

BAB IV

HASIL DAN ANALISA

4.1. Gambaran Umum Analisa

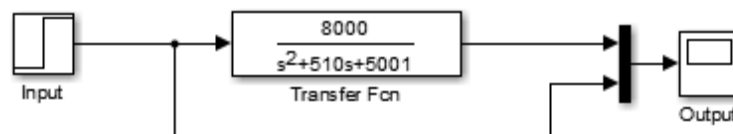
Pada bab sebelumnya telah diketahui langkah-langkah mendesain beserta proses *tunning* pengendali PID dan pengendali Fuzzy Logic pada kecepatan motor DC. Maka, yang dilakukan selanjutnya pada bab ini yaitu menganalisa sistem berupa simulasi. Pada bab ini akan dilakukan analisa menggunakan pengendali PID, Fuzzy Logic dan penggabungan pengendali PID dan Fuzzy Logic. Pengujian ini akan dilakukan menggunakan *software* MATLAB R2014a, dan pengujian ini bermaksud untuk melihat perbandingan perfomansi dari ketiga pengendali tersebut dalam mencapai *setpoint* kecepatan motor DC dan penambahan gangguan. Adapun beberapa analisa yang akan dilakukan pada pengujian ini yaitu :

1. Analisa sistem kecepatan motor DC secara *open loop* (tanpa pengendali).
2. Analisa sistem kecepatan motor DC dengan pengendali PID metode Quarter Decay.
3. Analisa sistem kecepatan motor DC dengan pengendali Fuzzy Logic.
4. Analisa sistem kecepatan motor DC dengan Pengendali Hybrid PID dan Fuzzy Logic.
5. Analisa kekokohan setiap pengendali dengan memberikan Gangguan.

4.2. Analisa Sistem Kecepatan Motor DC Secara Open Loop

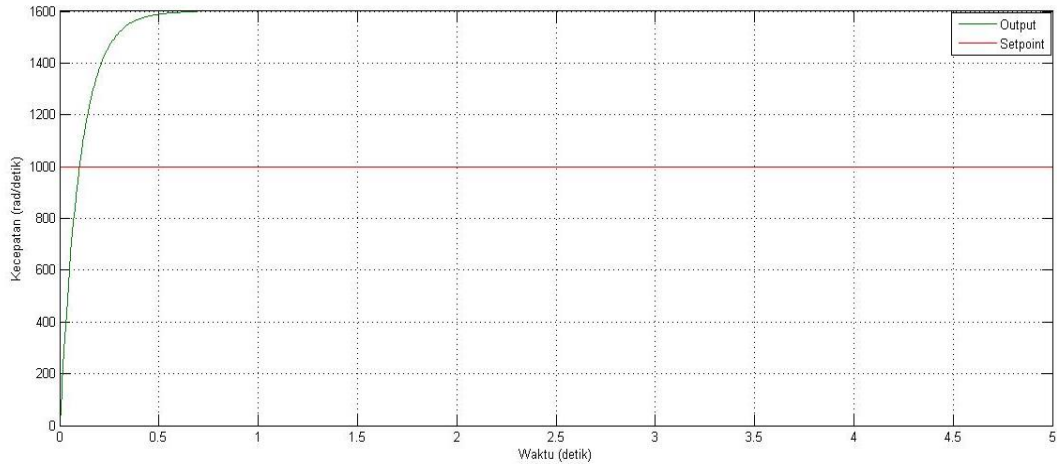
Untuk menganalisa respon keluaran dari sistem kecepatan motor DC maka perlu dilakukan pemodelan terlebih dahulu. Pemodelan dilakukan dengan menggunakan *simulink* di *software* Matlab R2014a, model matematis yang telah didapatkan pada persamaan 3.1 sebelumnya, kemudian dimasukkan ke dalam blok diagram yang ada di *simulink* Matlab, selanjutnya disimulasikan untuk melihat respon keluaran sistem.

Pengujian pertama yang dilakukan adalah pengujian sistem secara *open loop* atau tanpa pengendali. Pengujian ini bertujuan untuk mengetahui karakteristik dari sistem sebelum ditambahkan pengendali. Berikut adalah rangkain *simulink* respon keluaran sistem kecepatan motor DC tanpa pengendali:



Gambar 4.1. Rangkain *Open Loop*

Menghasilkan respon sebagai berikut:



Gambar 4.2 Respon Sistem Secara *Open Loop*

Gambar 4.2. merupakan respon sistem tanpa pengendali dengan *setpoint* sebesar 1000 rad/s dan waktu simulasi yang diberikan adalah 5 detik. Dari respon Sistem tersebut dapat dilihat bahwa *rise time* sistem adalah 0.52761108 detik, *settling time* sebesar 0.39105336 detik, dan *time delay* sebesar 0.07124921 detik, sistem tidak mengalami *overshoot* akan tetapi terdapat *error steady state* sebesar 599 rad/s.

Respon sistem *open loop* tidak mengalami osilasi sehingga dapat dinyatakan bahwa sistem tersebut adalah sistem yang stabil. Disamping tidak adanya *overshoot* pada sistem terdapat *steady state* sistem melebihi *setpoint* sehingga dapat dilihat sistem yang bekerja tidak sesuai dengan apa yang diinginkan, berikut dapat dilihat analisa *respons transient* dari keluarannya:

1. Waktu naik / *Rise time* (*tr*)

Waktu naik adalah waktu yang dibutuhkan oleh sistem untuk naik dari keadaan 5% ke 95%, adapun nilai mendapatkan nilai *tr* dihitung dari pengurangan 95% dan 5%.

$$\begin{aligned} Y(5\%) &= 5\% \times \text{steady state} \\ &= 5\% \times (1.599) \\ &= 77.95 \text{ rad / s} \end{aligned}$$

$Y(5\%)$ adalah *output* sistem saat 5%, langkah selanjutnya yaitu menentukan waktu $Y(5\%)$ yang disimbolkan dengan $t(5\%)$. Keluaran sistem *open loop* melewati nilai *setpoint* yang diberikan sehingga yang dipakai adalah nilai *steady state* dari sistem, adapun nilai yang menunjukkan kondisi respon $Y(5\%)$ dari *steady state* adalah 77.95 rad/s dengan $t(5\%)$ yang akan dilihat pada *workspace*. Namun nilai 77.95 rad/s tidak dijumpai di *workspace* sehingga dilakukan interpolasi dengan mengambil 2 nilai

terdekat dari 77.95 rad/s yaitu 53.81554476 rad/s dan 85.0928778 rad/s menggunakan persamaan (26). Untuk mempermudah penulis, penulis menggunakan Microsoft Excel untuk mendapatkan nilai interpolasinya.

