BAB 6
PENGUJIAN DAN ANALISIS

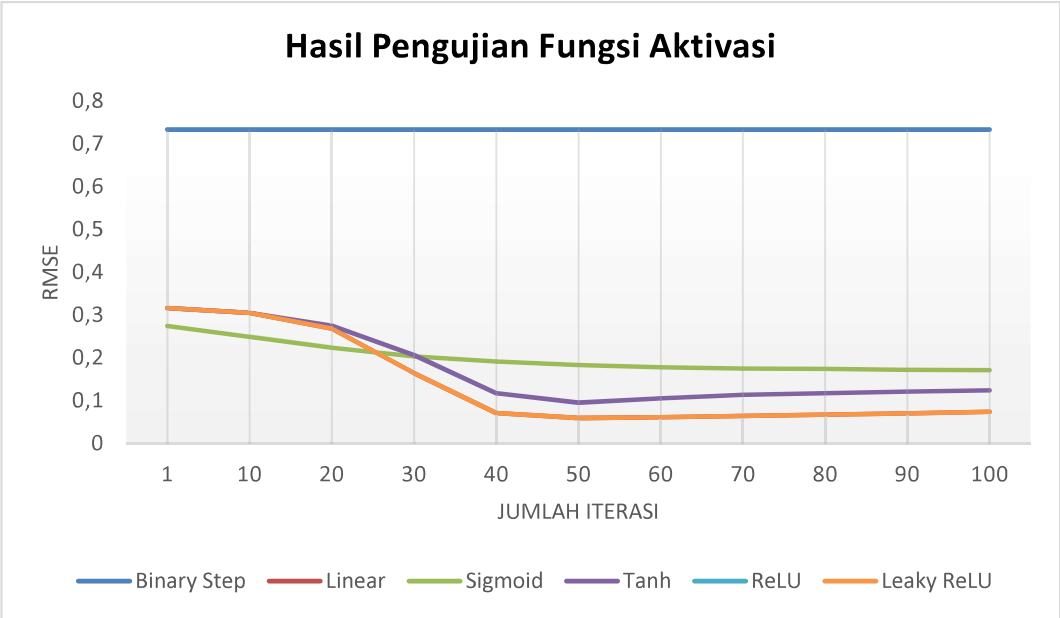
Bab ini membahas hasil dan pengujian program untuk penentuan durasi nyala lampu lalu lintas dengan menggunakan metode *Backpropagation*. Pada bab ini dilakukan 4 pengujian yaitu pengujian fungsi aktivasi, pengujian jumlah iterasi, pengujian jumlah node pada *hidden* layer, dan pengujian nilai *learning rate*.

6.1 Pengujian Fungsi Aktivasi

Pengujian fungsi aktivasi dilakukan untuk mengetahui fungsi akivasi mana yang optimal untuk mendapatkan nilai error terkecil pada penelitian ini. Fungsi aktivasi yang diuji yaitu *Binary Step Function*, *Linear Function*, *Sigmoid Functon*, *Tanh Function*, *ReLU Function*, dan *Leaky ReLU Function*. Setiap satu fungsi aktivasi dilakukan satu kali percobaan, sehingga pada pengujian ini diberikan nilai bobot awal yang sama pada setiap pengujian fungsi aktivasi yaitu 0,05. Pada pengujian ini, diberikan jumlah iterasi sebanyak 100 kali, jumlah *node* pada *hidden layer* sebanyak 2, dan nilai *learning rate* sebesar 0,02. Berdasarkan artikel yang ditulis oleh Surmenok (2017), pada pengujian ini nilai *a* pada fungsi linear diberikan nilai sebesar 1 dan nilai *a* pada fungsi Leaky ReLU sebesar 0,01. Tabel 6.1 menunjukan hasil dari pengujian fungsi aktivasi.

