

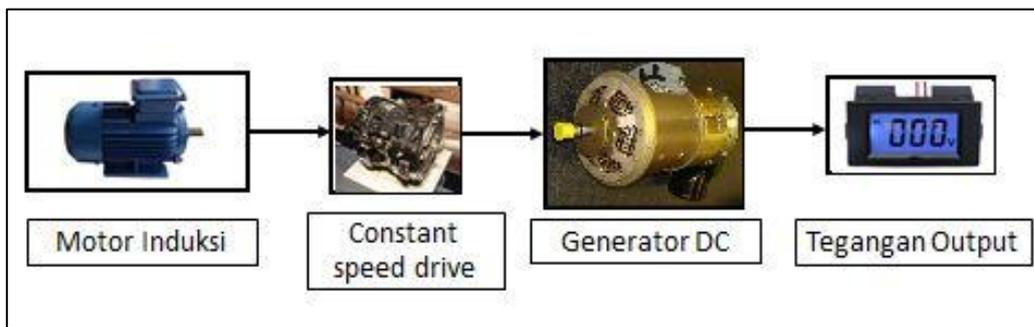
BAB III

METODE PENELITIAN

Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan informasi dengan tujuan dan manfaat tertentu. Dalam menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode penelitian pada skripsi ini meliputi:

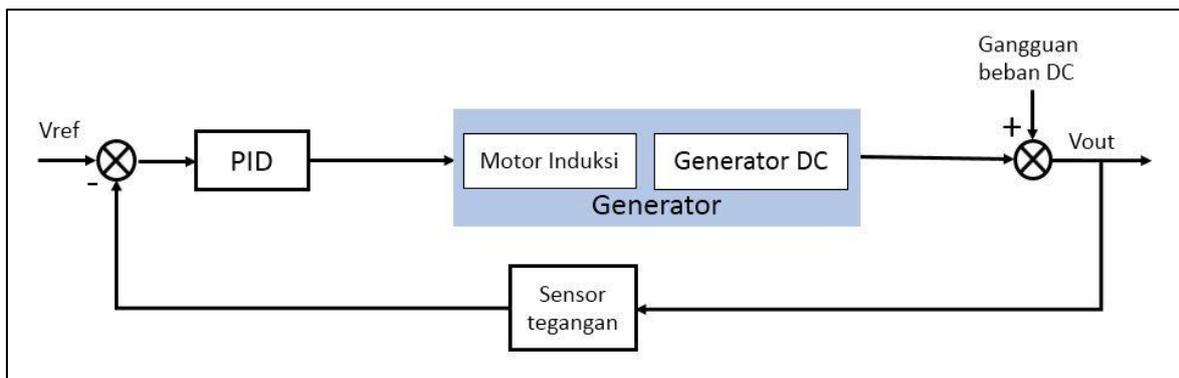
3.1 Penyusunan Model Sistem

Untuk blok diagram sistem untai terbuka dapat dilihat pada Gambar 3.1.



Gambar 3 . 1 Blok Diagram Untai Terbuka

Sedangkan untuk blok diagram sistem untai tertutup dapat dilihat pada Gambar 3.2.



Gambar 3 . 2 Blok Diagram Untai Tertutup

Langkah-langkah yang dilakukan untuk mendapatkan model sistem adalah sebagai berikut:

1. Menentukan sistem yang akan diteliti dan komponen-komponen yang menyusun sistem tersebut sesuai data yang didapatkan.

- Setelah diketahui komponen-komponen penyusun sistem beserta parameter-parameternya, maka sistem dapat dimodelkan menjadi sebuah persamaan matematis yang sesuai. Dengan memasukkan data-data sistem pada persamaan matematis tersebut, dan dengan transformasi Laplace, maka didapatkan fungsi alih sistem.

3.2 Spesifikasi Desain

Spesifikasi desain yang diinginkan pada perancangan dan simulasi sistem yaitu:

- Error steady state* < 3,5174 %, karena sistem yang baik memiliki respon dengan batas nilai akhir 3,5174 % dari *setpoint*.
- Maximum overshoot* < 12.5%, karena jika melebihi batas *overshoot* pada nilai tegangan bisa membahayakan sistem kelistrikan pesawat secara keseluruhan.
- Settling time* < 7 detik, karena *settling time* diharapkan lebih cepat daripada sebelum diberi kontroler yaitu < 7 detik.
- Recovery time* < 10 detik, karena *Recovery time* diharapkan lebih cepat agar sistem dapat kembali dengan semula.

Table 3. 1 Nilai Parameter Sistem

| Parameter | Nilai | Minimum | Maksimum |
|-----------|---------|---------|----------|
| K_A | 20.0000 | 10.0000 | 400.0000 |
| T_A | 0.0600 | 0.0200 | 0.1000 |
| K_E | 1.0000 | - | - |
| T_E | 0.2000 | - | - |
| K_G | 0.8000 | 0.7000 | 1.0000 |
| T_G | 1.5000 | 1.0000 | 2.0000 |
| K_R | 1.0000 | - | - |
| T_R | 0.0250 | - | - |

3.3 Pemodelan Matematis Sistem

Pemodelan (*modelling*) merupakan deskripsi dari prediksi kelakuan sistem atau komponen. Model matematis sistem diperlukan untuk simulasi yang bertujuan untuk mendapatkan respon dinamik sistem.

