

## **BAB III METODOLOGI PENELITIAN**

### **3.1 Objek dan Subjek Penelitian**

Objek dalam penelitian ini adalah permasalahan yang akan diteliti, yaitu penerapan algoritma *particle swarm optimization* dalam mengatasi kelemahan algoritma jaringan saraf tiruan *backpropagation* pada penentuan bobot. Subjek dalam penelitian ini yaitu data jumlah pasien penyakit ginjal kronis di RSUD Banyumas.

### **3.2 Alat dan Bahan Penelitian**

Penulis dalam melakukan penelitian prediksi terhadap pasien penyakit ginjal kronis memerlukan alat dan bahan untuk mencapai tujuan dari penelitian.

#### **3.2.1 Alat**

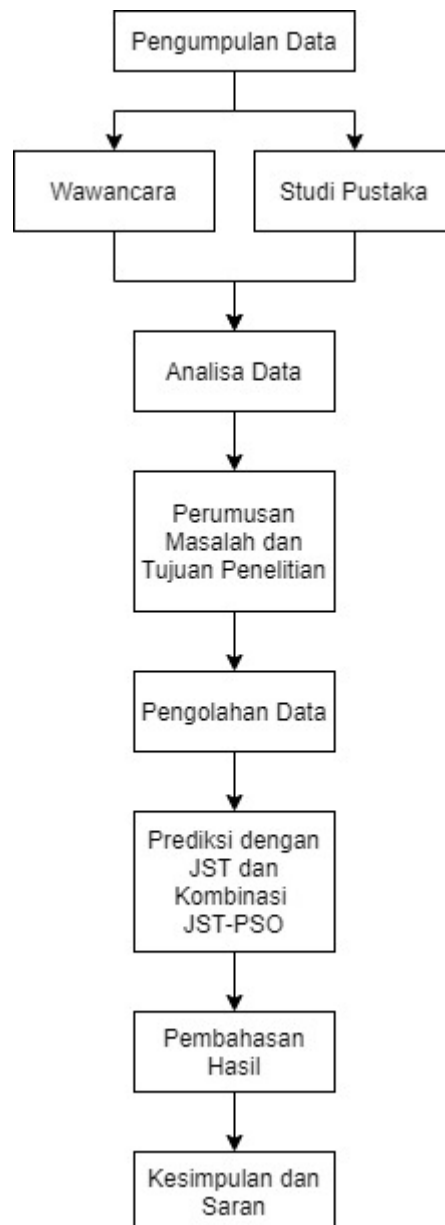
Alat yang digunakan dalam penelitian ini yaitu terdiri dari perangkat keras dan perangkat lunak. Perangkat keras yang digunakan dalam penelitian yaitu laptop 64-bit dengan RAM 8,00 GB. Kemudian, perangkat lunak yang digunakan dalam penelitian ini adalah Sistem Operasi Windows 10 dan Matlab.

#### **3.2.2 Bahan**

Bahan dalam penelitian ini yaitu data pasien penyakit ginjal kronis di RSUD Banyumas. Data pasien penyakit ginjal kronis di RSUD Banyumas yang digunakan yaitu data jumlah pasien penyakit ginjal kronis dari tahun 2011 sampai tahun 2020.

### 3.3 Diagram Alir Penelitian

Berikut adalah *flowchart* penelitian yang berisi tahapan-tahapan penelitian yang dilakukan penulis:



**Gambar 3.1 Tahapan-Tahapan Penelitian**

### 3.3.1 Pengumpulan Data

Pengumpulan data dilakukan dalam penelitian ini untuk mengetahui informasi mengenai pasien penderita penyakit ginjal kronis di RSUD Banyumas. Informasi-informasi tersebut digunakan sebagai data-data yang akan diproses pada tahap selanjutnya. Pengumpulan data dalam penelitian ini, dilakukan melalui salah satu teknik pengumpulan data yaitu wawancara dan salah satu proses dalam pengumpulan data yaitu studi pustaka, dimana keduanya merupakan tahapan selanjutnya dalam penelitian ini.

### 3.3.2 Wawancara

Wawancara dalam penelitian ini terhadap *staff* rumah sakit yang berperan langsung dalam mengatur data pasien penyakit ginjal kronis di RSUD Banyumas. Wawancara dilakukan melalui *email* sehingga data yang terekam berupa teks atau tulisan. Tujuan dari kegiatan wawancara ini yaitu untuk menghimpun sekumpulan informasi mengenai data pasien penderita penyakit ginjal kronis beserta dengan parameter yang terlibat didalamnya. Kemudian, penulis juga melakukan kegiatan studi pustaka dengan menggunakan jurnal dan buku yang berkaitan dengan penelitian ini. Data yang didapatkan dari proses pengumpulan data yaitu data pasien penyakit ginjal kronis di RSUD Banyumas pada tahun 2011 sampai 2020 seperti pada tabel 3.1.

**Tabel 3.1 Data Penyakit Ginjal Kronis Tahun 2011-2020**

Tahun	Jumlah Pasien											
	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
2011	8	17	11	12	32	19	10	20	9	45	12	33
2012	15	24	21	20	18	60	30	18	12	21	9	18
2013	35	25	15	9	50	22	25	73	8	25	55	29
2014	10	7	20	40	14	13	17	27	24	32	23	35
2015	12	31	19	29	28	22	20	19	195	20	15	16
2016	13	10	27	17	13	17	18	5	8	16	10	6
2017	10	5	6	8	5	7	7	10	13	9	4	8
2018	17	6	15	3	25	4	17	7	10	18	10	11
2019	13	18	27	19	9	6	13	5	21	10	18	32
2020	11	17	84	19	5	8	36	12	26	7	18	42

Keterangan: X1 (Januari), X2 (Februari), X3 (Maret), X4 (April), X5 (Mei). X6 (Juni), X7 (Juli), X8 (Agustus), X9 (September), X10 (Oktober), X11 (November), X12 (Desember).

### **3.3.3 Analisa Data**

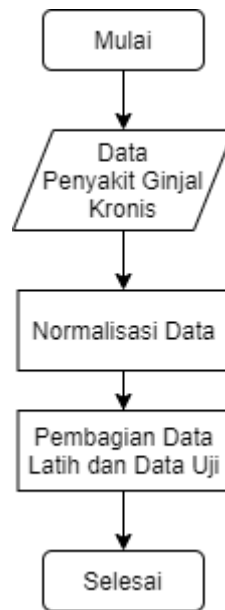
Analisa data pada penelitian ini dilakukan dengan pemahaman dan penelitian lebih dalam mengenai data-data yang telah terhimpun dari proses observasi dan wawancara yang didukung oleh kegiatan studi pustaka melalui jurnal maupun buku. Pemahaman dan penelitian lebih dalam tersebut dilakukan untuk dapat merumuskan masalah yang akan diteliti lebih lanjut mengenai penelitian ini serta dapat menjelaskan tujuan dari penelitian ini.

### **3.3.4 Perumusan Masalah dan Tujuan Penelitian**

Tahapan setelah penulis melakukan analisa terhadap data yang telah terhimpun yaitu perumusan masalah serta tujuan penelitian yang ingin dicapai dalam penelitian ini. Perumusan masalah dan tujuan penelitian yang ingin dicapai didasarkan pada hasil analisa data pada tahap sebelumnya.

### **3.3.5 Pengolahan Data**

Pengolahan data pada penelitian ini terdiri dari data *transformation* dan pembagian data latih dan data uji seperti pada Gambar 3.2.



**Gambar 3.2 Flowchart Pengolahan Data**

Tahap-tahap pengolahan data pada Gambar 3.2 diuraikan dengan penjelasan berikut:

1. Data penyakit ginjal kronis yang berupa jumlah pasien penyakit ginjal kronis di RSUD Banyumas dari tahun 2011 sampai dengan tahun 2020 dijadikan *input* dalam pengolahan data.
2. Tahap normalisasi data dilakukan terhadap data penyakit ginjal kronis yang berupa jumlah pasien penyakit ginjal kronis menggunakan rumus:

$$X' = \frac{0,8(X-b)}{a-b} + 0,1 \quad (3.1)$$

Dimana,

$X'$  = data ternormalisasi

$X$  = data aktual

$a$  = nilai maksimum data asli

$b$  = nilai minimum data asli

Contoh:

Normalisasi data jumlah pasien penyakit ginjal kronis pada tahun 2011:

$$X1' = \frac{0,8(X1 - b)}{a - b} + 0,1 = \frac{0,8(8 - 3)}{195 - 3} + 0,1 = 0,1208$$

$$X2' = \frac{0,8(X2 - b)}{a - b} + 0,1 = \frac{0,8(17 - 3)}{195 - 3} + 0,1 = 0,1583$$

$$X3' = \frac{0,8(X3 - b)}{a - b} + 0,1 = \frac{0,8(11 - 3)}{195 - 3} + 0,1 = 0,1333$$

$$X4' = \frac{0,8(X4 - b)}{a - b} + 0,1 = \frac{0,8(12 - 3)}{195 - 3} + 0,1 = 0,1375$$

$$X5' = \frac{0,8(X5 - b)}{a - b} + 0,1 = \frac{0,8(32 - 3)}{195 - 3} + 0,1 = 0,2208$$

$$X6' = \frac{0,8(X6 - b)}{a - b} + 0,1 = \frac{0,8(19 - 3)}{195 - 3} + 0,1 = 0,1667$$

$$X7' = \frac{0,8(X7 - b)}{a - b} + 0,1 = \frac{0,8(10 - 3)}{195 - 3} + 0,1 = 0,1292$$

$$X8' = \frac{0,8(X8 - b)}{a - b} + 0,1 = \frac{0,8(20 - 3)}{195 - 3} + 0,1 = 0,1708$$

$$X9' = \frac{0,8(X9 - b)}{a - b} + 0,1 = \frac{0,8(9 - 3)}{195 - 3} + 0,1 = 0,1250$$

$$X10' = \frac{0,8(X10 - b)}{a - b} + 0,1 = \frac{0,8(45 - 3)}{195 - 3} + 0,1 = 0,2750$$

$$X11' = \frac{0,8(X11 - b)}{a - b} + 0,1 = \frac{0,8(12 - 3)}{195 - 3} + 0,1 = 0,1375$$

$$X12' = \frac{0,8(X12 - b)}{a - b} + 0,1 = \frac{0,8(33 - 3)}{195 - 3} + 0,1 = 0,2250$$

$$X13' = \frac{0,8(X13 - b)}{a - b} + 0,1 = \frac{0,8(15 - 3)}{195 - 3} + 0,1 = 0,1500$$

### 3. Pembagian Data Latih dan Data Uji

Data latih yang digunakan oleh penulis yaitu data pasien ginjal kronis pada bulan Januari 2011 sampai dengan bulan November 2019, dimana data pasien ginjal kronis pada bulan Januari 2012 sampai dengan bulan Desember 2019 ditetapkan sebagai target latih seperti pada Tabel 3.2. Data uji yang digunakan oleh penulis yaitu data pasien penyakit ginjal kronis pada bulan Januari 2019 sampai dengan bulan November 2020, dimana data pasien ginjal kronis pada bulan Januari 2020 sampai dengan bulan Desember 2020 ditetapkan sebagai target uji seperti pada Tabel 3.3.