Tabel 6.1 Pengujian Fungsi Aktivasi

			Fungsi Aktivasi					
			Binary Step	*Linear*	*Sigmoid*	*Tanh*	*ReLU*	*Leaky ReLU*
Nilai RMSE	Iterasi ke-*i* (Iterasi = 100, *learning rate* = 0,02, jumlah node *hidden* = 2, dan *a* = 1)	1	0,733	0,316	0,274	0,316	0,316	0,316
		10	0,733	0,305	0,249	0,305	0,305	0,305
		20	0,733	0,268	0,223	0,275	0,268	0,268
		30	0,733	0,164	0,203	0,206	0,164	0,164
		40	0,733	0,071	0,191	0,117	0,071	0,071
		50	0,733	0,059	0,183	0,095	0,059	0,059
		60	0,733	0,061	0,178	0,105	0,061	0,061
		70	0,733	0,064	0,175	0,113	0,064	0,064
		80	0,733	0,067	0,174	0,117	0,067	0,067
		90	0,733	0,070	0,172	0,121	0,070	0,070
		100	0,733	0,074	0,171	0,124	0,074	0,074
Nilai RMSE terkecil			0,733	0,059	0,171	0,095	0,059	0,059



Gambar 6.1 Grafik Pengujian Fungsi Aktivasi

Pada Gambar 6.1 dapat dilihat bahwa fungsi aktivasi *Binary Step* menghasilkan grafik yang konvergen mulai dari awal sampai akhir sehingga bisa disimpulkan bahwa fungsi tersebut tidak cocok digunakan pada permasalahan ini. Satu satunya grafik yang memiliki grafik yang cenderung selalu turun adalah grafik milik fungsi aktivasi sigmoid. Tetapi fungsi aktivasi sigmoid memiliki nilai RMSE terkecil yang cukup besar yaitu 0,171.

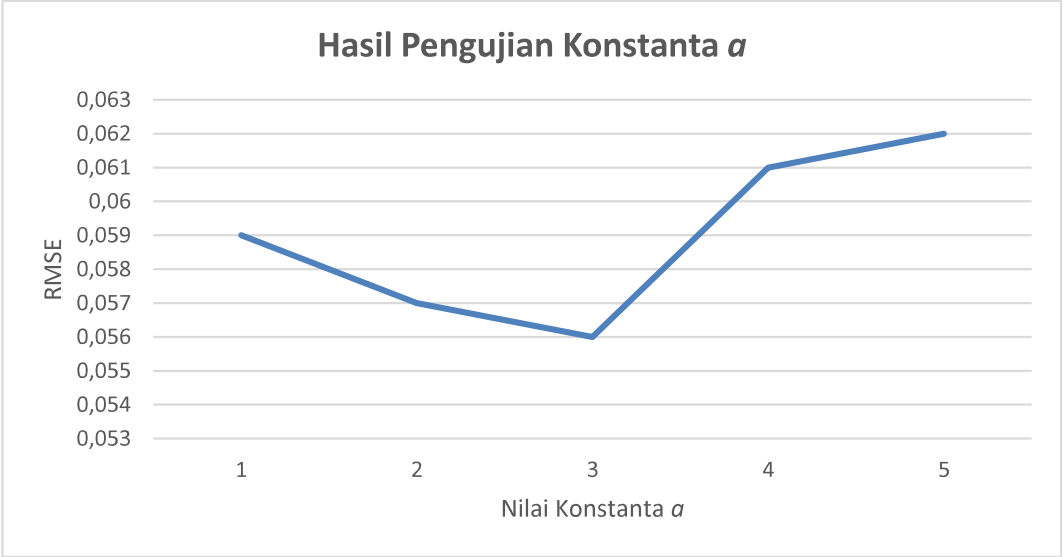
Berdasarkan grafik pengujian fungsi aktivasi yang ditunjukan pada Gambar 6.1 dapat diketahui bahwa nilai RMSE terkecil didapatkan dengan menggunakan fungsi aktivasi *linear, ReLU* dan *Leaky ReLU* dengan nilai 0,059 pada iterasi ke 50. Pada Tabel 6.1 terlihat ketiga fungsi memiliki nilai keluaran yang sama karena pada pada permasalahan ini nilai keluaran *node* yang dihasilkan dari sinyal-sinyal *input* pasti lebih dari nol dan penggunaan nilai a = 1 pada fungsi *linear* sehingga penggunaan fungsi pada ketiga fungsi aktivasi tersebut sama. Karena menghasilkan nilai yang sama pada ketiga fungsi, maka penulis memutuskan menggunakan fungsi aktivasi linear untuk proses berikutnya karena fungsi tersebut masih bisa ada kemungkinan lebih baik dengan menguji nilai *a* yang optimal.