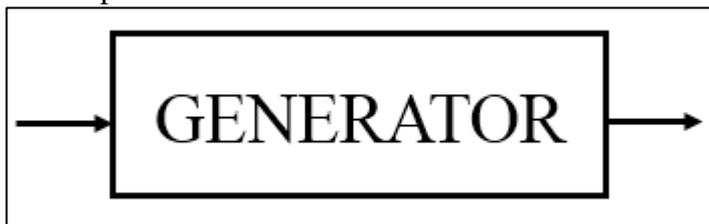
Jika model matematis sistem telah diketahui, maka grafik tanggapan sistem dapat diketahui pula dengan jalan melakukan simulasi terhadap model sistem tersebut. Kemudian langkah berikutnya adalah proses validasi. Pada proses ini data hasil simulasi tersebut

dibandingkan dengan data yang berasal dari proses yang sebenarnya. Jika model tersebut telah cocok (*valid*), proses dapat dilanjutkan ke arah desain dan *prototype*.

Proses penurunan model matematis sistem, terdapat tiga macam cara pendekatan yang dapat dipergunakan. Yang pertama adalah menyajikan model matematis sistem dengan menggunakan persamaan diferensial. Yang kedua adalah dengan menggunakan pendekatan fungsi alih (*transfer function*). Yang ketiga adalah pendekatan ruang keadaan (*state space*). Dalam masalah kendali PID, akan lebih menguntungkan jika menggunakan pendekatan fungsi alih.

3.3.1 Fungsi Alih dan Blok Diagram Generator DC

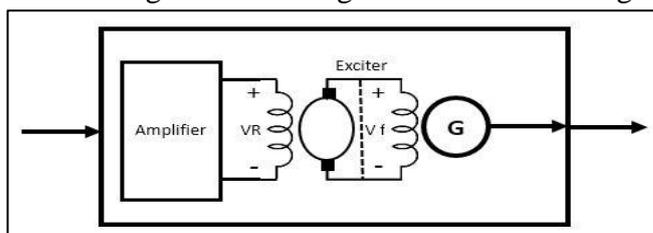
Sistem pasokan listrik arus searah sebagai penguatan pada generator listrik atau sebagai pembangkit medan magnet, sehingga suatu generator dapat menghasilkan energi listrik dengan besar tegangan keluaran generator bergantung pada besarnya arus eksitasinya. Hal ini dikenal sebagai sistem eksitasi (Graham, R: 1999). Diagram blok sistem generator dapat dilihat pada Gambar 3.3



Gambar 3 . 3 Blok Sistem Generator

Sumber Gunadin, 2016

Sedangkan untuk diagram blok Subsistem generator dapat dilihat pada Gambar 3.4.



Gambar 3 . 4 Blok Subsistem Generator

Sumber Gunadin, 2016

Kekokohan tanggapan tegangan sistem eksitasi generator merupakan kemampuan dari sistem eksitasi generator untuk meredam derau pada frekuensi tinggi, mempunyai tanggapan yang cepat terhadap masukan tertentu dan mampu menghilangkan gangguan pada saat beroperasi. (Skogestad, S. dan Postlethwaite: 1996).

Analisa kekokohan ini akan diamati tingkat kekokohan tanggapan tegangan sistem eksitasi generator terhadap perubahan parameter dari komponen – komponen sistem eksitasi. Adapun perubahan parameter yang diamati adalah perubahan konstanta penguatan generator

terhadap tanggapan tegangan sistem eksitasi. Untuk perubahan parameter komponen sistem eksitasi yang lain tidak dilakukan.

Pemodelan matematis sistem eksitasi generator yang dibahas meliputi pemodelan amplifier, pemodelan eksiter dan pemodelan generator. Tipe sistem eksitasi generator yang digunakan adalah sistem eksitasi generator tipe arus searah dengan umpan balik satu dengan bentuk diagram blok. Selain itu pemodelan matematis sistem eksitasi generator ini dilakukan dengan menggunakan persamaan linear diferensial dan transformasi Laplace. Hasil pemodelan masing – masing komponen ini berupa fungsi alih orde satu Fungsi alih dari masing-masing komponen ini kemudian digabungkan dan diperoleh fungsi alih lingkaran terbuka dan fungsi alih lingkaran tertutup dari sistem eksitasi generator.

Bagian ini membahas tentang pemodelan matematis komponen – komponen sistem eksitasi generator yang meliputi model amplifier, model eksiter dan model generator. Untuk model amplifier dinyatakan dalam bentuk persamaan(3-1).

$$\frac{V_R(s)}{V_E(s)} = \frac{K_A}{1+\tau_A s} \dots\dots\dots (3-1)$$

Nilai konstanta penguatan dari amplifier memiliki rentang nilai dari 10.0000 sampai 400.000 sedangkan nilai konstanta waktu amplifier memiliki rentang nilai dari 0.0200 detik sampai 0.1000 detik. Untuk model eksiter dinyatakan dalam bentuk persamaan(3-2).

$$\frac{V_F(s)}{V_R(s)} = \frac{K_E}{1+\tau_E s} \dots\dots\dots (3-2)$$

Untuk konstanta penguatan eksiter dan konstanta waktu eksiter ini mempunyai nilai yang kecil. Untuk model generator dinyatakan dalam bentuk persamaan(3-3).

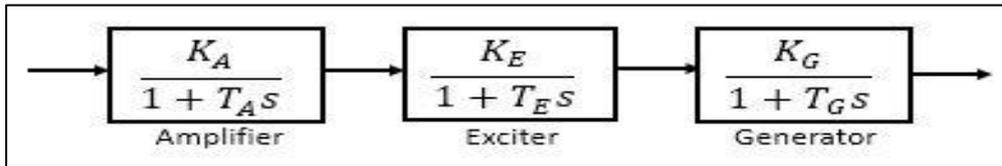
$$\frac{V_T(s)}{V_F(s)} = \frac{K_G}{1+\tau_G s} \dots\dots\dots (3-3)$$

dimana nilai konstanta penguatan generator memiliki rentang nilai dari 0.7000 sampai 1.0000 sedangkan untuk konstanta waktu generator memiliki rentang nilai dari 1.0000 detik sampai 2.0000 detik pada keadaan beban nol sampai keadaan beban penuh. Komponen – komponen sistem eksitasi generator ini kemudian digabungkan dan terbentuk diagram blok dari sistem eksitasi generator yang diperlihatkan pada Gambar 3.5. Berdasarkan diagram blok pada Gambar 3.5. Kemudian diperoleh fungsi alih lingkaran terbuka dan fungsi alih lingkaran tertutup dari sistem eksitasi generator. Untuk fungsi alih lingkaran terbuka diperlihatkan pada persamaan(3-4).