**Tabel 3.2 Pembagian Data Latih dan Target Latih**

Pola	Data Latih	Target Latih
1	Data pada bulan ke-1 s.d data pada bulan ke-12	Data pada bulan ke-13
2	Data pada bulan ke-2 s.d data pada bulan ke-13	Data pada bulan ke-14
3	Data pada bulan ke-3 s.d data pada bulan ke-14	Data pada bulan ke-15
4	Data pada bulan ke-4 s.d data pada bulan ke-15	Data pada bulan ke-16
5	Data pada bulan ke-5 s.d data pada bulan ke-16	Data pada bulan ke-17
6	Data pada bulan ke-6 s.d data pada bulan ke-17	Data pada bulan ke-18
.	.	.
.	.	.
.	.	.
96	Data pada bulan ke-96 s.d data pada bulan ke-107	Data pada bulan ke-108

**Tabel 3.3 Pembagian Data Uji dan Target Uji**

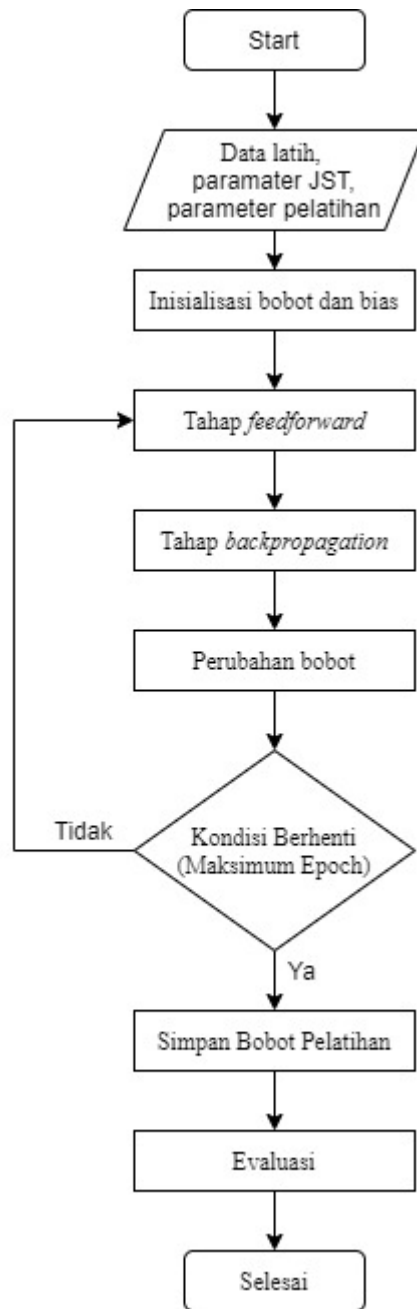
Pola	Data Latih	Target Latih
1	Data pada bulan ke-97 s.d data pada bulan ke-108	Data pada bulan ke-109
2	Data pada bulan ke-98 s.d data pada bulan ke-109	Data pada bulan ke-110
3	Data pada bulan ke-99 s.d data pada bulan ke-110	Data pada bulan ke-111
4	Data pada bulan ke-100 s.d data pada bulan ke-111	Data pada bulan ke-112
5	Data pada bulan ke-101 s.d data pada bulan ke-112	Data pada bulan ke-113
6	Data pada bulan ke-102 s.d data pada bulan ke-113	Data pada bulan ke-114
.	.	.
.	.	.
.	.	.
12	Data pada bulan ke-108 s.d data pada bulan ke-119	Data pada bulan ke-120

### **3.3.6 Prediksi dengan JST dan Kombinasi JST-PSO**

Pada tahap ini dilakukan dua proses prediksi, yaitu prediksi pasien penyakit ginjal kronis menggunakan jaringan saraf tiruan (JST) dan prediksi pasien penyakit ginjal kronis menggunakan kombinasi jaringan saraf tiruan dan *particle swarm optimization* (JST-PSO). Pada masing-masing proses prediksi, didalamnya dilakukan pelatihan, pengujian, dan evaluasi dengan perhitungan MSE. Berikut ini adalah penguraian tahap-tahap dalam melakukan prediksi pasien penyakit ginjal kronis menggunakan JST dan JST-PSO:

1. Prediksi Menggunakan Jaringan Saraf Tiruan (JST)





**Gambar 3.3 Flowchart Pelatihan dengan JST**

Gambar 3.3 di atas merupakan *flowchart* dari pelatihan data pasien penyakit ginjal kronis menggunakan jaringan saraf tiruan (JST). Berikut merupakan penguraian tahap-tahap pelatihan data pasien penyakit ginjal kronis menggunakan jaringan saraf tiruan (JST):

- a. Langkah pertama, yaitu memasukkan *input* berupa data latih, parameter JST, dan parameter pelatihan. Parameter JST dalam penelitian ini meliputi jumlah *input layer*, jumlah *hidden layer* beserta jumlah neuronnya, dan jumlah *output layer*. Pada penelitian ini, digunakan *input layer* dengan jumlah neuron sebanyak 12 yaitu bulan Januari sampai dengan bulan Desember, *hidden layer* yang digunakan yaitu 1 dengan beberapa jumlah neuron yang berbeda dalam setiap percobaan, dan *output layer* yang digunakan berjumlah 1 yaitu jumlah pasien ginjal kronis di tahun berikutnya. Parameter pelatihan yang digunakan yaitu *error goal*, *learning rate*, dan jumlah *epoch*.

Contoh:

Tabel 3.4 merupakan contoh data latih ternormalisasi yang digunakan, yaitu data jumlah pasien penyakit ginjal kronis pada tahun 2011 dari bulan Januari sampai bulan Desember (*input*) dengan target (*output*) jumlah pasien ginjal kronis pada bulan selanjutnya yaitu bulan Januari tahun 2012. Jumlah *hidden layer* yang digunakan yaitu 1 *hidden layer* dengan 1 neuron. Parameter pelatihan yang digunakan sebagai contoh yaitu 0,001 *learning rate*, 1 epoch, dan *error goal* sebesar 0,001.

**Tabel 3.4 Contoh Data Latih**

Tahun	Jumlah Pasien	
2011	X1	0,1208
	X2	0,1583
	X3	0,1333
	X4	0,1375
	X5	0,2208
	X6	0,1667
	X7	0,1292
	X8	0,1708
	X9	0,1250
	X10	0,2750
	X11	0,1375
	X12	0,2250
	T	0,15000

- b. Selanjutnya, dilakukan inialisasi bobot dan bias secara random, dimana jumlah bobot dan bias didapatkan berdasarkan arsitektur JST yang digunakan sesuai dengan parameter JST di langkah pertama.

Contoh:

Tabel 3.5 merupakan contoh inialisasi bobot dan bias yang menghubungkan *input layer* dan *hidden layer*. Tabel 3.6 merupakan contoh inialisasi bobot dan bias yang menghubungkan *hidden layer* dan *output layer*.

**Tabel 3.5 Contoh Inialisasi Bobot dan Bias *Input layer-Hidden Layer***

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	1
Z1	0,2	3,5	2,3	2,5	2,5	2,2	0,3	2,5	0,2	3,3	2,6	4,0	4,8

**Tabel 3.6 Contoh Inialisasi Bobot dan Bias *Hidden Layer-Output Layer***

	Z1	1
Y	-4,7	-3,5

- c. Setelah bobot dan bias sudah diinisialisasikan, maka dapat dilakukan perhitungan *feedforward* menggunakan rumus-rumus berikut ini:

Pada unit tersembunyi  $z_j$  ( $j = 1, \dots, p$ ) menghitung jumlah isyarat masukan sehingga menghasilkan keluaran :

$$z_{netj} = v_{j0} + \sum x_i v_{kj} \quad (3.1)$$

Lalu, melakukan perhitungan fungsi aktivasi sigmoid biner pada unit tersembunyi.

Contoh:

Keluaran di unit tersembunyi,

$$\begin{aligned} z_{net1} &= v_{01} + x_1 v_{11} + x_2 v_{21} + x_3 v_{31} + x_4 v_{41} + x_5 v_{51} + x_6 v_{61} + x_7 v_{71} + \\ & x_8 v_{81} + x_9 v_{91} + x_{10} v_{101} + x_{11} v_{111} + x_{12} v_{121} \\ &= 0,2 + 0,1208.3,5 + 0,1583.2,3 + 0,1333.2,5 + 0,1375.2,5 + 0,2208.2,2 + \\ & 0,1667.0,3 + 0,1292.2,5 + 0,1708.3,3 + 0,1250.2,6 + 0,2750.4,0 + 0,1375.4,8 \\ & + 0,2250.0,5 \end{aligned}$$

$$= 5,2838$$

Fungsi aktivasi di unit tersembunyi,

$$z_1 = f(z_{net1}) = \frac{1}{(1 + e^{-z_{net1}})} = \frac{1}{(1 + e^{-5,2838})} = 0,99495$$

Pada unit output  $y_k$  ( $k=1, \dots, m$ ) dihitung nilai keluarannya.

$$y_{netk} = w_{k0} + \sum z_j w_{kj} \quad (3.2)$$

Lalu, dilakukan perhitungan fungsi aktivasi di unit keluaran.

Contoh:

Keluaran di unit keluaran,

$$\begin{aligned} y_{net1} &= w_{01} + z_1 w_{11} \\ &= -4,7 + 0,99495 \cdot (-3,5) \\ &= -8,182325 \end{aligned}$$

Fungsi aktivasi di unit keluaran,

$$y_1 = f(y_{net1}) = \frac{1}{(1 + e^{-y_{net1}})} = \frac{1}{(1 + e^{8,182325})} = 2,7947$$

- d. Kemudian, setelah nilai keluaran didapatkan pada tahap *feedforward* dilakukan perhitungan *backpropagation* (propagasi balik) menggunakan rumus-rumus di bawah ini:

Nilai keluaran dari dari tiap-tiap unit output  $Y_k$  ( $k = 1, \dots, m$ ) dihitung nilai error:

$$\delta_k = (t_k - y_k) f'(y_{netk}) = (t_k - y_k) y_k (1 - y_k) \quad (3.3)$$

Contoh:

$$\begin{aligned} \delta_1 &= (t_1 - y_1) f'(y_{net1}) \\ &= (t_1 - y_1) y_1 (1 - y_1) \\ &= (0,15000 - 2,7947) \cdot 2,7947 \cdot (1 - 2,7947) \\ &= 13,265 \end{aligned}$$

Hitung nilai perubahan bobot  $W_{kj}$  dimana nilai tersebut digunakan untuk merubah bobot  $W_{kj}$  dengan laju pemahaman  $\alpha$ .

$$\Delta w_{kj} = \alpha \delta_k w_{kj} \quad (3.4)$$

Dengan :  $k = 1, \dots, m$  dan  $j = 0, \dots, p$

Contoh:

$$\Delta w_{11} = \alpha \delta_1 z_1 = 0,001. (13,265). 0,99495 = 0,013198$$

$$\Delta w_{01} = \alpha \delta_1 1 = 0,001. (13,265). 1 = 0,013265$$

Faktor  $\delta$  yang terdapat pada unit layar tersembunyi (hidden layer) dihitung berdasarkan error disetiap unit tersembunyi  $z_j$  ( $j = 1, \dots, p$ ).

$$\delta_{net_1} = \sum \delta_j w_{kj} = 1 \quad (3.5)$$

$$\delta_{net_1} = \delta_1 w_{11} = (13,265). -3,5 = -46,4275$$

Faktor error  $\delta$  di unit tersembunyi :

$$\delta_j = \delta_{net_1} f'(z_{netk}) = \delta_{net_1} z_j (1 - z_j) \quad (3.6)$$

