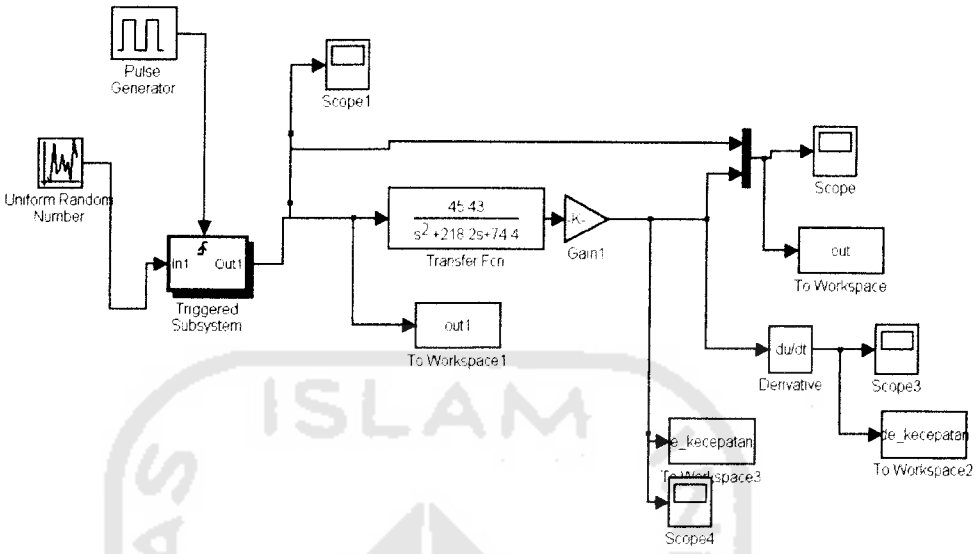
BAB IV

ANALISA DAN PEMBAHASAN SISTEM

Proses pelatihan dengan sistem automatisasi *fuzzy* metode gradien, terlebih dahulu dilakukan penentuan kondisi dari *plant* sesuai dengan keadaan yang ada pada motor sebenarnya. Masukan berupa tegangan (0 – 150 Volt) untuk mendapatkan keluaran berupa selisih kecepatan (*error*) dan perubahan kecepatan (*d_error*). Hasil simulasi disimpan kedalam *workspace* Matlab untuk dijadikan sebagai data masukan dan data keluaran pada pelatihan pengendali kecepatan motor DC.

4.1 Pelatihan *Fuzzy* dengan Metode Gradien

Pada pelatihan *fuzzy* metode gradien, data yang akan dilatih diambil dari hasil simulasi. Pada simulasi, masukan berupa nilai tegangan (0 – 150) pada blok *Uniform Random Number*, untuk memperoleh nilai kecepatan dan perubahan kecepatan. Hasil keluaran dari simulasi dalam bentuk gelombang kotak. Untuk membangkitkan gelombang kotak dapat diperoleh dengan menggunakan blok *Pulse Generator*. Simulasi menggunakan waktu 1000 detik. Hasil simulasi kemudian disimpan di *workspace* dengan menggunakan blok *from workspace,* yang disimpan dalam format *array*. Gambar 4.1 merupakan rangkaian blok *simulink* untuk pengambilan data pelatihan, seperti yang terlihat pada gambar dibawah ini :



Gambar 4.1. Simulink Pengambilan Data Pelatihan Open Loop

Data pelatihan masukan diambil dari nilai *error* kecepatan dan *derivative error* kecepatan, sedangkan data keluaran diambil dari nilai *out*1. Diperoleh data pelatihan sebanyak 64534. Data tersebut digunakan untuk melakukan pelatihan. dengan beragam nilai iterasi. Pada saat pelatihan dilakukan dengan semua data, membutuhkan waktu yang sangat lama dalam hitung jam. Namun tidak diperoleh nilai *error* yang diinginkan.

Pada pelatihan menggunakan 500 data berurutan diambil secara bebas. Proses pelatihan dilakukan dengan data pelatihan dan iterasi yang beragam, untuk memperoleh nilai *error* pelatihan yang diingkan. Jika *iterasi* diperbesar maka waktu pelatihan bertambah lama, *error* pelatihan bertambah kecil. Pelatihan menggunakan 200 *iterasi*, dapat dilakukan hanya dalam hitungan menit untuk menghasilkan nilai *error* yang diinginkan (mendekati 0). *Error* pelatihan menggunakan 3 aturan

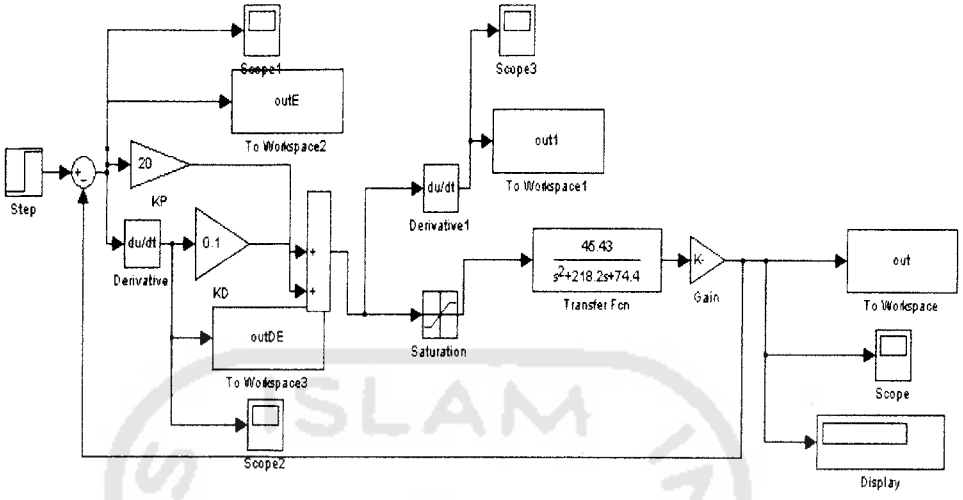
sebesaar 0 sampai 0.0278. *Error* pelatihan menggunakan 4 aturan sebesar 0 sampai 3.019. *Error* pelatihan menggunakan 5 aturan sebesaar 0 sampai 2.995. Aturan diperoleh secara *trial and error.*

Pada proses pelatihan data dengan nilai 0 tidak digunakan, karena akan menghasilkan nilai *error* 100% setelah dilakukan nilai *iterasi* yang berbeda – beda. Hal ini disebabkan pada keadaan sebenarnya, jika motor DC diberi masukan 0 volt maka tidak ada nilai tegangan yang dibangkitkan motor. Motor tidak akan mendapatkan keluaran berupa nilai kecepatan atau motor dalam keadaan diam.

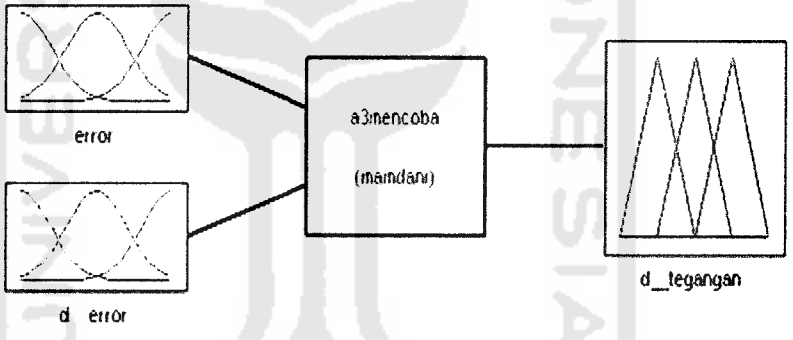
4.2 Simulasi Motor DC dengan Kendali *Fuzzy*

4.2.1 Simulasi Motor DC Sebelum Pelatihan

Tahap pertama adalah penentuan jangkauan untuk masukan dan keluaran pada sistem kendali logika *fuzzy* berdasarkan gambar 4.2. Jangkauan masukan dan keluaran ditentukan berdasarkan nilai maksimal dan minimal *error* dan *d_error* yang diperoleh saat simulasi. Nilai jangkauan *error* [-1750 , +1750], nilai jangkauan *d_error* [-650, +650]. Keluaran dari kendali *fuzzy* merupakan perubahan tegangan. Nilai jangkauan perubahan tegangan [-150, +150].