Untuk fungsi alih terbuka dinyatakan dengan persamaan (3-4).

$$\frac{V_T(s)}{V_{ref}(s)} = \frac{K_A K_E K_G}{(1+\tau_A s)(1+\tau_E s)(1+\tau_G s)} \dots\dots\dots(3-4)$$

$$\frac{V_T(s)}{V_{ref}(s)} = \frac{16,0000}{0,0180s^3+0,4020s^2+1,7600s+1,0000} \dots\dots\dots(3-5)$$



Gambar 3 . 5 Fungsi Alih Subsistem Blok Generator DC

3.3.2 Fungsi Alih dan Blok Diagram Sensor Tegangan

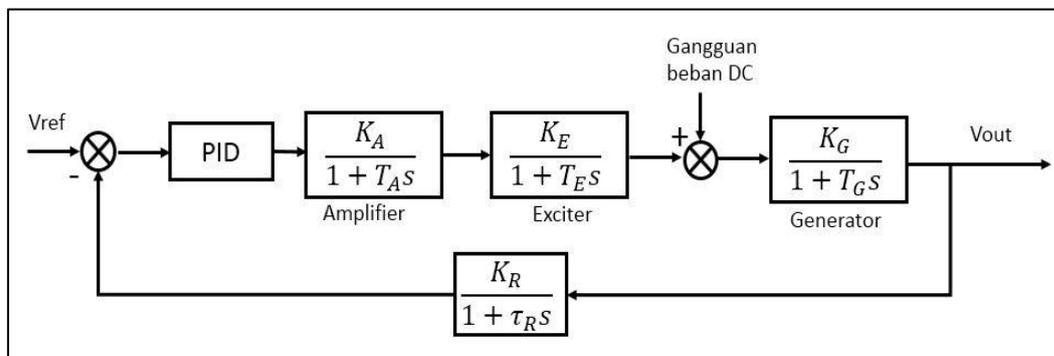
Sensor terdiri atas transformator tegangan (PT) dan sebuah penyearah. Sensor dapat direpresentasikan sebagai KR dengan konstanta waktu TR dan fungsi transfernya sebagai berikut:

$$\frac{V_S(s)}{V_t(s)} = \frac{K_R}{1+\tau_R s} \dots\dots\dots(3-6)$$

Hasil penyearahan lalu dibandingkan dengan tegangan referensi (Vref) apabila terjadi perbedaan maka kontroler akan memerintahkan amplifier untuk menaikkan atau menurunkan arus penguatan generator DC sehingga tegangan keluar dari generator tersebut juga berubah. Jika tegangan keluar generator DC berubah maka arus penguatan generator sinkron juga berubah, akibatnya tegangan keluaran generator kembali stabil.

3.4 Perancangan Diagram Sistem

Perancangan diagram sistem menjelaskan sistem yang direncanakan secara garis besar dan diharapkan sistem dapat menunjukkan hasil yang sesuai karakteristik komponen asli dan sesuai dengan aturan yang digunakan. Perancangan diagram proses sistem dan diagram blok sistem dapat dilihat pada gambar 3.6.



Gambar 3 . 6 Blok Diagram Sistem Kendali

3.5 Perancangan Algoritma

Terdapat tiga tahapan pada proses perancangan algoritma, yaitu:

1. Desain Menggunakan Matlab
2. Penalaan Kontroler menggunakan penala PIDF - pidTuner
3. Penentuan model referensi
4. Penetapan parameter kontroler

Adapun penjelasan ketiga tahapan tersebut adalah sebagai berikut:

3.5.1 Desain Menggunakan Matlab

A. Pendahuluan

Fungsi alih untai tertutup sebuah sistem kontrol dengan umpan balik negatif secara matematik dapat dituliskan sebagai:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s) \cdot H(s)}$$

$P(s)$ adalah fungsi alih dari plant, $G(s)$ adalah fungsi alih untai maju (*feed forward*) pada umumnya adalah perkalian dari fungsi alih kontroler, aktuator dan plant. Sedangkan $H(s)$ adalah fungsi alih dari umpan balik yaitu sensor. Agar sistem kontrol mempunyai respon keluaran yang stabil, secara teori mempersyaratkan $1 + G(s) \cdot H(s) = 0$. Semakin rumit fungsi alih plant maka semakin rumit pula proses desainnya, namun dengan tersedianya perangkat lunak analisis sistem kontrol yaitu matlab – proses desain sistem kontrol akan menjadi lebih cepat, mudah dan akurat. Pada sesi ini hanya menjelaskan tentang proses desain kontrol analog dengan menggunakan fungsi-fungsi yang tersedia di Matlab dengan menggunakan fungsi alih plant yang digunakan dalam pembahasan skripsi ini sehingga dapat melakukan analisa secara cepat.