6.1.1 Pengujian Konstanta *a*

Karena pada pengujian fungsi aktivasi didapatkan fungsi linear, maka perlu adanya pengujian tambahan untuk menghitung konstanta *a* yang ada pada persamaan fungsi *linear*. Pengujian ini diharapkan akan membuat fungsi linear dapat menghasilkan nilai yang lebih baik dan dengan waktu yang lebih cepat. Pada pengujian ini nilai *a* yang diuji yaitu 1, 2, 3, 4 dan 5. Seperti pada pengujian fungsi aktivasi, setiap nilai akan diuji satu kali percobaan sehingga nilai bobot awal jaringan akan diinisialisasi dengan nilai yang sama yaitu 0,05.

Tabel 6.2 Pengujian Konstanta a

Jumlah Iterasi	Nilai RMSE				
	Konstanta *a* (Fungsi Aktivasi = *Linear*, *learning rate* = 0,02, dan jumlah node *hidden* = 2)				
	1	2	3	4	5
1	0,316	0,298	0,264	0,215	0,153
10	0,305	0,157	0,056	0,061	0,062
20	0,268	0,057	0,063	0,071	0,078
30	0,164	0,062	0,073	0,084	0,095
40	0,071	0,069	0,083	0,097	0,109
50	0,059	0,075	0,093	0,109	0,122
60	0,061	0,082	0,103	0,120	0,134
70	0,064	0,089	0,111	0,129	0,146
80	0,067	0,095	0,119	0,139	0,157
90	0,070	0,102	0,127	0,148	0,168
100	0,074	0,107	0,134	0,158	0,179
Nilai RMSE Terkecil	0,059	0,057	0,056	0,061	0,062



Gambar 6.2 Hasil Pengujian Konstanta a

Dari Tabel 6.2 menunjukkan bahwa nilai *RMSE* terkecil didapatkan ketika penggunaan konstanta *a* sebesar 3 dengan nilai 0,056 pada iterasi ke 10. Pada sub-bab 2.6 dijelakan bahwa konstanta *a* pada fungsi aktivasi linear lebih

berpengaruh pada perbaikan error. Hal ini mengakibatan penggunaan konstanta *a* yang terlalu besar bisa mengakibatkan perbaikan error yang melewati atas minimum sehingga hasil perbaikan malah semakin buruk. Sedangkan penggunan konstanta *a* yang terlalu kecil menyebabkan perbaikan *error* yang terlalu kecil sehingga proses perbaikan menuju titik optimal yang terlalu lama. Karena nilai RMSE terkecil ditunjukan ketika nilai konstanta *a* sebesar 3 maka pada proses selanjutnya akan digunakan nilai *a* sebesar 3.

6.2 Pengujian Jumlah Iterasi

Pengujian jumlah iterasi dilakukan untuk mengetahui berapa jumlah iterasi yang optimal untuk mendapatkan nilai error terkecil pada penelitian ini. Jumlah iterasi yang digunakan yaitu 10, 20, 30, 40, 50, 60, 70, 80, 90, dan 100. Setiap 1 nilai yang diuji akan dilakukan 5 kali percobaan. Ini dilakukan karena pada penelitian ini, bobot awal jaringan diinisialisasi dengan nilai random, sehingga setiap dilakukan percobaan dengan masukan yang sama, hasilnya belum tentu sama. Setelah dilakukan 5 kali percobaan, kemudian nilai yang dihasilkan akan dicari nilai rata-ratanya. Jumlah iterasi dengan rata-rata *error* terkecil akan diambil untuk dipakai pada proses selanjutnya. Pada pengujian ini digunakan fungsi aktivasi *Linear*, jumlah node hidden layer sebanyak 2, dan nilai learning rate sebesar 0,02. Tabel 6.3 menunjukan hasil dari pengujian jumlah iterasi.

Tabel 6.3 Pengujian Jumlah Iterasi

Jumlah Iterasi	Nilai RMSE					Rata-rata RMSE
	Percobaan ke-*i* (Fungsi Aktivasi = *Linear*, *learning rate* = 0,02, jumlah node *hidden* = 2, dan *a* = 3)					
	1	2	3	4	5	
10	0,055	0,056	0,060	0,058	0,070	0,060
20	0,061	0,060	0,064	0,065	0,068	0,063
30	0,071	0,069	0,073	0,075	0,078	0,073
40	0,082	0,080	0,084	0,085	0,088	0,083
50	0,091	0,090	0,093	0,095	0,097	0,093
60	0,101	0,099	0,102	0,103	0,106	0,102
70	0,109	0,108	0,111	0,112	0,114	0,111
80	0,117	0,116	0,119	0,120	0,122	0,119
90	0,125	0,124	0,126	0,127	0,130	0,126
100	0,132	0,131	0,134	0,134	0,137	0,134

