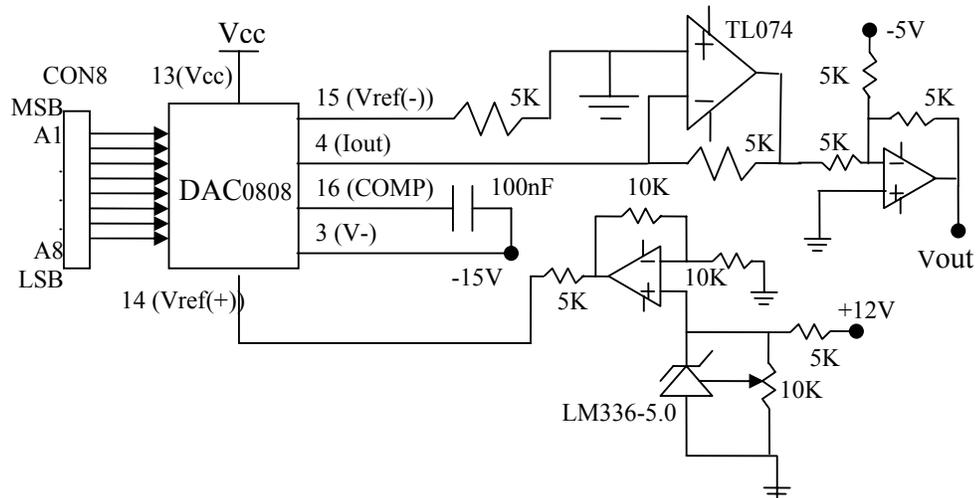


4. PENGUJIAN SISTEM

Pengujian alat dilakukan perbagian, tujuannya agar setiap rangkaian dapat diketahui apakah bekerja dengan baik dan pemeriksaan rangkaian dapat dilakukan dengan mudah. Bila satu bagian telah bekerja dengan baik, tahap berikutnya akan diuji, sampai tahap terakhir keseluruhan rangkaian. Dari setiap pengujian dapat diperoleh data-data.

4.1. Pengujian Rangkaian DAC0808

Pengujian terhadap rangkaian DAC0808 ini dimaksudkan untuk mengetahui besarnya tegangan keluaran analog jika diberikan suatu masukan dalam bentuk data binari. Pengujian dilakukan dengan cara memasukkan kombinasi bilangan binari pada masukan DAC dan kemudian mengukur tegangan yang dihasilkan dengan menggunakan multimeter digital. Karena data masukan DAC0808 ini adalah delapan bit, maka diperoleh sebanyak $2^8 = 256$ data kombinasi binari, yaitu dari 0 (00000000₂) sampai dengan 255 (11111111₂).



Gambar 4.1. Rangkaian DAC

Dengan bantuan perangkat lunak, dapat dilakukan pengiriman data binari secara paralel delapan bit melalui LPT1 (dari PC) menuju rangkaian DAC.

Kemudian besarnya tagangan analog yang dihasilkan dibandingkan dengan besarnya tegangan analog yang diperoleh dari perhitungan teori. Dari perbandingan ini dapat diketahui besar faktor kesalahan yang terjadi pada rangkaian ini.

Tegangan referensi yang diberikan adalah +10 Volt, kemudian digunakan *op-amp* sebagai penjumlah dan pengurang untuk dapat menghasilkan tegangan -5 Volt hingga +5 Volt. Kemudian daerah ini dapat digunakan sebagai masukan pembanding untuk mengatur *duty cycle* rangkaian PWM.

Hasil dari pengujian DAC0808 dapat dilihat pada table 4.1.

Tabel 4.1. Hasil Pengujian DAC0808

DAC Input	Vout Perhitungan	Vout Pengujian (dari gambar 4.1)	Verror
0	5	5,05	0.05
10	4.60938	4,68	0.07062
20	4.21875	4,26	0.04125
30	3.82813	3,89	0.06187
40	3.4375	3,50	0.0625
50	3.04688	3,11	0.06312
60	2.65625	2,71	0.05375
70	2.26563	2,32	0.05437
80	1.875	1,93	0.055
90	1.48438	1,54	0.05562
100	1.09375	1,14	0.04625
110	0.70313	0,75	0.04687
120	0.3125	0,36	0.0475
128	0	0,05	0.05
130	-0.078125	-0,02	0.058125
140	-0.46875	-0,41	0.05875
150	-0.859375	-0,80	0.059375
160	-1.25	-1,19	0.06
170	-1.640625	-1,59	0.050625
180	-2.03125	-2,03	0.00125
190	-2.421875	-2,37	0.051875
200	-2.8125	-2,77	0.0425
210	-3.203125	-3,16	0.043125
220	-3.59375	-3,55	0.04375
230	-3.984375	-3,95	0.034375
240	-4.375	-4,34	0.035
250	-4.765625	-4,74	0.025625
255	-4.960938	-4,94	0.020938

Untuk mendapatkan V_{out} perhitungan, yaitu:

Range masukan DAC adalah 256 (0 hingga 255)

Range keluaran DAC adalah 10 Volt (-5 hingga +5 Volt)

$$V_{out \text{ perhitungan}} = - \left[\left(10 \times \left(\frac{A1}{2} + \frac{A2}{4} + \dots + \frac{A8}{256} \right) \right) - 5 \right] \quad (4.1)$$

Dari hasil pengujian diperoleh:

$$V_{error \text{ rata rata}} = \frac{\sum V_{error}}{n} \quad (4.2)$$

$$V_{error \text{ rata rata}} = \frac{1.344033}{28} = 0.048001$$

$$\%Full \text{ Scale Error} = \frac{V_{Error}}{Full \text{ Scale Voltage}} \times 100\% \quad (4.3)$$

$$\%Full \text{ Scale Error} = \frac{0.048001}{10} \times 100\% = 0,48001\%$$

4.2. Pengujian Rangkaian Tachometer

Pada tahap ini dilakukan pengujian terhadap *tachometer* untuk menentukan kecepatan putaran dari motor DC. Pengujian terhadap rangkaian ini dimaksudkan agar pembacaan nilai kecepatan putaran motor sesuai dengan kecepatan putaran motor yang terjadi, artinya untuk mengecek apakah rangkaian *tachometer* dapat bekerja dengan baik.

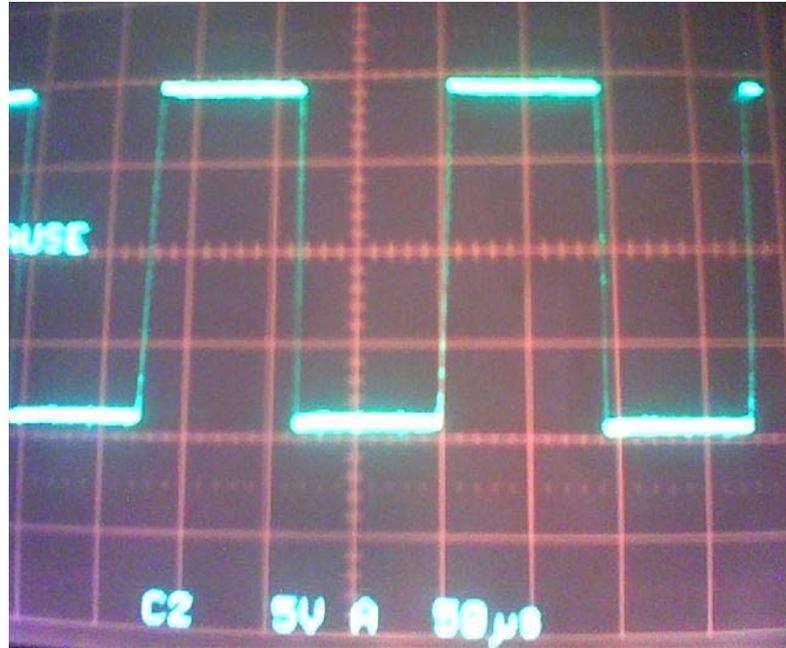
Pengujian dilakukan dengan cara memberi masukan (*input*) pada DAC0808, yaitu dengan mengubah-ubah kecepatan putaran motor DC dengan cara memberikan *input* yang bervariasi, dan hasil perhitungan kecepatan putaran oleh *microcontroller* AT89C2051 (dalam satuan Rps) dibandingkan dengan hasil pengukuran dengan menggunakan *tachometer* digital (dalam satuan Rpm). Karena berbeda satuan, maka disamakan bahwa perbandingannya dilakukan dalam satuan Rpm.

Selain melakukan penyetelan *tachometer* pada tabel 4.2 berikut ini juga disertakan pengujian PWM (*Pulse Width Modulation*), dimana pengujian dilakukan dengan cara melihat perubahan keluaran dari rangkaian PWM pada layar osiloskop.

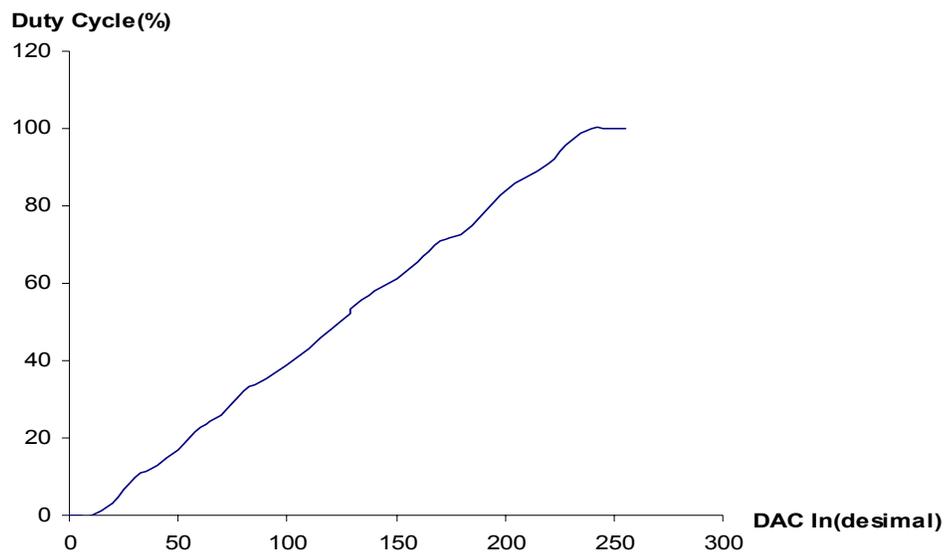
Tabel 4.2. Hasil Pengujian Rangkaian Tachometer

DAC In (decimal)	Duty Cycle PWM (%)	Kecepatan dari Tachometer(RPM)	Kecepatan dari AT89C2051 (RPM)
0	0	0	0
10	0	0	0
20	3,13	730	720
30	9,68	738	720
40	12,9	790	780
50	16,67	1384	1380
60	22,58	1458	1440
70	25,8	1500	1500
80	32,26	1760	1740
90	35,48	1866	1860
100	38,71	1989	1980
110	43,33	2100	2100
120	48,39	2176	2160
128	51,61	2224	2220
130	53,92	2280	2280
140	58,06	2341	2340
150	61,29	2431	2460
160	65,63	2489	2520
170	70,97	2559	2580
180	72,73	2626	2640
190	78,13	2684	2700
200	83,87	2746	2760
210	87,5	2788	2820
220	90,91	2800	2820
230	96,88	2822	2880
240	100	2856	2880
250	100	2886	2940
255	100	2924	2940

Dari hasil pengujian terlihat adanya sedikit perbedaan antara hasil pengukuran kecepatan putaran antara *tachometer* digital dengan hasil pengukuran yang dihasilkan oleh AT89C2051. Perbedaan ini disebabkan oleh adanya perbedaan waktu sampling yang digunakan antara *tachometer* digital dengan AT89C2051.

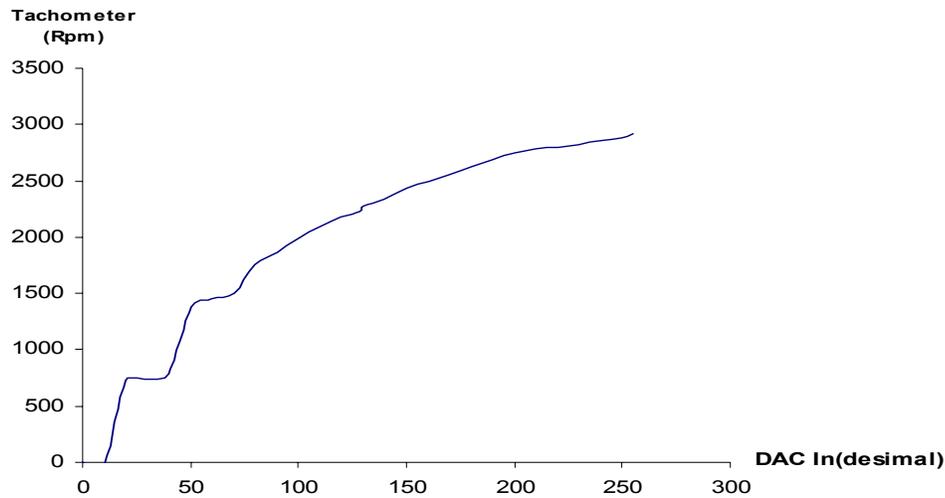


Gambar 4.2. Duty Cycle PWM Pada Input DAC = 128(Desimal)



Gambar 4.3. Grafik Input DAC Terhadap Duty Cycle

Dari grafik di atas, dapat dilihat bahwa *duty cycle* yang dihasilkan dengan melalui *input* DAC adalah membentuk sebuah garis yang dapat dikatakan linier, atau besarnya nilai *input* dari DAC yang diberikan sebanding dengan besarnya *duty cycle* yang dihasilkan.