B. Desain Sistem Kontrol Secara Manual

Mendesain sistem kontrol secara manual menggunakan Matlab adalah cara mendesain dengan menggunakan fungsi-fungsi secara terpisah seperti yang di jelaskan pada sub bab sebelumnya. Langkah-langkah mengikuti metode desain yang lazim dalam sistem kontrol, fungsi-fungsi yang ada di Matlab untuk membantu menganalisis secara cepat. Fungsi-fungsi dieksekusi dengan urutan tergantung kebutuhan dan situasi dalam langkah-

langkah proses desain sistem kontrolnya. Langkah-langkah umumnya mendesain sistem kontrol adalah:

1. Mendefinisikan fungsi alih plant

Fungsi alih plant yang digunakan sebagai berikut:

$$G(s) = \frac{16,0000}{0,0180s^3 + 0,4020s^2 + 1,7600s + 1,0000}$$

Jika konstanta polinomial fungsi alih telah diketahui pada penyebut dan pembilang, maka pendefinisian secara langsung dapat memasukkan konstanta tersebut kedalam sebuah variabel dengan cara:

```
G = tf(16.0000,[0.0180 0.4020 1.7600 1.0000])
```

```
G =
```

```
16
```

```
-----
```

```
0.018 s^3 + 0.402 s^2 + 1.76 s + 1
```

```
Continuous-time transfer function.
```

Melalui cara ini pendefinisian variabel polinomial s tidak perlu dilakukan.

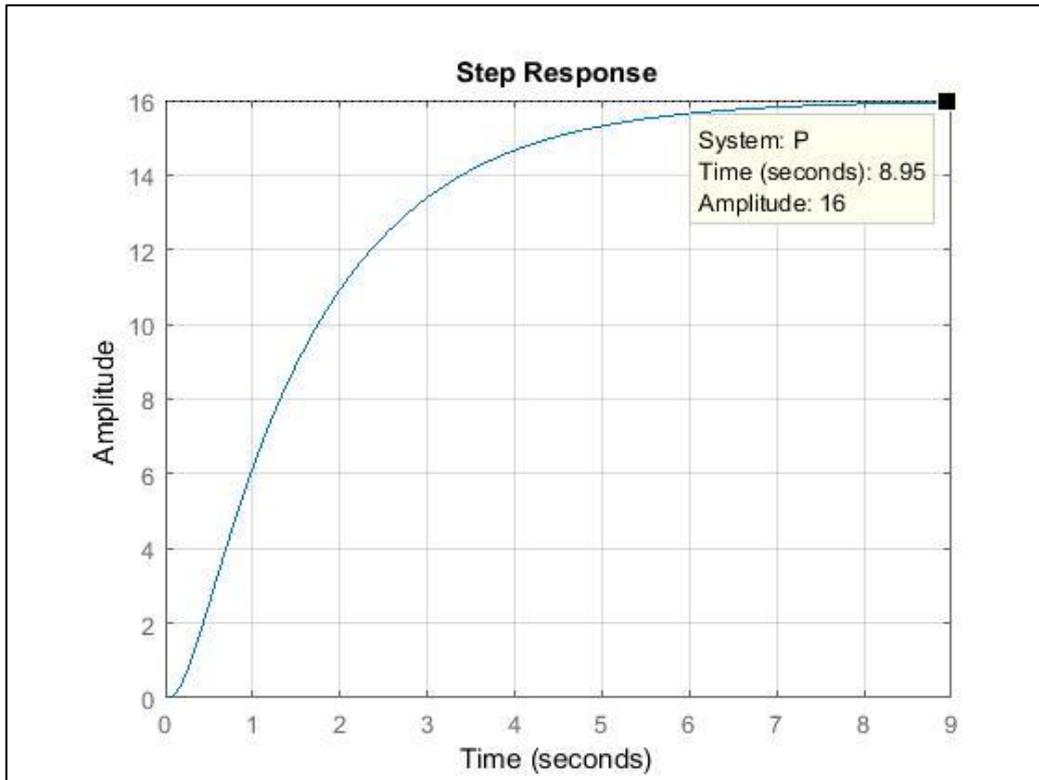
2. Melihat Karakteristik plant

Sebelum mendesain sistem kontrol, pengujian karakteristik plant sangat diperlukan untuk memutuskan jenis kontroler yang diperlukan. Untuk melihat karakteristik fungsi alih plant diantaranya melakukan pengujian respon keluarannya menggunakan fungsi step melalui *stepplot*.

```
stepplot(G)
```

```
grid on
```

pada baris 1 menguji keluaran menggunakan sisnyal uji step, baris 2 memasang grid pada grafik. Hasil pengujian karakteristik fungsi alih plant diperlihatkan pada Gambar 3.7.



Gambar 3 . 7 Respon Fungsi Alih Plant

Gambar diatas karakteristik dari fungsi alih: $\frac{16,0000}{0,0180s^3+0,4020s^2+1,7600s+1,0000}$. respon stabil pada amplitudo 16 dengan waktu 8,95

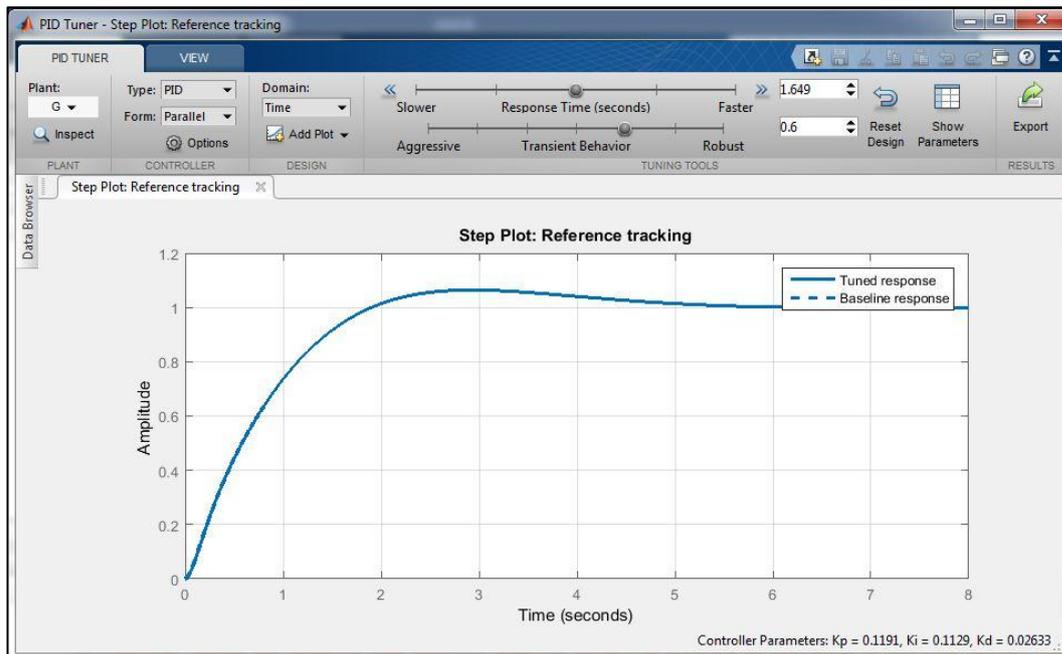
3.5.2 Penalaan Kontroler menggunakan penala PIDF - pidTuner

