivasi. Dalam algoritma *backpropagation* ini digunakan fungsi aktivasi *sigmoid biner*, karena *output* yang diharapkan bernilai antara 0 sampai 1.



Gambar 1. Arsitektur ANN *backpropagation*

dengan:

- V_{ij} = Bobot pada lapisan tersembunyi (*hidden layer*)
- V_{oj} = Bias pada lapisan tersembunyi (*hidden layer*)
- W_{ij} = Bobot pada lapisan keluaran (*output layer*)
- W_{oj} = Bias pada lapisan keluaran (*output layer*)
- X = Lapisan masukan (*Input Layer*)
- Y = Lapisan keluaran (*Output Layer*)
- Z = Lapisan tersembunyi (*Hidden Layer*)

2.2 Algoritma *Backpropagation*

2.2.1 Inisialisasi Bobot

Terdapat dua cara untuk menginisialisasi bobot, yaitu inisialisasi secara *random* dan inisialisasi dengan menggunakan *Nguyen-Widrow*. Inisialisasi acak merupakan cara yang paling sering digunakan dalam inisialisasi bobot. Pada inisialisasi bobot secara *random*, bobot diinisialisasi secara acak tanpa menggunakan faktor skala [4]. Pada inisialisasi *Nguyen-Widrow*, inisialisasi dilakukan dengan memodifikasi inisialisasi acak dengan menggunakan faktor skala β dengan tujuan untuk mempercepat proses pelatihan. Algoritma inisialisasi dengan *Nguyen-Widrow* adalah sebagai berikut:

- a. Menentukan besarnya skala β

$$\beta = 0.7(p)^{1/n} \dots\dots\dots(1)$$

dengan p: jumlah unit *hidden* dan n: jumlah unit *input*.

- b. Inisialisasi bobot V_{ij} secara *random* dengan nilai inisialisasi V_{ij} adalah

$$-0.5 \leq V_{ij} \leq 0.5 \dots\dots\dots(2)$$

- c. Menghitung besarnya *magnitude* bobot V_{ij}

$$\|V_{ij}\| = \sqrt{\left\{ \sum_{i=1}^p (V_{ij})^2 \right\}} \dots\dots\dots(3)$$

d. Meng-*update* bobot V_{ij}

$$V_{ij} = \frac{\beta \cdot V_{ij}}{\|V_j\|} \dots\dots\dots(4)$$

e. Mengatur nilai bias V_{oj} sebesar

$$-\beta \leq V_{oj} \leq \beta \dots\dots\dots(5)$$

2.2.2 Proses Feed Forward dan Backpropagation

Pada dasarnya proses algoritma *backpropagation* terdiri dari komputasi maju (*feed forward*) dan komputasi balik (*backpropagation*). Algoritma proses *feed forward* adalah sebagai berikut:

a. Unit *input* ($X_i, i=1,2,\dots,n$)

1. Menerima *input* X_i
2. Mengirimkannya ke semua unit *layer* di atasnya (*Hidden layer*).

b. Unit *Hidden* ($Z_j, j=1,2,\dots,n$)

1. Menghitung semua sinyal *input* dengan bobotnya:

$$z_in_j = V_{oj} + \sum X_i V_{ij} \dots\dots\dots(6)$$

2. Menghitung nilai aktivasi setiap unit *hidden* sebagai *output* unit *hidden*

$$Z_j = f(z_in_j) = \frac{1}{1 + e^{-z_in_j}} \dots\dots\dots(7)$$

3. Mengirim nilai aktivasi ke unit *output*.

c. Unit *Output* ($Y_k, k=1,2,\dots,n$)

1. Menghitung semua sinyal *input*-nya dengan bobotnya:

$$y_in_k = W_{ok} + \sum Z_j W_{jk} \dots\dots\dots(8)$$

2. Menghitung nilai aktivasi setiap unit *output* sebagai *output* jaringan.

$$Y_k = f(y_in_k) = \frac{1}{1 + e^{-y_in_k}} \dots\dots\dots(9)$$

Algoritma proses *backpropagation*-nya adalah sebagai berikut :

a. Unit *Output* ($Y_k, k=1,2,\dots,m$)