10	0.0034	28.4476	1000
11	0.0053	53.8155	1000
12	0.0074	85.0929	1000
13	0.0100	122.8923	1000

Gambar 4.3. Nilai Y(5%) dari Respon Sistem *Open Loop di Workspace*

Setelah dilakukan interpolasi maka didapatkan $t(5\%)$ adalah:

$$t(5\%) = 0.00694575 \text{detik}$$

$$\begin{aligned} Y(95\%) &= 95\% \times \text{steady state} \\ &= 95\% \times 1.599 \\ &= 1519.05 \text{rad / s} \end{aligned}$$

$Y(95\%)$ adalah *output* sistem saat 95%, langkah selanjutnya yaitu menentukan waktu $Y(95\%)$ yang disimbolkan dengan $t(95\%)$. Nilai yang menunjukkan kondisi respon $Y(95\%)$ dari *steady state* adalah 1519.05 rad/s dengan $t(95\%)$ yang akan dilihat pada *workspace*. Nilai 1519.05 rad/s dapat dijumpai di *workspace* sehingga tidak perlu dilakukan interpolasi.

92	0.5291	1.5915e+03	1000
93	0.5346	1.5919e+03	1000
94	0.5409	1.5924e+03	1000

Gambar 4.4. Nilai 95% dari Respon Sistem *Open loop di Workspace*

Sehingga $t(95\%)$ sebagai berikut:

$$t(95\%) = 0.53455484 \text{detik}$$

Setelah didapatkan nilai pada saat $t(5\%)$ dan (95%) maka dapat ditentukan waktu *rise time* dari sistem secara *open loop* sebagai berikut ini:

$$\begin{aligned} tr &= t(5\%) - t(95\%) \\ &= 0.53455484 - 0.00694375 \\ &= 0.52761109 \text{detik} \end{aligned}$$

2. Waktu Tunak/Settling Time (ts)

Waktu tunak adalah waktu yang menyatakan respon telah masuk kedalam keadaan *steady state* yang biasanya dipersentasekan 2% atau 5% dari *steady state*.

$$\begin{aligned} Y(98\%) &= 98\% \times \text{steady state} \\ &= 1567.02 \text{rad / s} \end{aligned}$$

Y(98%) adalah *output* sistem saat 98%, langkah selanjutnya yaitu menentukan waktu Y(98%) yang disimbolkan dengan t(98%). t(98%) dilihat dari Y(98%) *steady stade* yaitu 1567.02 rad/s, akan tetapi nilai 1267.02 rad/s tidak ditemukan pada *workspace* sehingga diambil 2 nilai yang mendekati yaitu 1566.68166662 rad/s dan 1568.52215426 rad/s, selanjutnya dilakukan interpolasi dengan bantuan Microsoft Exel.

70	0.3819	1.5639e+03
71	0.3900	1.5667e+03
72	0.3958	1.5685e+03
73	0.4015	1.5703e+03

Gambar 4.5. Nilai *Settling Time* dari Respon Sistem *Open Loop* di *Workspace* Sehingga *settling timenya* adalah:

$$\begin{aligned}
 t_s &= t(98\%) \\
 &= 0.39105336 \text{detik}
 \end{aligned}$$

3. Waktu Tunda/*Time Delay*(td)

Time delay merupakan waktu dari setengah nilai *steady stade*, Berikut adalah bentuk perhtungan dari *time delay*:

$$\begin{aligned}
 Y(50\%) &= 50\% \times \textit{steady stade} \\
 &= 799.5 \text{rad / s}
 \end{aligned}$$

Y(50%) adalah *output* sistem saat 50% langkah selanjutnya menentukan waktu Y(50%) yang disimbolkan dengan t(50%), t(50%) dilihat dari Y(50%) *steady stade* yaitu 799.5 rad/s, akan tetapi nilai 799.5 rad/s tidak ditemukan pada *workspace* sehingga diambil 2 nilai yang mendekati yaitu 745.6657308 rad/s dan 792.4222733 rad/s, selanjutnya dilakukan interpolasi dengan bantuan Microsoft Exel.

21	0.0564	670.9628	1000
22	0.0648	745.6657	1000
23	0.0704	792.4223	1000
24	0.0760	836.6252	1000

Gambar 4.6. Nilai *Time Delay* dari Respon Sistem *Open Loop* di *Workspace* Sehingga *time delaynya* adalah:

$$t(50\%) = 0.07124921 \text{detik}$$

4. Maksimum overshoot

Maximum overshoot merupakan nilai puncak dari *overshoot* yang terjadi pada sistem, adapun respon sistem secara *open loop* tidak terjadi mengalami *overshoot* maka nilai m_p adalah 0.

5. Error Steady State

Error steady state merupakan nilai yang melewati *setpoint* yang diberikan, untuk mendapatkan nilai *ess* variabel yang digunakan R_{ss} yaitu nilai *error* dan C_{ss} diartikan sebagai *setpoint* sistem

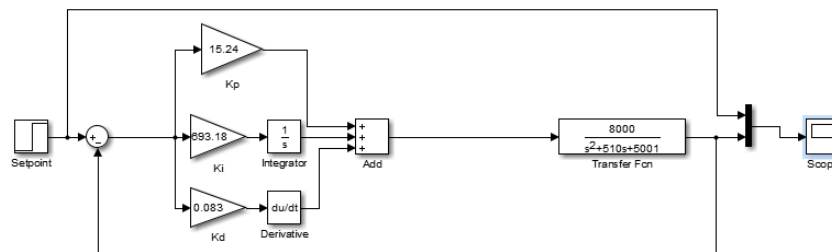
$$\begin{aligned} ess &= R_{ss} - C_{ss} \\ &= 1599 - 1.000 = 599 \text{ rad / s} \end{aligned}$$

Tabel 4.1. Respon Transiens Sistem secara *Open Loop*

Respons Transiens	Parameter Respon
Tr	0.52761109 detik
Ts	0.39105336 detik
Td	0.07124921 detik
Mp	0%
Ess	599 rad/s

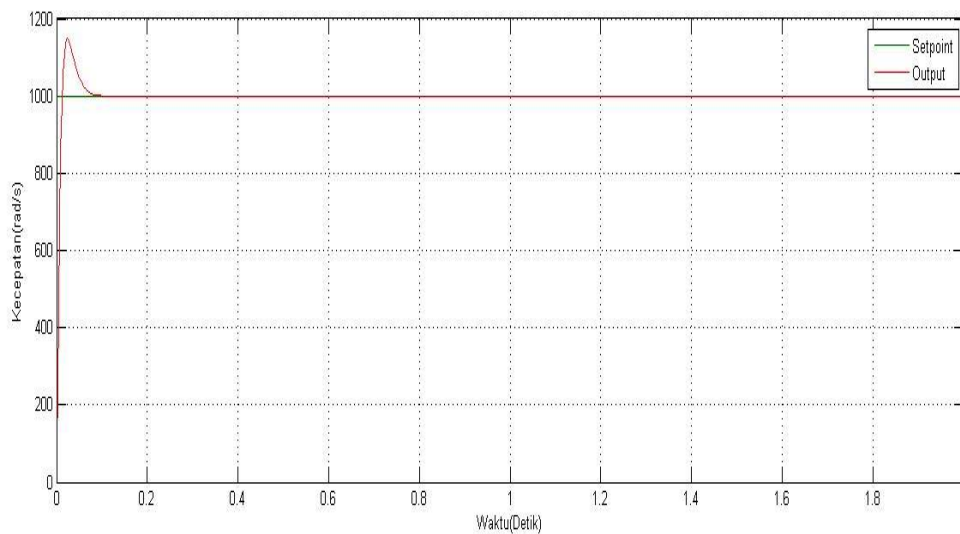
4.2. Analisa Sistem Kecepatan Motor DC dengan Pengendali PID