Penalaan kontroler menggunakan penala PIDF – pidTuner adalah metode penalaan untuk menentukan jenis kontroler dan konstanta penguatan dari fungsi alih C(s) dapat di peroleh secara cepat, mudah dan akurat dengan menggunakan aplikasi-aplikasi yang tersedia di matlab.

Plant diketahui memiliki fungsi alih

$$\frac{V_T(s)}{V_{ref}(s)} = \frac{16,0000}{0,0180s^3+0,4020s^2+1,7600s+1,0000} \dots\dots\dots(3-6)$$

Jika didesain menggunakan kontroler PIDF maka penalaannya akan di tampilkan pada jendela tampilan menu pidTuner pada Gambar 3.8.

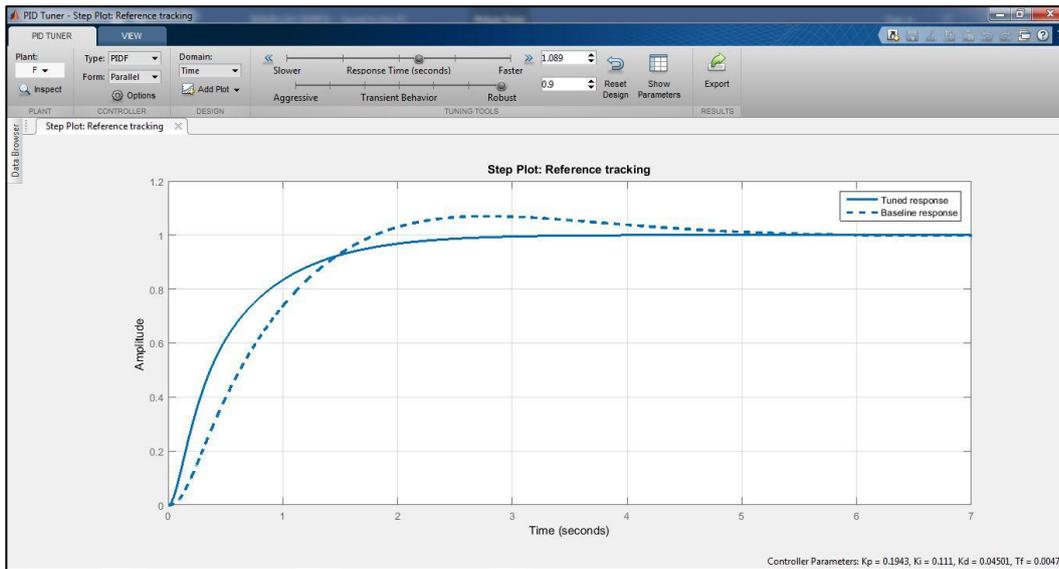


Gambar 3 . 8 Jendela Tampilan Menu pidTuner Sebelum Ditala

Untuk melihat parameter acuan dan hasil penalaan dilakukan dengan cara klik menu tombol *show parameter*, hasilnya diperlihatkan pada Gambar 3.10. Menu ini digunakan untuk melihat parameter secara realtime, karena setiap perubahan penalaan data parameter akan berubah secara otomatis.

Dalam penelitian ini desain dibutuhkan agar respon dipercepat sesuai spesifikasi desain yang telah ditetapkan pada sub bab yang sebelumnya. Dengan cara menggeser menu geser *Response Time* untuk menala waktu respon sistem kontrol dan menu geser *Transient Behavior* untuk menala perilaku transiennya.

Berikut tampilan jendela tampilan penalaan pidTuner yang telah dimodifikasi sesuai spesifikasi desain yang ditetapkan. Dapat dilihat pada Gambar 3.9



Gambar 3. 9 Jendela Tampilan Menu pidTuner Setelah Ditala

Selanjutnya dengan melihat di menu *show parameter* pada Gambar 3.10 kita akan melihat apakah penalaan sistem kita sudah sesuai dengan spesifikasi desain yang telah ditentukan sebelumnya. Ketika sudah memenuhi atau sesuai maka penelitian akan lanjut ke tahap berikutnya.

| Controller Parameters | | |
|-----------------------|-----------|----------|
| | Tuned | Baseline |
| Kp | 0.19425 | 0.11169 |
| Ki | 0.11098 | 0.10941 |
| Kd | 0.045014 | 0.016024 |
| Tf | 0.0047665 | 0.078599 |

| Performance and Robustness | | |
|----------------------------|----------------------|---------------------|
| | Tuned | Baseline |
| Rise time | 1.22 seconds | 1.2 seconds |
| Settling time | 2.27 seconds | 4.62 seconds |
| Overshoot | 0 % | 7.02 % |
| Peak | 1 | 1.07 |
| Gain margin | 40.1 dB @ 61.8 rad/s | 28 dB @ 15.3 rad/s |
| Phase margin | 90 deg @ 1.84 rad/s | 69 deg @ 1.21 rad/s |
| Closed-loop stability | Stable | Stable |