1. Menerima pola target yang bersesuaian dengan pola *input*
2. Menghitung informasi *error*:

$$\delta_k = (T_k - Y_k) f'(y_in_k) \dots\dots\dots(10)$$

3. Menghitung besarnya koreksi bobot unit *output*:

$$\Delta W_{jk} = \alpha \frac{\partial E(W_{jk})}{\partial W_{jk}} = \alpha \delta_k z_j \dots\dots\dots(11)$$

4. Menghitung besarnya koreksi bias *output*:

$$\Delta W_{ok} = \alpha \delta_k \dots\dots\dots(12)$$

5. Mengirimkan δ_k ke unit-unit yang ada pada *layer* di bawahnya, yaitu ke *hidden layer*.

b. Unit *Hidden* ($Z_j, j=1,2,\dots,p$)

1. Menghitung semua koreksi *error*:

$$\delta_{in_j} = \sum \delta_k W_{jk} \dots\dots\dots(13)$$

2. Menghitung nilai aktivasi koreksi *error*:

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \dots\dots\dots(14)$$

3. Menghitung koreksi bobot unit *hidden*:

$$\Delta V_{ij} = \alpha \frac{\partial E(V_{ij})}{\partial V_{ij}} = \alpha \delta_j X_i \dots\dots\dots(15)$$

4. Menghitung koreksi *error* bias unit *hidden*:

$$\Delta V_{0j} = \alpha \delta_j \dots\dots\dots(16)$$

c. *Update* bobot dan bias

1. Unit *Output* ($Y_k, k = 1,2,\dots,m$)

Meng-*update* bobot dan biasnya ($j = 0,\dots,p$):

$$\begin{aligned} W_{jk} &= W_{jk} + \Delta W_{jk} \dots\dots\dots(17) \\ W_{0k} &= W_{0k} + \Delta W_{0k} \end{aligned}$$

2. Unit *hidden* ($Z_j, j = 1,\dots,p$)

Meng-*update* bobot dan biasnya ($i = 0,\dots,n$):

$$\begin{aligned} V_{ij} &= V_{ij} + \Delta V_{ij} \dots\dots\dots(18) \\ V_{0j} &= V_{0j} + \Delta V_{0j} \end{aligned}$$

2.2.3 Stopping Condition

Terdapat dua kondisi *stopping* pada algoritma *backpropagation* ini, yaitu:

a. *Error* < *Error* maksimum

Error adalah perbedaan yang terjadi antara *output* terhadap target yang diinginkan. Proses ANN akan berhenti jika besarnya *error* yang terjadi telah bernilai lebih kecil dari nilai *error* maksimum yang telah ditetapkan. Besarnya nilai *error* dihitung dengan menggunakan fungsi *error* kuadratis.

$$E = 0.5 \sum_{k=0}^k (T_k - Y_k)^2 \dots\dots\dots(19)$$

b. *Epoch* > *Epoch* maksimum

Epoch adalah suatu langkah yang dilakukan dalam pembelajaran pada ANN. Jika besarnya *epoch* lebih besar dari besarnya *epoch* maksimum yang telah ditetapkan, maka proses pembelajaran akan berhenti.

Kedua kondisi *stopping* di atas digunakan dengan logika OR. Jadi kondisi *stopping* terjadi jika besarnya *Error* < *Error* maksimum atau *Epoch* > *Epoch* maksimum.

2.3 Faktor-Faktor Dalam Pembelajaran Backpropagation

Beberapa faktor yang mempengaruhi keberhasilan algoritma backpropagation, antara lain:

1. Inisialisasi bobot
Bobot awal menentukan apakah jaringan akan mencapai *global minima* atau *local minima* kesalahan, dan seberapa cepat jaringan akan konvergen.
2. Laju pembelajaran (*alpha*)
Laju pembelajaran merupakan parameter jaringan dalam mengendalikan proses penyesuaian bobot. Nilai laju pembelajaran yang optimal bergantung pada kasus yang dihadapi. Laju pembelajaran yang terlalu kecil menyebabkan konvergensi jaringan menjadi lebih lambat, sedangkan laju pembelajaran yang terlalu besar dapat menyebabkan ketidakstabilan pada jaringan.
3. Momentum (*miu*)
Momentum digunakan untuk mempercepat pelatihan jaringan. Metode momentum melibatkan penyesuaian bobot ditambah dengan faktor tertentu dari penyesuaian sebelumnya. Penyesuaian ini dinyatakan sebagai berikut:

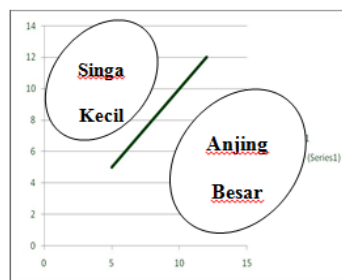
$$\begin{aligned} \Delta w_{jk}(t+1) &= \alpha \delta_k z_j + \mu \Delta w_{jk}(t) \\ \Delta w_{0k}(t+1) &= \alpha \delta_k z_j + \mu \Delta w_{0k}(t) \\ \Delta v_{ij}(t+1) &= \alpha \delta_j X_i + \mu \Delta v_{ij}(t) \\ \Delta v_{0j}(t+1) &= \alpha \delta_j X_i + \mu \Delta v_{0j}(t) \end{aligned} \dots\dots\dots(20)$$

Dengan menggunakan persamaan 17, 18, dan 20, *update* bobot dengan momentum dirumuskan sebagai berikut:

$$\begin{aligned} W_{jk}(t+1) &= W_{jk}(t) + \alpha \delta_k z_j + \mu \Delta W_{jk}(t-1) \\ W_{0k}(t+1) &= W_{0k}(t) + \alpha \delta_k z_j + \mu \Delta W_{0k}(t-1) \\ V_{ij}(t+1) &= V_{ij}(t) + \alpha \delta_j X_i + \mu \Delta V_{ij}(t-1) \\ V_{0j}(t+1) &= V_{0j}(t) + \alpha \delta_j X_i + \mu \Delta V_{0j}(t-1) \end{aligned} \quad (21)$$

3 PEMBAHASAN

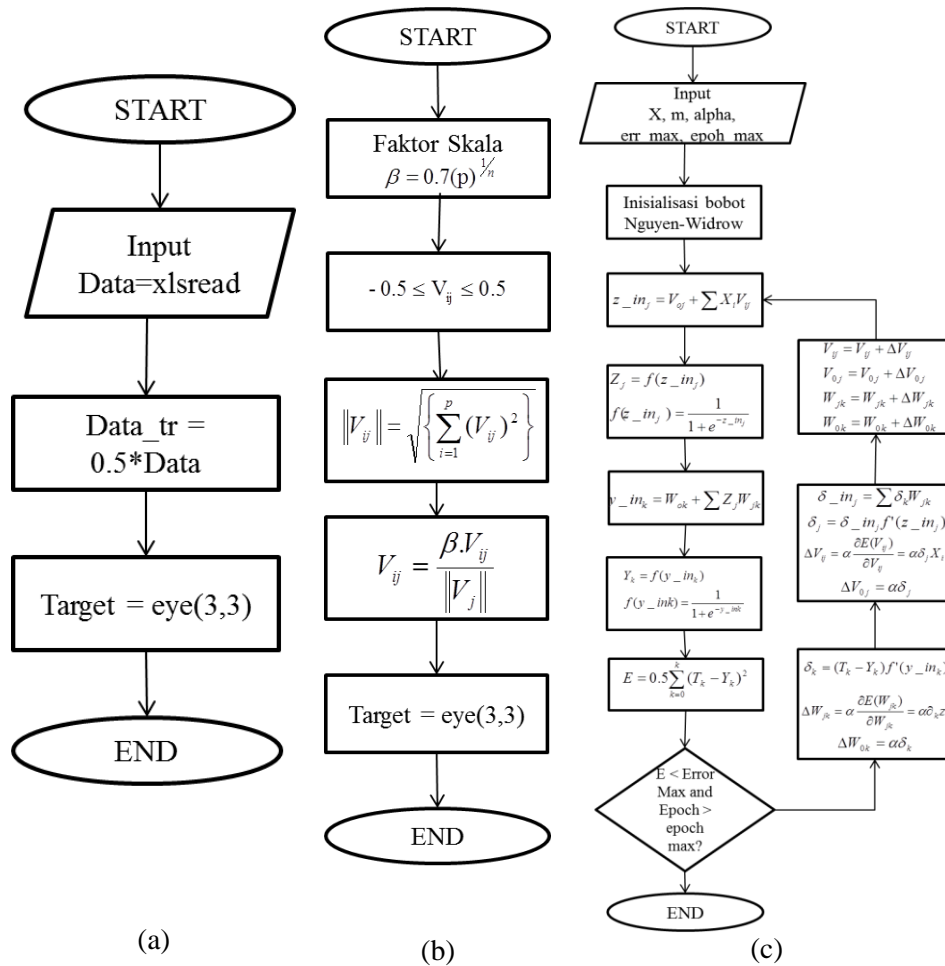
Jaringan syaraf tiruan digunakan untuk memodelkan suatu syaraf biologi pada otak manusia. Sebagai contoh, diberikan grafik seperti pada Gambar 2. JST ini digunakan untuk membuat sebuah perbedaan yang ditandai dengan sebuah *trendline* pada Gambar 2. Dengan demikian, anjing besar dan singa kecil berdasarkan ciri-ciri yang diberikan dapat dikenali.