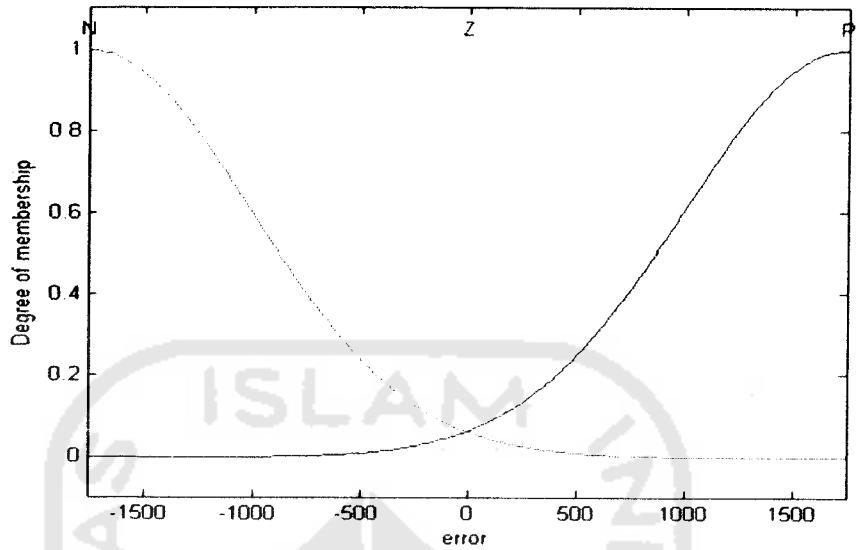


Gambar 4.2 Pengambilan jangkauan variabel masukan dan keluaran.

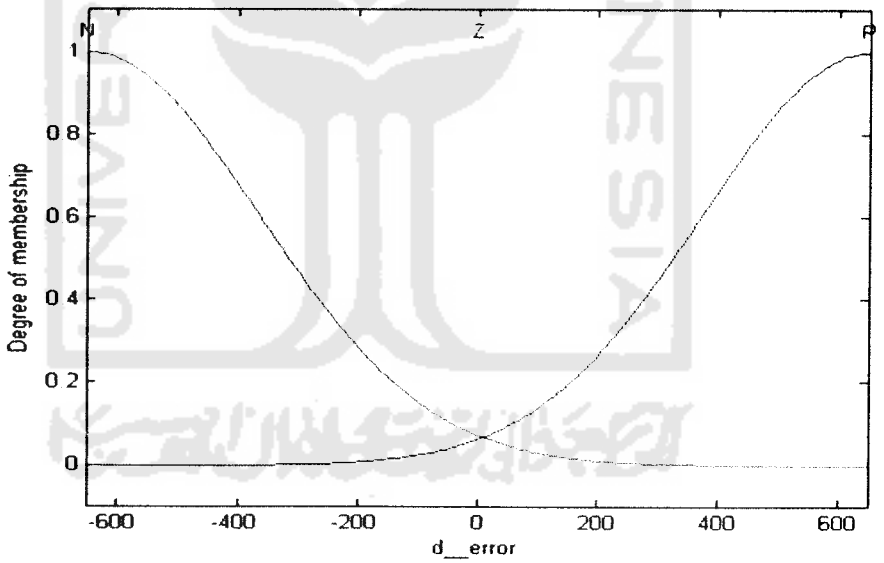


Gambar 4.3 Dua variabel masukan dan satu variabel keluaran

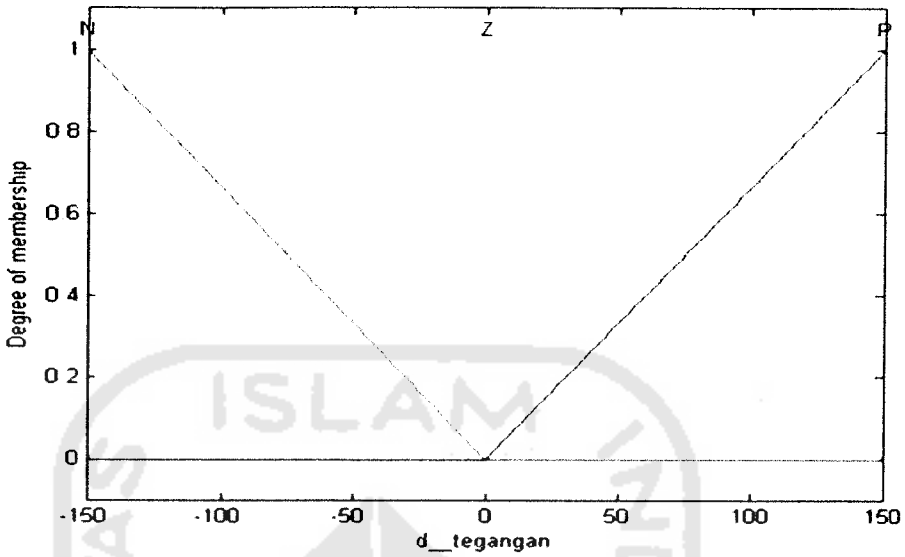
Tahap kedua yaitu penentuan fungsi keanggotaan untuk variabel masukan dan variabel keluaran. Variabel masukan menggunakan 3 fungsi keanggotaan Gaussian dan variabel keluaran menggunakan 3 fungsi keanggotaan segitiga.



Gambar 4.4 Fungsi keanggotaan untuk *error*



Gambar 4.5 Fungsi keanggotaan untuk *d _error*



Gambar 4.6 Fungsi keanggotaan untuk perubahan tegangan

Tahap ketiga adalah penentuan aturan *fuzzy*. Penentuan aturan *fuzzy* merupakan tahap akhir dari proses desain sistem *fuzzy*. Berdasarkan jumlah masukan (2) dan jumlah fungsi keanggotaan dari masing – masing masukan (3), diperoleh aturan maksimal yang dapat digunakan yaitu $3^2 = 9$ aturan. Jumlah 9 aturan tidak digunakan semua pada desain aturan *fuzzy*. Hanya 3, 4 dan 5 aturan yang digunakan dalam aturan *fuzzy*. Penggunaan variasi aturan diperoleh secara *trial and error* berdasarkan pada tabel 3.1.

Penggunaan 3 aturan seperti dibawah ini, setelah dilakukan beberapa kali percobaan.

1. if (er is N) and (d_er is N) then (tegangan is P)
2. if (er is P) and (d_er is Z) then (tegangan is N)
3. if (er is P) and (d_er is P) then (tegangan is N)

Penggunaan 4 aturan diperoleh dari beberapa percobaan seperti yang terlihat pada tabel 4.1.

Tabel 4.1 Tabel Percobaan 4 Aturan

Percobaan	Aturan yan digunakan	Keterangan
1	1,2,8,9	Parameter b no 2 tidak berubah.
2	1,3,8,9	Berhasil
3	1,4,8,9	Parameter b no 2 tidak berubah.
4	1,5,8,9	Parameter b no 2 tidak berubah.
5	1,6,8,9	Parameter b no 2 tidak berubah.

Berdasarkan pada tabel 4.1, maka 4 aturan seperti dibawah ini :

1. *if (er is N) and (d_er is N) then (d_tegangan is N)*
2. *if (er is N) and (d_er is P) then (d_tegangan is Z)*
3. *if (er is P) and (d_er is Z) then (d_tegangan is P)*
4. *if (er is P) and (d_er is P) then (d_tegangan is P)*

Penggunaan 5 aturan diperoleh dari beberapa percobaan seperti yang terlihat pada tabel 4.2.

Tabel 4.2 Tabel Percobaan 5 Aturan

Percobaan	Aturan yan digunakan	Keterangan
1	1,2,3,8,9	Berhasil.
2	1,3,4,8,9	Parameter b ke 2 tidak berubah.
3	1,3,5,8,9	Parameter b ke 2 dan 3 tidak berubah.
4	1,3,6,8,9	Parameter b ke 3 tidak berubah.