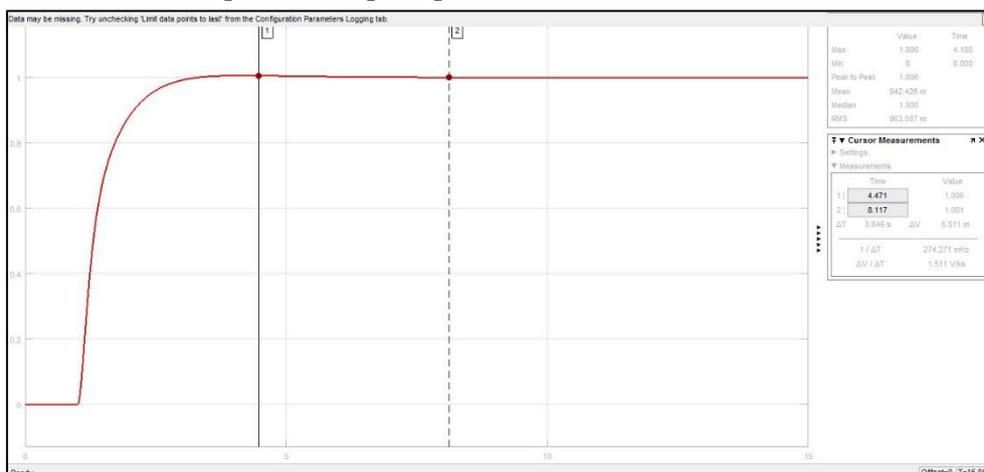
Gambar 3. 10 Jendela Tampilan Show Parameter Setelah Ditala

3.5.3 Penentuan Model Referensi

Penentuan model referensi dilakukan dengan cara *try and error* sehingga didapatkan respon yang stabil yang mengikuti karakteristik dari *plant*. Model referensi yang akan ditentukan tidak memiliki *error steady state*, tidak terdapat osilasi dan memiliki *settling time* yang cepat. Penentuan model referensi dapat ditentukan dengan cara melihat hasil respon saat simulasi melalui aplikasi matlab dan pada saat implementasi sistem. Fungsi alih dari model referensi adalah seperti pada persamaan (3-6)

$$\frac{V_T(s)}{V_{ref}(s)} = \frac{16,0000}{0,0180s^3 + 0,4020s^2 + 1,7600s + 1,0000} \dots\dots\dots(3-6)$$

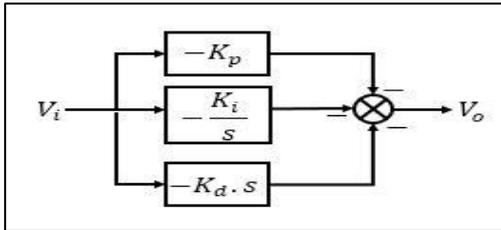
Dengan memberikan masukan unit step dari program matlab, maka dapat diketahui bahwa respon model referensi memiliki nilai time constant sebesar 1,511 detik dan nilai settling time sebesar 3.022 detik. Respon transien dari model referensi yang telah diberi masukan unit step adalah seperti pada Gambar 3.11.



Gambar 3 . 11 Model Referensi

Kontroler PID Paralel

Disebut sebagai kontroler PID paralel, karena masing-masing kontroler P,I dan D dihubungkan secara paralel dimana semua sinyal keluarannya dijumlah dengan menggunakan penjumlah membalikkan, sedangkan masukannya dihubungkan secara paralel. Secara blok diagram kontroler PID diperlihatkan pada Gambar 3.12.



Gambar 3. 12 Gambar 3. 12 Blok Diagram Kontroler PID Paralel

Sumber : Siswojo, 2017 : 129

Dari Gambar 3.12, fungsi alih jika disederhanakan ditambah dengan fungsi alih filter akan menjadi:

$$\frac{v_o(s)}{v_i(s)} = - \left(-K_p - \frac{K_i}{s} - K_d \cdot \frac{s}{T_f * s + 1} \right) \dots \dots \dots (3-7)$$

Eksperimen kontroler PID paralel

Tentang kontroler PID paralel telah di jelaskan sebelumnya. Eksperimen desain kontroler PID menggunakan fungsi alih sistem untai tertutup dalam persamaan yaitu

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s) \cdot H(s)} \dots \dots \dots (3-8)$$

Dengan:

$$G(s) = \frac{16,0000}{0,0180s^3 + 0,4020s^2 + 1,7600s + 1.0000}$$

$$H(s) = \frac{1}{0,025 + 1}$$

$$\frac{C(s)}{R(s)} = \frac{0,0072 s^4 + 0,7368 s^3 + 13,57 s^2 + 56,72 s + 32}{8,1e - 06 s^7 + 0,0006858 s^6 + 0,0201 s^5 + 0,2612 s^4 + 1,549 s^3 + 3,99 s^2 + 3,545 s + 1}$$

1. Memasukkan fungsi alih sistem

fungsi alih plant sebelum didesain kontrolernya harus didefinisikan sebagai fungsi s dengan memasukkan sebuah variabel G, dan fungsi alih umpan balik juga didefinisikan dengan variabel H dengan cara berikut

$$G = \text{tf}(16.0000,[0.0180 0.4020 1.7600 1.0000]);$$

G =

16

$$0.018 s^3 + 0.402 s^2 + 1.76 s + 1$$

Continuous-time transfer function.

Dan

$$H = \text{tf}(1,[0.025 \ 1]);$$

H =

1

$$0.025 s + 1$$

Continuous-time transfer function.