Contoh:

$$\begin{aligned} \delta_j &= \delta_{net_1} z_1 (1 - z_1) = (-46,4275). 0,99495 (1 - 0,99495) \\ &= -0,2344 \end{aligned}$$

Hitung selisih perubahan bobot  $V_{ji}$  :

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (3.7)$$

Contoh:

$$\Delta v_{11} = \alpha \delta_{-1} x_1 = 0,001. (-0,2344). 0,1208 = -0,000028$$

$$\Delta v_{21} = \alpha \delta_{-1} x_2 = 0,001. (-0,2344). 0,1583 = -0,000037$$

$$\Delta v_{31} = \alpha \delta_{-1} x_3 = 0,001. (-0,2344). 0,1333 = -0,000031$$

$$\Delta v_{41} = \alpha \delta_{-1} x_4 = 0,001. (-0,2344). 0,1375 = -0,000032$$

$$\Delta v_{51} = \alpha \delta_{-1} x_5 = 0,001. (-0,2344). 0,2208 = -0,000052$$

$$\Delta v_{61} = \alpha \delta_{-1} x_6 = 0,001. (-0,2344). 0,1667 = -0,000039$$

$$\Delta v_{71} = \alpha \delta_{-1} x_7 = 0,001. (-0,2344). 0,1292 = -0,000030$$

$$\Delta v_{81} = \alpha \delta_{-1} x_8 = 0,001. (-0,2344). 0,1708 = -0,000040$$

$$\Delta v_{91} = \alpha \delta_{-1} x_9 = 0,001. (-0,2344). 0,1250 = -0,000029$$

$$\Delta v_{101} = \alpha \delta_{-1} x_{10} = 0,001. (-0,2344). 0,2750 = -0,000032$$

$$\Delta v_{111} = \alpha \delta_{-1} x_{11} = 0,001. (-0,2344). 0,1375 = -0,000032$$

$$\Delta v_{121} = \alpha \delta_{-1} x_{12} = 0,001. (-0,2344). 0,2250 = -0,000053$$

$$\Delta v_{01} = \alpha \delta_{-1} 1 = 0,001. (-0,2344) = -0,0002344$$

- e. Selanjutnya, dilakukan perhitungan perubahan bobot menggunakan rumus-  
rumus di bawah ini:

Pergantian bobot garis yang mengarah ke unit output :

$$wkj(\text{baru}) = wkj(\text{lama}) + \Delta wkj \quad (3.8)$$

Dengan :  $k = 1, \dots, m$  dan  $j = 0, 1, \dots, p$

Contoh:

$$w_{11}(\text{baru}) = w_{11}(\text{lama}) + \Delta w_{11} = -3,5 + 0,013198 = -3,486802$$

$$w_{01}(\text{baru}) = w_{01}(\text{lama}) + \Delta w_{01} = -4,7 + 0,013265 = -4,686375$$

Pergantian bobot garis yang mengarah ke unit tersembunyi :

$$vji(\text{baru}) = vji(\text{lama}) + \Delta vji \quad (3.9)$$

$$v_{11}(\text{baru}) = v_{11}(\text{lama}) + \Delta v_{11} = 3,5 - 0,000028 = 3,499972$$

$$v_{21}(\text{baru}) = v_{21}(\text{lama}) + \Delta v_{21} = 2,3 - 0,000037 = 2,299963$$

$$v_{31}(\text{baru}) = v_{31}(\text{lama}) + \Delta v_{31} = 2,5 - 0,000031 = 2,499969$$

$$v_{41}(\text{baru}) = v_{41}(\text{lama}) + \Delta v_{41} = 2,5 - 0,000032 = 2,499968$$

$$v_{51}(\text{baru}) = v_{51}(\text{lama}) + \Delta v_{51} = 2,2 - 0,000052 = 2,199948$$

$$v_{61}(\text{baru}) = v_{61}(\text{lama}) + \Delta v_{61} = 0,3 - 0,000039 = 0,299961$$

$$v_{71}(\text{baru}) = v_{71}(\text{lama}) + \Delta v_{71} = 2,5 - 0,000030 = 3,29997$$

$$v_{81}(\text{baru}) = v_{81}(\text{lama}) + \Delta v_{81} = 3,3 - 0,000040 = 3,29996$$

$$v_{91}(\text{baru}) = v_{91}(\text{lama}) + \Delta v_{91} = 2,6 - 0,000029 = 2,599971$$

$$v_{101}(\text{baru}) = v_{101}(\text{lama}) + \Delta v_{101} = 4,0 - 0,000032 = 3,999968$$

$$v_{111}(\text{baru}) = v_{111}(\text{lama}) + \Delta v_{111} = 4,8 - 0,000032 = 4,799968$$

$$v_{121}(\text{baru}) = v_{121}(\text{lama}) + \Delta v_{121} = 0,5 - 0,000053 = 0,499947$$

$$v_{01}(\text{baru}) = v_{01}(\text{lama}) + \Delta v_{01} = 0,2 - 0,0002344 = 0,1997656$$

- f. Setelah seluruh perhitungan dalam algoritma jaringan saraf tiruan sudah dilakukan, maka dilakukan pengecekan kondisi berhenti, dimana kondisi berhenti disini merupakan jumlah maksimum epoch. Pengecekan dilakukan terhadap jumlah epoch apakah sudah mencapai jumlah yang ditetapkan sebagai parameter pelatihan pada *input* atau belum mencapai. Jika epoch belum sesuai, maka akan dilakukan perulangan perhitungan mulai dari tahap *feedforward*, *backpropagation*, dan perubahan bobot sampai jumlah epoch

terpenuhi. Apabila epoch sudah terpenuhi, maka langkah selanjutnya yaitu menyimpan bobot pelatihan, menghitung nilai prediksi hasil proses pelatihan menggunakan nilai output di unit keluaran dan melakukan denormalisasi, setelah itu melakukan evaluasi terhadap algoritma pelatihan dengan perhitungan MSE.

Contoh:

Pada contoh perhitungan jumlah epoch yang digunakan adalah 1. Berikut adalah contoh perhitungan nilai prediksi target yaitu jumlah pasien dibulan Januari 2012.

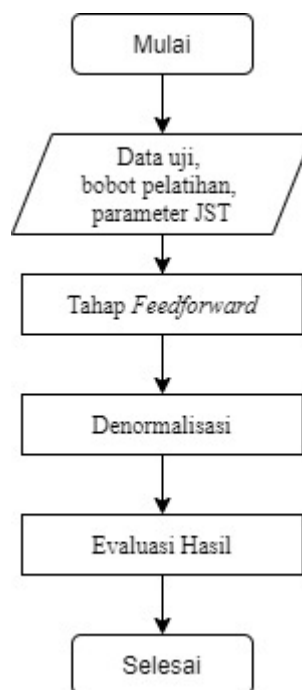
Denormalisasi nilai prediksi,

$$y' = \left( \frac{(y - 0,1)(max - min)}{0,8} \right) + min$$

$$= \left( \frac{(2,7947 - 0,1)(45 - 8)}{0,8} \right) + 8 = 132,629875$$

Perhitungan MSE pada proses pelatihan,

$$MSE = \frac{\sum_{t=1}^n |a - b|^2}{n} = \frac{|0,15000 - 2,7947|^2}{13} = 0,5380$$



**Gambar 3.4 Flowchart Pengujian dengan JST**

Gambar 3.4 di atas merupakan *flowchart* dari pengujian data pasien penyakit ginjal kronis menggunakan jaringan saraf tiruan (JST). Berikut ini adalah penguraian tahap-tahap pengujian data pasien pneyakit ginjal kronis menggunakan jaringan saraf tiruan (JST):

- a. Langkah pertama, yaitu memasukkan *input* berupa data uji, bobot pelatihan, dan parameter JST. Bobot pelatihan merupakan bobot hasil dari pelatihan data pasien penyakit ginjal kronis menggunakan jaringan saraf tiruan (JST). Parameter JST yang digunakan yaitu arsitektur terbaik dari proses pelatihan, arsitektur terbaik yaitu arsitektur JST dengan nilai MSE terkecil pada proses pelatihan.

Contoh:

Pada Tabel 3.4 data uji yang digunakan sebagai contoh yaitu jumlah pasien pemyakit ginjal kronis pada bulan Januari sampai Desember di tahun 2019 dengan target (*output*) jumlah pasien penyakit ginjal kronis pada bulan Januari 2020. Parameter JST yang digunakan yaitu arsitektur terbaik dari proses pelatihan JST, namun karena pada contoh hanya digunakan satu epoch, maka arsitektur yang dipakai sama dengan arsitektur di proses pelatihan.

**Tabel 3.7 Contoh Data Uji**

Tahun	Jumlah Pasien	
2019	X1	0,1583
	X2	0,1333
	X3	0,1375
	X4	0,2208
	X5	0,1667
	X6	0,1292
	X7	0,1708
	X8	0,1250
	X9	0,2750
	X10	0,1375
	X11	0,2250
	X12	0,1500
	T	0,1875



b. Selanjutnya, dilakukan perhitungan *feedforward* untuk mengetahui nilai prediksi pasien penyakit ginjal kronis menggunakan jaringan saraf tiruan (JST). Berikut ini merupakan rumus-rumus dalam tahap *feedforward*:

Pada unit tersembunyi  $z_j$  ( $j = 1, \dots, p$ ) menghitung jumlah isyarat masukan sehingga menghasilkan keluaran :

$$z_{netj} = v_j0 + \sum x_i v_i \quad (3.10)$$

Lalu, melakukan perhitungan fungsi aktivasi sigmoid biner pada unit tersembunyi.

Contoh:

Keluaran di unit tersembunyi,

$$\begin{aligned} z_{-net1} &= v_{01} + x_1 v_{11} + x_2 v_{21} + x_3 v_{31} + x_4 v_{41} + x_5 v_{51} + x_6 v_{61} + \\ & x_7 v_{71} + x_8 v_{81} + x_9 v_{91} + x_{10} v_{101} + x_{11} v_{111} + x_{12} v_{121} \\ &= 0,1997656 + 0,1583 \cdot 3,499972 + 0,1333 \cdot 2,299963 + \\ & 0,1375 \cdot 2,499969 + 0,2208 \cdot 2,499968 + 0,1667 \cdot 2,199948 + \\ & 0,1292 \cdot 0,299961 + 0,1708 \cdot 3,29997 + 0,1250 \cdot 3,29996 + \\ & 0,2750 \cdot 2,599971 + 0,1375 \cdot 3,999968 + 0,2250 \cdot 4,799968 + \\ & 0,1500 \cdot 0,499947 \\ &= 5,7577 \end{aligned}$$

Fungsi aktivasi di unit tersembunyi,

$$z_1 = f(z_{-net1}) = \frac{1}{(1 + e^{-z_{-net1}})} = \frac{1}{(1 + e^{-5,7577})} = 1,003$$

Pada unit output  $y_k$  ( $k=1, \dots, m$ ) dihitung nilai keluarannya.

$$y_{netk} = w_k0 + \sum z_j w_{kj} = 1 = f(y_{netk}) \quad (3.11)$$

Lalu, melakukan perhitungan fungsi aktivasi sigmoid biner pada unit keluaran.