Gambar 4.4. Grafik Input DAC Terhadap Kecepatan dari Tachometer

Dari Grafik tersebut dapat diketahui bahwa kecepatan putaran dari motor tidak linier dengan *duty cycle* dan juga dengan *input* yang diberikan.

4.3. Pengujian Dengan Sistem Kendali

Pengujian dengan sistem kendali ini dilakukan melalui dua tahap. Tahap pertama, yaitu pelatihan yang dilakukan selama proses *training* berjalan, sedangkan yang ke dua adalah pelatihan dengan menjalankan sistem secara keseluruhan dengan tujuan untuk mendapatkan respon *plant* secara keseluruhan apabila di kontrol oleh suatu sistem kendali dengan metode tertentu.

Sistem kontrol yang dipilih di sini adalah sistem kendali berbasis Jaringan Saraf Tiruan, yaitu dengan metode *training BackPropagation*.

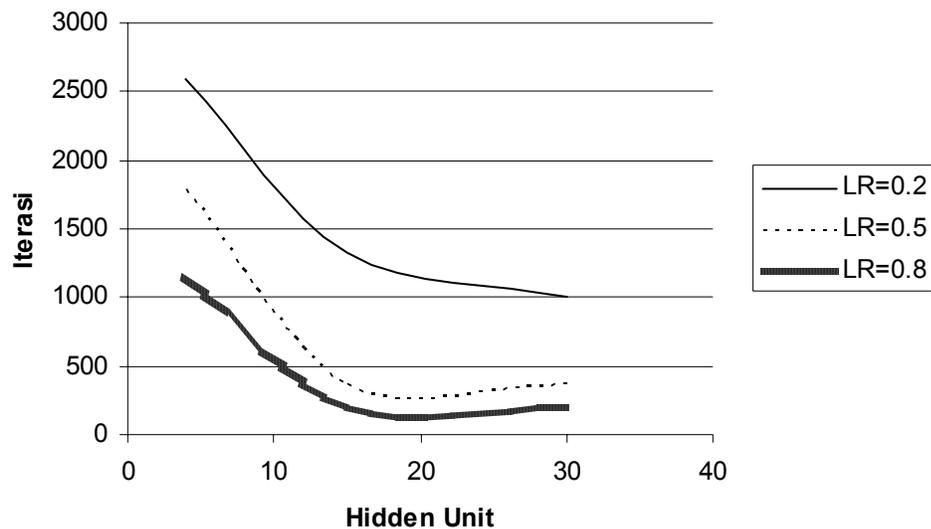
Pelatihan tahap pertama ini adalah pelatihan yang dilakukan melalui proses *training*, yang mana pengujian dimaksudkan untuk melihat kecenderungan perubahan iterasi terhadap perubahan *learning rate* dan juga terhadap perubahan banyaknya jumlah *hidden unit*. Pada pengujian ini ditentukan besarnya nilai *learning rate* adalah 0,2; 0,5; dan 0,8, dan banyaknya *hidden unit* adalah 4, 15, dan 30.

Dari proses pelatihan yang telah dilakukan dengan data *training* dengan nilai *output* antara -70 s/d 70 didapat hasil seperti yang terlihat pada tabel di berikut ini.

Tabel 4.3. Tabel Perubahan Nilai Learning Rate Dan Jumlah Hidden Unit Terhadap Banyaknya Iterasi

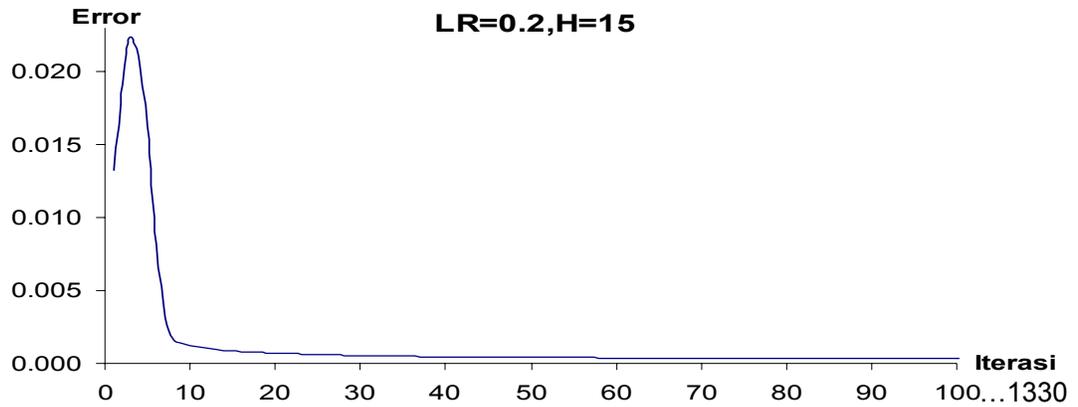
Learning Rate	Hidden Unit	Iterasi
0,2	4	2589
	15	1330
	30	1012
0,5	4	1790
	15	368
	30	366
0,8	4	1153
	15	211
	30	197

Secara rinci dari proses *training* dengan nilai-nilai *error* pada setiap iterasi dapat dilihat pada CD dengan nama *file* 'hasil *training*'. Dari pengujian yang dilakukan dapat diketahui bahwa semakin banyak jumlah *hidden unit*, maka jumlah iterasi yang diperlukan untuk mencapai kondisi *error* yang diinginkan semakin sedikit atau kecil. Begitu juga dengan perubahan *learning rate*, di mana semakin besar nilai *learning rate*, maka semakin sedikit iterasi yang dilakukan.

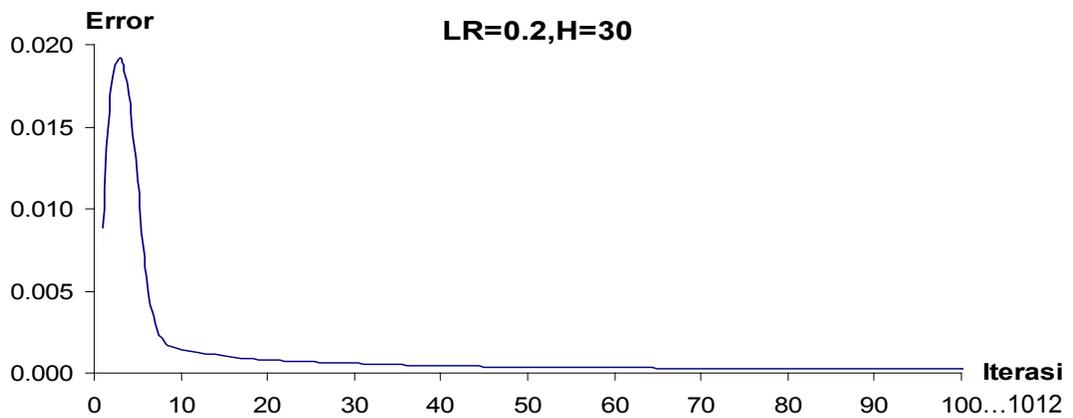


Gambar 4.5. Grafik Perubahan Nilai Learning Rate Dan Jumlah Hidden Unit terhadap Banyaknya Iterasi

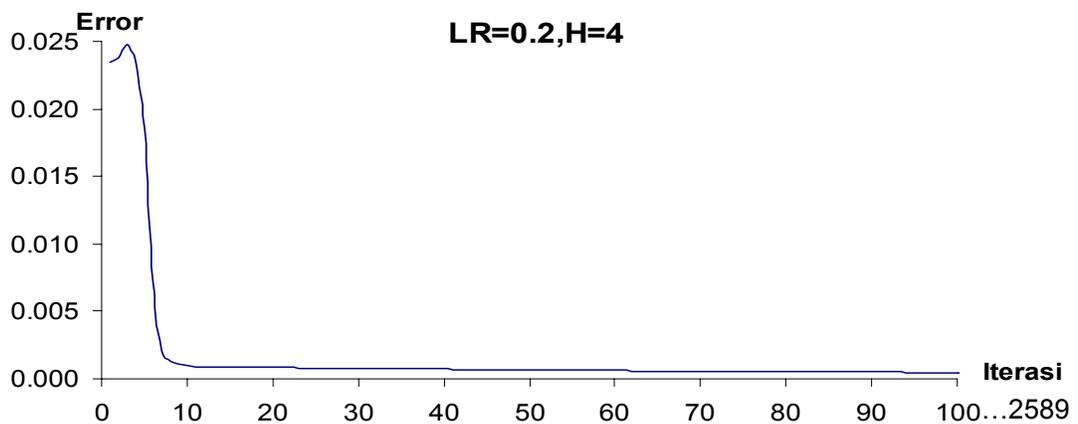
Berikut adalah grafik jumlah iterasi terhadap nilai *error* yang dihasilkan pada masing-masing nilai *learning rate*(LR) dan jumlah *hidden unit*(H).



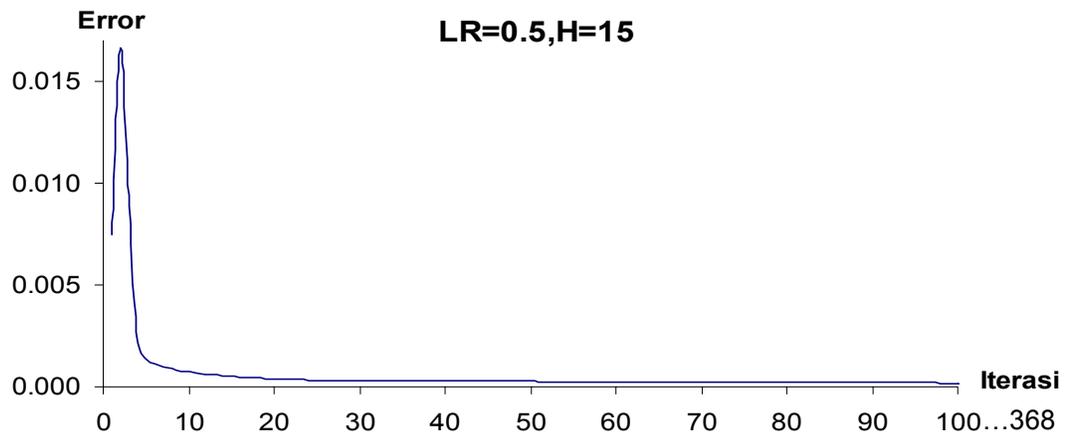
Gambar 4.6. Grafik Pencapaian Error Pada LR=0,2 Dan H=15



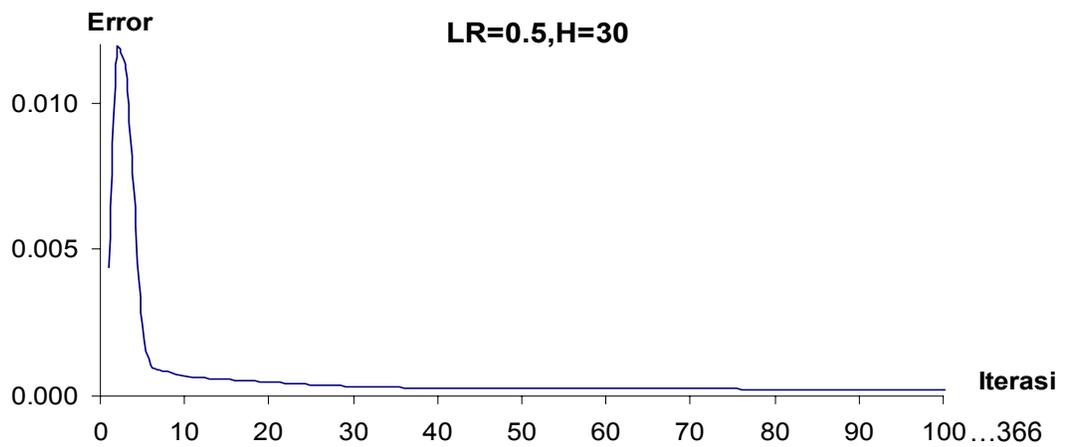
Gambar 4.7. Grafik Pencapaian Error Pada LR=0,2 Dan H=30