Gambar 2. Metode ANN untuk pengenalan pola antara Singa Kecil dan Anjing Besar

Penelitian ini menggunakan data IRIS (data bunga Iris) dengan 150 data dan tiga pembagian kelas. *Backpropagation* digunakan untuk mengolah data bunga dari tiga kelas tersebut sehingga spesies bunga dapat dikenali. Pada data tersebut, terdapat 150 data yang dibagi menjadi tiga kelas, yaitu kelas 1, kelas 2, dan kelas 3, dengan target yang berbeda untuk masing-masing kelas. Data tersebut berbentuk matriks baris dengan bentuk 150 x 5 dengan kolom 1 – 4 sebagai dimensi 1 - 4 dan kolom 5 merupakan kolom yang menunjukkan kelas data tersebut. Target untuk masing-masing kelas tersebut bernilai satu. Data *training* yang digunakan adalah 70% dari data IRIS. Percobaan ini memiliki variabel terikat (yang akan diamati), yaitu nilai presentase tingkat *Recognition Rate*, nilai *error*, dan *epoch*. Variabel bebas yang diubah adalah *alpha*, koefisien momentum, dan jumlah *hidden layer*. Nilai awal untuk masing-masing parameter adalah sebagai berikut.

$\alpha = 0.2$; *hidden neuron* = 2; koef. Momentum=0.2; *epoch max*= 10000; *error min*= 0.01; metode inisialisasi = *nguyen widrow*.



Gambar 3. (a) *Flowchart Input Data*, (b) *Flowchart Nguyen Widrow*, (c) *Flowchart ANN Backpropagation*

Pada penelitian ini, diberikan *flowchart* untuk mengambil data dari *file excel* menjadi matriks di Matlab (Gambar 3.a). Untuk inisialisasi *nguyen widrow*, diberikan *flowchart* seperti Gambar 3.b. ANN *Backpropagation* diberikan *flowchart* seperti Gambar 3.c.

Dengan *flowchart* yang telah diberikan, dibuat program dalam *software Matlab* dalam bentuk *mfiles*. Keluaran dari program tersebut adalah data dan grafik.

4 HASIL PENELITIAN

Penelitian yang telah dilakukan meliputi variasi *alpha*, *momentum*, dan *hidden layer*. Hasil penelitian diberikan sebagai berikut.