Contoh:

Keluaran di unit keluaran,

$$\begin{aligned} y_{net1} &= w_{01} + z_1 w_{11} \\ &= -4,686375 + 1,003 \cdot -3,486802 \\ &= -8,18364 \end{aligned}$$

Fungsi aktivasi di unit keluaran,

$$y_1 = f(y_{net1}) = \frac{1}{(1 + e^{-y_{net1}})} = \frac{1}{(1 + e^{8,18364})} = 2,79106$$

- c. Setelah hasil prediksi didapatkan, dilakukan denormalisasi terhadap hasil prediksi yang didapatkan dari perhitungan pada tahap *feedforward*.

Contoh:

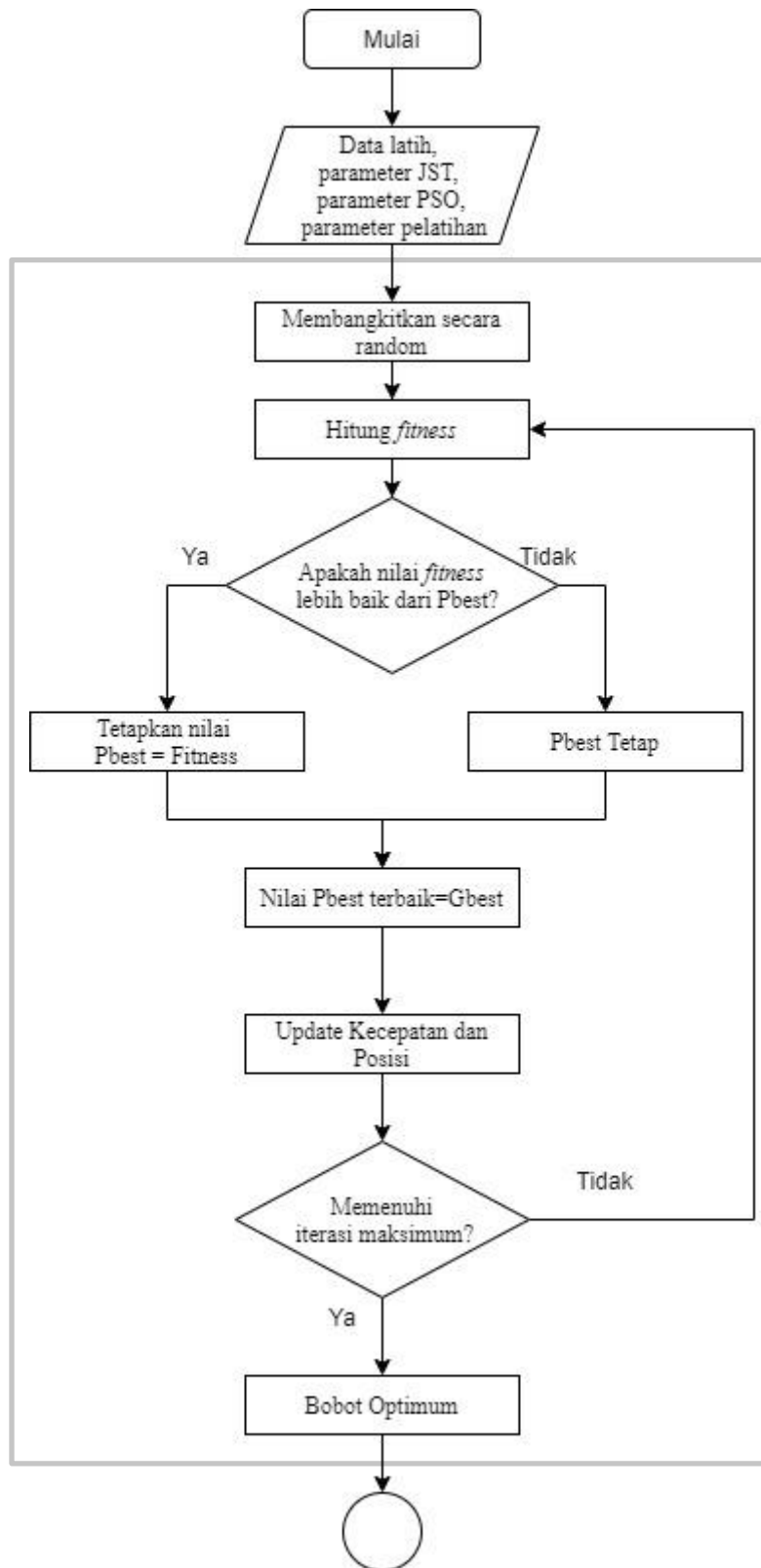
$$y' = \left( \frac{(y - 0,1)(max - min)}{0,8} \right) + min \frac{(2,79106 - 0,1)(60 - 9)}{0,8} + 9$$
$$= 180,555$$

- d. Langkah terakhir yaitu melakukan perhitungan evaluasi pengujian algoritma jaringan saraf tiruan (JST) dalam memprediksi pasien penyakit ginjal kronis menggunakan perhitungan MSE. Nilai MSE ini akan digunakan sebagai nilai performansi algoritma jaringan saraf tiruan (JST) dalam prediksi pasien penyakit ginjal kronis.

Contoh:

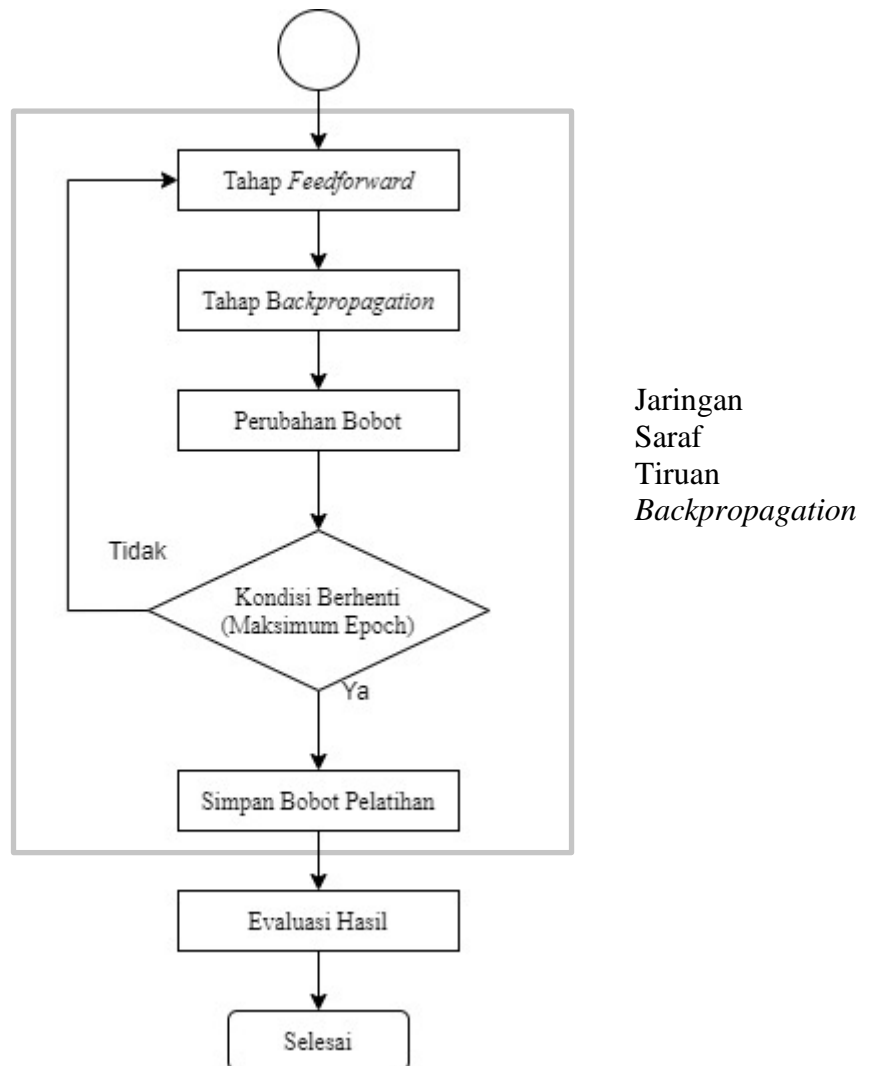
$$MSE = \frac{\sum_{t=1}^n |a - b|^2}{n} = \frac{|0,1875 - 2,79106|^2}{13} = 0,521$$

## 2. Prediksi Menggunakan Jaringan Saraf Tiruan dan *Particle Swarm Optimization* (JST-PSO)



*Particle  
Swarm  
Optimization*

**Gambar 3. 5** Flowchart Pelatihan dengan JST-PSO



**Gambar 3. 6** *Flowchart* Pelatihan dengan JST-PSO

Gambar 3.5 dan Gambar 3.6 yang merupakan lanjutan dari *flowchart* Gambar 3.5 di atas merupakan *flowchart* pelatihan data pasien penyakit ginjal kronis menggunakan kombinasi jaringan saraf tiruan *backpropagation* dan *particle swarm optimization* (JST-PSO). Berikut ini merupakan penguraian tahap-tahap pelatihan menggunakan kombinasi jaringan saraf tiruan dan *particle swarm optimization* (JST-PSO):

- a. Langkah pertama, yaitu memasukkan *input* berupa data latih, parameter JST, parameter PSO, dan parameter pelatihan. Parameter JST meliputi jumlah *input layer*, jumlah *hidden layer* beserta jumlah neuronnya, dan jumlah *output*

*layer*. Pada penelitian ini, digunakan *input layer* dengan jumlah neuron sebanyak 12 yaitu bulan Januari sampai dengan bulan Desember, *hidden layer* yang digunakan yaitu 1 dengan beberapa jumlah neuron yang berbeda dalam setiap percobaan, dan *output layer* yang digunakan berjumlah 1 yaitu jumlah pasien ginjal kronis di tahun berikutnya. Parameter PSO meliputi ukuran/jumlah swarm, c1,c2, w (bobot inersia), iterasi maksimum, fungsi tujuan, batas variabel atas, batas variabel bawah, batas kecepatan atas, dan batas kecepatan bawah. Parameter pelatihan yang digunakan yaitu *error goal*, *learning rate*, dan jumlah *epoch*.

Contoh:

Pada contoh perhitungan, data latih (*input*) yang digunakan sama dengan data latih yang digunakan pada pelatihan dengan JST yang ditunjukkan pada Tabel 3.4 yaitu jumlah pasien pada tahun 2011 dari bulan Januari sampai dengan bulan Februari dan contoh target yang digunakan yaitu jumlah pasien penyakit ginjal kronis di bulan Januari 2012. Jumlah *hidden layer* yang digunakan yaitu 1 *hidden layer* dengan 1 neuron. Parameter pelatihan yang digunakan sebagai contoh yaitu 0,001 *learning rate*, 1 epoch, dan *error goal* sebesar 0,001. Parameter PSO yang digunakan yaitu,

$$nPop=2$$

$$c1=1,5$$

$$c2=2,0$$

$$w=1$$

$$MaxIt=1$$

$$\text{Fungsi tujuan} = f(X) = \frac{\sum error^2}{n} \quad (3.12)$$

$$VarMax = 5, VarMin = -5$$

$$VelMax = \frac{(0,1*(VarMax-VarMin))}{2} = 0,5 \quad (3.13)$$

$$VelMin = -VelMax = -0,5 \quad (3.14)$$

- b. Selanjutnya, dilakukan pembangkitan secara random terhadap posisi awal dan kecepatan awal. Dimana posisi awal merupakan nilai acak dari setiap





partikel dan kecepatan awal bernilai 0. Setiap partikel mewakili jumlah seluruh bobot pada jaringan saraf tiruan *backpropagation*.