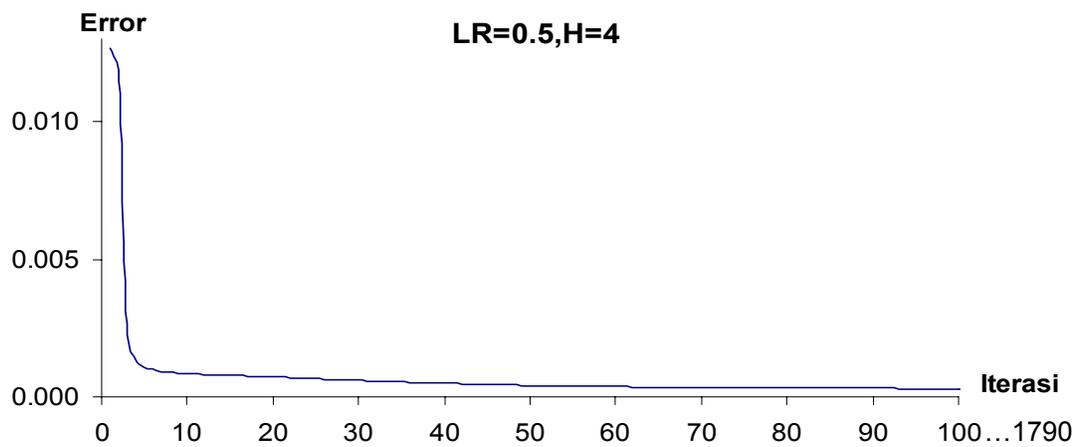
Gambar 4.8. Grafik Pencapaian Error Pada LR=0,2 Dan H=4



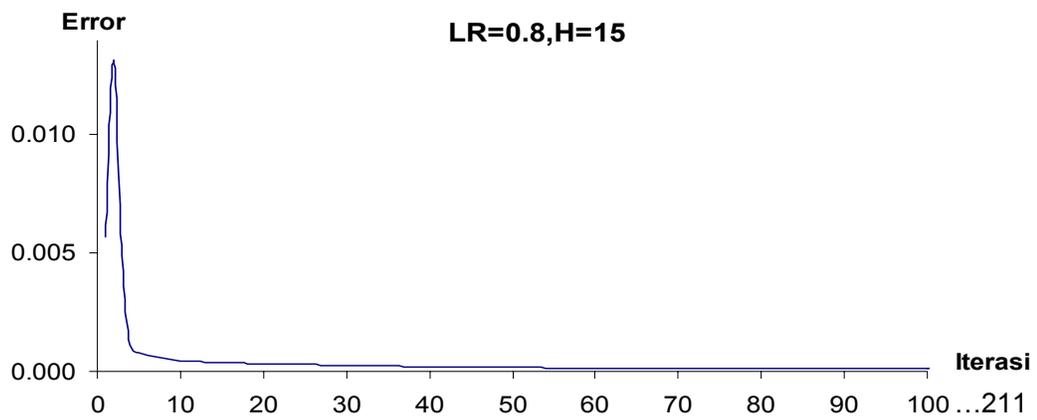
Gambar 4.9. Grafik Pencapaian Error Pada LR=0,5 Dan H=15



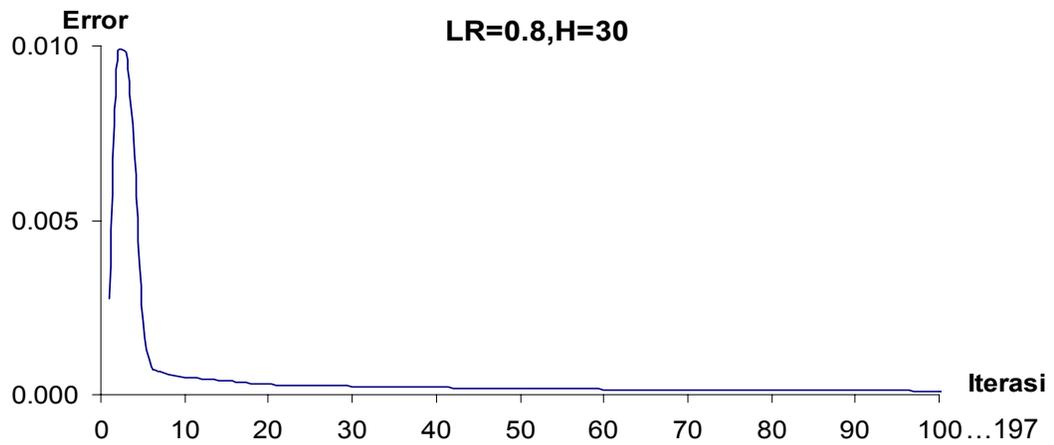
Gambar 4.10. Grafik Pencapaian Error Pada LR=0,5 Dan H=30



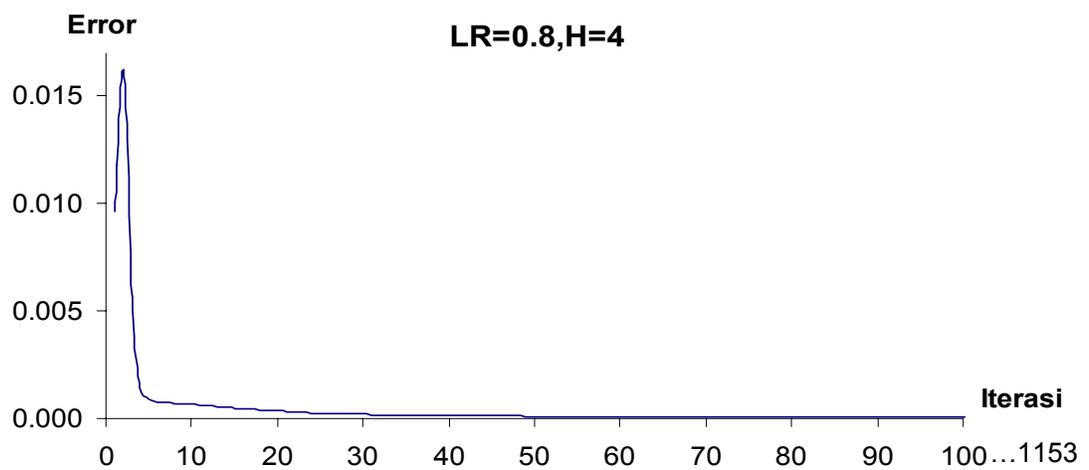
Gambar 4.11. Grafik Pencapaian Error Pada LR=0,5 Dan H=4



Gambar 4.12. Grafik Pencapaian Error Pada LR=0,8 Dan H=15



Gambar 4.13. Grafik Pencapaian Error Pada LR=0,8 Dan H=30

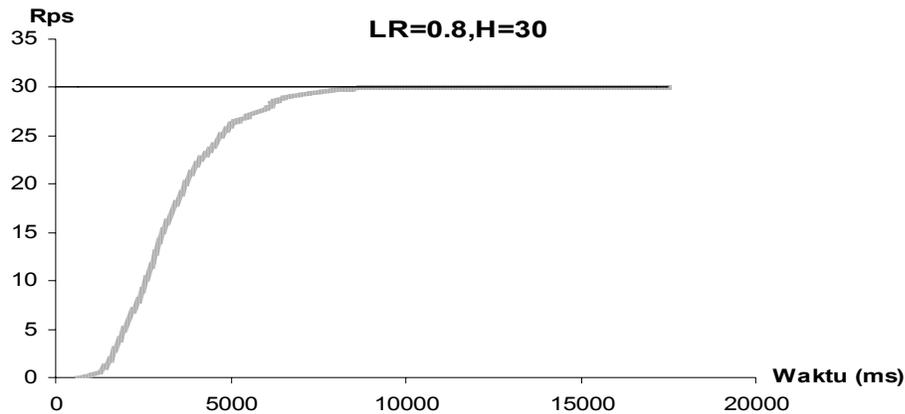


Gambar 4.14. Grafik Pencapaian Error Pada LR=0,8 Dan H=4

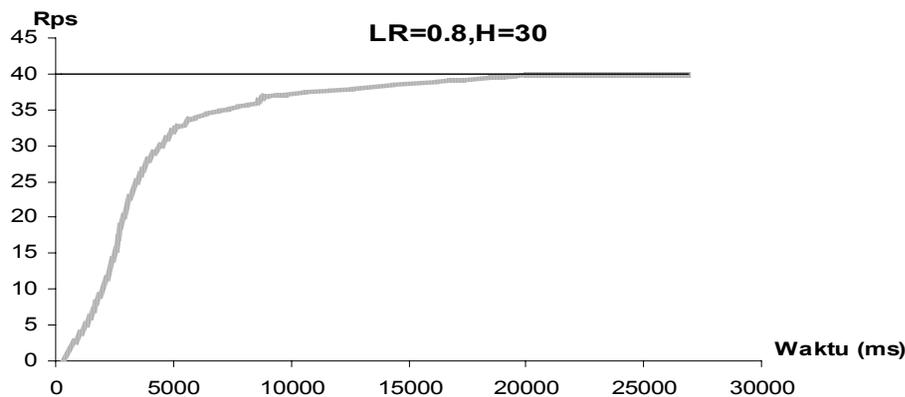
Dari percobaan tersebut ditentukan jumlah *hidden unit* yang efektif dan efisien adalah 30 karena pada saat itu jumlah iterasi menurun secara drastis. Dan demikian juga dalam menentukan besarnya konstanta belajar (LR) dilihat pada kecenderungan bahwa semakin besar nilai LR, maka jumlah iterasi cenderung menjadi lebih kecil atau sedikit, sehingga besarnya nilai konstanta efektif yang di ambil adalah 0,8.

Pelatihan selanjutnya adalah pelatihan dengan menjalankan modul untuk mengendalikan kecepatan putar motor. *Setting point* (kecepatan) dari pelatihan kali ini ditentukan sebesar 30 dan 40 Rps.

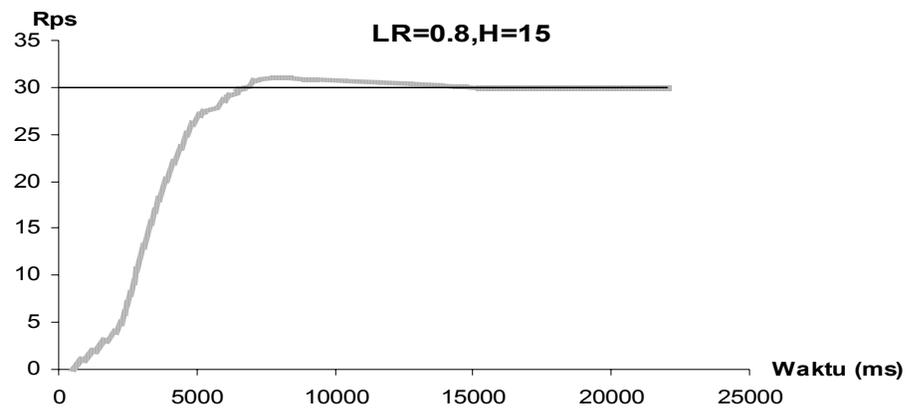
Dari pelatihan didapat hasil seperti terlihat pada grafik berikut ini.