Setelah melakukan Simulasi secara *open loop* maka langkah selanjutnya yaitu menambahkan pengendali PID yang telah dirancang dan *tunning* menggunakan metode Quarter Decay pada bab sebelumnya, adapun *setpoint* yang penulis tetapkan adalah 1000 rad/s dengan waktu simulasi 2 detik, untuk rangkaian *simulink* dan respon sistemnya dapat dilihat sebagai berikut:



Gambar 4.7. Rangkaian Sistem dengan Pengendali PID

Menghasilkan respon sistem berikut:



Gambar 4.8 Respon Sistem dengan Pengendali PID

Gambar 4.8. merupakan respon sistem dengan pengendali PID. Dari respon Sistem tersebut dapat dilihat bahwa *rise time* sistem adalah 0.105744635 detik, *settling time* sebesar 0.011924547 detik, *time delay* sebesar 0.0049560 detik, sistem mengalami *overshoot* sebesar 14.835%, dan terdapat *error steady state* sebesar 0.17 rad/s.

Metode Quarter Decay menghasilkan respon sistem cukup baik, nilai proposional yang didapatkan dapat dikatakan pas sehingga waktu naik sistem cepat tanpa adanya osilasi karena osilasi tersebut dapat diredam oleh nilai integratif, akan tetapi metode ini menghasilkan nilai integratif yang terlalu besar oleh karena itu *overshoot* yang dihasilkan menjadi besar karena nilai derivatif yang didapatkan belum mampu meminimalisir *overshoot* yang terjadi, akan tetapi respon sistem adalah sistem yang stabil walaupun masih terdapat *error* pada *steady state*. Berdasarkan respon tersebut dapat dianalisa *respons transient* dari keluarannya, berikut identifikasi keluarannya:

1. Waktu naik / *Rise time* (t_r)

$$\begin{aligned}
 Y(5\%) &= 5\% \times \text{steady state} \\
 &= 5\% \times (1000) \\
 &= 50 \text{ rad / s}
 \end{aligned}$$

Keluaran sistem dengan pengendali PID didapatkan nilai *steady state* mengikuti nilai *setpoint* yang diberikan sehingga nilai yang dipakai adalah nilai *setpointnya*, adapun nilai yang menunjukkan kondisi respon $Y(5\%)$ adalah 50 rad/s dan selanjutnya menentukan waktu saat $Y(50\%)$ yaitu $t(5\%)$ yang dilihat pada *workspace*, namun nilai

50 rad/s tidak dijumpai di *workspace* sehingga dilakukan interpolasi dengan mengambil 2 nilai terdekat dari 50 yaitu 29.97127638 rad/s dan 97.49742348 rad/s.

12	3.2435e-04	1000	5.9429
13	7.6783e-04	1000	29.9713
14	0.0015	1000	97.4974
15	0.0027	1000	233.0950

Gambar 4.9. Nilai 5% dari Respon Sistem dengan Pengendali PID

Sehingga $t(5\%)$ adalah:

$$t(5\%) = 0.0069437 \text{ detik}$$

$$\begin{aligned} Y(95\%) &= 95\% \times \text{steady state} \\ &= 95\% \times (1000) \\ &= 950 \text{ rad / s} \end{aligned}$$

Nilai yang menunjukkan kondisi respon $Y(95\%)$ dari *steady state* adalah 950 rad/s dan selanjutnya menentukan waktu saat $Y(50\%)$ yaitu $t(5\%)$. Namun nilai 950 rad/s tidak dijumpai di *workspace* sehingga dilakukan interpolasi dengan mengambil 2 nilai terdekat dari 950 rad/s yaitu 860.6020 rad/s dan 977.966 rad/s.

17	0.0066	1000	661.8988
18	0.0094	1000	860.6020
19	0.0119	1000	977.9660
20	0.0143	1000	1.0562e+03

Gambar 4.10. Nilai 95% dari Respon Sistem dengan Pengendali PID

Sehingga $t(95\%)$ adalah:

$$t(95\%) = 0.0054424 \text{ detik}$$

Setelah nilai $t(5\%)$ dan $t(95\%)$ didapatkan maka dapat ditentukan waktu *rise time* dari sistem sebagai berikut ini:

$$\begin{aligned} tr &= t(95\%) - t(5\%) \\ &= 0.53455484 - 0.0069437 \\ &= 0.5276114 \text{ detik} \end{aligned}$$

2. Waktu Tunak/*Settling Time* (t_s)

$$\begin{aligned} Y(98\%) &= 98\% \times \text{steady state} \\ &= 980 \text{ rad / s} \end{aligned}$$

$t(98\%)$ dilihat dari $Y(98\%)$ *steady state* yaitu 980 rad/s, akan tetapi nilai 980 rad/s tidak ditemukan pada *workspace* sehingga diambil 2 nilai yang mendekati yaitu

977.9660145 rad/s dan 1056.186768 rad/s. Setelah diinterpolasi, maka dapat ditentukan waktu disaat Y(98%) atau t(98%)nya.

18	0.0094	1000	860.6020
19	0.0119	1000	977.9660
20	0.0143	1000	1.0562e+03
21	0.0171	1000	1.1087e+03

Gambar 4.11. Nilai *Settling Time* dari Respon Sistem Sistem dengan Pengendali PID Sehingga didapat *settling timenya* adalah:

$$ts = t(98\%)$$

$$= 0.0119245 \text{ detik}$$

3. Waktu Tunda/*Time Delay* (td)

$$Y(50\%) = 50\% \times \text{steady stade}$$

$$= 500 \text{ rad / s}$$

t(50%) dilihat dari Y(50%) *steady stade* yaitu 500 rad/s, akan tetapi nilai 500 rad/s tidak ditemukan pada *workspace* sehingga diambil 2 nilai yang mendekati yaitu 460.95772970rad/s dan 661.89876449rad/s. Setelah diinterpolasi, maka dapat ditentukan waktu disaat Y(50%) atau t(50%)nya.