Berdasarkan pada tabel 4.2, maka 5 aturan seperti terlihat dibawah ini :

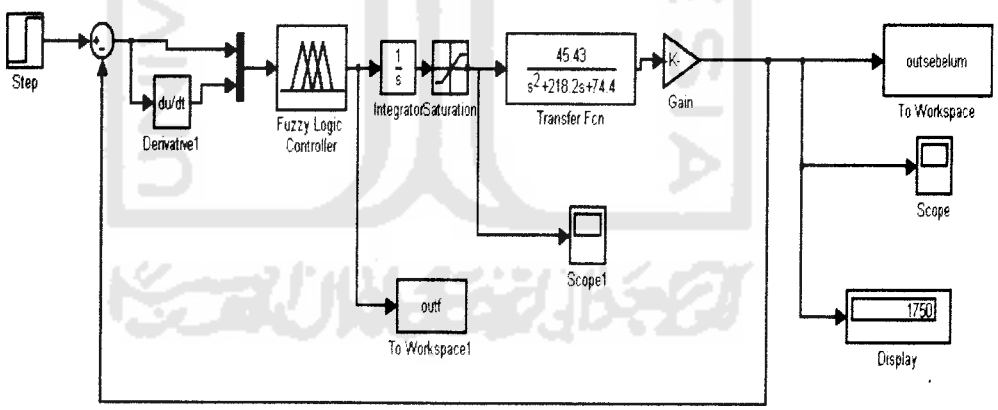
1. *if (er is N) and (d_er is N) then (d_tegangan is N)*
2. *if (er is N) and (d_er is Z) then (d_tegangan is N)*
3. *if (er is N) and (d_er is P) then (d_tegangan is Z)*
4. *if (er is P) and (d_er is Z) then (d_tegangan is P)*
5. *if (er is P) and (d_er is P) then (d_tegangan is P)*

Tabel 4.1 dan 4.2 merupakan contoh untuk memperoleh 4 dan 5 aturan *fuzzy*. Karena tidak semua variasi aturan dapat digunakan dalam perancangan *fuzzy* yang

baru untuk mendapatkan *error* sistem kecil pada saat pengujian. Beberapa hal dapat disebabkan karena parameter b dari nomor aturan tertentu tidak berada dalam nilai jangkaun setelah dilakukan pelatihan dengan *iterasi* yang beragam dan memperlebar jangkauan parameter b.

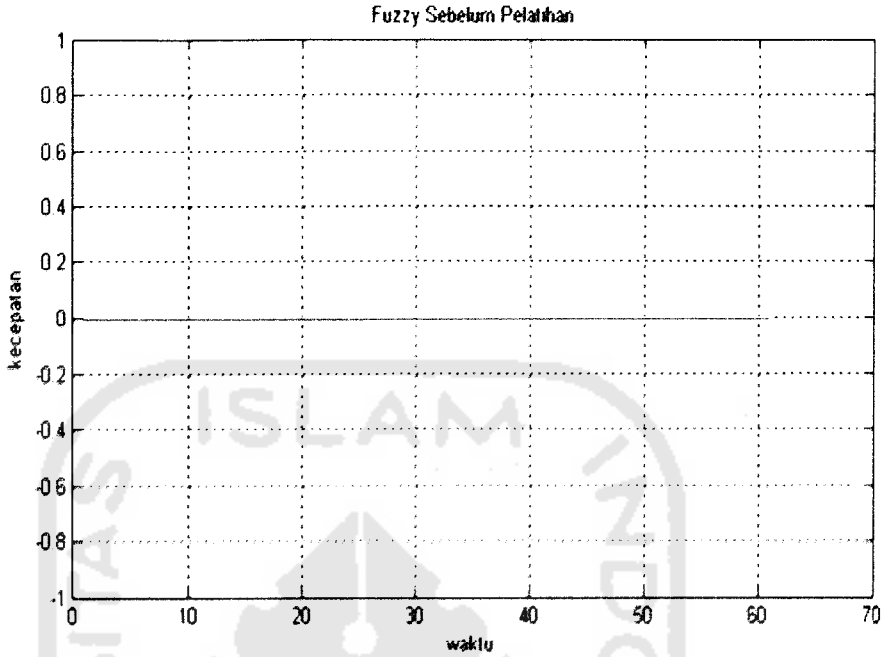
Penggunaan aturan *fuzzy* yang baru dapat diperoleh dengan metode *trial and error* untuk mendapatkan variasi aturan yang beragam. Metode tersebut membutuhkan waktu yang lama untuk memperoleh aturan yang diinginkan.

Proses tahapan perancangan *fuzzy* selesai dilakukan, *fuzzy* disimpan dengan format .fis, langkah selanjutnya *export* data kendali *fuzzy* ke *workspace.* Untuk dapat menjalankan simulasi sistem dengan menggunakan *simulink* seperti pada gambar 4.7, blok FLC diberi nama sesuai dengan nama *fuzzy* yang disimpan.



Gambar 4.7 Rangkaian simulasi motor DC sebelum pelatihan

Simulasi kendali motor DC menggunakan waktu simulasi selama 100 detik. Setelah di jalankan didapatkan grafik respon sistem seperti pada gambar 4.8.



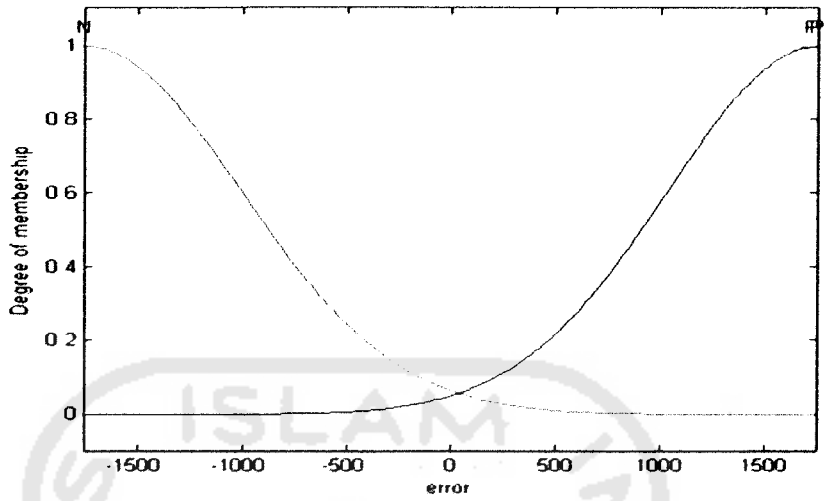
Gambar 4.8 Keluaran sistem *fuzzy* sebelum pelatihan

4.2.2 Simulasi Motor DC Setelah Pelatihan

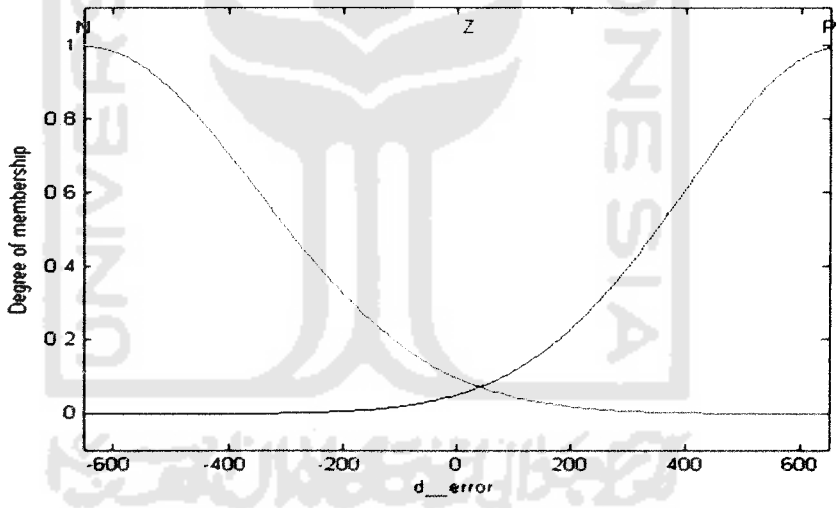
Proses perancangan *fuzzy* yang baru dilakukan setelah tahap pelatihan *fuzzy* seperti pada subbab 4.1. Pada saat diperoleh *error* yang diinginkan dan perubahan parameter fungsi keanggotaan (bi, c_j^i, σ_j^i). Langkah selanjutnya melakukan perancangan *fuzzy* yang baru. Tahap melakukan perancangan *fuzzy* setelah pelatihan, hampir sama seperti perancangan *fuzzy* sebelum pelatihan. Perbedaannya adalah pada parameter fungsi keanggotaan masukan (c_j^i, σ_j^i) dan keluaran (bi). Perubahan hasil parameter fungsi keanggotaan (bi, c_j^i, σ_j^i) dimasukan ke perancangan *fuzzy*.