Gambar 6.3 Grafik Pengujian Jumlah Iterasi

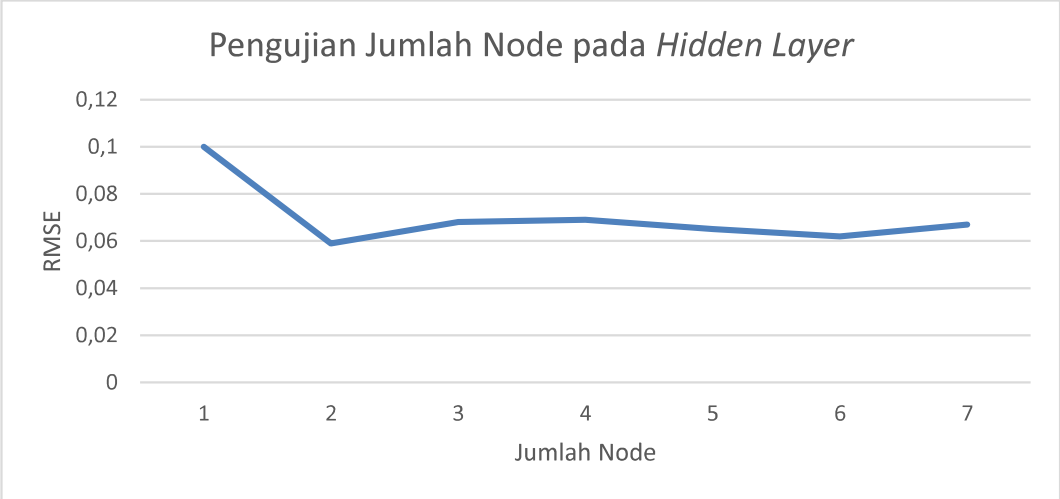
Berdasarkan grafik pengujian jumlah iterasi yang ditunjukan pada Tabel 6.3 dan Gambar 6.3 dapat dilihat bahwa nilai rata-rata RMSE terkecil didapatkan ketika iterasi = 10 yaitu sebesar 0,06. Pada Gambar 6.3 terlihat grafik yang konsisten menanjak setelah iterasi ke 10. Hal ini dikarenakan pada iterasi ke 10, nilai *error* sudah mencapai titik optimal sehingga perbaikan pada iterasi berikutnya menyebabkan *error* yang lebih buruk. Menurut Sharma (2017), jika suatu parameter terlalu besar maka akan menyebabkan *cost* atau *error* semakin buruk. Sehingga pada proses pengujian berikutnya akan digunakan iterasi sebanyak 10.

6.3 Pengujian Jumlah Node Pada *Hidden Layer*

Pada penelitian ini digunkan satu hidden layer. Hal ini dikarenakan dalam banyak masalah, neural network hanya membutuhkan satu *hidden layer*, sedangkan dua *hidden layer* diperlukan untuk pemodelan data dengan diskontinuitas seperti pola gelombang gigi gergaji namun memiliki resiko tercapainya *local optimum* sehingga tidak ada alasan yang untuk memakai lebih dari dua *hidden layer* (Panchal, et al., 2011). Pengujian jumlah node pada *hidden* layer dilakukan untuk mengetahui berapa jumlah node yang optimal pada *hidden* layer untuk mendapatkan nilai error terkecil pada penelitian ini. Menurut Heaton (2005) jumlah neuron hidden layer yang baik adalah 2/3 dari jumlah *input* ditambah dengan jumlah *output*. Karena jumlah *input* adalah 4 dan jumlah *output* adalah 4 maka menurut Heaton jumlah *node* pada *hidden layer* adalah antara 6 sampai 7 buah node. Sehingga jumlah node yang diuji yaitu 1, 2, 3, 4, 5, 6, dan 7. Setiap 1 nilai yang diuji akan dilakukan 5 kali percobaan. Ini dilakukan karena pada penelitian ini, bobot awal jaringan diinisialisasi dengan nilai random, sehingga setiap dilakukan percobaan dengan masukkan yang sama, hasilnya belum tentu sama. Setelah dilakukan 5 kali percobaan, kemudian nilai yang dihasilkan akan dicari nilai rata-ratanya. Jumlah node dengan rata-rata error terkecil akan diambil untuk dipakai pada proses selanjutnya. Tabel 6.4 menunjukan hasil dari pengujian jumlah *node* pada *hidden layer*.