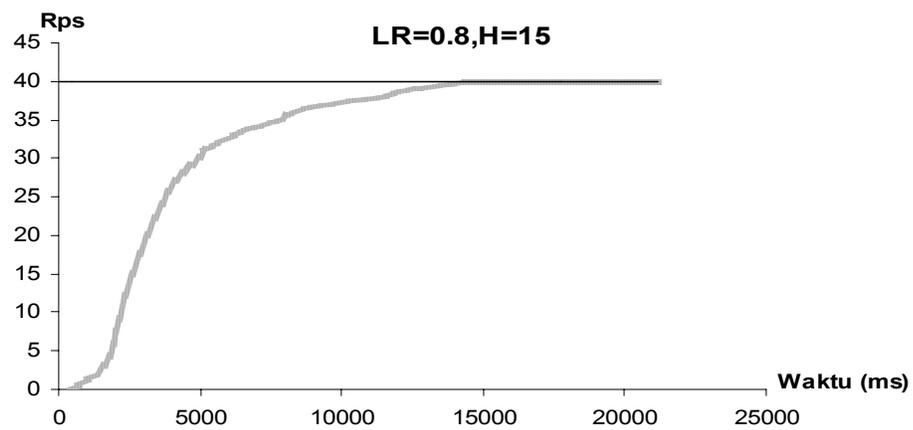
Gambar 4.15. Grafik Pengendalian Kecepatan Pada 30 Rps (LR=0.8,H=30)



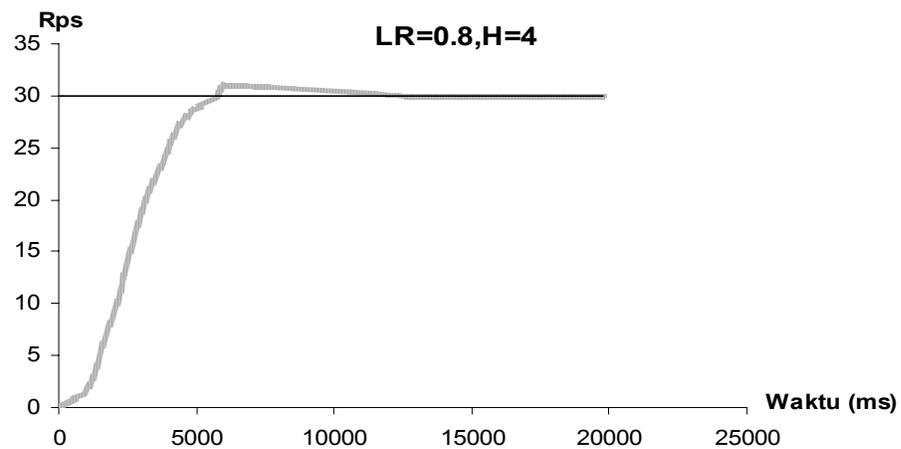
Gambar 4.16. Grafik Pengendalian Kecepatan Pada 40 Rps (LR=0.8,H=30)



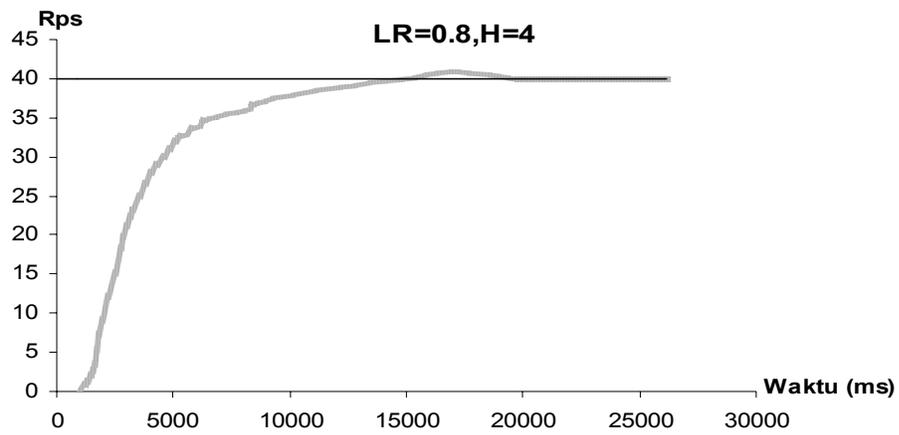
Gambar 4.17. Grafik Pengendalian Kecepatan Pada 30 Rps (LR=0.8,H=15)



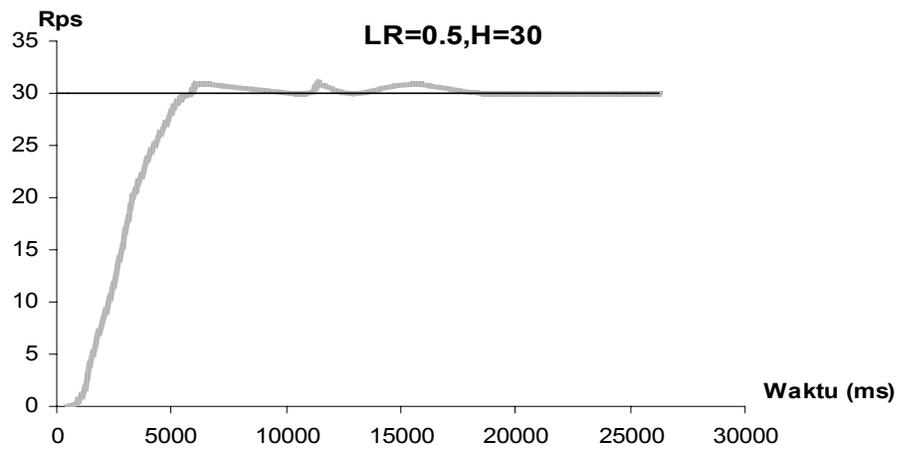
Gambar 4.18. Grafik Pengendalian Kecepatan Pada 40 Rps (LR=0.8,H=15)



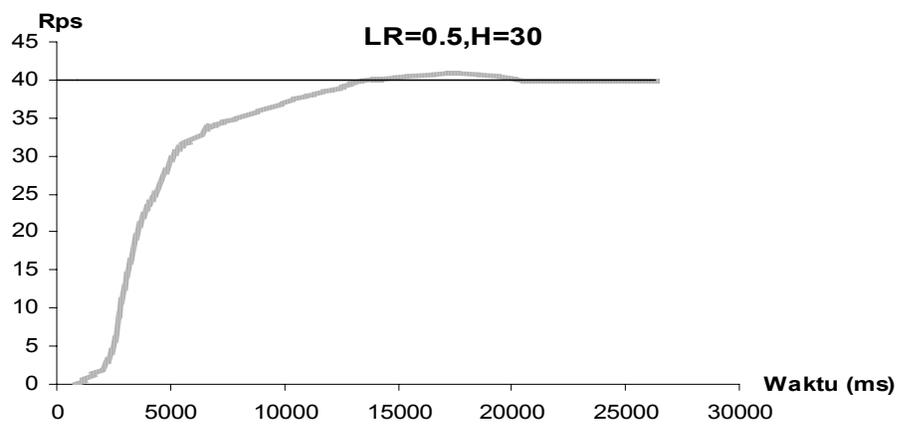
Gambar 4.19. Grafik Pengendalian Kecepatan Pada 30 Rps (LR=0.8,H=4)



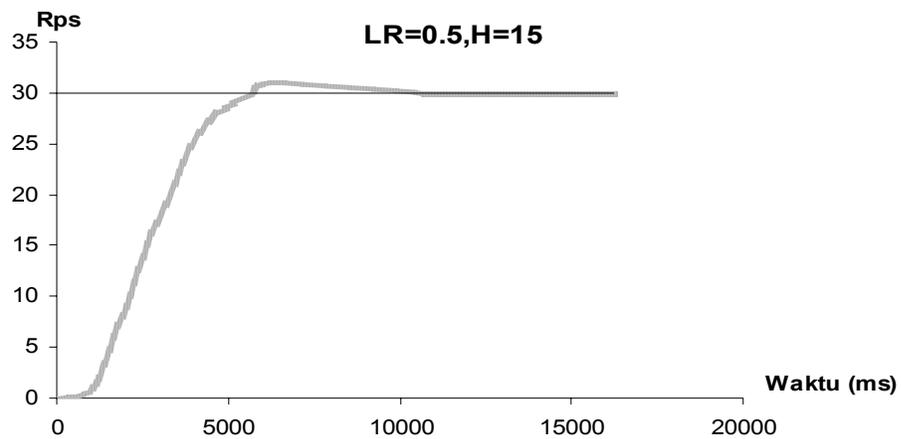
Gambar 4.20. Grafik Pengendalian Kecepatan Pada 40 Rps (LR=0.8,H=4)



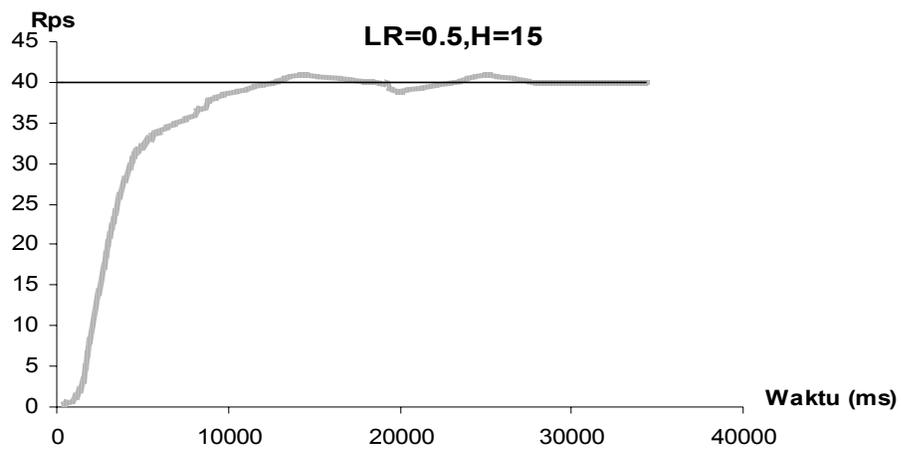
Gambar 4.21. Grafik Pengendalian Kecepatan Pada 30 Rps (LR=0.5,H=30)



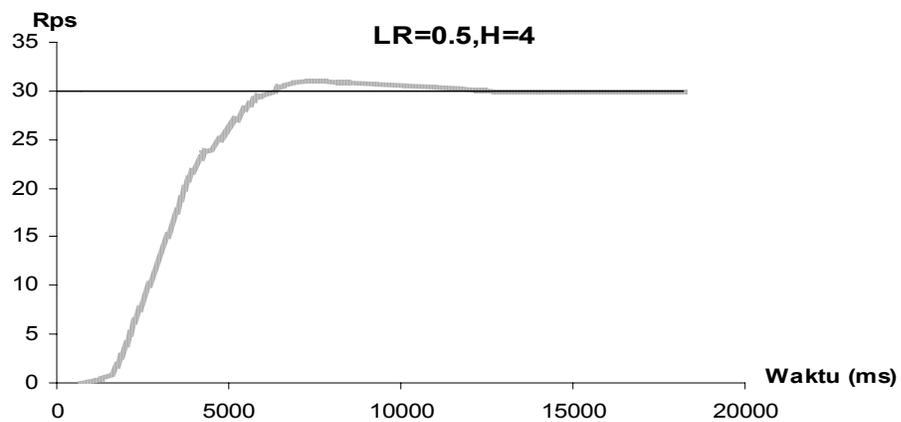
Gambar 4.22. Grafik Pengendalian Kecepatan Pada 40 Rps (LR=0.5,H=30)



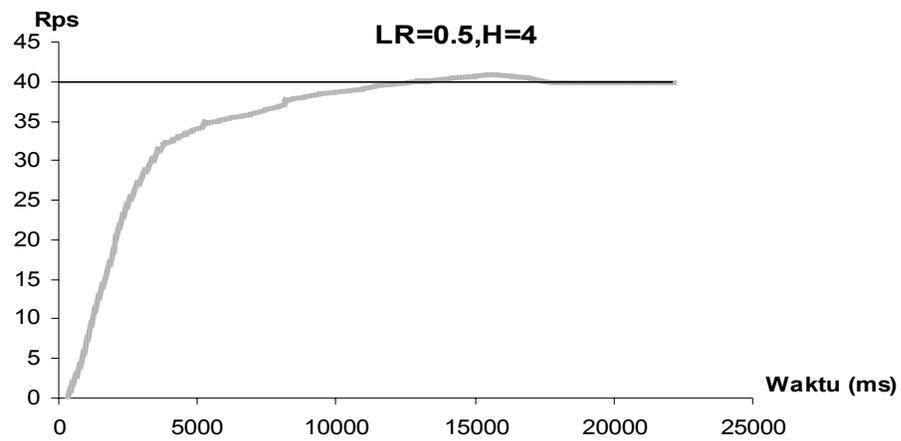
Gambar 4.23. Grafik Pengendalian Kecepatan Pada 30 Rps (LR=0.5,H=15)



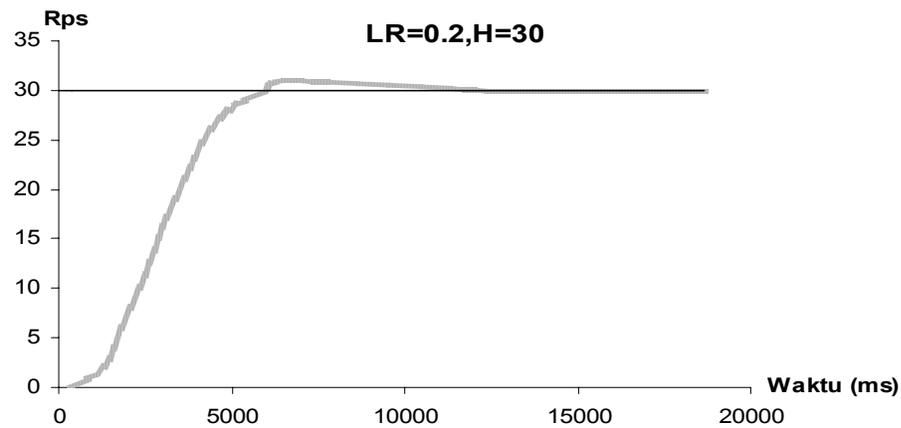
Gambar 4.24. Grafik Pengendalian Kecepatan Pada 40 Rps (LR=0.5,H=15)