15	0.0027	1000	233.0950
16	0.0046	1000	460.9577
17	0.0066	1000	661.8988
18	0.0094	1000	860.6020

Gambar 4.12. Nilai *Time Delay* dari Respon Sistem Sistem dengan Pengendali PID Sehingga nilai *settling timenya* adalah:

$$ts = t(98\%)$$

$$= 0.00495 \text{ detik}$$

4. *Maksimum overshoot*

Adapun untuk mendapatkan nilai dari *Maximum overshoot* dapat dilakukan dengan menggunakan persamaan (25) dengan mengambil nilai tertinggi pada *output* yang ada di *workspace*, dapat dilihat sebagai berikut:

21	0.0171	1000	1.1087e+03
22	0.0200	1000	1.1379e+03
23	0.0230	1000	1.1484e+03
24	0.0262	1000	1.1461e+03

Gambar 4.13. Nilai *Maximum Overshoot* dari Respon Sistem dengan Pengendali PID

Untuk mendapatkan nilai m_p variabel yang digunakan t_p yaitu nilai *overshoot* tertinggi dan ∞ diartikan sebagai *steady state* sistem. Nilai *overshoot* tertinggi yang didapatkan adalah 1148.355826 rad/s tersebut dimasukkan ke dalam rumus:

$$\begin{aligned} m_p &= \frac{c(tp) - c^\infty}{c^\infty} (100\%) \\ &= \frac{1148.3558 - 1000}{1000} 100\% \\ &= 14.83558\% \end{aligned}$$

5. Error Steady State

Adapun untuk mendapatkan *error steady state* dapat dilihat sebagai berikut:

86	0.4375	1000	1.0001e+03
87	0.4455	1000	1.0001e+03
88	0.4568	1000	1.0002e+03
89	0.4630	1000	1.0001e+03

Gambar 4.14. Nilai *Error Steady State* dari Respon Sistem dengan Pengendali PID

Untuk mendapatkan nilai e_{ss} variabel yang digunakan R_{ss} yaitu nilai *error* dan C_{ss} diartikan sebagai *setpoint* sistem. Nilai *error* yang didapatkan adalah 1000.179034 rad/s tersebut dimasukkan ke dalam rumus:

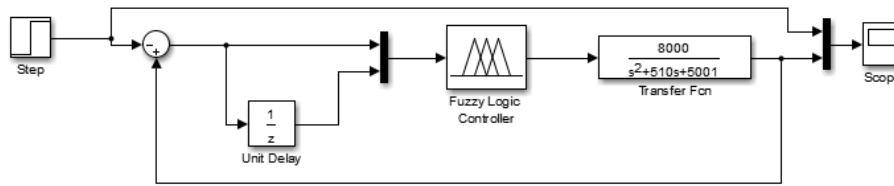
$$\begin{aligned} e_{ss} &= R_{ss} - C_{ss} \\ &= 1000.17903 - 1000 \\ &= 0.17903 \text{ rad / s} \end{aligned}$$

Tabel 4.2. *Respon Transiens* Sistem dengan Pengendali PID

<i>Respons Transiens</i>	Parameter Respon
Tr	0.105744635 detik
Ts	0.011924547 detik
Td	0.004956093 detik
Mp	14.835 %
Ess	0.17 rad/s

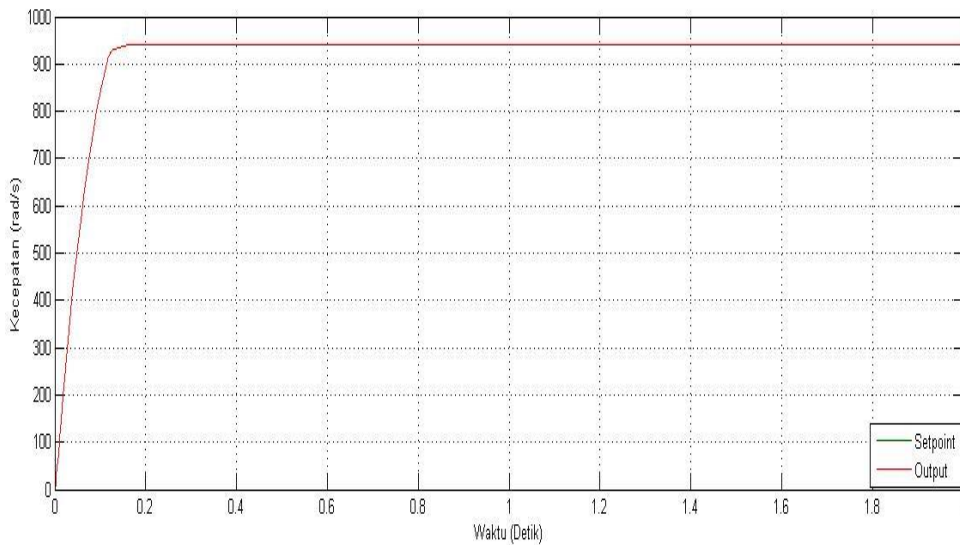
4.4. Analisa Sistem Kecepatan Motor DC dengan Pengendali Fuzzy Logic

Selanjutnya yaitu menambahkan pengendali Fuzzy Logic yang telah dirancang dan *ditunning* menggunakan metode Mamdani pada bab sebelumnya, adapun *setpoint* yang penulis tetapkan adalah 1000 rad/s dengan waktu simulasi 2 detik, untuk rangkaian *simulink* dan respon sistemnya dapat dilihat sebagai berikut:



Gambar 4.15. Rangkain Sistem Kendali Fuzzy Logic

Menghasilkan respon sistem berikut:



Gambar 4.16 Respon Sistem dengan Pengendali Fuzzy Logic

Gambar 4.16. merupakan respon sistem dengan pengendali Fuzzy Logic. Dari respon Sistem tersebut dapat dilihat bahwa *rise time* sistem adalah 0.106802033 detik, *settling time* sebesar 0.1209311 detik, *time delay* sebesar 0.0451557 detik, sistem tidak mengalami *overshoot*, dan terdapat *error steady stade* sebesar 0.0007 rad/s.

Dari Hasil respon terlihat bahwa sistem belum mampu mencapai *setpoint* yang diberikan walaupun sistem tersebut adalah sistem yang stabil, dan juga respon pengendali Fuzzy logic menghasilkan *rise time* yang lebih lama dari pengendali PID. Adapun kelebihan pengendali Fuzzy dari pengendali PID adalah *error steady stade* yang kecil dan tidak mengalami *overshoot*. Berdasarkan respon tersebut dapat dianalisa *respons transient* dari keluarannya, berikut identifikasi keluarannya:

1. Waktu naik / *Rise time* (t_r)

$$\begin{aligned}
 Y(5\%) &= 5\% \times \text{steady stade} \\
 &= 5\% \times (941.627) \\
 &= 47.0813 \text{rad} / \text{s}
 \end{aligned}$$

Keluaran sistem Pengendali Fuzzy Logic tidak mencapai nilai *setpoint* yang diberikan sehingga yang dipakai adalah nilai *steady state* dari sistem, adapun nilai yang menunjukkan kondisi respon $Y(5\%)$ dari *steady state* adalah 47.0813 rad/s dan selanjutnya menentukan waktu saat $Y(5\%)$ yaitu $t(5\%)$, namun nilai 47.0813 rad/s tidak dijumpai di *workspace* sehingga dilakukan interpolasi dengan mengambil 2 nilai terdekat dari 47.0813 rad/s yaitu 26.68502139 rad/s dan 48.77646473rad/s.