A. Penggunaan 3 Aturan

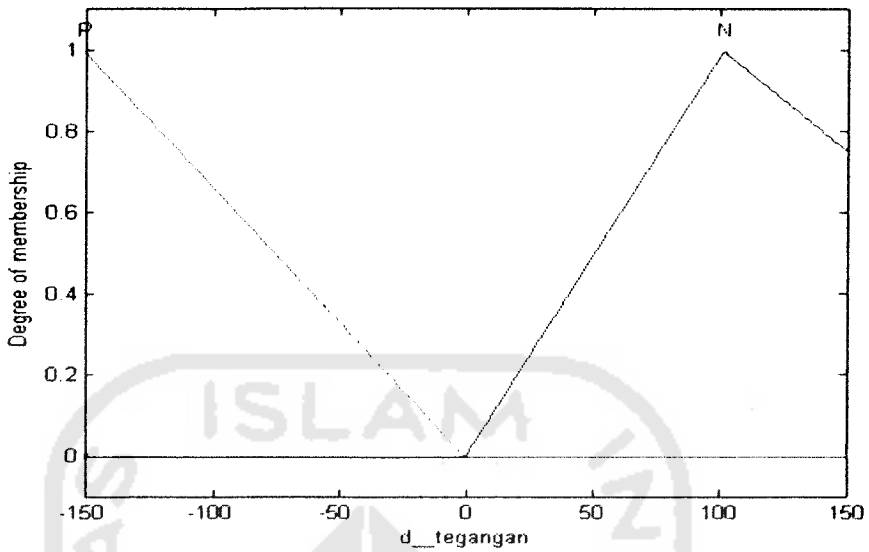
Gambar 4.9, 4.10, 4.11 merupakan hasil perubahan parameter fungsi keanggotaan *fuzzy* dengan 3 aturan.



Gambar 4.9 Fungsi keanggotaan masukan *error fuzzy* setelah Pelatihan



Gambar 4.10 Fungsi keanggotaan masukan *derivative error fuzzy* setelah Pelatihan



Gambar 4.11 Fungsi keanggotaan keluaran *fuzzy* setelah Pelatihan

Gambar 4.9, 4.10, 4.11 diperoleh dari tabel 4.3, 4.4, 4.5 yang merupakan tabel perbandingan antara parameter fungsi keanggotaan masukan dan keluaran sebelum dan setelah pelatihan.

Tabel 4.3 Parameter fungsi keanggotaan masukan *error*

Perubahan Parameter C		Perubahan Parameter σ	
Sebelum Pelatihan	**Setelah Pelatihan**	**Sebelum Pelatihan**	**Setelah Pelatihan**
-1750	-1749.7	750	750.8557
1741	1725.6	750	778.4098
1741	1757.8	750	717.3058

Tabel 4.4 Parameter fungsi keanggotaan masukan *d_error*

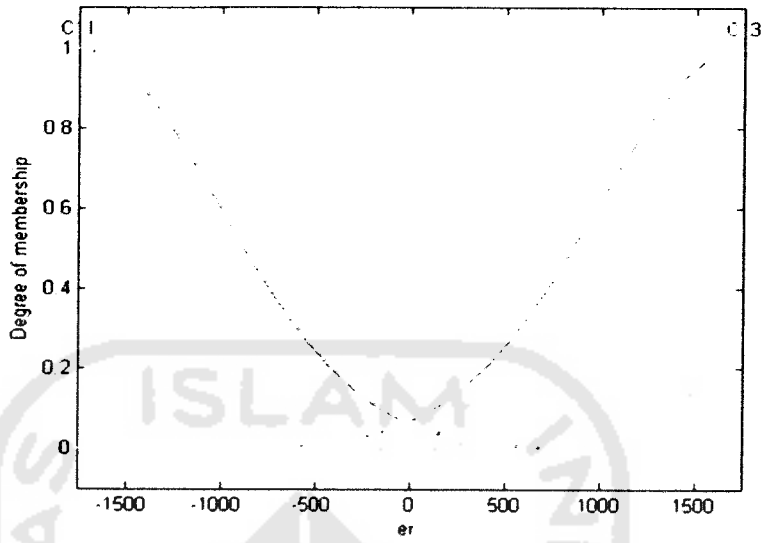
Perubahan Parameter C		Perubahan Parameter σ	
Sebelum Pelatihan	**Setelah Pelatihan**	**Sebelum Pelatihan**	**Setelah Pelatihan**
-650	-649.2	300	302.3384
0	19.9	300	317.4746
650	674.6	300	275.7636

Tabel 4.5 Parameter fungsi keanggotaan keluaran (b)

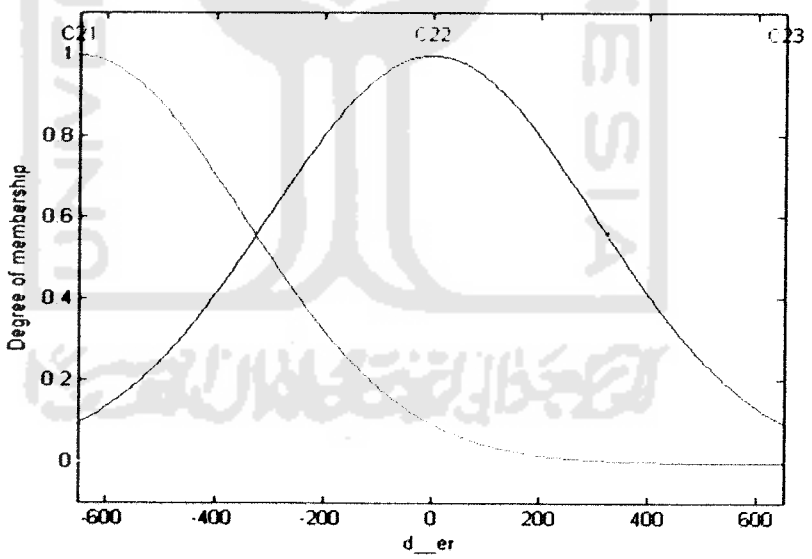
Sebelum Pelatihan	**Setelah Pelatihan**
150	150.5427
-150	101.8506
-150	101.3777

B. Penggunaan 4 Aturan

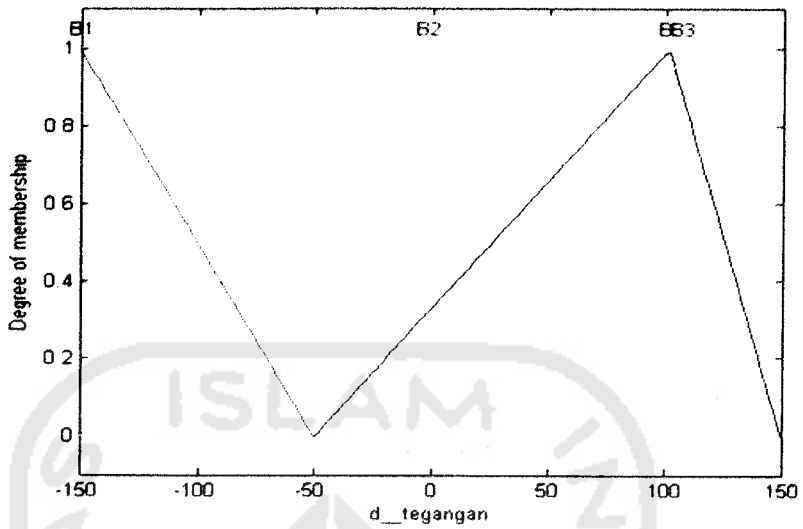
Gambar 4.12, 4.13, 4.14 merupakan hasil perubahan parameter fungsi keanggotaan *fuzzy* dengan 4 aturan.



Gambar 4.12 Fungsi keanggotaan masukan *error fuzzy* setelah Pelatihan



Gambar 4.13 Fungsi keanggotaan masukan *derivative error fuzzy* setelah Pelatihan



Gambar 4.14 Fungsi keanggotaan keluaran *fuzzy* setelah Pelatihan

Gambar 4.12, 4.13, 4.14 diperoleh dari tabel 4.6, 4.7, 4.8 yang merupakan tabel perbandingan antara parameter fungsi keanggotaan masukan dan keluaran sebelum dan setelah pelatihan.