Contoh:

**Tabel 3.8 Contoh Inisialisasi Posisi Awal Partikel**

	Posisi (0)	
	x1	x2
j1	0.2	0.4
j2	0.3	-0.1
j3	-2.3	1.3
j4	2.5	2.6
j5	2.2	3.6
j6	0.4	0.2
j7	-0.1	0.2
j8	1.3	0.3
j9	2.6	-2.3
j10	3.6	2.5
j11	0.2	1.2
j12	0.2	4.5
j13	0.3	2.1
j14	-2.3	2.4
j15	2.5	2.2

Keterangan:





	Bobot dari <i>input layer</i> ke <i>hidden layer</i>
	Bias dari <i>input layer</i> ke <i>hidden layer</i>
	Bobot dari <i>hidden layer</i> ke <i>output layer</i>
	Bias dari <i>hidden layer</i> ke <i>output layer</i>

**Tabel 3.9 Contoh Inisialisasi Kecepatan Awal Partikel**

	Kecepatan (0)	
	V1	V2
j1	0	0
j2	0	0
j3	0	0
j4	0	0
j5	0	0
j6	0	0

	Kecepatan (0)	
	V1	V2
j7	0	0
j8	0	0
j9	0	0
j10	0	0
j11	0	0
j12	0	0
j13	0	0
j14	0	0
j15	0	0

Keterangan:

	Bobot dari <i>input layer</i> ke <i>hidden layer</i>
	Bias dari <i>input layer</i> ke <i>hidden layer</i>
	Bobot dari <i>hidden layer</i> ke <i>output layer</i>
	Bias dari <i>hidden layer</i> ke <i>output layer</i>

- c. Kemudian, dilakukan perhitungan *fitness* (nilai kemampuan berpindah posisi) dari masing-masing partikel yang didapatkan menggunakan nilai fungsi tujuan. Setelah didapatkan nilai *fitness* dari masing-masing partikel, dilakukan pemilihan *fitness* terbaik yaitu *fitness* dengan nilai terkecil. Posisi partikel pada *fitness* dengan nilai terkecil dijadikan nilai *Pbest*.

Contoh:

Fungsi tujuan yang digunakan dalam perhitungan *fitness* menggunakan nilai *error* didalamnya, maka diperlukan perhitungan *feedforward* terlebih dahulu, seperti berikut ini:

$$\begin{aligned}
 z_{-net1} &= v_{01} + x_1v_{11} + x_2v_{21} + x_3v_{31} + x_4v_{41} + x_5v_{51} + x_6v_{61} + \\
 &x_7v_{71} + x_8v_{81} + x_9v_{91} + x_{10}v_{101} + x_{11}v_{111} + x_{12}v_{121} \\
 &= 0,3 + 0,1583 \cdot 0,2 + 0,1333 \cdot 0,3 + 0,1375 \cdot -2,3 + 0,2208 \cdot 2,5 + 0,1667 \cdot 2,2 \\
 &+ 0,1292 \cdot 0,4 + 0,1708 \cdot -0,1 + 0,1250 \cdot 1,3 + 0,2750 \cdot 3,6 + 0,1375 \cdot 3,6 + \\
 &0,2250 \cdot 0,2 + 0,1500 \cdot 0,2 \\
 &= 3,196373
 \end{aligned}$$

$$\begin{aligned}
 z_{-net2} &= v_{01} + x_1v_{12} + x_2v_{22} + x_3v_{32} + x_4v_{42} + x_5v_{52} + x_6v_{62} + \\
 &x_7v_{72} + x_8v_{82} + x_9v_{92} + x_{10}v_{102} + x_{11}v_{112} + x_{12}v_{122}
 \end{aligned}$$

$$\begin{aligned}
&= 2,1 + 0,1583 \cdot 0,4 + 0,1333 \cdot -0,1 + 0,1375 \cdot 1,3 + 0,2208 \cdot 2,6 + 0,1667 \cdot 3,6 + \\
&0,1292 \cdot 0,2 + 0,1708 \cdot 0,2 + 0,1250 \cdot 0,3 + 0,2750 \cdot -2,3 + 0,1375 \cdot 2,5 + \\
&0,2250 \cdot 1,2 + 0,1500 \cdot 4,5 \\
&= 4,25669
\end{aligned}$$

Fungsi aktivasi di unit tersembunyi,

$$\begin{aligned}
z_1 &= f(z_{net1}) = \frac{1}{(1 + e^{-z_{net1}})} = \frac{1}{(1 + e^{-3,196373})} = 0,960 \\
z_2 &= f(z_{net2}) = \frac{1}{(1 + e^{-z_{net2}})} = \frac{1}{(1 + e^{-4,25669})} = 0,986
\end{aligned}$$

Keluaran di unit keluaran,

$$\begin{aligned}
y_{net1} &= w_{01} + z_1 w_{11} \\
&= 2,5 + 0,960 \cdot -2,3 \\
&= 0,292 \\
y_{net2} &= w_{02} + z_1 w_{12} \\
&= 2,2 + 0,986 \cdot 2,4 \\
&= 4,6
\end{aligned}$$

Fungsi aktivasi di unit keluaran,

$$\begin{aligned}
y_1 &= f(y_{net1}) = \frac{1}{(1 + e^{-y_{net1}})} = \frac{1}{(1 + e^{-0,292})} = 0,572 \\
y_2 &= f(y_{net2}) = \frac{1}{(1 + e^{-y_{net2}})} = \frac{1}{(1 + e^{-4,6})} = 0,990
\end{aligned}$$

Perhitungan *fitness*,

$$\begin{aligned}
F(x_1^0) &= \frac{(0,15 - 0,572)^2}{13} \\
&= 0,014 \\
F(x_2^0) &= \frac{(0,15 - 0,990)^2}{13} \\
&= 0,054
\end{aligned}$$

Pada iterasi ke-0 ( $t=0$ ), nilai Pbest disamakan dengan nilai posisi awal partikel, yaitu ( $Pbest_{i,j}(t) = x_{i,j}(t)$ ).

**Tabel 3.10 Contoh Pbest Awal**

	Pbest(0)	
	Pbest1	Pbest2
j1	0.2	0.4
j2	0.3	-0.1



	Pbest(0)	
	Pbest1	Pbest2
j3	-2.3	1.3
j4	2.5	2.6
j5	2.2	3.6
j6	0.4	0.2
j7	-0.1	0.2
j8	1.3	0.3
j9	2.6	-2.3
j10	3.6	2.5
j11	0.2	1.2
j12	0.2	4.5
j13	0.3	2.1
j14	-2.3	2.4
j15	2.5	2.2

- d. Setelah mendapatkan nilai Pbest, maka dilakukan penetapan nilai Gbest, dimana nilai Gbest merupakan nilai terbaik Pbest.

**Tabel 3.11 Contoh Gbest Awal**

	Gbest (0)
j1	0.2
j2	0.3
j3	-2.3
j4	2.5
j5	2.2
j6	0.4
j7	-0.1
j8	1.3
j9	2.6
j10	3.6
j11	0.2
j12	0.2
j13	0.3
j14	-2.3
j15	2.5

e. Kemudian, dilakukan *update* kecepatan dan posisi menggunakan rumus di bawah ini:

Perhitungan *update* kecepatan:

$$v_{i,j}^{t+1} = w * v_{i,j}^t + c1 * R * (Pbest_{i,j}^t - x_{i,j}^t) + c2 * R * (Gbest_{g,j}^t - x_{i,j}^t) \quad (3.15)$$

Contoh:

$$\begin{aligned} v_{1,1}^{0+1} &= w * v_{1,1}^0 + c1 * R * (Pbest_{1,1}^0 - x_{1,1}^0) + c2 * R * (Gbest_{1,1}^0 - x_{1,1}^0) \\ &= (1)(0) + (1,5)(0,1)(0,2 - 0,2) + (2)(0,1)(0,2 - 0,2) = 0 \\ v_{1,2}^{0+1} &= w * v_{1,2}^0 + c1 * R * (Pbest_{1,2}^0 - x_{1,2}^0) + c2 * R * (Gbest_{1,2}^0 - x_{1,2}^0) \\ &= (1)(0) + (1,5)(0,1)(0,3 - 0,3) + (2)(0,1)(0,3 - 0,3) = 0 \\ v_{1,3}^{0+1} &= w * v_{1,3}^0 + c1 * R * (Pbest_{1,3}^0 - x_{1,3}^0) + c2 * R * (Gbest_{1,3}^0 - x_{1,3}^0) \\ &= (1)(0) + (1,5)(0,1)(-2,3 + 2,3) + (2)(0,1)(-2,3 + 2,3) = 0 \\ v_{1,4}^{0+1} &= w * v_{1,4}^0 + c1 * R * (Pbest_{1,4}^0 - x_{1,4}^0) + c2 * R * (Gbest_{1,4}^0 - x_{1,4}^0) \\ &= (1)(0) + (1,5)(0,1)(2,5 - 2,5) + (2)(0,1)(2,5 - 2,5) = 0 \\ v_{1,5}^{0+1} &= w * v_{1,5}^0 + c1 * R * (Pbest_{1,5}^0 - x_{1,5}^0) + c2 * R * (Gbest_{1,5}^0 - x_{1,5}^0) \\ &= (1)(0) + (1,5)(0,1)(2,2 - 2,2) + (2)(0,1)(2,2 - 2,2) = 0 \\ v_{1,6}^{0+1} &= w * v_{1,6}^0 + c1 * R * (Pbest_{1,6}^0 - x_{1,6}^0) + c2 * R * (Gbest_{1,6}^0 - x_{1,6}^0) \\ &= (1)(0) + (1,5)(0,1)(0,4 - 0,4) + (2)(0,1)(0,4 - 0,4) = 0 \\ v_{1,7}^{0+1} &= w * v_{1,7}^0 + c1 * R * (Pbest_{1,7}^0 - x_{1,7}^0) + c2 * R * (Gbest_{1,7}^0 - x_{1,7}^0) \\ &= (1)(0) + (1,5)(0,1)(-0,1 + 0,1) + (2)(0,1)(-0,1 + 0,1) = 0 \\ v_{1,8}^{0+1} &= w * v_{1,8}^0 + c1 * R * (Pbest_{1,8}^0 - x_{1,8}^0) + c2 * R * (Gbest_{1,8}^0 - x_{1,8}^0) \\ &= (1)(0) + (1,5)(0,1)(1,3 - 1,3) + (2)(0,1)(1,3 - 1,3) = 0 \\ v_{1,9}^{0+1} &= w * v_{1,9}^0 + c1 * R * (Pbest_{1,9}^0 - x_{1,9}^0) + c2 * R * (Gbest_{1,9}^0 - x_{1,9}^0) \\ &= (1)(0) + (1,5)(0,1)(2,6 - 2,6) + (2)(0,1)(2,6 - 2,6) = 0 \\ v_{1,10}^{0+1} &= w * v_{1,10}^0 + c1 * R * (Pbest_{1,10}^0 - x_{1,10}^0) + c2 * R * (Gbest_{1,10}^0 - x_{1,10}^0) \\ &= (1)(0) + (1,5)(0,1)(3,6 - 3,6) + (2)(0,1)(3,6 - 3,6) = 0 \\ v_{1,11}^{0+1} &= w * v_{1,11}^0 + c1 * R * (Pbest_{1,11}^0 - x_{1,11}^0) + c2 * R * (Gbest_{1,11}^0 - x_{1,11}^0) \\ &= (1)(0) + (1,5)(0,1)(0,2 - 0,2) + (2)(0,1)(0,2 - 0,2) = 0 \\ v_{1,12}^{0+1} &= w * v_{1,12}^0 + c1 * R * (Pbest_{1,12}^0 - x_{1,12}^0) + c2 * R * (Gbest_{1,12}^0 - x_{1,12}^0) \\ &= (1)(0) + (1,5)(0,1)(-2,3 + 2,3) + (2)(0,1)(0,2 - 0,2) = 0 \\ v_{1,13}^{0+1} &= w * v_{1,13}^0 + c1 * R * (Pbest_{1,13}^0 - x_{1,13}^0) + c2 * R * (Gbest_{1,13}^0 - x_{1,13}^0) \\ &= (1)(0) + (1,5)(0,1)(0,3 - 0,3) + (2)(0,1)(0,3 - 0,3) = 0 \\ v_{1,14}^{0+1} &= w * v_{1,14}^0 + c1 * R * (Pbest_{1,14}^0 - x_{1,14}^0) + c2 * R * (Gbest_{1,14}^0 - x_{1,14}^0) \\ &= (1)(0) + (1,5)(0,1)(-2,3 + 2,3) + (2)(0,1)(-2,3 + 2,3) = 0 \end{aligned}$$