8	0.0022	1000	11.7715
9	0.0037	1000	26.6850
10	0.0056	1000	48.7765
11	0.0078	1000	75.8341

Gambar 4.17. Nilai 5% dari Respon Sistem dengan Pengendali Fuzzy Logic
Sehingga $t(5\%)$ adalah:

$$t(5\%) = 0.00544240 \text{ detik}$$

$$\begin{aligned} Y(95\%) &= 95\% \times \text{steady state} \\ &= 95\% \times (941.627) \\ &= 894.54565 \text{ rad / s} \end{aligned}$$

Nilai yang menunjukkan kondisi respon $Y(95\%)$ dari *steady state* adalah 894.54565 rad/s dan selanjutnya menentukan waktu saat $Y(5\%)$ yaitu $t(5\%)$, namun nilai 894.54565 rad/s tersebut tidak dijumpai di *workspace* sehingga dilakukan interpolasi dengan mengambil 2 nilai terdekat dari 894.54565 rad/s yaitu 892 rad/s dan 896 rad/s.

59	0.1117	1000	892.0000
60	0.1117	1000	892.0000
61	0.1126	1000	896.0000
62	0.1126	1000	896.0000

Gambar 4.18. Nilai 95% dari Respon Sistem dengan Pengendali Fuzzy Logic
Sehingga $t(95\%)$ adalah:

$$t(95\%) = 0.11224443 \text{ detik}$$

Setelah nilai $t(5\%)$ dan $t(95\%)$ didapatkan maka dapat ditentukan waktu *rise time* dari sistem sebagai berikut ini:

$$\begin{aligned} tr &= t(95\%) - t(5\%) \\ &= 0.11224443 - 0.00544240 \\ &= 0.106802033 \text{ detik} \end{aligned}$$

2. Waktu Tunak/*Settling Time* (ts)

$$Y(98\%) = 98\% \times \text{steady state} \\ = 922.794 \text{ rad / s}$$

t(98%) dilihat dari Y(98%) *steady state* yaitu 922.794 rad/s, akan tetapi nilai 922.794 rad/s tidak ditemukan pada *workspace* sehingga diambil 2 nilai yang mendekati yaitu 920 rad/s dan 923.24233899rad/s dan dilakukan interpolasi. Setelah diinterpolasi, maka dapat ditentukan waktu disaat Y(98%) atau t(98%)nya.

74	0.1194	1000	920.0000
75	0.1194	1000	920.0000
76	0.1212	1000	923.2423
77	0.1217	1000	924.0000

Gambar 4.19. Nilai *Settling Time* dari Respon Sistem dengan Pengendali Fuzzy Logic

maka dapat ditentukan waktu *settling time* dari sistem yaitu sebagai berikut ini:

$$ts = t(98\%)$$

$$ts = 0.12093112 \text{ detik}$$

3. Waktu Tunda/*Time Delay*(td)

$$Y(50\%) = 50\% \times \text{steadystate} \\ = 470.8135 \text{ rad / s}$$

t(50%) dilihat dari Y(50%) *steady state* yaitu 470.8135 rad/s, akan tetapi nilai 470.8135 rad/s tidak ditemukan pada *workspace* sehingga diambil 2 nilai yang mendekati yaitu 467.76902233 rad/s dan 512.94312580 rad/s dan dilakukan interpolasi. Setelah diinterpolasi, maka dapat ditentukan waktu disaat Y(50%) atau t(50%)nya.

17	0.0395	1000	420.1500
18	0.0448	1000	467.7690
19	0.0501	1000	512.9431
20	0.0572	1000	569.8132

Gambar 4.20. Nilai *Time Delay* dari Respon Sistem dengan Pengendali Fuzzy Logic

Maka didapatkan *time delay* dari sistem sebagai berikut:

$$td = t(50\%)$$

$$= 0.04515577 \text{ detik}$$

4. Maksimum overshoot

Maximum overshoot merupakan nilai puncak dari *overshoot* yang terjadi pada sistem, adapun respon sistem pengendali Fuzzy Logic tidak terjadi mengalami *overshoot* maka nilai *mp* adalah 0.

5. Error Steady State

Adapun untuk mendapatkan *error steady state* dapat dilihat sebagai berikut:

176	0.7804	1000	941.6073
177	0.7866	1000	941.6234
178	0.7931	1000	941.6279
179	0.8011	1000	941.6261

Gambar 4.21. Nilai *ess* dari Respon Sistem dengan Pengendali Fuzzy Logic
Nilai *error* yang didapatkan adalah 941.6279 tersebut dimasukkan ke dalam rumus menggunakan variabel *Rss* dan *Css*, untuk *Css* pada sistem ini menggunakan *steady state* karena sistem ini tidak mencapai *setpoint*:

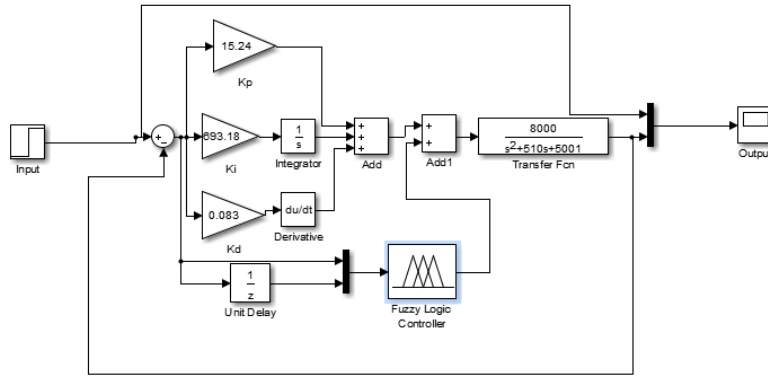
$$\begin{aligned} ess &= Rss - Css \\ &= 941.6279 - 941.6277 \\ &= 0.0007 \text{ rad / s} \end{aligned}$$

Tabel 4.3. Respon Transiens Sistem dengan Pengendali Fuzzy

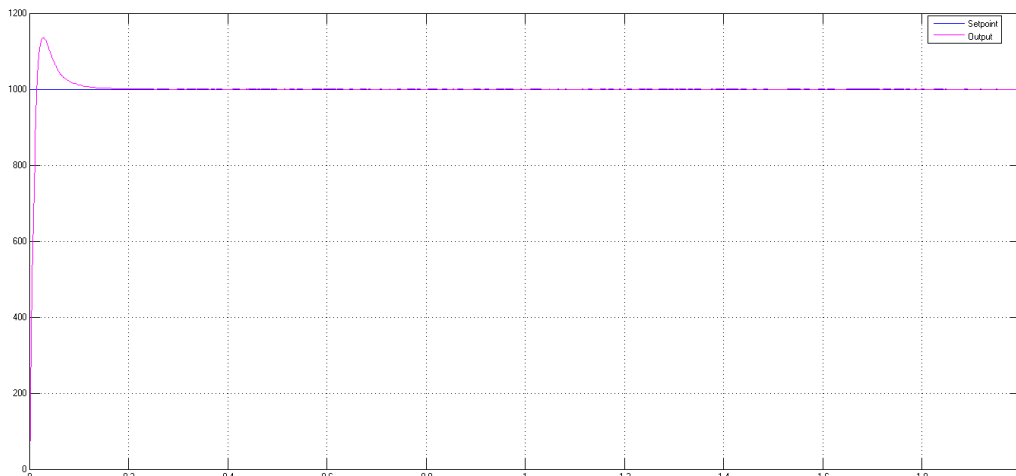
Respons Transiens	Parameter Respon
Tr	0.106802033 detik
Ts	0.120931122 detik
Td	0.04515577 detik
Mp	0%
Ess	0.0007 rad/s