Tabel 4.6 Parameter fungsi keanggotaan masukan *error*

Perubahan Parameter C		Perubahan Parameter σ	
Sebelum Pelatihan	**Setelah Pelatihan**	**Sebelum Pelatihan**	**Setelah Pelatihan**
-1750	-1749.9	750	750.2174
-1750	-1749.3	750	752.0831
1741	1740.7	750	750.6396
1741	1741.6	750	748.9025

Tabel 4.7 Parameter fungsi keanggotaan masukan *d_error*

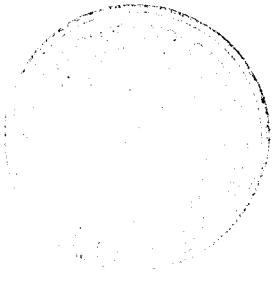
Perubahan Parameter C		Perubahan Parameter σ	
Sebelum Pelatihan	**Setelah Pelatihan**	**Sebelum Pelatihan**	**Setelah Pelatihan**
-650	-649.8	300	300.4882
650	649.2	300	301.2250
0	0.5	300	300.4372
650	651.2	300	298.6372

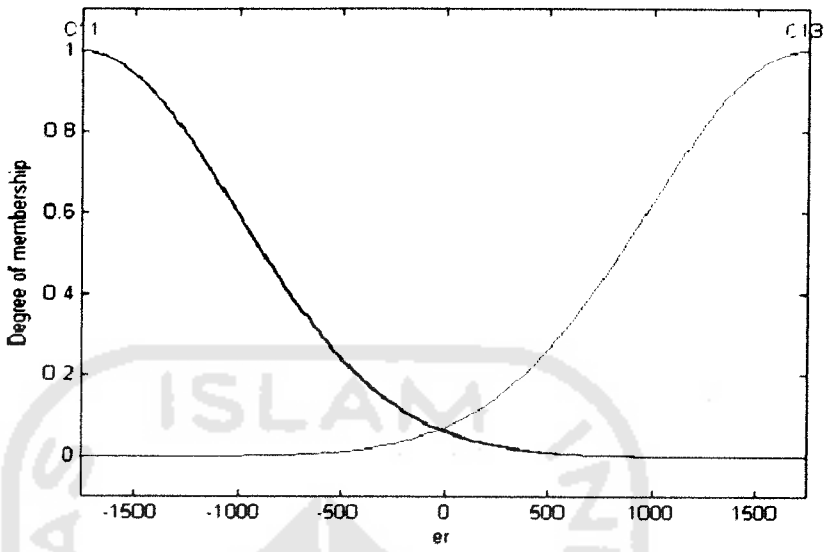
Tabel 4.8 Parameter fungsi keanggotaan keluaran (b)

Sebelum Pelatihan	**Setelah Pelatihan**
-150	-150.0631
-150	-0.8612
0	101.3855
150	106.2847

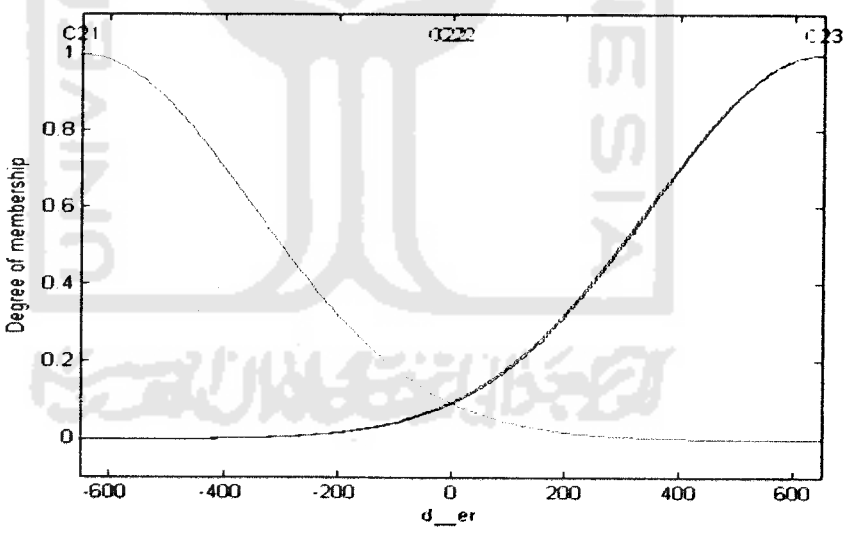
C. Penggunaan 5 Aturan

Gambar 4.15, 4.16, 4.17 merupakan hasil perubahan parameter fungsi keanggotaan *fuzzy* dengan 5 aturan.

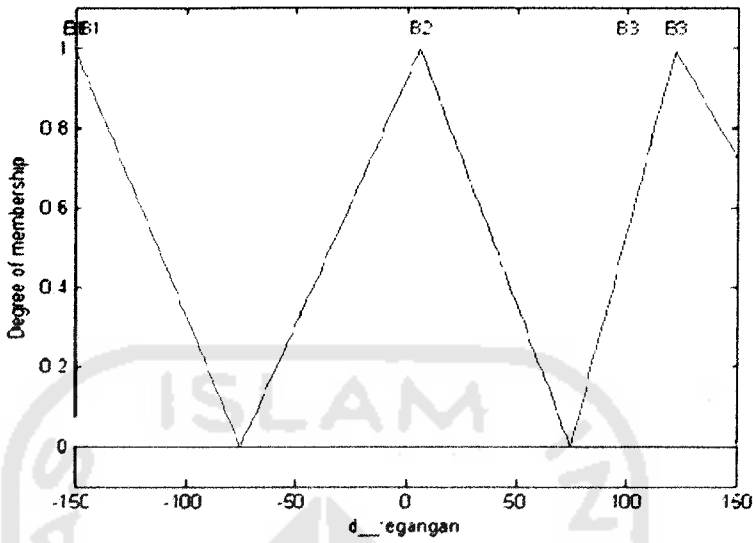




Gambar 4.15 Fungsi keanggotaan masukan *error fuzzy* setelah Pelatihan



Gambar 4.16 Fungsi keanggotaan masukan *derivative error fuzzy* setelah Pelatihan



Gambar 4.17 Fungsi keanggotaan keluaran *fuzzy* setelah Pelatihan

Gambar 4.15, 4.16, 4.17 diperoleh dari tabel 4.9, 4.10, 4.11 yang merupakan tabel perbandingan antara parameter fungsi keanggotaan masukan dan keluaran sebelum dan setelah pelatihan.

Tabel 4.9 Parameter fungsi keanggotaan masukan *error*

Perubahan Parameter C		Perubahan Parameter σ	
Sebelum Pelatihan	**Setelah Pelatihan**	**Sebelum Pelatihan**	**Setelah Pelatihan**
-1750	-1750	750	750.1349
-1750	-1753.9	750	741.5995
-1750	-1751.9	750	744.9383
1741	1737.9	750	755.5481
1741	1738.8	750	755.6004

Tabel 4.10 Parameter fungsi keanggotaan masukan *d_error*

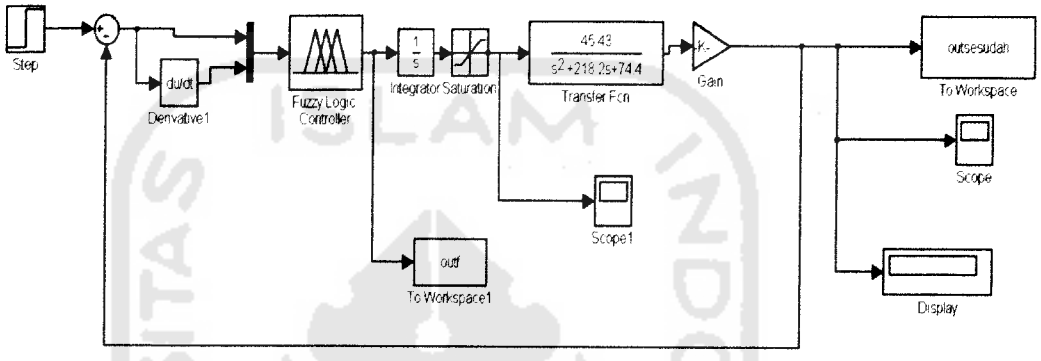
Perubahan Parameter C		Perubahan Parameter σ	
Sebelum Pelatihan	**Setelah Pelatihan**	**Sebelum Pelatihan**	**Setelah Pelatihan**
-650	-650	300	299.8790
0	-5.9	300	293.1283
650	652	300	297.9885
0	4.1	300	303.3245
650	648.9	300	299.8649

Tabel 4.11 Parameter fungsi keanggotaan keluaran (b)

Sebelum Pelatihan	**Setelah Pelatihan**
-150	-150.0013
-150	-144.1906
0	6.5860
0	99.8263
150	121.8568