Tabel 6.4 Pengujian Jumlah Node Pada *Hidden Layer*

Jumlah Node	Nilai RMSE					Rata-rata RMSE
	Percobaan ke-*i* (Fungsi Aktivasi = *Linear*, *learning rate* = 0,02, jumlah iterasi = 10, dan *a* = 3)					
	1	2	3	4	5	
1	0,108	0,069	0,084	0,089	0,148	0,100
2	0,055	0,057	0,065	0,058	0,062	0,059
3	0,069	0,063	0,068	0,066	0,075	0,068
4	0,063	0,070	0,071	0,076	0,065	0,069
5	0,060	0,062	0,063	0,073	0,067	0,065
6	0,063	0,060	0,062	0,061	0,065	0,062
7	0,072	0,089	0,059	0,057	0,060	0,067



Gambar 6.4 Grafik Pengujian Jumlah Node pada *Hidden Layer*

Berdasarkan grafik pengujian jumlah node pada hidden layer yang ditunjukkan pada Gambar 6.4 dapat dilihat bahwa mulai dengan jumlah *node* sebanyak 2 menghasilkan grafik yang cenderung naik turun dan pada jumlah *node* sebanyak 2 memiliki nilai rata-rata *RMSE* yang terkecil dengan nilai error 0,059. Sedangkan jumlah node sebanyak 1 memiliki nilai *RMSE* yang berbeda dan jauh lebih besar dari yang lainnya dikarenakan kurangnya variasi. Selain itu, jika jumlah node kurang dari 2 akan terjadi kondisi *underfitting*, yaitu kondisi dimana jaringan tidak dapat dapat memepelajari pola yang ada dengan baik yang mengakibatkan kurang baiknya hasil peramalan. Tetapi ketika jumlah node terlalu banyak maka akan terjadi kondisi *overfitting,* yaitu kondisi dimana jaringan kehilangan kemampuan dalam generalisasi dan mempelajari pola-pola

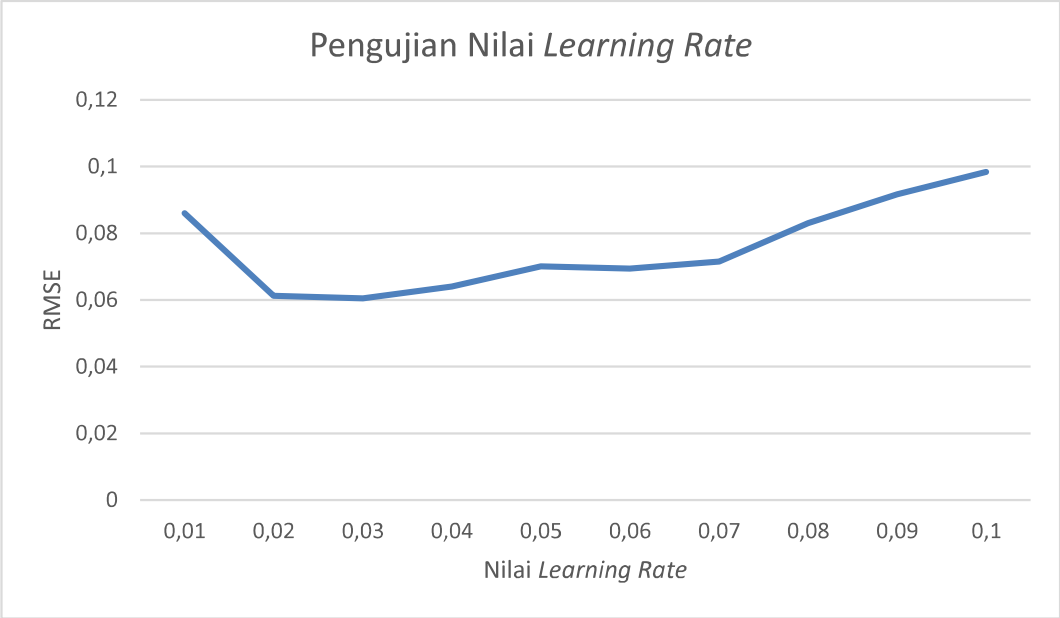
yang diuji. Sehingga untuk proses selanjutnya akan digunakan jumlah node pada hidden layer sebanyak 2.