$$\begin{aligned}
v_{1,15}^{0+1} &= w * v_{1,15}^0 + c1 * R * (Pbest_{1,15}^0 - x_{1,15}^0) + c2 * R * (Gbest_{1,15}^0 - x_{1,15}^0) \\
&= (1)(0) + (1,5)(0,1)(2,5 - 2,5) + (2)(0,1)(2,5 - 2,5) = 0 \\
v_{2,1}^{0+1} &= w * v_{2,1}^0 + c1 * R * (Pbest_{2,1}^0 - x_{2,1}^0) + c2 * R * (Gbest_{1,1}^0 - x_{2,1}^0) \\
&= (1)(0) + (1,5)(0,1)(0,4 - 0,4) + (2)(0,1)(0,2 - 0,4) = -0,04 \\
v_{2,2}^{0+1} &= w * v_{2,2}^0 + c1 * R * (Pbest_{2,2}^0 - x_{2,2}^0) + c2 * R * (Gbest_{1,2}^0 - x_{2,2}^0) \\
&= (1)(0) + (1,5)(0,1)(-0,1 + 0,1) + (2)(0,1)(0,3 + 0,1) = 0,08 \\
v_{2,3}^{0+1} &= w * v_{2,3}^0 + c1 * R * (Pbest_{2,3}^0 - x_{2,3}^0) + c2 * R * (Gbest_{1,3}^0 - x_{2,3}^0) \\
&= (1)(0) + (1,5)(0,1)(1,3 - 1,3) + (2)(0,1)(-2,3 + 1,3) = -0,26 \\
v_{2,4}^{0+1} &= w * v_{2,4}^0 + c1 * R * (Pbest_{2,4}^0 - x_{2,4}^0) + c2 * R * (Gbest_{1,4}^0 - x_{2,4}^0) \\
&= (1)(0) + (1,5)(0,1)(2,6 - 2,6) + (2)(0,1)(2,5 - 2,6) = -0,02 \\
v_{2,5}^{0+1} &= w * v_{2,5}^0 + c1 * R * (Pbest_{2,5}^0 - x_{2,5}^0) + c2 * R * (Gbest_{1,5}^0 - x_{2,5}^0) \\
&= (1)(0) + (1,5)(0,1)(3,6 - 3,6) + (2)(0,1)(2,2 - 3,6) = -0,28 \\
v_{2,6}^{0+1} &= w * v_{2,6}^0 + c1 * R * (Pbest_{2,6}^0 - x_{2,6}^0) + c2 * R * (Gbest_{1,6}^0 - x_{2,6}^0) \\
&= (1)(0) + (1,5)(0,1)(0,2 - 0,2) + (2)(0,1)(0,4 - 0,2) = 0,04 \\
v_{2,7}^{0+1} &= w * v_{2,7}^0 + c1 * R * (Pbest_{2,7}^0 - x_{2,7}^0) + c2 * R * (Gbest_{1,7}^0 - x_{2,7}^0) \\
&= (1)(0) + (1,5)(0,1)(0,2 - 0,2) + (2)(0,1)(-0,1 - 0,2) = -0,06 \\
v_{2,8}^{0+1} &= w * v_{2,8}^0 + c1 * R * (Pbest_{2,8}^0 - x_{2,8}^0) + c2 * R * (Gbest_{1,8}^0 - x_{2,8}^0) \\
&= (1)(0) + (1,5)(0,1)(0,3 - 0,3) + (2)(0,1)(1,3 - 3,3) = -0,4 \\
v_{2,9}^{0+1} &= w * v_{2,9}^0 + c1 * R * (Pbest_{2,9}^0 - x_{2,9}^0) + c2 * R * (Gbest_{1,9}^0 - x_{2,9}^0) \\
&= (1)(0) + (1,5)(0,1)(-2,3 + 2,3) + (2)(0,1)(2,6 + 2,3) = 0,98 \\
v_{2,10}^{0+1} &= w * v_{2,10}^0 + c1 * R * (Pbest_{2,10}^0 - x_{2,10}^0) + c2 * R * (Gbest_{1,10}^0 - x_{2,10}^0) \\
&= (1)(0) + (1,5)(0,1)(2,5 - 2,5) + (2)(0,1)(3,6 - 2,5) = 0,22 \\
v_{2,11}^{0+1} &= w * v_{2,11}^0 + c1 * R * (Pbest_{2,11}^0 - x_{2,11}^0) + c2 * R * (Gbest_{1,11}^0 - x_{2,11}^0) \\
&= (1)(0) + (1,5)(0,1)(1,2 - 1,2) + (2)(0,1)(0,2 - 1,2) = 0,2 \\
v_{2,12}^{0+1} &= w * v_{2,12}^0 + c1 * R * (Pbest_{2,12}^0 - x_{2,12}^0) + c2 * R * (Gbest_{1,12}^0 - x_{2,12}^0) \\
&= (1)(0) + (1,5)(0,1)(4,5 + 4,5) + (2)(0,1)(0,2 - 4,5) = -0,86 \\
v_{2,13}^{0+1} &= w * v_{2,13}^0 + c1 * R * (Pbest_{2,13}^0 - x_{2,13}^0) + c2 * R * (Gbest_{1,13}^0 - x_{2,13}^0) \\
&= (1)(0) + (1,5)(0,1)(2,1 - 2,1) + (2)(0,1)(0,3 - 2,1) = -0,36 \\
v_{2,14}^{0+1} &= w * v_{2,14}^0 + c1 * R * (Pbest_{2,14}^0 - x_{2,14}^0) + c2 * R * (Gbest_{1,14}^0 - x_{2,14}^0) \\
&= (1)(0) + (1,5)(0,1)(2,4 + 2,4) + (2)(0,1)(-2,3 - 2,4) = -0,94 \\
v_{2,15}^{0+1} &= w * v_{2,15}^0 + c1 * R * (Pbest_{2,15}^0 - x_{2,15}^0) + c2 * R * (Gbest_{1,15}^0 - x_{2,15}^0) \\
&= (1)(0) + (1,5)(0,1)(2,2 - 2,2) + (2)(0,1)(2,5 - 2,2) = 0,06
\end{aligned}$$

**Tabel 3.12 Contoh Hasil *Update* Kecepatan Sebelum Pengecekan Batas**

	Kecepatan(1)	
	V1	V2
j1	0	-0,04
j2	0	0,08
j3	0	-0,26
j4	0	-0,02
j5	0	-0,28
j6	0	0,04
j7	0	-0,06
j8	0	-0,4
j9	0	0,98
j10	0	0,22
j11	0	0,2
j12	0	-0,86
j13	0	-0,36
j14	0	-0,94
j15	0	0,06

Setelah *update kecepatan* selesai dilakukan, perlu dilakukan pengecekan nilai kecepatan berdasarkan batas kecepatan atas dan kecepatan bawah. Tabel 3.13 adalah hasil *update* kecepatan sesuai dengan batas kecepatan atas dan bawah.

**Tabel 3.13 Contoh Hasil *Update* Kecepatan Setelah Pengecekan Batas**

	Kecepatan(1)	
	V1	V2
j1	0	-0,04
j2	0	0,08
j3	0	-0,26
j4	0	-0,02
j5	0	-0,28
j6	0	0,04
j7	0	-0,06
j8	0	-0,4
j9	0	0,5
j10	0	0,22
j11	0	0,2
j12	0	-0,5

	Kecepatan(1)	
	V1	V2
j13	0	-0,36
j14	0	-0,5
j15	0	0,06

Perhitungan *update* posisi:

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \quad (3.16)$$

Contoh:

$$\begin{aligned} x_{1,1}^{0+1} &= x_{1,1}^0 + v_{1,1}^{0+1} = 0,2 + 0 = 0,2 \\ x_{1,2}^{0+1} &= x_{1,2}^0 + v_{1,2}^{0+1} = 0,3 + 0 = 0,3 \\ x_{1,3}^{0+1} &= x_{1,3}^0 + v_{1,3}^{0+1} = -2,3 + 0 = -2,3 \\ x_{1,4}^{0+1} &= x_{1,4}^0 + v_{1,4}^{0+1} = 2,5 - 0 = 2,5 \\ x_{1,5}^{0+1} &= x_{1,5}^0 + v_{1,5}^{0+1} = 2,2 + 0 = 2,2 \\ x_{1,6}^{0+1} &= x_{1,6}^0 + v_{1,6}^{0+1} = 0,4 + 0 = 0,4 \\ x_{1,7}^{0+1} &= x_{1,7}^0 + v_{1,7}^{0+1} = -0,1 + 0 = -0,1 \\ x_{1,8}^{0+1} &= x_{1,8}^0 + v_{1,8}^{0+1} = 1,3 + 0 = 1,3 \\ x_{1,9}^{0+1} &= x_{1,9}^0 + v_{1,9}^{0+1} = 3,6 - 0 = 3,6 \\ x_{1,10}^{0+1} &= x_{1,10}^0 + v_{1,10}^{0+1} = 3,6 - 0 = 3,6 \\ x_{1,11}^{0+1} &= x_{1,11}^0 + v_{1,11}^{0+1} = 0,2 + 0 = 0,2 \\ x_{1,12}^{0+1} &= x_{1,12}^0 + v_{1,12}^{0+1} = 0,2 + 0 = 0,2 \\ x_{1,13}^{0+1} &= x_{1,13}^0 + v_{1,13}^{0+1} = 0,3 + 0 = 0,3 \\ x_{1,14}^{0+1} &= x_{1,14}^0 + v_{1,14}^{0+1} = -2,3 + 0 = -2,3 \\ x_{1,15}^{0+1} &= x_{1,15}^0 + v_{1,15}^{0+1} = 2,5 + 0 = 2,5 \\ x_{2,1}^{0+1} &= x_{2,1}^0 + v_{2,1}^{0+1} = 0,4 - 0,04 = 0,36 \\ x_{2,2}^{0+1} &= x_{2,2}^0 + v_{2,2}^{0+1} = -0,1 + 0,08 = -0,02 \\ x_{2,3}^{0+1} &= x_{2,3}^0 + v_{2,3}^{0+1} = 1,3 - 0,26 = 1,04 \\ x_{2,4}^{0+1} &= x_{2,4}^0 + v_{2,4}^{0+1} = 2,6 - 0,02 = 2,58 \\ x_{2,5}^{0+1} &= x_{2,5}^0 + v_{2,5}^{0+1} = 3,6 - 0,28 = 3,32 \\ x_{2,6}^{0+1} &= x_{2,6}^0 + v_{2,6}^{0+1} = 0,2 + 0,04 = 0,24 \\ x_{2,7}^{0+1} &= x_{2,7}^0 + v_{2,7}^{0+1} = 0,2 - 0,06 = 0,14 \\ x_{2,8}^{0+1} &= x_{2,8}^0 + v_{2,8}^{0+1} = 0,3 - 0,4 = -0,1 \\ x_{2,9}^{0+1} &= x_{2,9}^0 + v_{2,9}^{0+1} = -2,3 + 0,5 = -1,8 \\ x_{2,10}^{0+1} &= x_{2,10}^0 + v_{2,10}^{0+1} = 2,5 + 0,22 = 2,72 \\ x_{2,11}^{0+1} &= x_{2,11}^0 + v_{2,11}^{0+1} = 1,2 + 0,2 = 1,4 \end{aligned}$$