Lalu fungsi alih sistem $\frac{C(s)}{R(s)}$ sebelum didesain kontrolernya harus didefinisikan sebagai fungsi s dengan memasukkan kesebuah variabel F, dengan cara berikut

$$F = \text{tf}(G/1+H*G);$$

F =

$$0.0072 s^4 + 0.7368 s^3 + 13.57 s^2 + 56.72 s + 32$$

$$8.1e-06 s^7 + 0.0006858 s^6 + 0.0201 s^5 + 0.2612 s^4 + 1.549 s^3 + 3.99 s^2 + 3.545 s + 1$$

Continuous-time transfer function.

Nama variabel untuk menyimpan fungsi alih sistem adalah F jika lebih suka menggunakan persamaan matematik polinomial s, harus mendefinisikan dengan memasukkan perintahs = *tf('s')* terlebih dahulu.

2. Inisialisasi kontroler PIDF

Awalnya kontroler PI belum didefinisikan, untuk memprediksi kontroler PIDF yang cocok dengan lant tersebut, harus didefinisikan terlebih dahulu dengan memberikan perintah *pidtune*. Cara mengawali prediksi kontroler PIDF:

```
C=pidtune(F,'pidf')
```

C =

$$K_p + K_i * \frac{1}{s} + K_d * \frac{s}{T_f * s + 1}$$

with $K_p = 0.112$, $K_i = 0.109$, $K_d = 0.016$, $T_f = 0.0786$

Continuous-time PIDF controller in parallel form.

Parameter awal adalah fungsi alih sistem dalam hal ini adalah variabel F sebagai fungsi alih sistem yang telah didefinisikan pada langkah 1, parameter kedua adalah pemilihan jenis kontroler PIDF. Sedangkan hasil prediksi kontroler PIDF diarahkan sebagai variabel C. sebagai awalan kontroler PIDF ditala menjadi $K_p = 0.0566$, $K_i = 0.0523$, $K_d = 0.0129$, dan $T_f = 0.0786$. selanjutnya konfigurasi kontroler PID ini akan digunakan sebagai referensi untuk melakukan proses penalaan guna memperbaiki respon sesuai kebutuhan.

3. Mengeksekusi pidTuner untuk menala PIDF

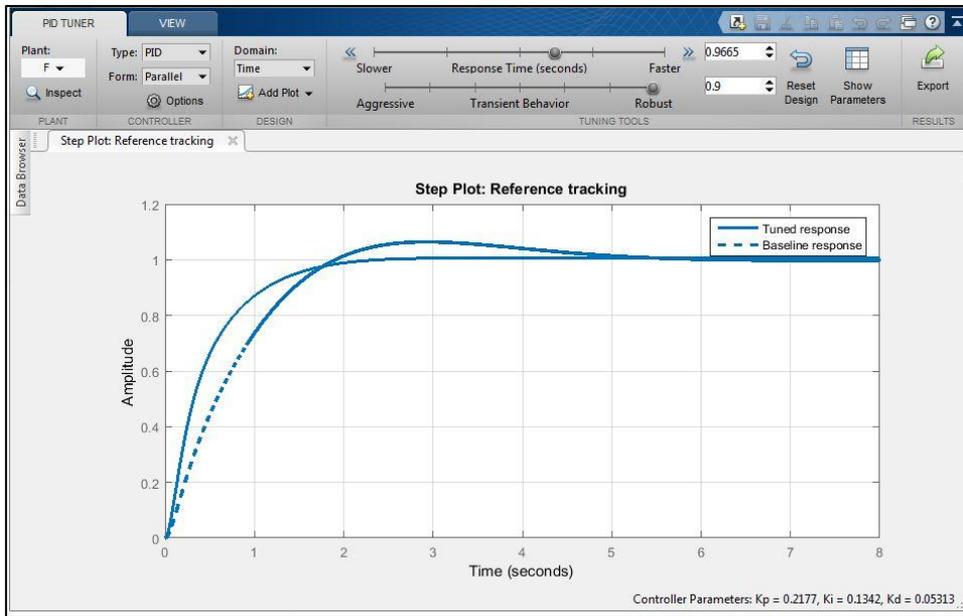
Kontroler PID ini dapat ditala ulang dengan menggunakan perintah perintah *pidTuner*, dengan memasukkan perintah sebagai berikut:

```
1 C=pidtune(P, 'pid')
```

Setelah perintah *pidTuner* diberikan, akan muncul tampilan menu grafis aplikasi untuk melakukan penalaan kontroler PIDF seperti pada Gambar 3.13 .respon diperlihatkan berdasarkan prediksi awal kontrolernya, untuk selanjutnya rpson tersebut dapat diperbaiki dengan sangat mudah dan cepat dengan menggunakan menu slider. Menu slider yang disediakan berfungsi untuk melakukan penalaan perbaikan respon kontroler PID. Dua setelan respon yang disediakan yaitu:

- (a) **Setelan perilaku waktu** (*response time*): digunakan untuk memperbaiki respon waktu agar menjadi lebih lambat (*slower*) atau lebih cepat (*faster*) dengan menggeser slider atau dengan cara memasukkan nilainya melalui isian.
- (b) **Setelan perilaku transien** (*response behavior*): digunakan untuk memperbaiki perilaku transiennya menjadi lebih responsif (*aggressive*) atau lebih tegar (*robust*) dengan menggeser-geser slider atau dengan cara memasukkan nilainya melalui isian

sebagai contoh pernalaannya dengan menentukan nilai perilaku transien dibuat 0,9 dan nilai respon waktu 0,9665. setelah dilakukan penalaan responnya berubah, respon awal diperlihatkan sebagai grafik yang terputus dapat dilihat pada Gambar 3.13.