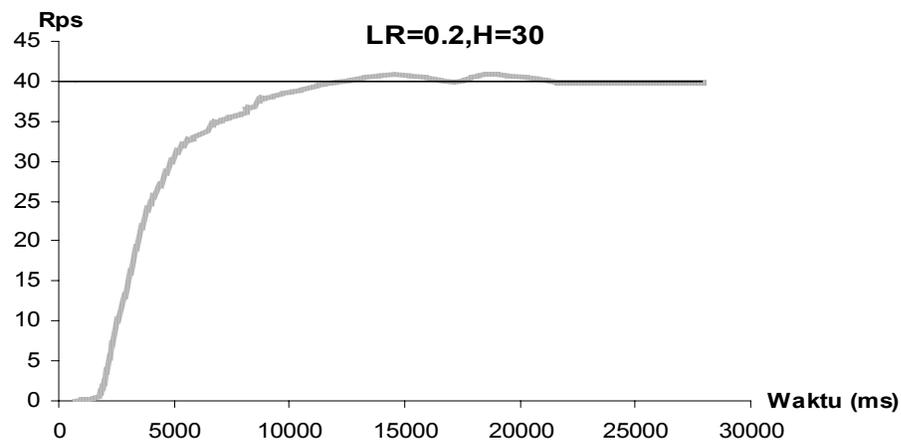
Gambar 4.25. Grafik Pengendalian Kecepatan Pada 30 Rps (LR=0.5,H=4)



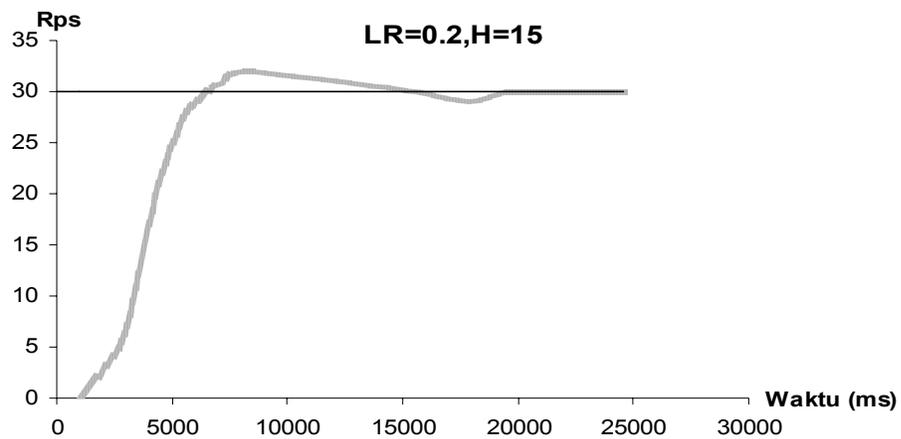
Gambar 4.26. Grafik Pengendalian Kecepatan Pada 40 Rps (LR=0.5,H=4)



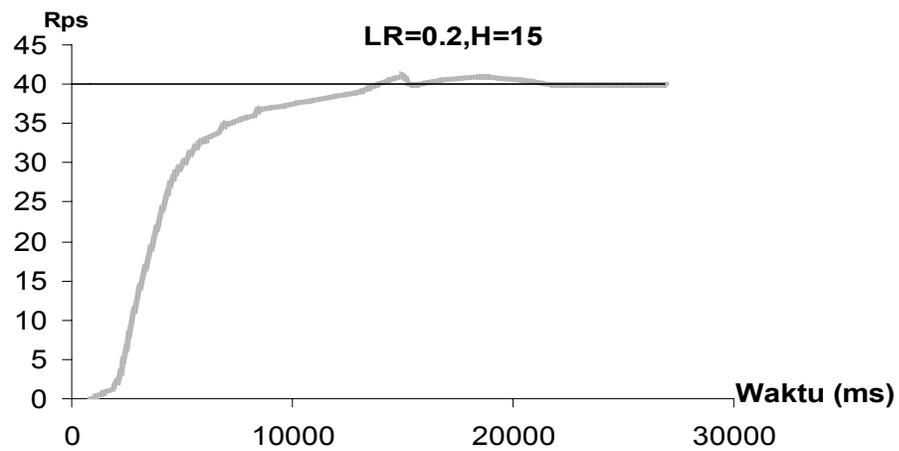
Gambar 4.27. Grafik Pengendalian Kecepatan Pada 30 Rps (LR=0.2,H=30)



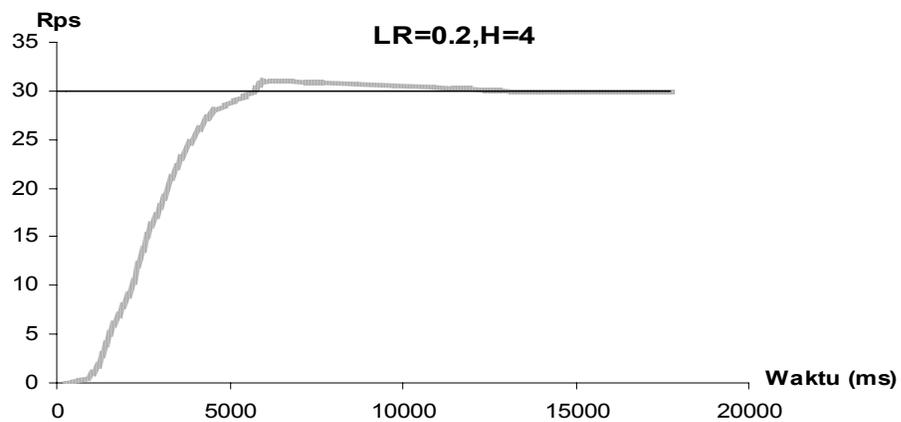
Gambar 4.28. Grafik Pengendalian Kecepatan Pada 40 Rps (LR=0.2,H=30)



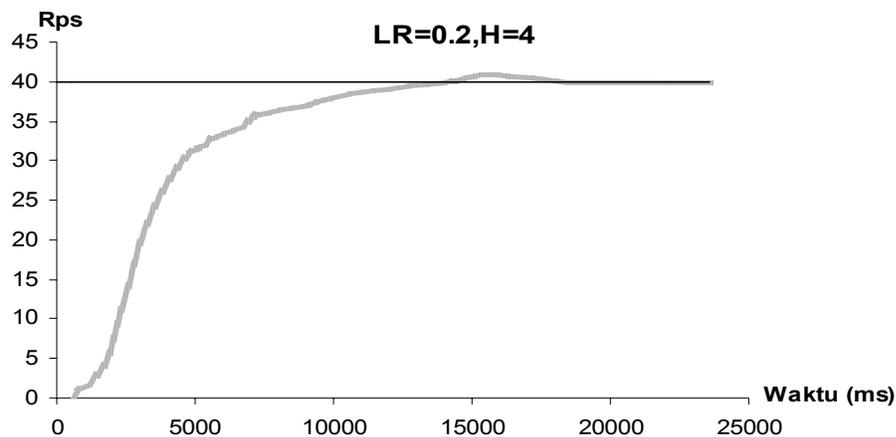
Gambar 4.29. Grafik Pengendalian Kecepatan Pada 30 Rps (LR=0.2,H=15)



Gambar 4.30. Grafik Pengendalian Kecepatan Pada 40 Rps (LR=0.2,H=15)



Gambar 4.31. Grafik Pengendalian Kecepatan Pada 30 Rps (LR=0.2,H=4)



Gambar 4.32. Grafik Pengendalian Kecepatan Pada 40 Rps (LR=0.2,H=4)

Dari hasil pelatihan di atas dapat diperoleh informasi-informasi sebagai berikut:

1. Pelatihan yang dilakukan pada JST secara terus-menerus, untuk suatu waktu tertentu dan suatu keadaan tertentu akan menghasilkan suatu respon kestabilan yang baik. Dalam arti bahwa, semakin banyak dilakukan pelatihan, respon kecepatan motor akan semakin mendekati *setting point* yang diinginkan dan *error* yang terjadi tentu saja akan semakin berkurang.
2. Dari hasil pelatihan tersebut di atas diperoleh nilai-nilai parameter pengendalian sebagai berikut:

Keterangan :

- *Delay time* (waktu tunda), adalah waktu yang diperlukan untuk mencapai setengah dari nilai target/tujuan (0 % sampai dengan 50 %).
- *Rise time* (waktu naik), adalah waktu dari saat sistem mencapai 10 % hingga sistem berada pada kondisi 90 % dari target yang diinginkan.
- *Peak time* (waktu puncak), waktu untuk mencapai kondisi maksimum.
- *Percent maximum overshoot* (lewatan maksimum), adalah besarnya lewatan/*maximum overshoot* (kondisi yang melebihi target atau jarak antara nilai puncak dengan target) terhadap nilai target.

$$\%Maximum\ overshoot = \frac{maksimum\ overshoot}{final\ value} \times 100\% \quad (4.4)$$

- *Settling time* (waktu penetapan), adalah waktu yang diperlukan sistem untuk mencapai kondisi *steady state*, atau kondisi yang sama dengan target hingga waktu $t = \infty$.

Tabel 4.4. Waktu Pencapaian Setting Point Pada 30 Rps

Learning Rate = 0,2						Learning Rate = 0,5						Learning Rate = 0,8					
Hidden = 4		Hidden = 15		Hidden = 30		Hidden = 4		Hidden = 15		Hidden = 30		Hidden = 4		Hidden = 15		Hidden = 30	
t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps
280	0	960	0	370	0	680	0	130	0	480	0	130	0	500	0	640	0
1050	1	2450	4	1310	2	1560	1	1060	1	1080	1	1070	2	2040	4	1410	1
1540	5	3000	7	1800	6	2110	5	1610	6	1580	5	1560	6	2580	8	1910	5
2040	9	3490	12	2350	10	2650	10	2100	10	2130	9	2110	10	3080	13	2460	9
2580	15	3980	17	2850	15	3200	15	2600	15	2680	14	2610	15	3570	18	2950	14
3080	19	4530	22	3340	19	3700	20	3150	19	3170	19	3100	20	4120	22	3440	18
3570	23	5030	25	3890	23	4190	23	3640	23	3660	22	3650	23	4620	25	3990	22
4120	26	5580	28	4380	26	4740	25	4130	26	4210	25	4140	26	5110	27	4490	24
4620	28	6070	29	4930	28	5240	27	4680	28	4710	27	4640	28	5660	28	4980	26
5170	29	6570	30	5430	29	5730	29	5180	29	5200	29	5190	29	6150	29	5530	27
5660	30	7110	31	5920	30	6280	30	5670	30	5750	30	5680	30	6700	30	6030	28
6150	31	8100	32	6470	31	7270	31	6220	31	6250	31	6180	31	7690	31	6520	29
13400	30	12720	31	12620	30	12930	30	10840	30	10860	30	12930	30	15440	30	8660	30
13460	30	15850	30	12680	30	12980	30	10890	30	11410	31	12990	30	15490	30	8720	30
13510	30	17880	29	12730	30	13040	30	10950	30	12950	30	13040	30	15550	30	8770	30
13570	30	19420	30	12790	30	13090	30	11000	30	15470	31	13100	30	15600	30	8830	30
13620	30	19470	30	12840	30	13150	30	11060	30	18550	30	13150	30	15660	30	8880	30
13680	30	19530	30	12900	30	13200	30	11110	30	18600	30	13210	30	15710	30	8940	30