$$x_{2,12}^{0+1} = x_{2,12}^0 + v_{2,12}^{0+1} = 4,5 - 0,5 = 3,64$$

$$x_{2,13}^{0+1} = x_{2,13}^0 + v_{2,13}^{0+1} = 2,1 - 0,36 = 1,74$$

$$x_{2,14}^{0+1} = x_{2,14}^0 + v_{2,14}^{0+1} = 2,4 - 0,5 = 1,9$$

$$x_{2,15}^{0+1} = x_{2,15}^0 + v_{2,15}^{0+1} = 2,2 + 0,06 = 2,26$$

**Tabel 3.14 Contoh Hasil *Update* Posisi Sebelum Pengecekan Batas**

	Posisi(1)	
	x1	x2
j1	0,2	0,36
j2	0,3	-0,92
j3	-2,3	1,04
j4	2,5	2,58
j5	2,2	3,32
j6	0,4	0,24
j7	-0,1	0,14
j8	1,3	-0,1
j9	3,6	-1,8
j10	3,6	2,72
j11	0,2	1,22
j12	0,2	3,64
j13	0,3	1,74
j14	-2,3	1,9
j15	2,5	2,26

Karena seluruh posisi masuk diantara batas atas dan batas bawah variabel atau posisi yaitu di antara -5 dan 5, maka hasil *update* posisi sesudah pengecekan batas nilai posisi setiap partikel sama dengan hasil *update* posisi sebelum pengecekan batas nilai posisi setiap partikel.

- f. Setelah perhitungan selesai, dilakukan pengecekan apakah sudah mencapai iterasi maksimum yang telah ditetapkan sebagai parameter PSO pada *input*. Apabila belum mencapai iterasi maksimum, maka akan dilakukan perulangan perhitungan kembali dari hitung *fitness* sampai mencapai iterasi maksimum. Namun, apabila sudah mencapai iterasi maksimum, bobot yang dihasilkan oleh algoritma PSO disimpan sebagai bobot optimum.

Contoh:

Pada contoh perhitungan dilakukan perhitungan *fitness*, *update Pbest* dan *Gbest* kembali untuk mendapatkan bobot optimum.

- Perhitungan *fitness*

$$\begin{aligned} z_{-net1} &= v_{01} + x_1 v_{11} + x_2 v_{21} + x_3 v_{31} + x_4 v_{41} + x_5 v_{51} + x_6 v_{61} + \\ & x_7 v_{71} + x_8 v_{81} + x_9 v_{91} + x_{10} v_{101} + x_{11} v_{111} + x_{12} v_{121} \\ &= 0,3 + 0,1583 \cdot 0,2 + 0,1333 \cdot 0,3 + 0,1375 \cdot -2,3 + 0,2208 \cdot 2,5 + 0,1667 \cdot 2,2 \\ &+ 0,1292 \cdot + 0,1708 \cdot 0,4 + 0,1250 \cdot -0,1 + 0,2750 \cdot 1,3 + 0,1375 \cdot 3,6 + \\ &0,2250 \cdot 3,6 + 0,1500 \cdot 0,2 \\ &= 3,196373 \end{aligned}$$

$$\begin{aligned} z_{-net2} &= v_{01} + x_1 v_{12} + x_2 v_{22} + x_3 v_{32} + x_4 v_{42} + x_5 v_{52} + x_6 v_{62} + \\ & x_7 v_{72} + x_8 v_{82} + x_9 v_{92} + x_{10} v_{102} + x_{11} v_{112} + x_{12} v_{122} \\ &= 1,74 + 0,1583 \cdot 0,36 + 0,1333 \cdot -0,92 + 0,1375 \cdot 1,04 + 0,2208 \cdot 2,58 + \\ &0,1667 \cdot 3,32 + 0,1292 \cdot 0,24 + 0,1708 \cdot 0,14 + 0,1250 \cdot -0,1 + 0,2750 \cdot -1,8 + \\ &0,1375 \cdot 2,72 + 0,2250 \cdot 1,22 + 0,1500 \cdot 3,64 \\ &= 3,68238 \end{aligned}$$

Fungsi aktivasi di unit tersembunyi,

$$\begin{aligned} z_1 &= f(z_{-net1}) = \frac{1}{(1 + e^{-z_{-net1}})} = \frac{1}{(1 + e^{-3,196373})} = 0,961 \\ z_2 &= f(z_{-net2}) = \frac{1}{(1 + e^{-z_{-net2}})} = \frac{1}{(1 + e^{-3,68238})} = 0,975 \end{aligned}$$

Keluaran di unit keluaran,

$$\begin{aligned} y_{net1} &= w_{01} + z_1 w_{11} \\ &= 2,5 + 0,961 \cdot -2,3 \\ &= 0,2897 \\ y_{net2} &= w_{02} + z_1 w_{12} \\ &= 2,26 + 0,975 \cdot 1,9 \\ &= 4,1125 \end{aligned}$$

Fungsi aktivasi di unit keluaran,

$$\begin{aligned} y_1 &= f(y_{net1}) = \frac{1}{(1 + e^{-y_{net1}})} = \frac{1}{(1 + e^{-0,2897})} = 0,572 \\ y_2 &= f(y_{net2}) = \frac{1}{(1 + e^{-y_{net2}})} = \frac{1}{(1 + e^{-4,1125})} = 0,984 \end{aligned}$$

*Fitness,*

$$F(x_1^1) = \frac{(0,15-0,572)^2}{13}$$

$$=0,014$$

$$F(x_2^1) = \frac{(0,15-0,984)^2}{13}$$

$$=0,053$$

- *Update Pbest* dengan membandingkan antara Pbest pada iterasi sebelumnya dengan hasil dari *update* posisi, Pbest dengan nilai *fitness* lebih kecil akan menjadi hasil *update* Pbest pada iterasi 1.

**Tabel 3.15 Hasil *Update* Pbest pada iterasi 1**

	Pbest(1)	
	Pbest1	Pbest2
j1	0,2	0,36
j2	0,3	-0,92
j3	-2,3	1,04
j4	2,5	2,58
j5	2,2	3,32
j6	0,4	0,24
j7	-0,1	0,14
j8	1,3	-0,1
j9	3,6	-1,8
j10	3,6	2,72
j11	0,2	1,22
j12	0,2	3,64
j13	0,3	1,74
j14	-2,3	1,9
j15	2,5	2,26

- *Update Gbest* dilakukan dengan cara memilih Pbest terbaik (Pbest dengan nilai *fitness* terkecil). Tabel 3.16 merupakan hasil dari proses *update* Gbest sebagai bobot optimum untuk digunakan dalam proses selanjutnya.

**Tabel 3.16 Hasil *Update* Gbest pada iterasi 1**

	Gbest (1)
j1	0,2
j2	0,3



	Gbest (1)
j3	-2,3
j4	2,5
j5	2,2
j6	0,4
j7	-0,1
j8	1,3
j9	3,6
j10	3,6
j11	0,2
j12	0,2
j13	0,3
j14	-2,3
j15	2,5

g. Selanjutnya, dilakukan perhitungan *feedforward*, *backpropagation*, dan perubahan bobot seperti dalam pelatihan menggunakan jaringan saraf tiruan, namun pada proses ini bobot yang digunakan yaitu bobot bobot optimum yang sudah didapatkan dari perhitungan menggunakan PSO.

Contoh:

- Tahap *feedforward*

Keluaran di unit tersembunyi,

$$\begin{aligned}
 z_{net1} &= v_{01} + x_1v_{11} + x_2v_{21} + x_3v_{31} + x_4v_{41} + x_5v_{51} + x_6v_{61} \\
 &\quad + x_7v_{71} + x_8v_{81} + x_9v_{91} + x_{10}v_{101} + x_{11}v_{111} + x_{12}v_{121} \\
 &= 0,3 + 0,1208 \cdot 0,2 + 0,158 \cdot 0,3 + 0,1333 \cdot (-2,3) + 0,1375 \cdot 2,5 + \\
 &\quad 0,2208 \cdot 2,2 + 0,1667 \cdot 0,4 + 0,1292 \cdot -0,1 + 0,1708 \cdot 1,3 + 0,1250 \cdot 3,6 \\
 &\quad + 0,2750 \cdot 3,6 + 0,1375 \cdot 0,2 + 0,2250 \cdot 0,2 \\
 &= 2,68278
 \end{aligned}$$

Fungsi aktivasi di unit tersembunyi,

$$z_1 = f(z_{net1}) = \frac{1}{(1 + e^{-z_{net1}})} = \frac{1}{(1 + e^{-2,68278})} = 0,936$$

Keluaran di unit keluaran,

$$\begin{aligned}
 y_{net1} &= w_{01} + z_1w_{11} \\
 &= 2,5 + 0,936 \cdot (-2,3) \\
 &= 0,3472
 \end{aligned}$$

Fungsi aktivasi di unit keluaran,

$$y_1 = f(y_{net1}) = \frac{1}{(1 + e^{-y_{net1}})} = \frac{1}{(1 + e^{-0,3472})} = 0,5859$$

- Tahap *Backpropagation*

Error dari nilai keluaran di unit keluaran,

$$\begin{aligned} \delta_1 &= (t_1 - y_1)f'(y_{net1}) = (t_1 - y_1)y_1(1 - y_1) \\ &= (0,15000 - 0,5859) \cdot 0,5859 \cdot (1 - 0,5859) \\ &= -0,1057 \end{aligned}$$

- Perubahan Bobot

Perubahan bobot  $W_{kj}$  dengan laju pembelajaran  $\alpha$ ,

$$\begin{aligned} \Delta w_{11} &= \alpha \delta_1 z_1 = 0,001 \cdot (-0,1057) \cdot 0,936 = -0,0000989352 \\ \Delta w_{01} &= \alpha \delta_1 \cdot 1 = 0,001 \cdot (-0,1057) \cdot 1 = -0,0001057 \end{aligned}$$

Faktor  $\delta$  yang terdapat pada unit layar tersembunyi (hidden layer) dihitung berdasarkan error disetiap unit tersembunyi  $z_j$  ( $j = 1, \dots, p$ ).

$$\delta_{net\ 1} = \delta_1 w_{11} = (-0,1057) \cdot -2,3 = 0,24311$$

Faktor error  $\delta$  di unit tersembunyi,

$$\begin{aligned} \delta_{-1} &= \delta_{net\ 1} z_1 (1 - z_1) \\ &= (0,24311) \cdot 0,936 (1 - 0,936) \\ &= 0,064 \end{aligned}$$