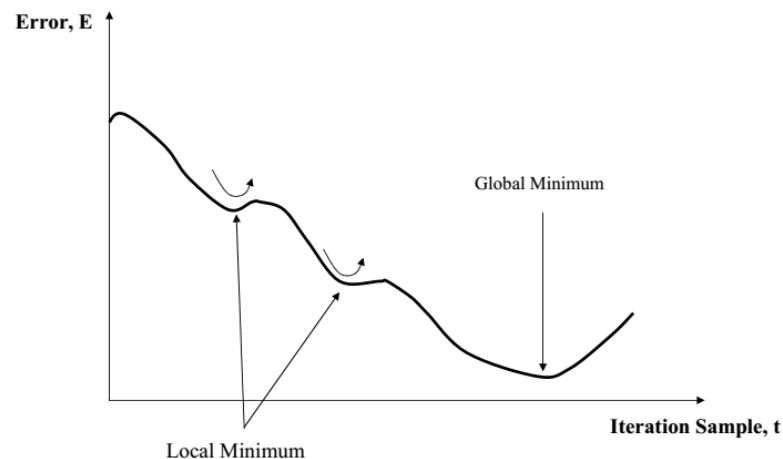
4.1 Pengaruh Perubahan *Alpha* (Laju Pembelajaran)

Pada percobaan ini dilakukan variasi *alpha* dari 0,1 sampai dengan 0,9 dengan penambahan 0,1 dan hal ini dilakukan sebanyak 10 kali percobaan. Nilai parameter-parameter lainnya diberikan nilai awal. Hasil percobaannya adalah sebagai berikut.

Table 1. Percobaan dengan variasi *alpha*

alpha	perc ke-1	perc ke-2	perc ke-3	perc ke-4	perc ke-5	perc ke-6	perc ke-7	perc ke-8	perc ke-9	perc ke-10	rata RR	rata waktu (sec)	rata epoch
0.1	98	96	96	97	97	97	97	99	96	99	97.2	11.89978988	213.6
0.2	97	97	98	98	97	98	100	98	97	98	97.8	5.985935313	108
0.3	96	96	96	98	97	96	99	97	97	98	97	3.485574809	61.4
0.4	99	96	96	99	96	97	96	98	98	96	97.1	2.652323335	46.7
0.5	99	99	97	98	97	98	98	99	96	97	97.8	1.877426958	32.3
0.6	99	96	99	98	98	97	98	98	98	96	97.7	1.49382942	26.7
0.7	96	97	98	100	97	99	96	99	99	98	97.9	1.010620758	18.1
0.8	100	100	98	98	98	94	99	97	98	88	97	0.852669048	15.1
0.9	97	100	99	98	96	98	98	98	99	96	97.9	0.890626544	15.9

Dari data di atas terlihat bahwa *alpha* 0,7 dan 0,9 memiliki *rate recognition* yang sama. Dengan nilai *alpha* 0,9 yang terlalu besar, dikhawatirkan akan membuat sistem divergen dan membuat *error* mencapai *local minima*. Gambar 4 mendeskripsikan *local minima* dan *global minima*. Dengan demikian, *alpha* yang memiliki *Recognition Rate* maksimal adalah 0.7.



Gambar 4. Grafik *Error* terhadap iterasi

4.2 Pengaruh Perubahan Miu (Momentum)

Pada percobaan ini dilakukan variasi miu dari 0,1 sampai dengan 0,9 dengan penambahan 0,1. Hal ini dilakukan sebanyak 10 kali percobaan. Nilai parameter-parameter lainnya diberikan nilai awal. Hasil percobaannya adalah sebagai berikut.

Table 2. Percobaan dengan variasi koefisien momentum

miu	perc ke-1	perc ke-2	perc ke-3	perc ke-4	perc ke-5	perc ke-6	perc ke-7	perc ke-8	perc ke-9	perc ke-10	rata RR	rata waktu (sec)	rata epoch
0.1	95.3333	94.6667	95.3333	94	96	94	96	96	95.3333	95.3333	95.2	12.94158509	845.5
0.2	96	94.6667	95.3333	96	95.3333	94.6667	96	94.6667	96	96.6667	95.5333	18.48745035	1217.5
0.3	96.6667	96	96	96	94.6667	96	96.6667	95.3333	96	96	95.9333	18.87520661	1208.5
0.4	96	96	96	96	96.6667	96.6667	96.6667	96	96.6667	96	96.2667	18.90556317	1212.1
0.5	96	96	96.6667	96.6667	96.6667	94.6667	96.6667	95.3333	96.6667	96.6667	96.2	16.90089268	1084.7
0.6	96.6667	96.6667	96	96	96	96	96.6667	96.6667	96.6667	96.6667	96.4	23.76551137	1534.9
0.7	96.6667	96.6667	96.6667	96.6667	96.6667	96.6667	96.6667	96.6667	96.6667	96.6667	96.6667	19.52245066	1259.4
0.8	96.6667	96.6667	96.6667	96.6667	96	94.6667	96.6667	94	96.6667	96.6667	96.1333	29.42129859	1794.5
0.9	96.6667	96.6667	96.6667	94.6667	96.6667	96.6667	96.6667	96.6667	94.6667	96.6667	96.2667	23.21013791	1516.7