Tabel 4.5. Waktu Pencapaian Setting Point Pada 40 Rps

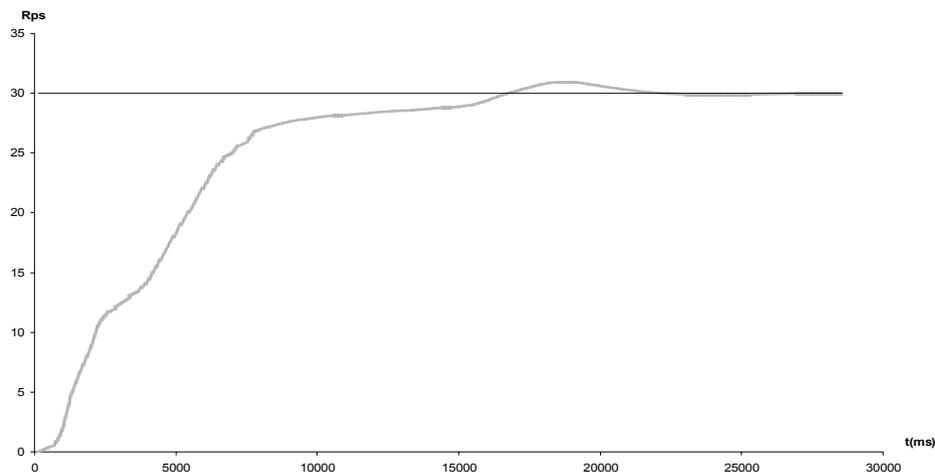
Learning Rate = 0,2						Learning Rate = 0,5						Learning Rate = 0,8					
Hidden = 4		Hidden = 15		Hidden = 30		Hidden = 4		Hidden = 15		Hidden = 30		Hidden = 4		Hidden = 15		Hidden = 30	
t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps	t(ms)	Rps
510	0	870	0	720	0	260	0	370	0	920	0	860	0	490	0	210	0
1720	4	2020	2	1820	1	760	4	1360	2	2240	3	1470	2	1590	3	1310	5
2270	11	2520	8	2310	7	1310	11	1860	8	2740	9	1960	9	2090	9	1860	9
2760	17	3070	14	2810	13	1800	17	2410	14	3230	16	2460	15	2580	15	2360	14
3260	22	3560	19	3360	19	2300	23	2900	20	3780	22	3000	21	3130	20	2850	20
3810	26	4060	24	3850	24	2850	27	3400	24	4270	25	3500	25	3620	24	3400	25
4300	29	4600	28	4350	27	3340	30	3940	28	4770	28	3990	28	4120	27	3890	28
4850	31	5100	30	4900	30	3830	32	4440	31	5320	31	4540	30	4670	29	4440	30
5340	32	5590	32	5390	32	4380	33	4930	32	5810	32	5040	32	5160	31	4940	32
5840	33	6140	33	5940	33	4880	34	5480	33	6310	33	5530	33	5710	32	5430	33
6390	34	6640	34	6430	34	5430	35	5980	34	6860	34	6080	34	6210	33	5930	34
6880	35	7130	35	6930	35	6970	36	7020	35	7900	35	6570	35	6700	34	6970	35
7380	36	8180	36	7970	36	7950	37	8010	36	8890	36	8110	36	7740	35	8510	36
8910	37	8670	37	8470	37	8500	38	8560	37	9930	37	8660	37	8240	36	9060	37
9960	38	10810	38	9020	38	10540	39	9050	38	10980	38	10200	38	9280	37	12630	38
11500	39	12840	39	10550	39	12570	40	10590	39	12510	39	12230	39	11310	38	16250	39
14080	40	13890	40	12090	40	15750	41	12620	40	13500	40	14810	40	12360	39	20320	40
15620	41	14930	41	14620	41	17790	40	14220	41	17680	41	16900	41	14390	40	20370	40
18690	40	15430	40	17200	40	17840	40	18830	40	20750	40	19980	40	14440	40	20430	40
18750	40	18500	41	18740	41	17900	40	19820	39	20810	40	20030	40	14500	40	20480	40
18800	40	22070	40	21810	40	17950	40	22950	40	20860	40	20090	40	14550	40	20540	40
18860	40	22130	40	21870	40	18010	40	24980	41	20920	40	20140	40	14610	40	20590	40
18910	40	22180	40	21920	40	18060	40	28110	40	20970	40	20200	40	14660	40	20650	40

Tabel 4.6. Nilai-Nilai Parameter Pengendalian

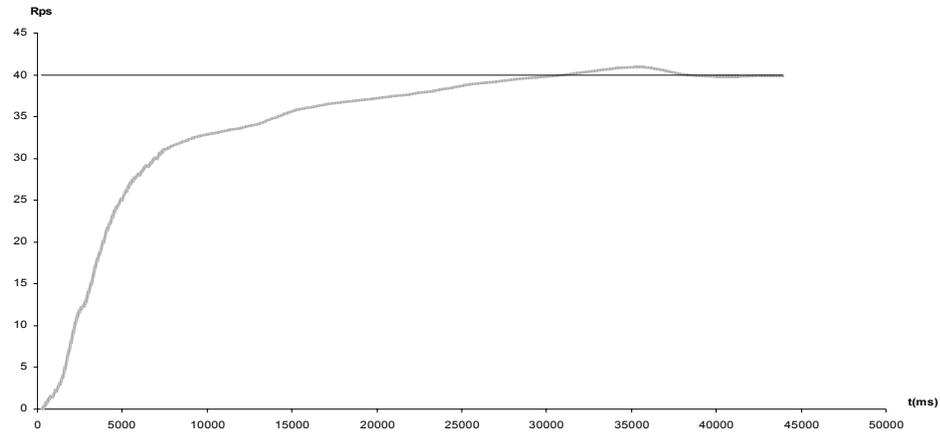
Parameter	30 Rps									40Rps								
	LR = 0,2			LR = 0,5			LR = 0,8			LR = 0,2			LR = 0,5			LR = 0,8		
	Hidden Unit			Hidden Unit			Hidden Unit			Hidden Unit			Hidden Unit			Hidden Unit		
	4	15	30	4	15	30	4	15	30	4	15	30	4	15	30	4	15	30
Delay Time (ms)	2580	3784	2850	3200	2600	2778	2610	3276	3073	3060	3660	3458	2050	2900	3597	2910	3130	2850
Rise Time (ms)	3075	3319	3223	3405	3125	3380	3198	3455	3870	5660	5993	5905	6210	6483	6567	6500	6567	7420
Peak Time (ms)	6150	8100	6470	7270	6220	6250	6180	7690	8660	15620	14930	14620	15750	14220	17680	16900	14390	20320
Maximum Overshoot (%)	3,33	6,67	3,33	3,33	3,33	3,33	3,33	3,33	0,00	2,50	2,50	2,50	2,50	2,50	2,50	2,50	0,00	0,00
Settling Time (ms)	13400	19420	12620	12930	10840	18550	12930	15440	8660	18690	22070	21810	17790	28110	20750	19980	14390	20320

3. Kecepatan sistem dalam mencapai kestabilan untuk *setting point* yang berbeda mengalami perbedaan. Hal ini dapat disebabkan oleh arsitektur JST yang dipergunakan, seperti banyaknya lapis *hidden* dan banyaknya sel *hidden* yang dipergunakan, banyaknya pelatihan yang diberikan pada keadaan tersebut, LR yang dipergunakan, oleh karena itu waktu penetapan untuk keadaan yang satu mengalami perbedaan dengan keadaan yang lain.
4. Karena kecepatan pencapaian yang berbeda-beda, maka pencapaian *error* minimum juga mengalami perbedaan untuk keadaan yang satu dengan yang lainnya. Pada satu keadaan, *error* minimum dapat dicapai setelah beberapa iterasi, tetapi untuk keadaan yang lain *error* minimum dapat dicapai setelah banyak iterasi.
5. Respon kestabilan untuk keadaan-keadaan di atas belum dapat dikatakan baik, karena waktu pencapaian sampai keadaan sistem stabil masih membutuhkan waktu yang relatif cukup lama untuk ukuran sistem stabil, untuk itu masih membutuhkan pelatihan lebih lanjut agar jaringan saraf tiruan dapat mengidentifikasi sistem dengan cepat.

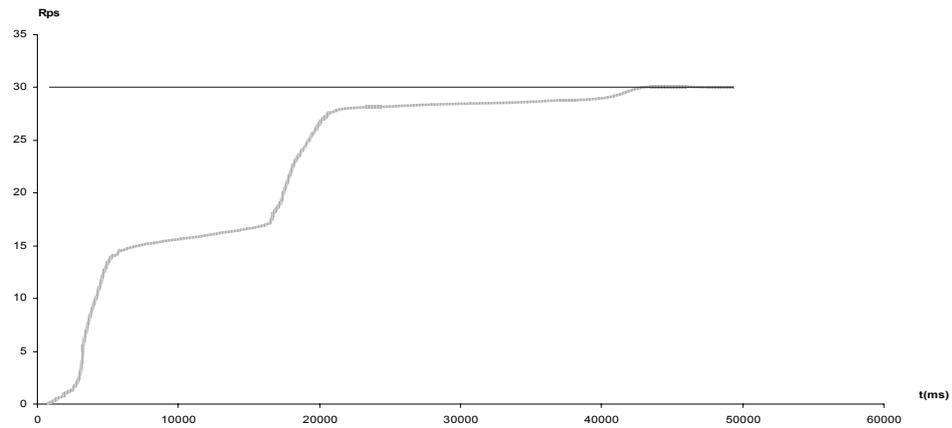
Selanjutnya adalah pengujian dengan data *training* yang berbeda, yaitu dimana pada nilai *output* nya mempunyai nilai antara -35 sampai dengan 35 dan antara -14 s/d 14, berbeda dengan data sebelumnya yang nilai *output* nya berkisar antara -70 s/d 70. Dan percobaan dilakukan pada LR=0.8 dan H=30, maka diperoleh hasil sebagai berikut:



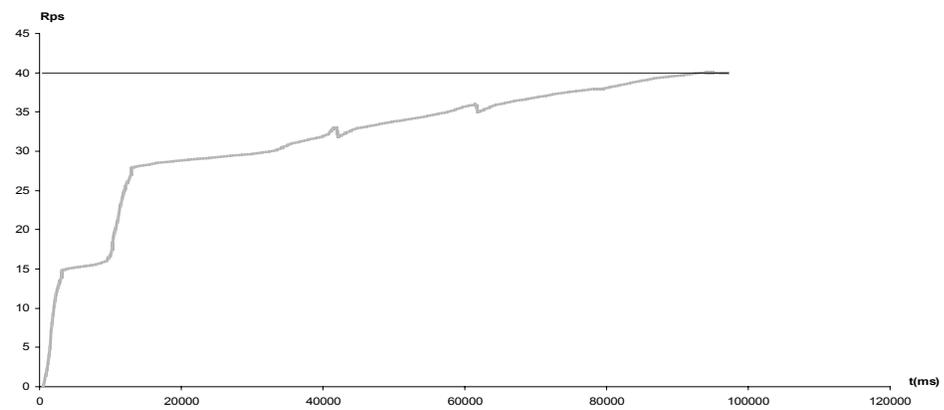
Gambar 4.33. Grafik Pengendalian Kecepatan Pada 30 Rps (-35 s/d 35)



Gambar 4.34. Grafik Pengendalian Kecepatan Pada 40 Rps (-35 s/d 35)

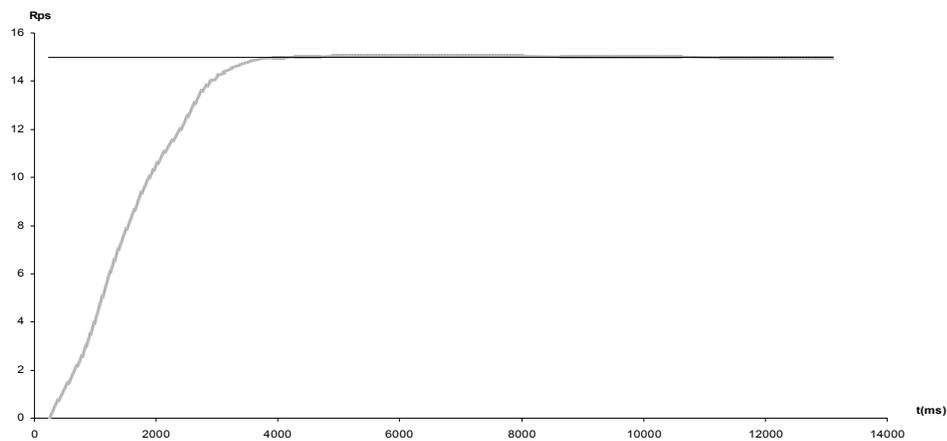


Gambar 4.35. Grafik Pengendalian Kecepatan Pada 30 Rps (-14 s/d 14)

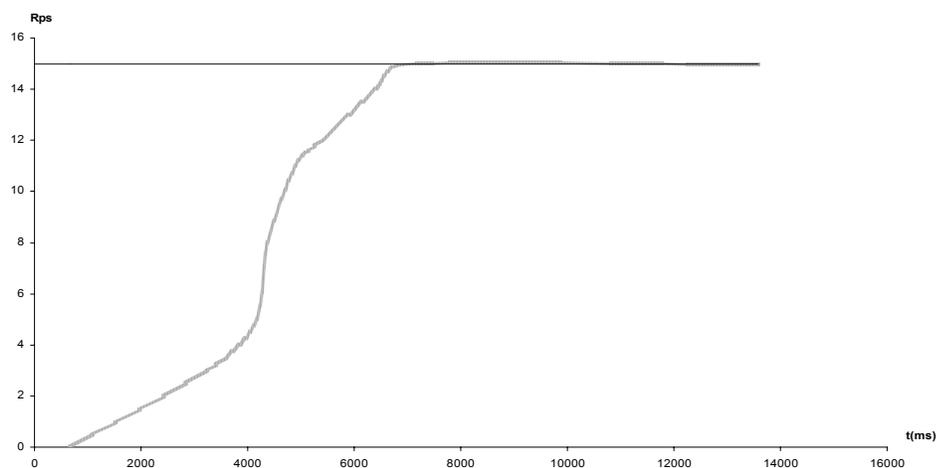


Gambar 4.36. Grafik Pengendalian Kecepatan Pada 40 Rps (-14 s/d 14)