6.4 Pengujian Nilai Learning Rate

Pengujian nilai *learning rate* dilakukan untuk mengetahui berapa nilai learning rate yang optimal untuk mendapatkan nilai error terkecil pada penelitian ini. Menurut Surmenok (2017) jika nilai *learning rate* terlalu rendah, maka pelatihan lebih dapat diandalkan, namun optimasi akan memakan banyak waktu karena langkah-langkah menuju *error* yang sangat lambat, namun Jika nilai *learning rate* terlalu tinggi, maka pelatihan mungkin tidak menyatu atau bahkan menyimpang, perubahan bobot jaringan bisa begitu besar sehingga pengoptimal melampaui batas minimum dan membuat nilai *error* semakin buruk. Seperti penelitian yang ditulis Smith (2017) menerangkan bahwa teknik yang ampuh untuk memilih rentang *learning rate* untuk jaringan syaraf tiruan caranya adalah dengan melatih jaringan mulai dari tingkat belajar rendah dan meningkat secara terus menerus, sehingga nilai *learning rate* yang digunakan yaitu mulai 0,01; 0,02; 0,03; 0,04; 0,05; 0,06; 0,07; 0,08; 0,09; dan 0,1. Setiap 1 nilai yang diuji akan dilakukan 5 kali percobaan. Ini dilakukan karena pada penelitian ini, bobot awal jaringan diinisialisasi dengan nilai random, sehingga setiap dilakukan percobaan dengan masukkan yang sama, hasilnya belum tentu sama. Setelah dilakukan 5 kali percobaan, kemudian nilai yang dihasilkan akan dicari nilai rata-ratanya. Tabel 6.5 menunjukan hasil dari 6.4 pengujian nilai *Learning Rate*.

Tabel 6.5 Pengujian Nilai *Learning Rate*

Nilai Learning Rate	Nilai RMSE					Rata-rata RMSE
	Percobaan ke-*i* (Fungsi Aktivasi = *Linear*, jumlah iterasi = 10, jumlah Node Hidden = 2, dan *a* = 3)					
	1	2	3	4	5	
0,01	0,0925	0,0689	0,0975	0,0873	0,0838	0,0860
0,02	0,0597	0,0674	0,0603	0,0599	0,0586	0,0612
0,03	0,0641	0,0630	0,0623	0,0591	0,0599	0,0617
0,04	0,0650	0,0686	0,0604	0,0636	0,0626	0,0640
0,05	0,0679	0,0712	0,0696	0,0743	0,0675	0,0701
0,06	0,0709	0,0675	0,0692	0,0700	0,0692	0,0694
0,07	0,0722	0,0704	0,0723	0,0732	0,0693	0,0715
0,08	0,0820	0,0842	0,0849	0,0813	0,0824	0,0830
0,09	0,0915	0,0910	0,0930	0,0896	0,0932	0,0917
0,1	0,1019	0,0974	0,0930	0,0999	0,0999	0,0984



Gambar 6.5 Grafik Pengujian Nilai *Learning Rate*

Berdasarkan grafik pengujian nilai *learning rate* yang ditunjukkan pada Gambar 6.5 dapat dilihat bahwa *learning rate* dengan nilai 0,02 memiliki nilai rata-rata RMSE yang paling kecil dengan nilai 0, 0612. Menurut Sumernok seperti yang sudah dijelaskan sebelumnya, hal ini disebabkan penggunaan nilai *learning rate* diatas 0,02 mengalami kenaikan *RMSE* karena semakin tinggi nilai *learning rate* mengakibatkan pengoptimalan atau perbaikan bobot melewati batas minimum perbaikan sehingga menjadikan *error* atau jarak dengan target semakin besar, sedangkan pada nilai *learning rate* dengan nilai dibawah 0,02 mengalami proses perbaikan *error* yang terlalu kecil sehingga proses perbaikan terlalu lama dan hasilnya juga kurang baik. Sehingga pada penelitian ini didapatkan nilai *learning rate* paling optimal sebesar 0,02.