4.5. Analisa Sistem Kecepatan Motor DC dengan Hybrid PID dan Fuzzy Logic

Selanjutnya yaitu menggabungkan pengendali PID dan pengendali Fuzzy Logic yang telah dirancang pada bab sebelumnya, adapun *setpoint* yang penulis tetapkan adalah 1000 rad/s dengan waktu simulasi 2 detik, untuk rangkain *Simulink* respon sistemnya dapat dilihat sebagai berikut:



Gambar 4.22. Rangkaian Sistem Kendali Hybrid PID dan Fuzzy Logic



Gambar 4.23 Respon Sistem dengan Pengendali Hybrid PID dan Fuzzy Logic

Gambar 4.23. merupakan respon sistem dengan pengendali Hybrid PID dan Fuzzy Logic. Dari respon Sistem tersebut dapat dilihat bahwa *rise time* sistem adalah 0.12548523 detik, *settling time* sebesar 0.0143862 detik, *time delay* sebesar 0.005894 detik, sistem mengalami *overshoot* sebesar 13.631%, dan tidak terdapat *error steady state*.

Dari Hasil respon terlihat bahwa sistem mampu mencapai *setpoint* yang ditentukan, respon sistem memiliki *overshoot* yang lebih kecil dari pada pengendali PID Quarter Decay, tidak ada *error steady state*, dan respon sistem juga menunjukkan respon yang stabil. Efek penggabungan kedua pengendali ini adalah pengendali Fuzzy dapat mengurangi *overshoot* yang dihasilkan kelebihan nilai integratif dari metode Quarter Decay dan menghilangkan *error steady state*, akan tetapi penggabungan tersebut sedikit memperlambat waktu naik sistem walaupun tidak terlalu signifikan.

Berdasarkan respon tersebut dapat dianalisa *respons transient* dari keluarannya, berikut identifikasi keluarannya:

1. Waktu naik / *Rise time* (tr)

$$\begin{aligned}
 Y(5\%) &= 5\% \times \text{steady state} \\
 &= 5\% \times (1000) \\
 &= 50 \text{ rad / s}
 \end{aligned}$$

Keluaran sistem dengan pengendali Hybrid didapatkan nilai *steady state* mengikuti nilai *setpoint* yang diberikan sehingga nilai yang dipakai adalah nilai *setpoint*nya, adapun nilai yang menunjukkan $Y(5\%)$ dari *setpoint* adalah 50 rad/s dan selanjutnya menentukan waktu saat $Y(5\%)$ yaitu $t(5\%)$, Namun nilai 50 rad/s tidak dijumpai di *workspace* sehingga dilakukan interpolasi dengan mengambil 2 nilai terdekat dari 50 rad/s yaitu 45.86165116 dan 104.82042570 rad/s.

13	5.4173e-04	1000	15.8535
14	9.8997e-04	1000	45.8617
15	0.0017	1000	104.8204
16	0.0026	1000	196.6604

Gambar 4.24. Nilai 5% dari Respon Sistem dengan Pengendali Hybrid

Sehingga $t(5\%)$ adalah:

$$t(5\%) = 0.001036801 \text{ detik}$$

$$\begin{aligned}
 Y(95\%) &= 95\% \times \text{steady state} \\
 &= 95\% \times (1000) \\
 &= 950 \text{ rad / s}
 \end{aligned}$$

Nilai yang menunjukkan kondisi respon $Y(95\%)$ dari *steady state* adalah 950 rad/s dan selanjutnya menentukan waktu saat $Y(95\%)$ yaitu $t(95\%)$, namun nilai 950 rad/s tidak dijumpai di *workspace* sehingga dilakukan interpolasi dengan mengambil 2 nilai terdekat dari 950 rad/s yaitu 948 rad/s dan 952 rad/s.

87	0.0135	1000	948.0000
88	0.0135	1000	948.0000
89	0.0136	1000	952.0000
90	0.0136	1000	952.0000
91	0.0137	1000	956

Gambar 4.25. Nilai 95% dari Respon Sistem dengan Pengendali Hybrid

Sehingga $t(95\%)$ adalah:

$$t(95\%) = 0.01358532 \text{ detik}$$

Setelah nilai $t(5\%)$ dan $t(95\%)$ didapatkan maka dapat ditentukan waktu *rise time* dari sistem sebagai berikut ini:

$$\begin{aligned}
 tr &= t(95\%) - t(5\%) \\
 &= 0.01358532 - 0.00103680 \\
 &= 0.1254852 \text{ detik}
 \end{aligned}$$

2. Waktu Tunak/*Settling Time* (t_s)

$$Y(98\%) = 98\% \times \text{steady state} \\ = 980 \text{ rad / s}$$

$t(98\%)$ dilihat dari $Y(98\%)$ *steady state* yaitu 980 rad/s dengan nilai yang dapat dilihat pada *workspace*, Nilai 980 rad/s dapat dijumpai di *workspace* sehingga tidak perlu dilakukan interpolasi. Setelah diinterpolasi, maka dapat ditentukan waktu disaat $Y(98\%)$ atau $t(98\%)$ nya. .

102	0.0143	1000	976.0000
103	0.0144	1000	980.0000
104	0.0144	1000	980.0000
105	0.0145	1000	984.0000
106	0.0145	1000	984.0000

Gambar 4.26. Nilai *Settling Time* dari Respon Sistem dengan Pengendali Hybrid.

Adapun *time settling*nya adalah:

$$t_s = t(98\%) \\ = 0.01438624 \text{ detik}$$

3. Waktu Tunda/*Time Delay*(t_d)

$$Y(50\%) = 50\% \times \text{steady state} \\ = 500 \text{ rad / s}$$

$t(50\%)$ dilihat dari $Y(50\%)$ *steady state* yaitu 500 rad/s, akan tetapi nilai 500 rad/s tidak ditemukan pada *workspace* sehingga diambil 2 nilai yang mendekati yaitu 497.16951316 rad/s dan 659.738289291 rad/s dan diinterpolasi. Setelah diinterpolasi, maka dapat ditentukan waktu disaat $Y(50\%)$ atau $t(50\%)$ nya.