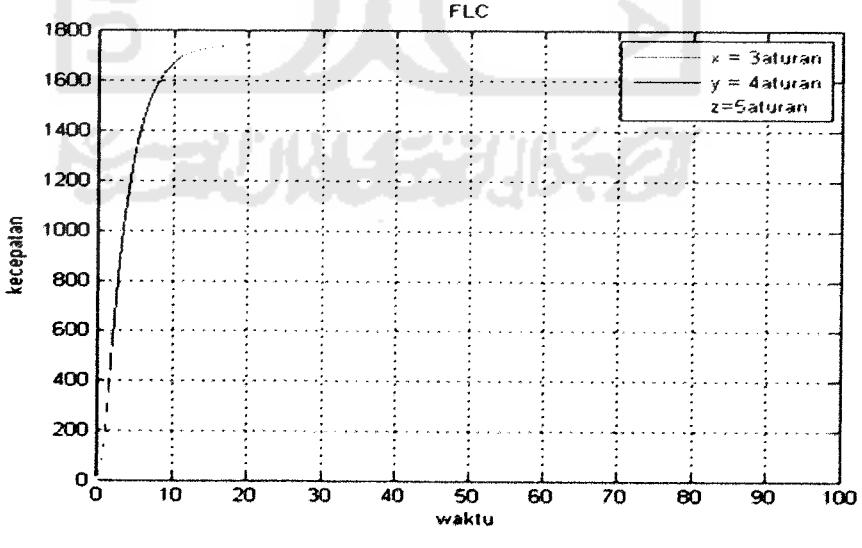
4.3 Pengujian Sistem

Perancangan *fuzzy* selesai dilakukan, langkah selanjutnya *export* data kendali *fuzzy* ke *workspace* untuk dapat menjalankan simulasi sistem dengan menggunakan *simulink* seperti pada gambar 4.18.



Gambar 4.18 Rangkaian simulasi motor DC setelah pelatihan

Simulasi kendali motor DC menggunakan waktu simulasi selama 100 detik. Pengujian dilakukan dengan masukan 1750, didapatkan respon sistem dan keluaran *step respon* setelah pelatihan seperti terlihat pada gambar 4.19.



Gambar 4.19 Hasil keluaran setelah pelatihan

Tabel 4.12 Respon Sistem Sebelum dan Setelah Pelatihan dengan masukan 1750 rad / s.

Karakteristik	**Sebelum**	**Setelah**		
		3 Aturan	4 Aturan	5 Aturan
Rise time	-	6.7192	6.5987	6.5633
Settling time	-	12.4683	12.3139	12.2748
Settling minimal	-	1.5755e+003	1.5755e+003	1.5757e+003
Settling maximal	-	1.7502e+003	1.7502e+003	1.7502e+003
Overshoot (%)	-	0.0104	0.0104	0.0104
Undershoot (%)	-	0	0	0
Peak	-	1.7502e+003	1.7502e+003	1.7502e+003
Peak time	-	99.8007	99.9484	99.4044

Berdasarkan tabel 4.12, respon sistem setelah pelatihan (menggunakan 3, 4, 5 aturan) lebih baik dibandingkan sebelum pelatihan. Hal ini disebabkan oleh penggunaan data pelatihan pada proses pelatihan untuk memperbaiki parameter masukan dan keluaran.

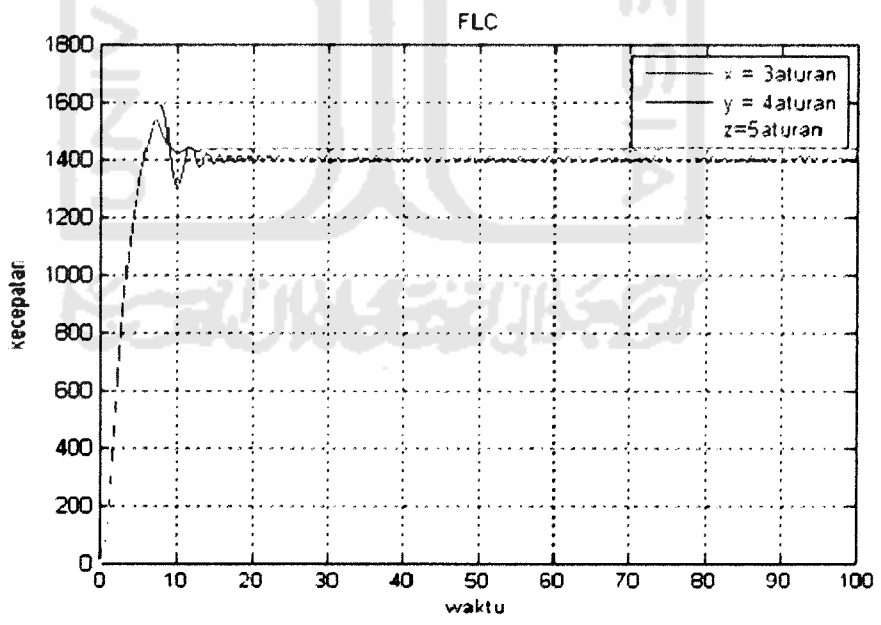
Hasil pengujian menggunakan 3, 4 dan 5 aturan dilakukan dengan memberikan nilai masukan (*setpoint*) yang berbeda.

1. Pengambilan data 1400 rad / s.

   Hasil respon sistem seperti terlihat pada table 4.13

Tabel 4.13 Respon Sistem Setelah Pelatihan dengan masukan 1400 rad / s.

Karakteristik	**Setelah**		
	3 Aturan	4 Aturan	5 Aturan
Rise time	3.8324	3.7038	3.6642
Settling time	8.0210	12.4936	21.1036
Settling minimal	1.2621e+003	1.2613e+003	1.2620e+003
Settling maximal	1.5380e+003	1.5917e+003	1.6206e+003
Overshoot (%)	9.8558	13.6929	15.7553
Undershoot (%)	0	0	0
Peak	1.5380e+003	1.5917e+003	1.6206e+003
Peak time	7.3275	7.9811	8.5293



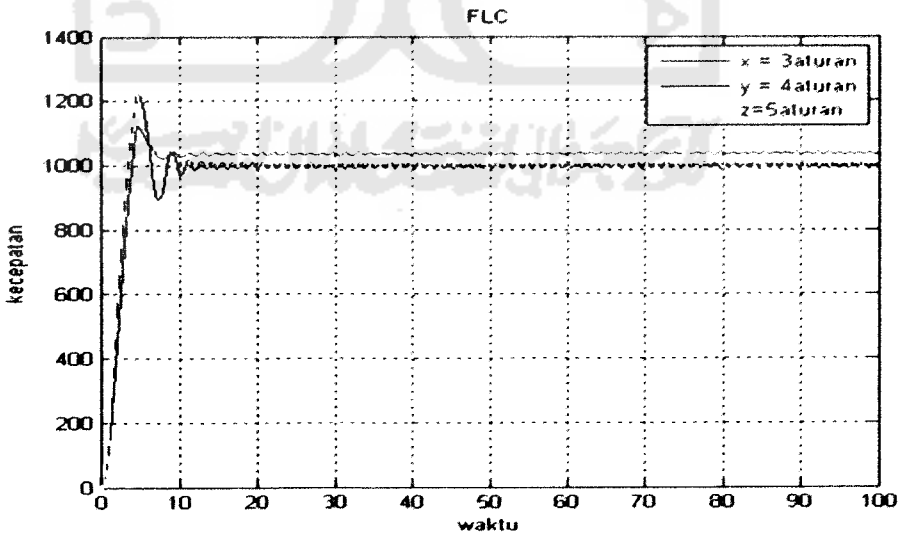
Gambar 4.20 *Step Respon* Pengambilan data 1400 rad / s

2. Pengambilan data 1000 rad / s.

Hasil respon sistem seperti terlihat pada table 4.14.

Tabel 4.14 Respon Sistem Setelah Pelatihan dengan masukan 1000 rad / s.