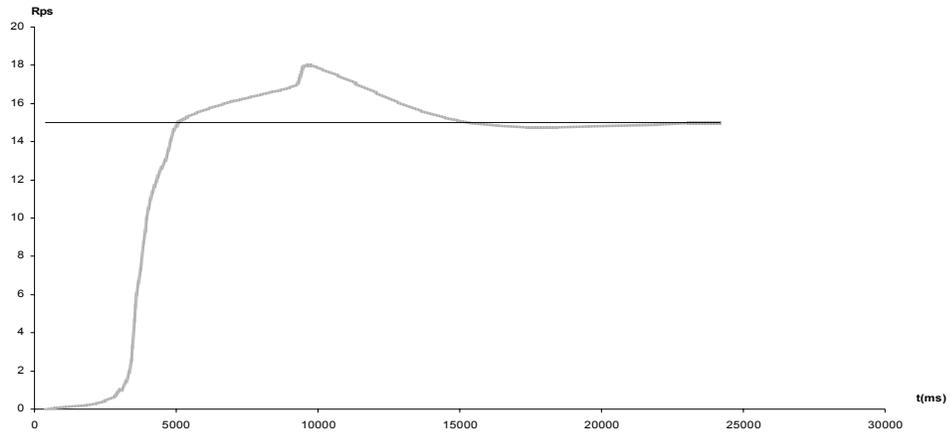
Dari grafik tersebut terlihat bahwa pencapaian *setting point* pada data *training* yang nilai *output* nya lebih kecil adalah lebih lama. Ini disebabkan karena nilai *output* yang lebih kecil akan menyebabkan perubahan kecepatan maupun selisih nilai *output* yang terjadi lebih lambat jika dibandingkan dengan data *training* dengan nilai *output* yang lebih besar. Secara logika nilai *output* yang lebih kecil akan lebih baik hasilnya jika dijalankan untuk *setting point* yang selisih *error* (SP-PV) nya kecil. Untuk itu dilakukan percobaan dengan memilih nilai *setting point* 15 dan 18. Dan didapat :



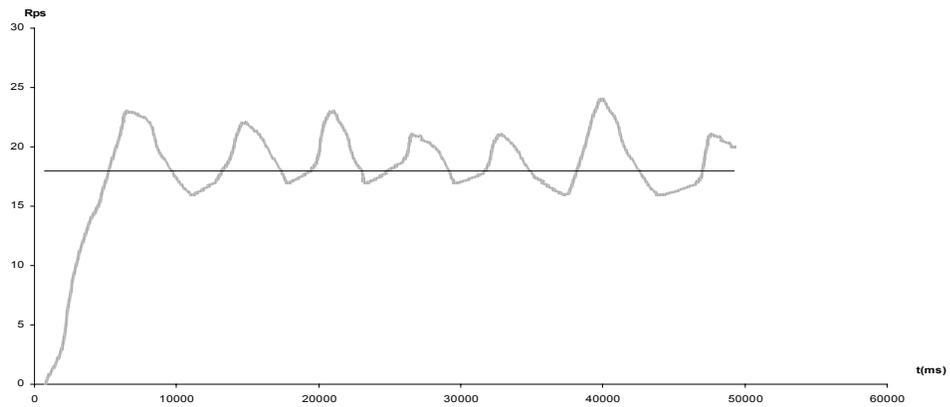
Gambar 4.37. Grafik Pengendalian Kecepatan Pada 15 Rps (-70 s/d 70)



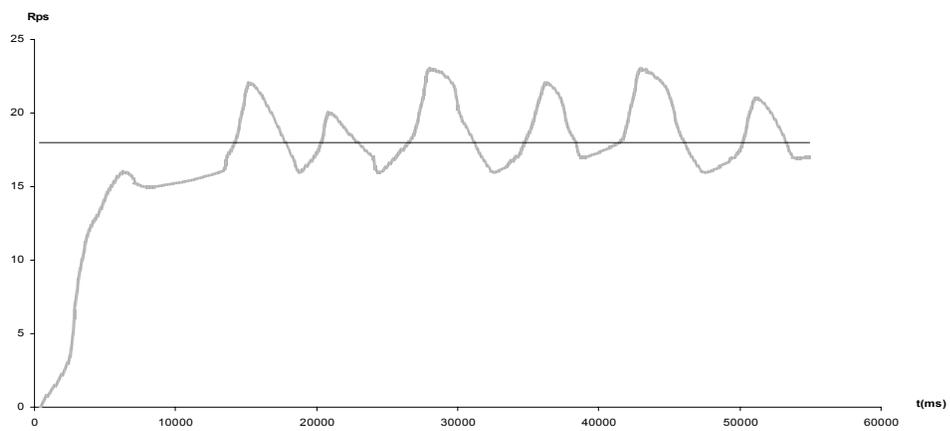
Gambar 4.38. Grafik Pengendalian Kecepatan Pada 15 Rps (-35 s/d 35)



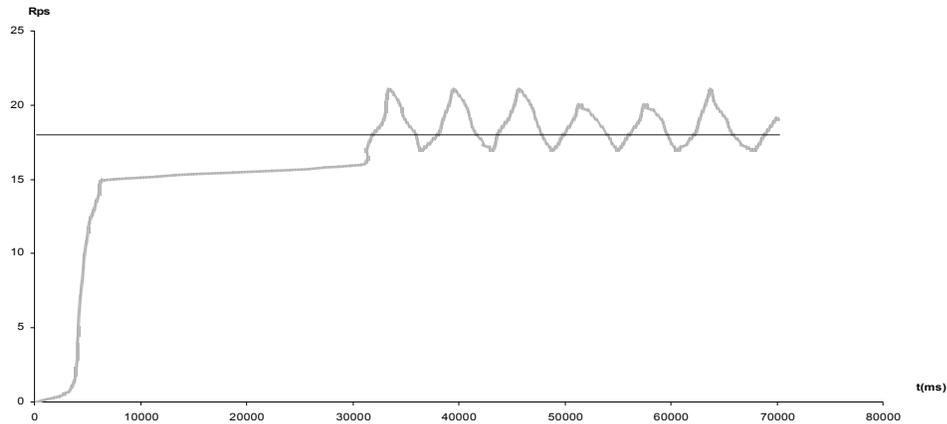
Gambar 4.39. Grafik Pengendalian Kecepatan Pada 15 Rps (-14 s/d 14)



Gambar 4.40. Grafik Pengendalian Kecepatan Pada 18 Rps (-70 s/d 70)



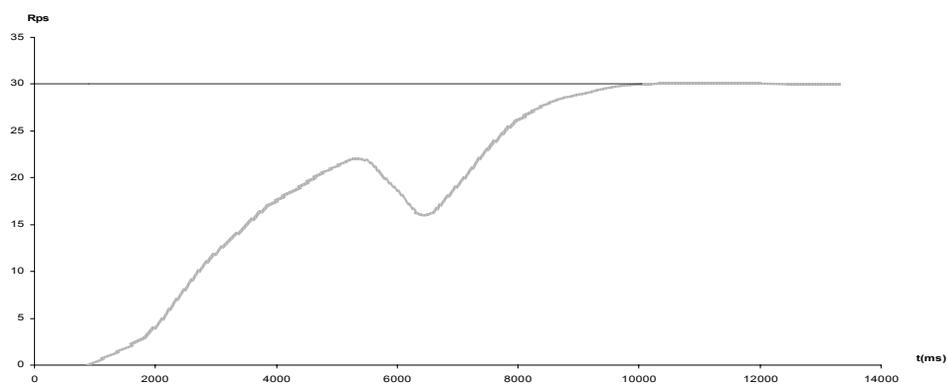
Gambar 4.41. Grafik Pengendalian Kecepatan Pada 18 Rps (-35 s/d 35)



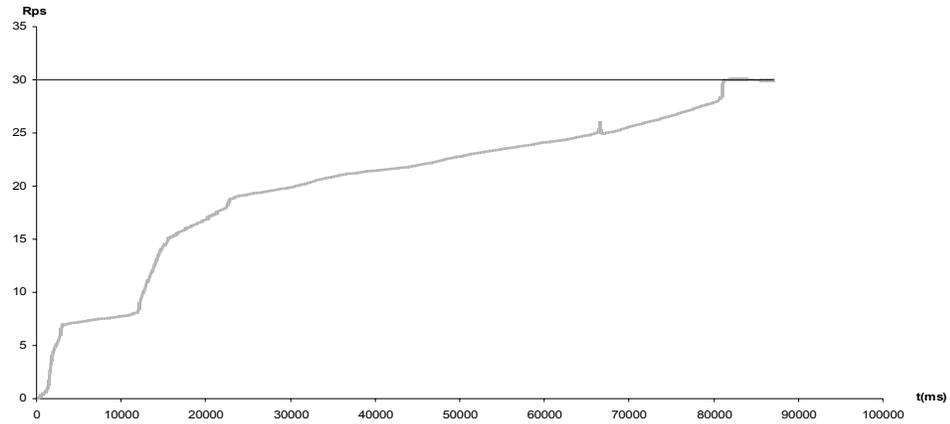
Gambar 4.42. Grafik Pengendalian Kecepatan Pada 18 Rps (-14 s/d 14)

Dari percobaan tersebut didapatkan bahwa pada *setting point* 15, data *training* dengan nilai *output* -70 s/d 70 lebih baik dari pada data *training* dengan nilai *output* yang lebih kecil. Tetapi pada $SP = 18$, data *training* dengan nilai *output* -14 s/d 14 lebih baik. Ini disebabkan karena perubahan kecepatannya sangat cepat setiap perubahan satu bit *input* nya, seperti pada *setting point* 18 Rps, jadi diperlukan nilai *output* yang perubahannya kecil. Dan juga pada $SP = 18$ tidak tercapai kondisi yang diinginkan karena sifat motornya yang tidak linier dengan *input* yang diberikan, dimana kenaikan satu bit *input* dihasilkan *output* kecepatan motor yang meningkat drastis.

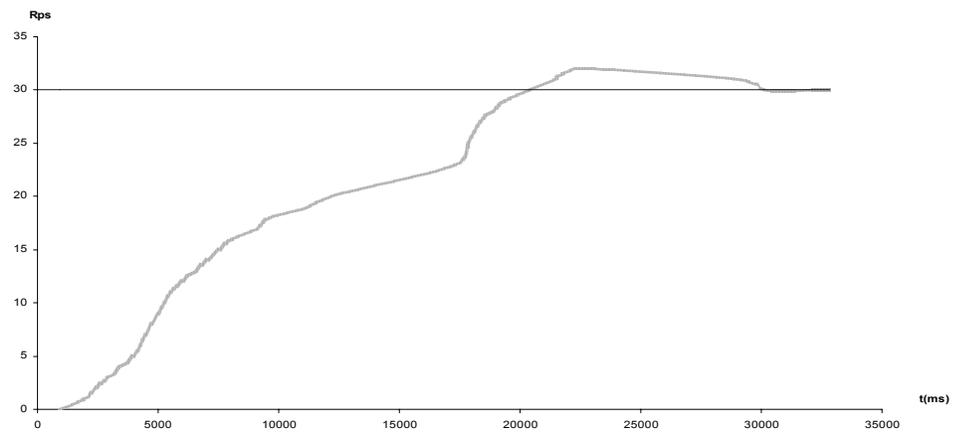
Selanjutnya adalah percobaan mengendalikan kecepatan dengan beban 100 gram.