Gambar 3 . 13 Jendela Penalaan

4. Melihat parameter penalaan

Untuk melihat parameter kontroler respon referensi dan respon setelah dilakukan penalaan klik tombol menu *Show Parameters*, hasilnya diperlihatkan dalam gambar (). Ketika dilakukan penalaan grafik respon, parameter-parameter ini juga berubah secara realtime untuk dapat memastikan parameter-parameter kontrolnya. Parameter-parameter kontroler yang ditampilkan:

- (a) $K_p = 0,11169$ menjadi $K_p = 0,19425$
- (b) $K_i = 0,10941$ menjadi $K_i = 0,11098$
- (c) $K_d = 0,016024$ menjadi $K_d = 0,045014$
- (d) $T_f = 0,078599$ menjadi $T_f = 0,0047665$

Sedangkan parameter untuk kerja dan ketegaran yang di tampilkan:

- (a) *Rise time*=1,2 detik menjadi *Rise time*=1,22 detik
- (b) *Settling time*= 4,62 detik menjadi *settling time*= 2,27 detik
- (c) *Overshoot* = 7,02% menjadi *Overshoot* = 0%
- (d) *Peak* = 1,07 menjadi *Peak* menjadi *Peak* = 1
- (e) *Gain margin*=28 dB @ 15,3 rad/s menjadi *Gain margin* = 40,1 dB @ 61,8 rad/s
- (f) *Phase margin* =69 deg @1,21 Rad/s menjadi *Phase margin* = 90 deg @1,84rad/s

(g) *Closed – loop stability = Stable* menjadi *Closed – loop stability = Stable*

5. Mengekpor parameter kontroler

Parameter hasil penalaan kontroler dapat diekspor ke konsol untuk keperluan pemrosesan lebih lanjut jika diperlukan. Untuk mengekspor parameter kontroler yaitu variabel C cukup klik tombol menu *Export*, selanjutnya pilih variabel yang akan diekspor. Ketika kembali ke konsol fungsi alih kontroler dapat ditampilkan sebagai berikut. Dan dapat dilihat pada Gambar 3.14.

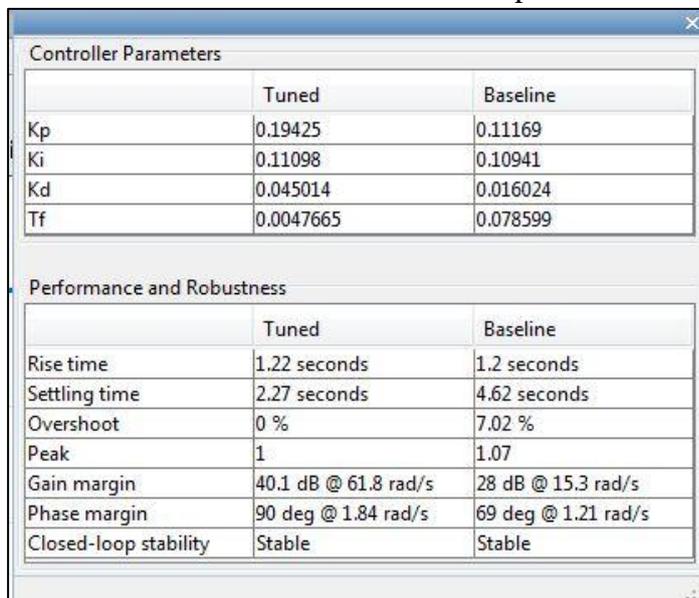
>> C

C =

$$K_p + K_i * \frac{1}{s} + K_d * \frac{s}{T_f * s + 1}$$

with $K_p = 0.112$, $K_i = 0.109$, $K_d = 0.016$, $T_f = 0.0786$

Continuous-time PIDF controller in parallel form.



| Controller Parameters | | |
|-----------------------|-----------|----------|
| | Tuned | Baseline |
| Kp | 0.19425 | 0.11169 |
| Ki | 0.11098 | 0.10941 |
| Kd | 0.045014 | 0.016024 |
| Tf | 0.0047665 | 0.078599 |

| Performance and Robustness | | |
|----------------------------|----------------------|---------------------|
| | Tuned | Baseline |
| Rise time | 1.22 seconds | 1.2 seconds |
| Settling time | 2.27 seconds | 4.62 seconds |
| Overshoot | 0 % | 7.02 % |
| Peak | 1 | 1.07 |
| Gain margin | 40.1 dB @ 61.8 rad/s | 28 dB @ 15.3 rad/s |
| Phase margin | 90 deg @ 1.84 rad/s | 69 deg @ 1.21 rad/s |
| Closed-loop stability | Stable | Stable |

Gambar 3. 14 Jendela Show Parameter

Dengan demikian dari hasil proses penalaan, fungsi alih kontroler PID menjadi:

$$C(s) = 0,11169 + 0,10941 \cdot \frac{1}{s} + 0,016024 \cdot \frac{s}{0,078599s + 1}$$

Menjadi

$$C(s) = 0,19425 + 0,11098 \cdot \frac{1}{s} + 0,045014 \cdot \frac{s}{0,0047665s + 1}$$

3.5.4 Penetapan Parameter Kontroler

Penetapan parameter kontroler pada penelitian ini dapat dilihat pada hasil penalaan menggunakan pidTuner yang dapat dilihat pada menu *Show Parameter* pada gambar 3.13 Dan didapatkan nilai parameter kontrolernya sebagai berikut:

$$K_p = 0,19425$$

$$K_i = 0,11098$$

$$K_d = 0,045014$$

$$T_f = 0,0047665$$

Kemudian nilai parameter yang didapat dimasukkan ke dalam parameter blok kontroler PID yang terdapat pada blok diagram sistem yang akan diteliti

Halaman ini sengaja dikosongkan