Karakteristik	**Setelah**		
	3 Aturan	4 Aturan	5 Aturan
Rise time	2.7494	2.3078	2.2198
Settling time	8.4045	11.3915	18.6656
Settling minimal	903.6929	896.8396	902.5205
Settling maximal	1.1277e+003	1.2265e+003	1.2816e+003
Overshoot (%)	12.7705	22.6508	28.1552
Undershoot (%)	0	0	0
Peak	1.1277e+003	1.2265e+003	1.2816e+003
Peak time	4.9202	4.7657	4.9580



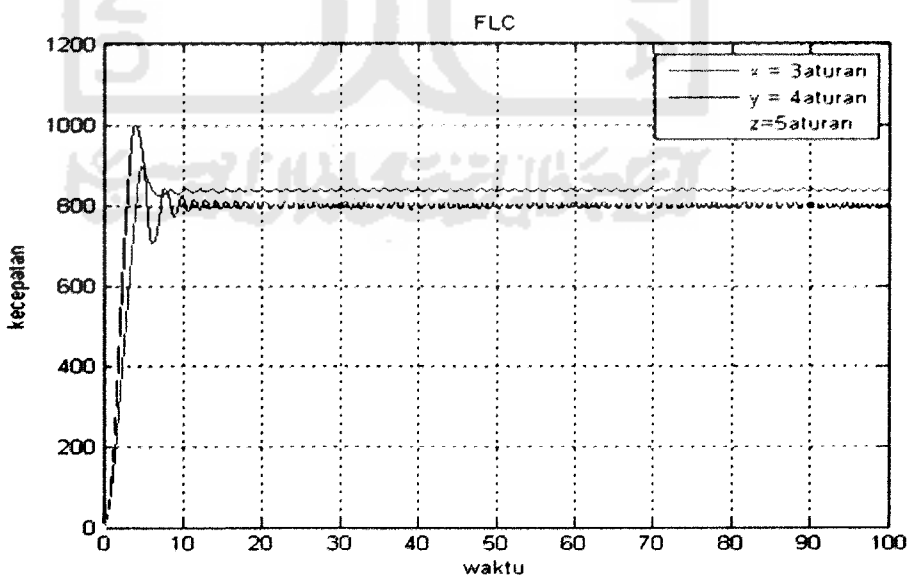
Gambar 4.21 *Step Respon* Pengambilan data 1000 rad / s.

3. Pengambilan data 800 rad / s.

Hasil respon sistem seperti terlihat pada table 4.15.

Tabel 4.15 Respon Sistem Setelah Pelatihan dengan masukan 800 rad / s.

Karakteristik	**Setelah**		
	3 Aturan	4 Aturan	5 Aturan
Rise time	2.7983	1.9207	1.7688
Settling time	8.3013	10.3492	18.1977
Settling minimal	720.2586	705.6440	725.9562
Settling maximal	901.5193	1.0012e+003	1.0614e+003
Overshoot (%)	12.6899	25.1533	32.6688
Undershoot (%)	0	0	0
Peak	901.5193	1.0012e+003	1.0614e+003
Peak time	4.8999	3.9329	3.9390



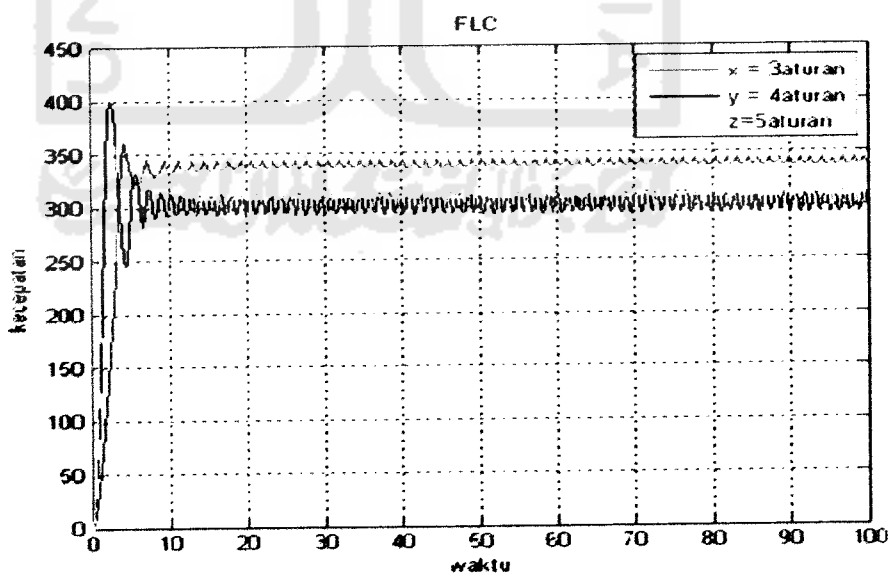
Gambar 4.22 *Step Respon* Pengambilan data 800 rad / s.

4. Pengambilan data 300 rad / s.

Hasil respon sistem seperti terlihat pada table 4.16.

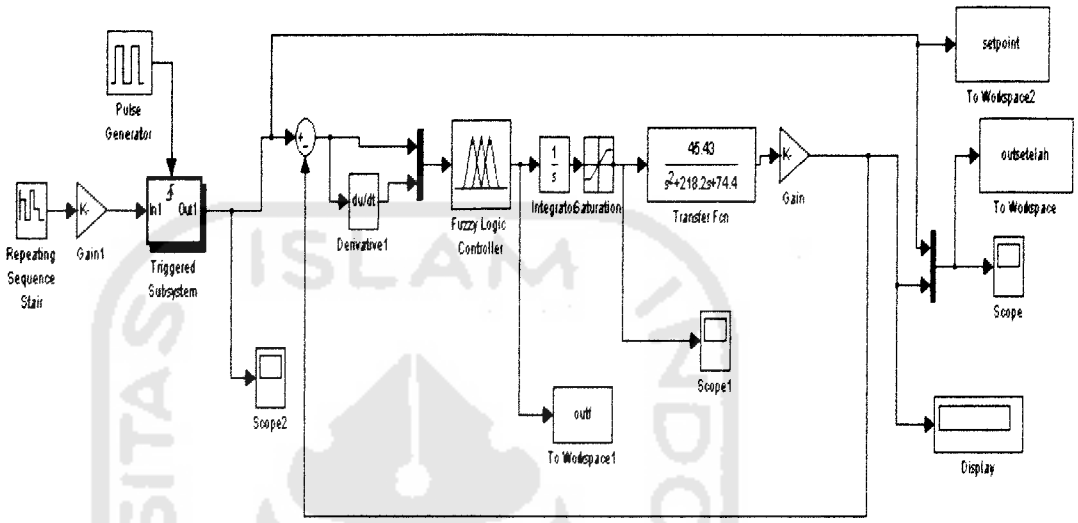
Tabel 4.16 Respon Sistem Setelah Pelatihan dengan masukan 300 rad / s.

Karakteristik	**Setelah**		
	3 Aturan	4 Aturan	5 Aturan
Rise time	2.4619	1.1634	0.9710
Settling time	NaN	NaN	NaN
Settling minimal	270.0671	245.7802	272.3831
Settling maximal	361.2320	400.8773	440.7591
Overshoot (%)	20.4107	33.6258	46.9197
Undershoot (%)	0	0	0
Peak	361.2320	400.8773	440.7591
Peak time	4.3774	2.4820	2.2566

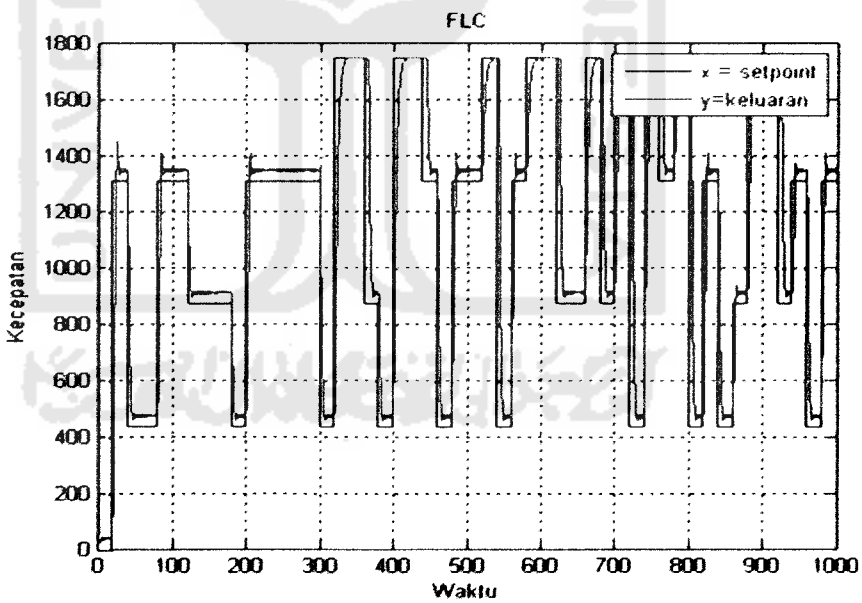


Gambar 4.23 *Step Respon* Pengambilan data 300 rad / s.