Dari data di atas terlihat bahwa miu 0,7 memiliki *Recognition Rate* yang paling besar walaupun waktu dan *epoch*-nya bukan yang terkecil. Pada nilai miu 0,7, sistem terlihat menghasilkan *error* yang lebih stabil dibandingkan yang lain. Oleh karena itu, digunakan konstanta momentum sebesar 0,7.

4.3 Pengaruh Perubahan Jumlah Unit *Hidden Layer*

Pada percobaan ini dilakukan variasi jumlah unit *hidden layer* dan hal ini dilakukan sebanyak 10 kali percobaan. Nilai parameter-parameter lainnya diberikan nilai awal. Hasil percobaannya adalah sebagai berikut.

Table 3. Percobaan dengan variasi jumlah unit *hidden layer*

hidden	perc ke-1	perc ke-2	perc ke-3	perc ke-4	perc ke-5	perc ke-6	perc ke-7	perc ke-8	perc ke-9	perc ke-10	rata RR	rata waktu (sec)	rata epoch
1	64.6667	64.6667	63.3333	64.6667	64.6667	63.3333	63.3333	64.6667	63.3333	64.6667	64.1333	49.92413943	4000
2	96.6667	96.6667	96.6667	96.6667	70.6667	96.6667	96.6667	96.6667	96.6667	96.6667	94.0667	13.98248318	1058.9
3	96	70	96.6667	96	96	96	96	96	96.6667	96.6667	93.6	15.53132918	1077.3
4	96	96.6667	96.6667	96	76	96	72	78	96.6667	96.6667	90.0667	12.79636358	824.7

5 KESIMPULAN

Dari percobaan yang telah dilakukan, diperoleh kesimpulan sebagai berikut:

1. Semakin besar *alpha*, semakin cepat proses dilakukan. Selain itu, sistem memiliki *epoch* yang semakin kecil, dan *Recognition Rate* yang semakin besar. Dari percobaan variasi *alpha*, *alpha* dengan *Recognition Rate* yang terbaik adalah *alpha* 0,7 dan 0,9. Namun, dipilih nilai *alpha* 0,7 karena *alpha* tidak boleh bernilai terlalu besar untuk menghindari *error* yang jatuh di *local minima*.
2. Semakin besar koefisien momentum, semakin besar *Recognition Rate*. Pada percobaan variasi koefisien momentum, diperoleh koefisien momentum 0,7 dengan *Recognition Rate* yang terbaik, yaitu 96,6667.

3. Semakin banyak jumlah *neuron* pada *hidden layer*, semakin kecil *epoch* dan *error*. Dengan demikian, semakin besar *Recognition Rate* yang diperoleh. Pada percobaan variasi jumlah unit *hidden layer*, diperoleh jumlah *hidden layer* 2 sebagai jumlah *hidden layer* yang memiliki *Recognition Rate* yang terbaik, yaitu 94,0667.
4. Pengenalan pola dapat dilakukan dengan menggunakan metode ANN *Backpropagation*.

6 REFERENSI

- [1] Fisher, R. A. "The Use of Multiple Measurements in Taxonomic Problems." 1936: 466-475.
- [2] Kusumadewi, Sri. "Analisis Jaringan Syaraf Tiruan dengan Metode Backpropagation untuk Mendeteksi Gangguan Psikologi." *Media Informatika*, 2004: 1-14.
- [3] Kusumoputro, Benyamin. *Jaringan Neural Buatan, Ed. 1*. Jakarta: Universitas Indonesia, 2001.
- [4] Marzuki. "Multilayer Neural Network and the Backpropagation Algorithm." *Lecture Material*. Kuala Lumpur: UTM, n.d.