17	0.0040	1000	339.6501
18	0.0059	1000	497.1695
19	0.0081	1000	659.7383
20	0.0108	1000	821.5297

Gambar 4.27. Nilai *Time Delay* dari Respon Sistem dengan Pengendali Hybrid

Sehingga *time delay*nya adalah :

$$t_d = t(50\%) \\ = 0.00589246 \text{ detik}$$

4. *Maximum overshoot*

Adapun untuk mendapatkan nilai dari *Maximum overshoot* dapat dilakukan dengan menggunakan persamaan (25) dengan mengambil nilai tertinggi pada *output* yang ada di *workspace*, dapat dilihat sebagai berikut:

363	0.0275	1000	1.1354e+03
364	0.0295	1000	1.1356e+03
365	0.0321	1000	1132
366	0.0321	1000	1.1320e+03

Gambar 4.28. Nilai mp dari Respon Sistem dengan Pengendali Hybrid

Nilai tertinggi yang didapatkan adalah 1135.6315 rad/s tersebut dimasukkan ke dalam rumus menggunakan variabel tp dan ∞ :

$$\begin{aligned}
 mp &= \frac{c(tp) - \infty}{c\infty} (100\%) \\
 &= \frac{1135.631 - 1000}{1000} \times 100\% \\
 &= 13.631\%
 \end{aligned}$$

5. Error Steady State

Adapun untuk mendapatkan *error steady state* dapat dilihat sebagai berikut:

715	0.3626	1000	1.0000e+03
716	0.3626	1000	1.0000e+03
717	0.3626	1000	1000.0000
718	0.3626	1000	1000.0000

Gambar 4.29. Nilai ess dari Respon Sistem dengan Pengendali Hybrid

Nilai *error* yang didapatkan adalah 1000 rad/s tersebut dimasukkan ke dalam rumus dengan variabel Rss dan C_{ss}:

$$\begin{aligned}
 ess &= R_{ss} - C_{ss} \\
 &= 1000 - 1000 \\
 &= 0
 \end{aligned}$$

Tabel 4.4. Respon Transiens Sistem dengan Pengendali Hybrid PID dan Fuzzy Logic

Respons Transiens	Parameter Respon
Tr	0.12548523 detik
Ts	0.014386249 detik
Td	0.005892468 detik
Mp	13.631 %
Ess	0

Tabel 4.5. Perbandingan Pengendali

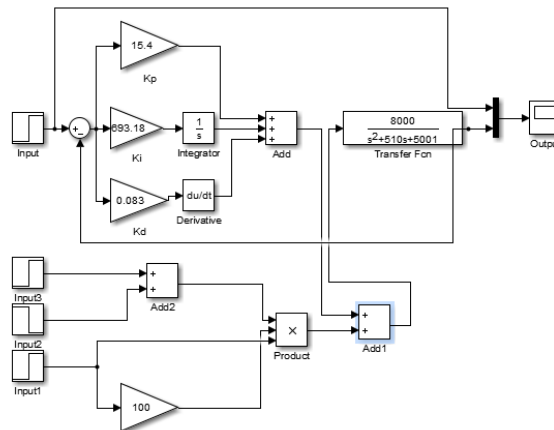
Pengendali	Rise Time	Maximum Overshoot	Error Steady State
PID	0.105744635 detik	14.835 %	0.17
Fuzzy	0.106802033 detik	0%	0.0007
PID-Fuzzy	0.12548523 detik	13.631%	0

Berdasarkan Tabel 4.5. dapat dilihat bahwa pengendali yang memiliki respon sistem yang paling bagus adalah pengendali Hybrid PID dan Fuzzy Logic karena memiliki

overshoot yang paling kecil yaitu sebesar 13% dan tidak mengalami *error steady state* walaupun memiliki rise time yang paling lambat daripada pengendali PID Quarter Decay dan pengendali Fuzzy Logic.

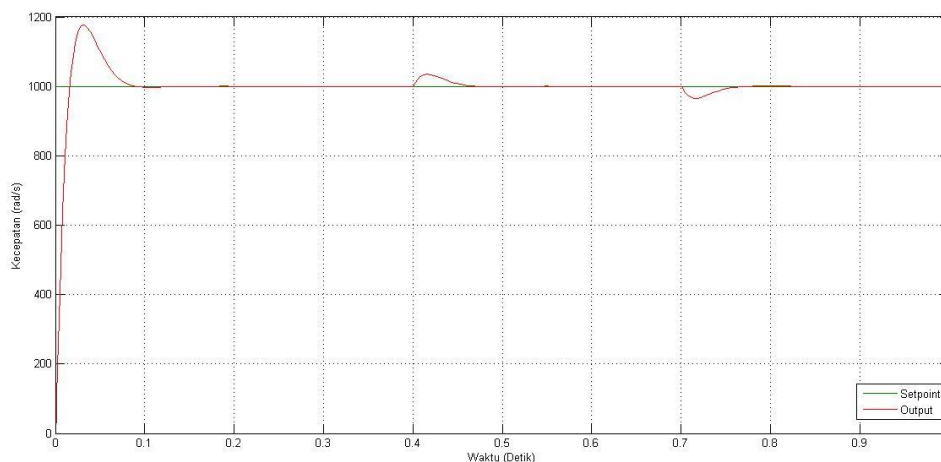
4.6. Analisa kekokohan setiap pengendali dengan memberikan Gangguan.

Pengujian selanjutnya yaitu menguji performa pengendali PID, Fuzzy Logic, dan Hybrid PID dan Fuzzy Logic pada pengendalian kecepatan motor DC dengan menambahkan gangguan pada sinyal kendali. Gangguan pada motor DC didefinisikan sebagai perubahan beban yang naik dan turun [34]. Gangguan yang diberikan sebesar 10% dari *setpoint* yang diberikan yang mana *setpoint*nya adalah 1000 rad/s maka 10% dari 1000 rad/s adalah 100 rad/s. Sinyal gangguan diberikan pada detik 0.4 sampai 0.7 dari waktu simulasi 1 detik, kemudian menganalisa dampak dan perubahan respon sistem dari pengendali tersebut. Berikut adalah analisa dari sistem dengan penambahan sinyal kendali 10%:



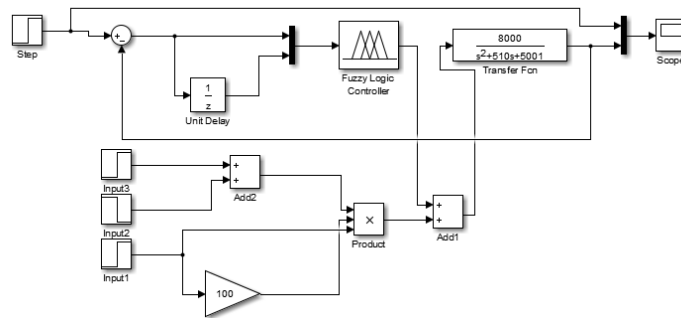
Gambar 3.30 Rangkaian Pengendali PID dengan Gangguan

Menghasilkan respon sistem berikut:



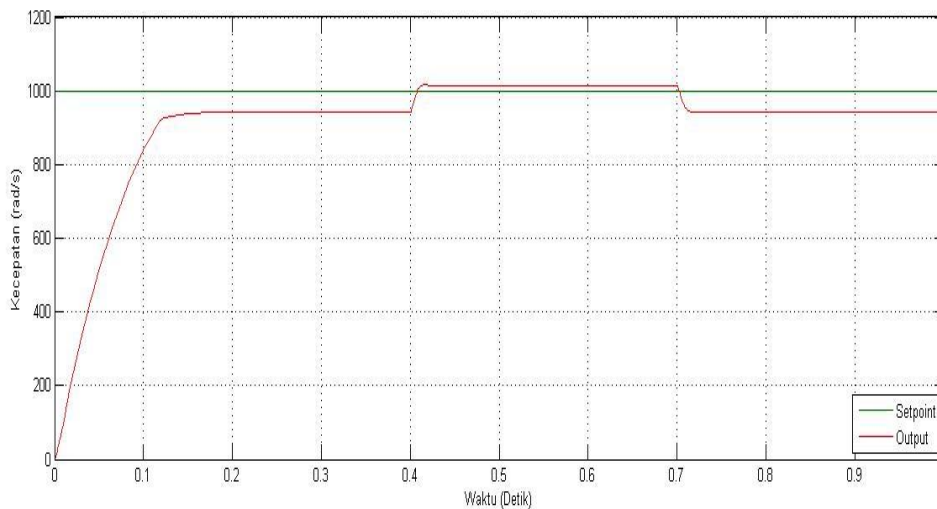
Gambar 4.31 Respon Sistem PID dengan Gangguan 10%

Gambar 4.31 merupakan hasil respon sistem dari pengendali PID yang ditambahkan dengan gangguan sebesar 10% dari *setpoint*, dari respon tersebut dapat dilihat pada saat diberi gangguan pada detik ke 0.4 dan 0.7 terjadi lonjakan *overshoot* sebesar 34.502 rad/s dari kondisi *steady state* sistem 1000 rad/s. Sistem masih dalam keadaan stabil saat diberikan gangguan karena tidak terjadi osilasi dan dalam mempertahankan kondisi *steady state*.



Gambar32 Rangkain Fuzzy Logic dengan Gangguan 10%

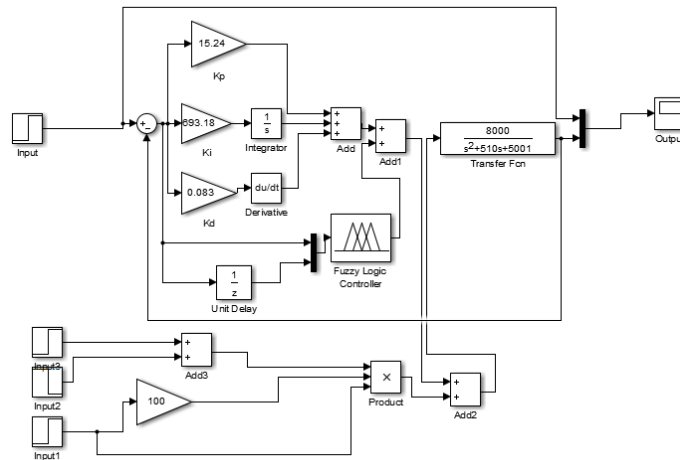
Menghasilkan respon sistem berikut:



Gambar 4.33 Respon Sistem Fuzzy Logic dengan Gangguan 10%

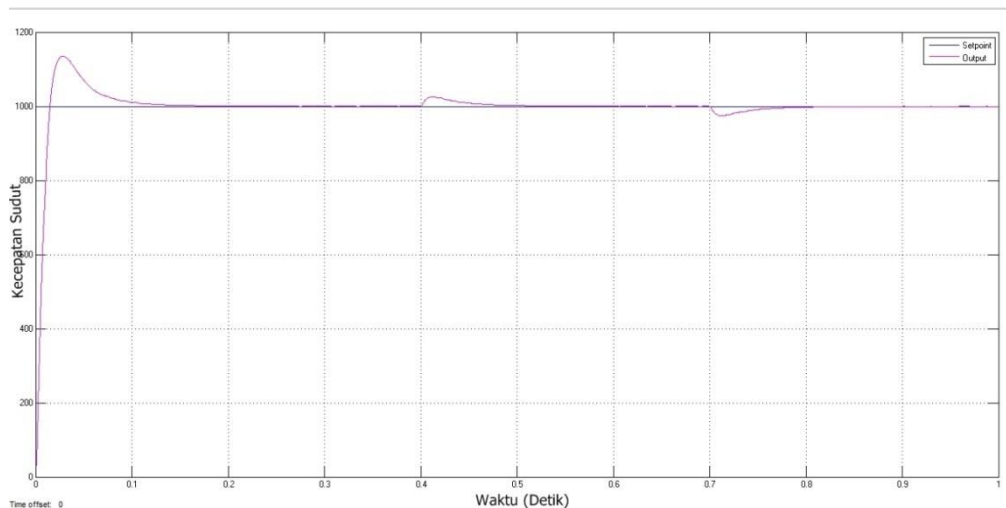
Gambar 4.33 merupakan hasil respon sistem dari pengendali Fuzzy Logic yang ditambahkan dengan gangguan sebesar 10% dari *setpoint*, dari respon tersebut dapat dilihat pada saat diberi gangguan terjadi perubahan keadaan *steady state* pada detik ke 0.4 sampai detik 0.7 hingga baru kembali keadaan *steady state* sebelumnya, adapun keadaan *steady state* saat di detik 0.4 sampai 0.7 adalah 1013,879 rad/s dari keadaan *steady state* 941.627

rad/s. Sistem tidak mengalami *overshoot* dan osilasi tapi sistem tidak stabil karena nilai *steady state*nya berubah-ubah.



Gambar 3.34. Rangkaian Pengendali Hybrid dengan Gangguan 10%

Menghasilkan respon sistem berikut:



Gambar 4.35 Respon Sistem Hybrid PID dan Fuzzy Logic dengan Gangguan 10%

Gambar 4.35 merupakan hasil respon sistem dari pengendali Hybrid PID dan Fuzzy Logic yang ditambahkan dengan gangguan sebesar 10% dari *setpoint*, dari respon tersebut dapat dilihat pada saat diberi gangguan pada detik ke 0.4 dan 0.7 terjadi lonjakan *overshoot* sebesar 24 rad/s dari kondisi *steady state* sistem 1000 rad/s. Sistem masih dalam keadaan stabil saat diberikan gangguan karena tidak terjadi osilasi dan dalam mempertahankan kondisi *steady state*.

Dari setiap respon sistem dengan gangguan 10% didapatkan respon sistem yang paling bagus adalah dengan menggunakan pengendali Hybrid PID dan Fuzzy Logic karena pengendali ini memiliki *overshoot* paling kecil di detik 0.4 dan 0.7 yaitu sebesar 24 rad/s

dan dapat mempertahankan keadaan *steady state* saat ditambahkan gangguan serta stabil. Dari Pengujian ini pengendali Hybrid PID dan Fuzzy Logic adalah Pengendali yang cocok digunakan pada sistem kecepatan motor DC.