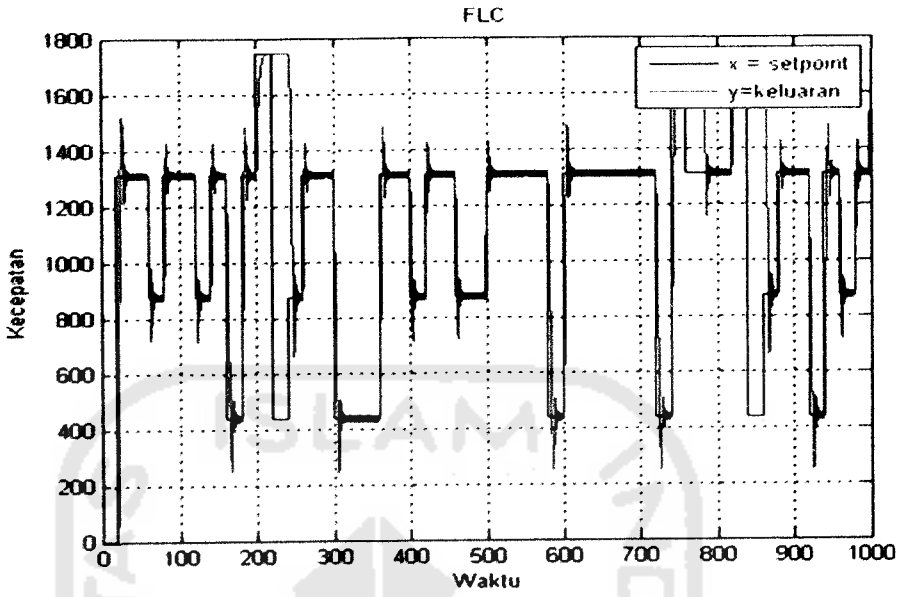
Setelah nilai karakteristik dari respon sistem diketahui, kemudian dilakukan pengujian menggunakan rangkaian simulasi menggunakan gambar 4.24.



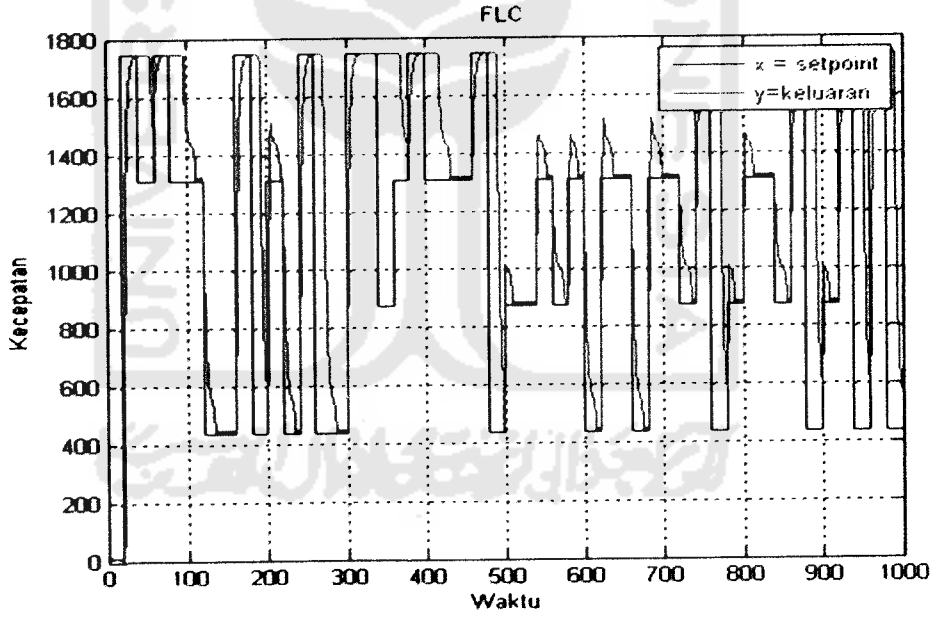
Gambar 4.24 Rangkaian simulasi pengujian terakhir.



Gambar 4.25 Hasil keluaran pengujian dengan 3 aturan



Gambar 4.26 Hasil keluaran pengujian dengan 4 aturan.



Gambar 4.27 Hasil keluaran pengujian dengan 5 aturan.

Pada hasil *step respon* dengan penggunaan 5 aturan menghasilkan respon sistem yang lebih cepat dari penggunaan 3 aturan dan 4 aturan. Berdasarkan hasil respon sistem, kendali *fuzzy* dapat meredam *overshoot* yang dihasilkan sistem pada nilai

akhir 1750, sedangkan pada nilai lain terdapat *overshoot*. Pada Gambar 4.25 diatas, penggunaan 3 aturan, grafik keluaran (berwarna biru) sudah hampir mengikuti grafik *set point* (berwarna merah). Setiap mencapai puncak nilai 1750 tidak ada osilasi, sedangkan pada saat keadaan turun untuk mencapai puncak pada nilai yang lain terdapat osilasi dan pada waktu turun ada beberapa grafik keluaran yang tidak mengikuti dengan grafik *set point*. Begitu juga pada penggunaan 4 aturan dan 5 aturan. Hal ini karena pengaruh dari jangkauan dan perubahan parameter fungsi keanggotaan (masukan dan keluaran) dan aturan *fuzzy* yang digunakan. Namun secara keseluruhan grafik tersebut membuktikan bahwa pengujian sistem sudah cukup baik namun bukan yang terbaik.

Berikut ini adalah tabel perbandingan hasil pengujian sebelum dan setelah pelatihan *fuzzy*, dimana nilai masukan kecepatan dipilih secara bebas.

Tabel 4.17 Perbandingan Pengujian Sebelum dan Setelah Pelatihan

NO	Masukan (rad / s)	Sebelum (rad / s)	Setelah (rad / s)		
			3 Aturan	4 Aturan	5 Aturan
1	1750	-	1750	1750	1750
2	1400	-	1439	1406	1411
3	1000	-	1041	995.4	1011
4	800	-	839	794.4	808.4
5	300	-	338.7	297.3	314.8

Tabel 4.18 Pengujian kecepatan *set point* dengan kecepatan motor setelah pelatihan (3 aturan)

NO	Masukan (rad / s)	Keluaran (rad / s)	Selisih (rad / s)
1	1750	1750	0
2	1400	1439	39
3	1000	1041	41
4	800	839	39.2
5	300	338.7	38.7

Dari tabel 4.18, rata-rata selisih kecepatan setelah pelatihan sebesar 31.58 rad / s.

Tabel 4.19 Pengujian kecepatan *set point* dengan kecepatan motor setelah pelatihan (4 aturan)

NO	Masukan (rad / s)	Keluaran (rad / s)	Selisih (rad / s)
1	1750	1750	0
2	1400	1406	6
3	1000	995.4	4.6
4	800	794.4	5.6
5	300	297.3	2.7

Dari tabel 4.19, rata-rata selisih kecepatan setelah pelatihan sebesar 3.15 rad / s.

Tabel 4.20 Pengujian kecepatan *set point* dengan kecepatan motor setelah pelatihan (5 aturan)

NO	Masukan (rad / s)	Keluaran (rad / s)	Selisih (rad / s)
1	1750	1750	0
2	1400	1411	11
3	1000	1011	11
4	800	808.4	8.4
5	300	314.8	14.8

Dari tabel 4.20, rata-rata selisih kecepatan setelah pelatihan sebesar 7.53 rad / s.

Berdasarkan tabel 4.18, 4.19, dan 4.20 diperoleh nilai rata – rata selisih kecepatan. Nilai tersebut dapat diperoleh dari persamaan berikut :

$$selisih\ rata - rata = \frac{\Sigma \text{ selisih}}{\text{banyaknya data}} \qquad (4.1)$$

Penggunaan 4 aturan, selisih rata – rata kecepatannya lebih kecil dari penggunaan 3 dan 5 aturan. Hasil penelitian ini membuktikan bahwa simulasi dengan sistem automatisasi *fuzzy* metode gradien sebagai kendali kecepatan motor DC sudah cukup baik namun bukan yang terbaik. Penggunaan variasi aturan berpengaruh terhadap perubahan parameter masukan dan keluaran, hasil respon sistem, dan selisih rata – rata kecepatan

Dalam pelatihan tidak menggunakan seluruh data. Data yang digunakan sebanyak 500 data berurutan. Hal ini juga disebabkan keterbatasan komputer yang digunakan saat pelatihan dan simulasi, sehingga dengan data masukan yang banyak, perlu dicoba menggunakan komputer dengan tingkat proses komputasi yang lebih tinggi.