Selisih perubahan bobot,

$$\begin{aligned} \Delta v_{11} &= \alpha \delta_{-1} x_1 = 0,001 \cdot (0,064) \cdot 0,1208 = 0,0000077312 \\ \Delta v_{21} &= \alpha \delta_{-1} x_2 = 0,001 \cdot (0,064) \cdot 0,1583 = 0,0000101312 \\ \Delta v_{31} &= \alpha \delta_{-1} x_3 = 0,001 \cdot (0,064) \cdot 0,1333 = 0,0000085312 \\ \Delta v_{41} &= \alpha \delta_{-1} x_4 = 0,001 \cdot (0,064) \cdot 0,1375 = 0,0000088 \\ \Delta v_{51} &= \alpha \delta_{-1} x_5 = 0,001 \cdot (0,064) \cdot 0,2208 = 0,0000141312 \\ \Delta v_{61} &= \alpha \delta_{-1} x_6 = 0,001 \cdot (0,064) \cdot 0,1667 = 0,0000106688 \\ \Delta v_{71} &= \alpha \delta_{-1} x_7 = 0,001 \cdot (0,064) \cdot 0,1292 = 0,0000082688 \\ \Delta v_{81} &= \alpha \delta_{-1} x_8 = 0,001 \cdot (0,064) \cdot 0,1708 = 0,0000109312 \\ \Delta v_{91} &= \alpha \delta_{-1} x_9 = 0,001 \cdot (0,064) \cdot 0,1250 = 0,000008 \\ \Delta v_{101} &= \alpha \delta_{-1} x_{10} = 0,001 \cdot (0,064) \cdot 0,2750 = 0,0000176 \\ \Delta v_{111} &= \alpha \delta_{-1} x_{11} = 0,001 \cdot (0,064) \cdot 0,1375 = 0,0000088 \end{aligned}$$

$$\Delta v_{121} = \alpha \delta_{-1} x_{12} = 0,001 \cdot (0,064) \cdot 0,2250 = 0,0000144$$

$$\Delta v_{01} = \alpha \delta_{-1} 1 = 0,001 \cdot (0,064) = 0,000064$$

Pergantian bobot yang mengarah ke unit keluaran,

$$\begin{aligned} w_{11}(\text{baru}) &= w_{11}(\text{lama}) + \Delta w_{11} \\ &= 2,5 - 0,0000989352 = -2,3000989352 \end{aligned}$$

$$\begin{aligned} w_{01}(\text{baru}) &= w_{01}(\text{lama}) + \Delta w_{01} \\ &= 2,5 - 0,0001057 = 2,4998943 \end{aligned}$$

Pergantian bobot yang mengarah ke unit tersembunyi,

$$\begin{aligned} v_{11}(\text{baru}) &= v_{11}(\text{lama}) + \Delta v_{11} \\ &= 0,2 - 0,0000077312 = 0,1999922688 \end{aligned}$$

$$\begin{aligned} v_{21}(\text{baru}) &= v_{21}(\text{lama}) + \Delta v_{21} \\ &= 0,3 - 0,0000101312 = 0,2999898688 \end{aligned}$$

$$\begin{aligned} v_{31}(\text{baru}) &= v_{31}(\text{lama}) + \Delta v_{31} \\ &= -2,3 + 0,0000085312 = -2,2999914688 \end{aligned}$$

$$\begin{aligned} v_{41}(\text{baru}) &= v_{41}(\text{lama}) + \Delta v_{41} \\ &= 2,5 - 0,0000088 = 2,4999912 \end{aligned}$$

$$\begin{aligned} v_{51}(\text{baru}) &= v_{51}(\text{lama}) + \Delta v_{51} \\ &= 2,5 - 0,0000141312 = 2,4999858688 \end{aligned}$$

$$\begin{aligned} v_{61}(\text{baru}) &= v_{61}(\text{lama}) + \Delta v_{61} \\ &= 0,4 - 0,0000106688 = 0,3999893312 \end{aligned}$$

$$\begin{aligned} v_{71}(\text{baru}) &= v_{71}(\text{lama}) + \Delta v_{71} \\ &= -0,1 - 0,0000082688 = -0,1000082688 \end{aligned}$$

$$\begin{aligned} v_{81}(\text{baru}) &= v_{81}(\text{lama}) + \Delta v_{81} \\ &= 1,3 - 0,0000109312 = 1,2999890688 \end{aligned}$$

$$\begin{aligned} v_{91}(\text{baru}) &= v_{91}(\text{lama}) + \Delta v_{91} \\ &= 3,6 + 0,000008 = 3,600008 \end{aligned}$$

$$\begin{aligned} v_{101}(\text{baru}) &= v_{101}(\text{lama}) + \Delta v_{101} \\ &= 3,6 + 0,0000176 = 3,6000176 \end{aligned}$$

$$\begin{aligned} v_{111}(\text{baru}) &= v_{111}(\text{lama}) + \Delta v_{111} \\ &= 0,2 + 0,0000088 = 0,2000088 \end{aligned}$$

$$\begin{aligned} v_{121}(\text{baru}) &= v_{121}(\text{lama}) + \Delta v_{121} \\ &= 0,2 + 0,0000144 = 0,2000144 \end{aligned}$$

$$\begin{aligned} v_{01}(\text{baru}) &= v_{01}(\text{lama}) + \Delta v_{01} \\ &= 0,3 + 0,000064 = 0,300064 \end{aligned}$$

- h. Langkah terakhir yaitu menyimpan bobot hasil pelatihan, denormalisasi nilai prediksi, dan melakukan evaluasi terhadap performansi kombinasi

jaringan saraf tiruan dan *particle swarm optimization* (JST-PSO) melalui perhitungan MSE sebagai performansi pelatihan.

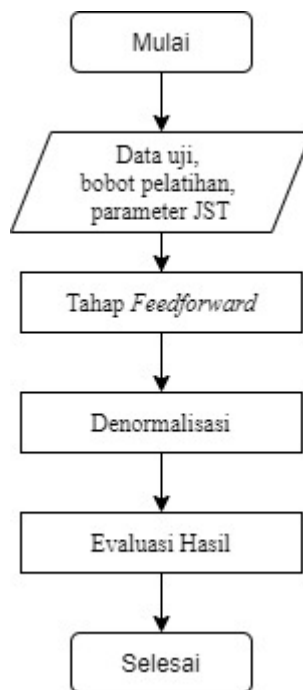
Denormalisasi nilai prediksi,

$$y' = \left( \frac{(y-0,1)(max-min)}{0,8} \right) + \min$$

$$= \left( \frac{(0,5859 - 0,1)(45 - 8)}{0,8} \right) + 8 = 33,500975$$

Evaluasi performansi kombinasi algoritma jaringan saraf tiruan dan *particle swarm optimization* pada tahap pelatihan,

$$MSE = \frac{\sum_{t=1}^n |a - b|^2}{n} = \frac{|0,15000 - 0,5859|^2}{13} = 0,014616$$



**Gambar 3.6 Flowchart Pengujian dengan JST-PSO**

Gambar 3.6 di atas merupakan *flowchart* dari pengujian data pasien penyakit ginjal kronis menggunakan kombinasi jaringan saraf tiruan dan *particle swarm optimization* (JST-PSO). Berikut ini adalah penguraian tahap-tahap pengujian data pasien pneyakit ginjal kronis menggunakan kombinasi jaringan saraf tiruan dan *particle swarm optimization* (JST-PSO):

- a. Langkah pertama, yaitu memasukkan *input* berupa data uji, bobot pelatihan, dan parameter JST. Bobot pelatihan merupakan bobot hasil dari pelatihan data pasien penyakit ginjal kronis menggunakan jaringan saraf tiruan (JST). Parameter JST yang digunakan yaitu arsitektur terbaik dari proses pelatihan, arsitektur terbaik yaitu arsitektur JST dengan nilai MSE terkecil pada proses pelatihan.

Contoh:

Data uji yang digunakan sebagai contoh pengujian menggunakan kombinasi JST dan PSO sama dengan data uji yang digunakan pada contoh pengujian menggunakan JST yang ditunjukkan pada Tabel 3.7. Data uji tersebut yaitu jumlah pasien penyakit ginjal kronis pada bulan Januari sampai Desember di tahun 2019 dengan target (*output*) jumlah pasien penyakit ginjal kronis pada bulan Januari 2020. Parameter JST yang digunakan yaitu arsitektur terbaik dari proses pelatihan JST, namun karena pada contoh hanya digunakan satu epoch, maka arsitektur yang dipakai sama dengan arsitektur di proses pelatihan.

- b. Selanjutnya, dilakukan perhitungan *feedforward* untuk mengetahui nilai prediksi pasien penyakit ginjal kronis menggunakan jaringan saraf tiruan (JST).

Contoh:

Keluaran di unit tersembunyi,

$$\begin{aligned}
 z_{net1} &= v_{01} + x_1v_{11} + x_2v_{21} + x_3v_{31} + x_4v_{41} + x_5v_{51} + x_6v_{61} + \\
 &x_7v_{71} + x_8v_{81} + x_9v_{91} + x_{10}v_{101} + x_{11}v_{111} + x_{12}v_{121} \\
 &= 0,300064+0,1583. 0,1999922688+ \\
 &0,1333. 0,2999898688+0,1375. (-2,2999914688) \\
 &+ 0,2208. 2,4999912+0,1667. 2,4999858688 \\
 &+ 0,1292. 0,3999893312+0,1708. -0,1000082688 \\
 &+ 0,1250. 1,2999890688+ 0,2750. 3,600008 \\
 &+ 0,1375.3,6000176+0,2250. 0,2000088 \\
 &+ 0,1500. 0,2000144 \\
 &=2,78131
 \end{aligned}$$

Fungsi aktivasi di unit tersembunyi,

$$z_1 = f(z_{net1}) = \frac{1}{(1 + e^{-z_{net1}})} = \frac{1}{(1 + e^{-2,78131})} = 0,94166$$

Nilai keluaran di unit keluaran,

$$\begin{aligned} y_{net1} &= w_{01} + z_1 w_{11} \\ &= 2,4998943 + 0,94166 \cdot (-2,3000989352) \\ &= 0,333983 \end{aligned}$$

Fungsi aktivasi di unit keluaran,

$$y_1 = f(y_{net1}) = \frac{1}{(1 + e^{-0,333983})} = 0,58273$$

- c. Setelah hasil prediksi didapatkan, dilakukan denormalisasi terhadap hasil prediksi yang didapatkan dari perhitungan pada tahap *feedforward*.

Contoh:

$$\begin{aligned} y' &= \left( \frac{(y - 0,1)(max - min)}{0,8} \right) + min \\ &= \left( \frac{(0,58273 - 0,1)(60 - 9)}{0,8} \right) + 9 \\ &= 39,774 \end{aligned}$$

- d. Langkah terakhir yaitu melakukan perhitungan evaluasi pengujian algoritma jaringan saraf tiruan (JST) dalam memprediksi pasien penyakit ginjal kronis menggunakan perhitungan MSE. Nilai MSE ini akan digunakan sebagai nilai performansi algoritma jaringan saraf tiruan (JST) dalam prediksi pasien penyakit ginjal kronis.

Contoh:

Evaluasi pengujian prediksi pasien penyakit ginjal kronis menggunakan MSE,

$$MSE = \frac{\sum_{t=1}^n |a - b|^2}{n} = \frac{|0,1875 - 0,58273|^2}{13} = 0,01202$$