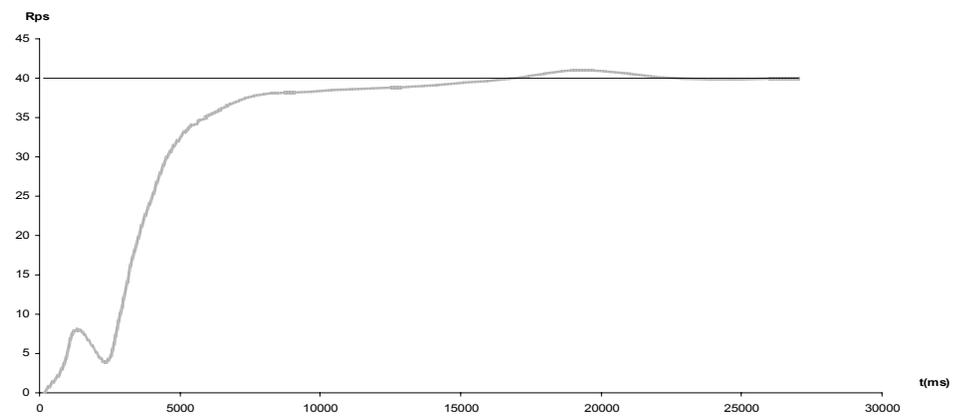
Gambar 4.43. Grafik Setting Point 30 Rps Dengan Beban (-70 s/d 70)



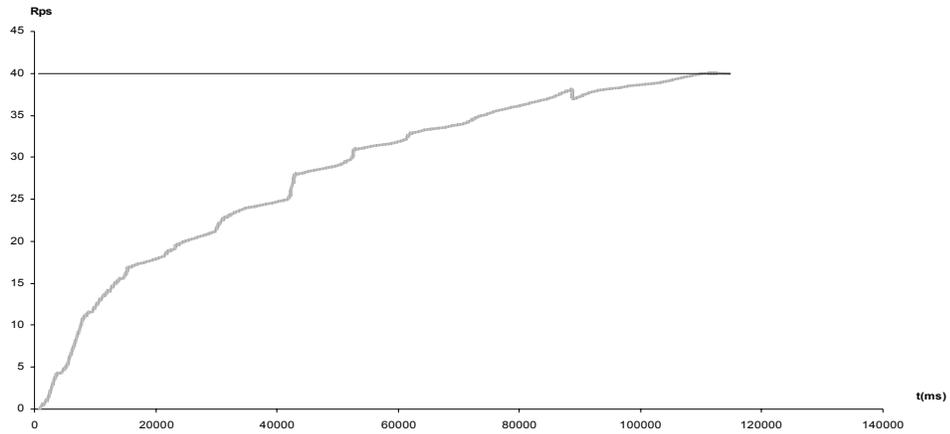
Gambar 4.44. Grafik Setting Point 30 Rps Dengan Beban (-14 s/d 14)



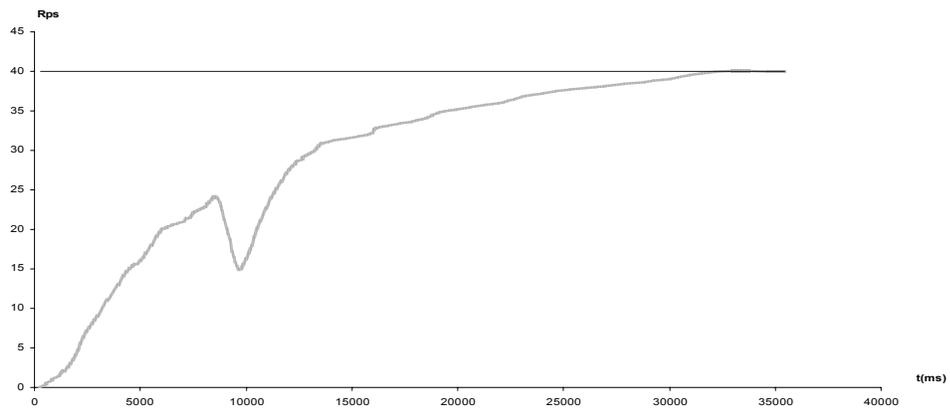
Gambar 4.45. Grafik Setting Point 30 Rps Dengan Beban (-35 s/d 35)



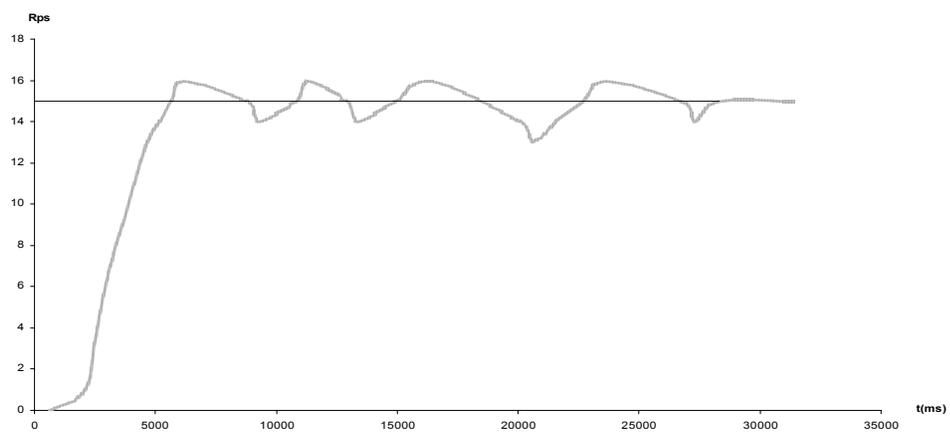
Gambar 4.46. Grafik Setting Point 40 Rps Dengan Beban (-70 s/d 70)



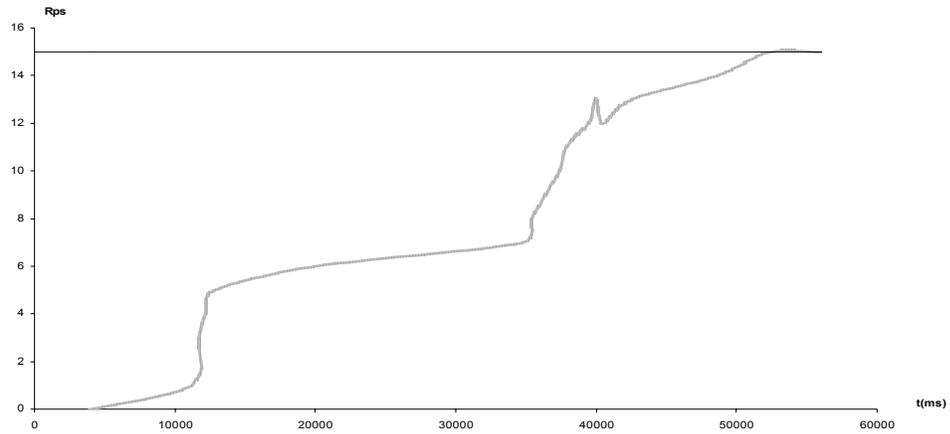
Gambar 4.47. Grafik Setting Point 40 Rps Dengan Beban (-14 s/d 14)



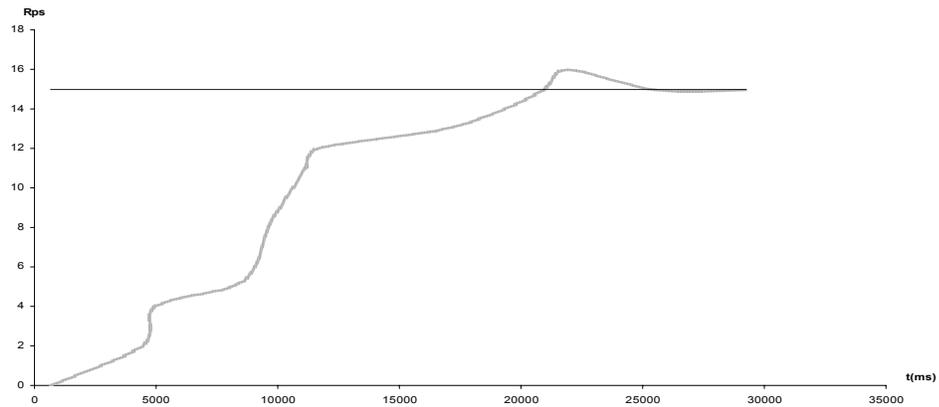
Gambar 4.48. Grafik Setting Point 40 Rps Dengan Beban (-35 s/d 35)



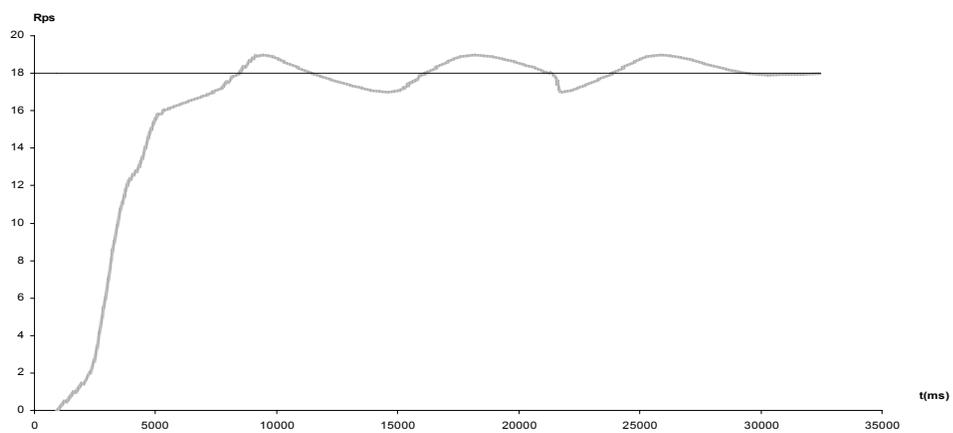
Gambar 4.49. Grafik Setting Point 15 Rps Dengan Beban (-70 s/d 70)



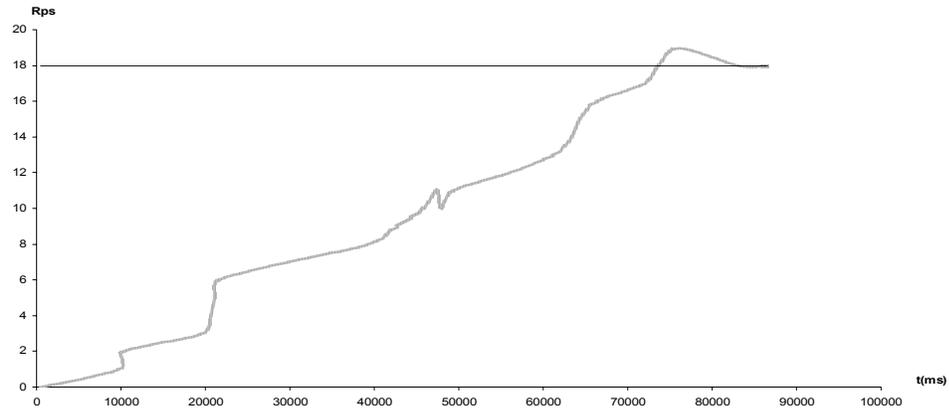
Gambar 4.50. Grafik Setting Point 15 Rps Dengan Beban (-14 s/d 14)



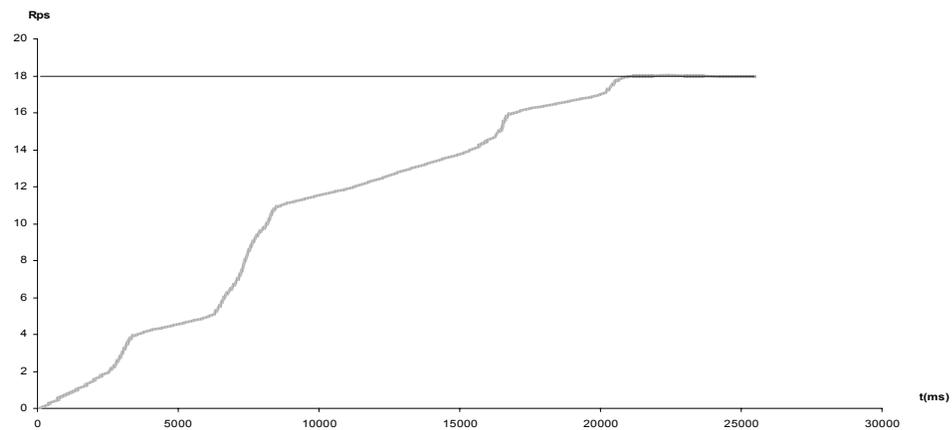
Gambar 4.51. Grafik Setting Point 15 Rps Dengan Beban (-35 s/d 35)



Gambar 4.52. Grafik Setting Point 18 Rps Dengan Beban (-70 s/d 70)



Gambar 4.53. Grafik Setting Point 18 Rps Dengan Beban (-14 s/d 14)



Gambar 4.54. Grafik Setting Point 18 Rps Dengan Beban (-35 s/d 35)

Dari grafik pada percobaan dengan menggunakan beban, didapat hasil adalah pada saat *setting point* 30 dan 40, data *training* dengan nilai *output* -70 s/d 70 memberikan respon yang lebih baik. Tetapi pada saat nilai dari *setting point* 15 dan 18, data *training* yang memberikan respon yang baik adalah data *training* yang mempunyai nilai *output* berkisar antara -35 s/d 35.