

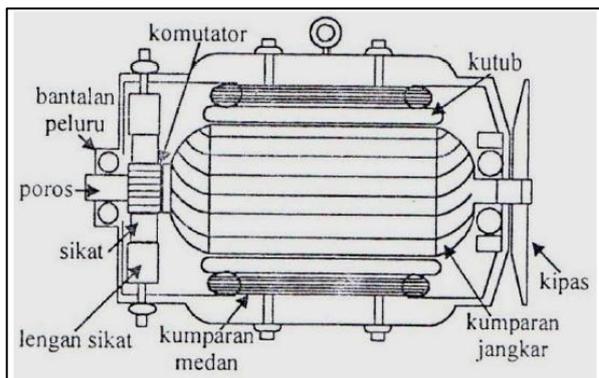
## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Generator Direct Current (DC)

Generator *Direct Current* (DC) merupakan sebuah piranti elektronik yang mengubah energi mekanis menjadi energi listrik. Energi mekanis di pergunakan untuk memutar kumparan kawat penghantar di dalam medan magnet. Energi yang dipergunakan bisa berbentuk uap, turbin air, mesin bensin, atau motor listrik.

Generator DC memiliki konstruksi yang terdiri atas dua bagian yaitu bagian yang berputar (rotor) dan bagian yang diam (stator). Contoh dari stator adalah rangka, komponen magnet dan komponen sikat. Sedangkan yang termasuk rotor adalah jangkar, kumparan jangkar dan komutator (Ardian, 2010: 6). Secara umum, konstruksi generator DC dapat dilihat pada Gambar 2.1.



Gambar 2 . 1 Kontruksi Generator

Sumber : Niko

Prinsip kerja dari suatu generator DC adalah berdasarkan hukum Faraday mengenai induksi elektromagnetik. Hukum Faraday menyebutkan “Apabila suatu konduktor digerakkan dan memotong garis-garis gaya magnetik (fluksi), maka pada konduktor tersebut akan timbul suatu gaya gerak listrik (g.g.l.)” (Soemarwanto, 2010: 79).

Jika kumparan kawat dihubungkan dengan rangkaian listrik tertutup, maka akan timbul pula arus listrik yang mengalir pada rangkaian. Berdasarkan Hukum Induksi Faraday/ Hukum Faraday besarnya g.g.l. yang diinduksikan satu belitan adalah:

$$e = -N \frac{d\phi}{dt} \times 10^{-8} \text{ Volt } (\phi \text{ dalam Maxwell}) \dots\dots\dots(2-1)$$

Untuk N belitan maka:

$$e = -N \frac{d\phi}{dt} \times 10^{-8} \text{ Volt} \dots\dots\dots (2-2)$$

Apabila kecepatan putaran =  $\frac{n \text{ putaran}}{\text{menit}}$ , maka untuk 1 putaran =  $\frac{1}{n}$  menit =  $\frac{60}{n}$  detik, konduktor akan memotong P x  $\phi$  Maxwell, sehingga:

$$e_{rata-rata} = \frac{p \cdot \phi}{60/n} \times 10^{-8} \text{ Volt} = \frac{p \cdot n \cdot \phi}{60} \times 10^{-8} \dots\dots\dots (2-3)$$

Karena ada  $\frac{z}{a}$  konduktor pada setiap cabang jangkar paralel, maka:

$$E = \frac{z}{a} \frac{p \cdot n \cdot \phi}{60} \times 10^{-8} \text{ Volt} \dots\dots\dots (2-4)$$

$$E = \frac{p}{a} \frac{n}{60} \cdot z \cdot \phi \times 10^{-8} \text{ Volt} \dots\dots\dots (2-5)$$

Dengan:

P = Jumlah kutub.

a = Jumlah cabang jangkar paralel.

n = Jumlah putaran (putaran/menit).

z = Jumlah konduktor di jangkar.

$\phi$  = Fluksi tiap kutub (Maxwell).

Di dalam generator DC besar p, a, dan z adalah tertentu/ tetap/ konstan. Maka dari persamaan 2-5, apabila  $\frac{p}{a} \frac{n}{60} \cdot z \cdot \phi \times 10^{-8} C$  konstan, maka:

$$E = C \cdot n \cdot \phi \text{ Volt} \dots\dots\dots (2-6)$$

Dengan:

E = Tegangan induksi.

C = Konstanta.

n = Jumlah putaran (putaran/menit).

$\phi$  = Fluksi tiap kutub (Maxwell).

## 2.2 Kontroler

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolak ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata K.,1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana

kecepatan sistem dalam merespons masukan, dan bagaimana peredaman terhadap adanya lonjakan (*over shoot*).

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil jika *outputnya* berosilasi terus menerus ketika dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu masukan, tidak dapat mengikuti masukan secara serentak, tapi menunjukkan respons transien berupa suatu osilasi teredam sebelum mencapai *steady state*.

Dalam sistem pengendalian terdapat 2 macam *loop*:

1. Pengendalian dengan *loop* terbuka

Sistem kontrol *loop* terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolan. Jadi pada sistem kontrol *loop* terbuka, keluaran tidak diukur atau diumpan balik untuk dibandingkan dengan masukan.

2. Pengendalian dengan *loop* tertutup

Sistem kontrol *loop* tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Disebut juga sistem kontrol yang menggunakan umpan balik untuk memperkecil kesalahan sistem.

### 2.2.1 Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding dengan sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler  $m(t)$  dan sinyal kesalahan penggerak  $e(t)$  adalah:

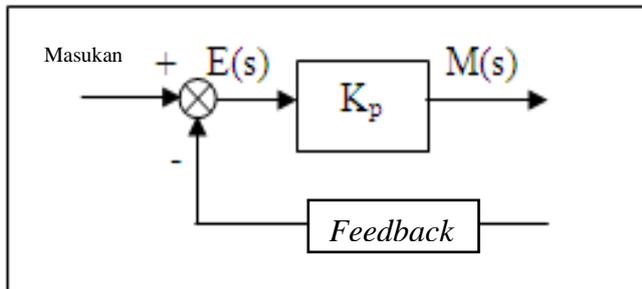
$$m(t) = K_p \cdot e(t) \dots\dots\dots (2-7)$$

atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = K_p \dots\dots\dots (2-8)$$

Di mana  $K_p$  adalah kepekaan proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerak, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997). Diagram blok kontroler proporsional ditunjukkan dalam Gambar 2.2.



Gambar 2 . 2 Diagram Blok Kontroler Proporsional

Sumber : Ogata K, 1997

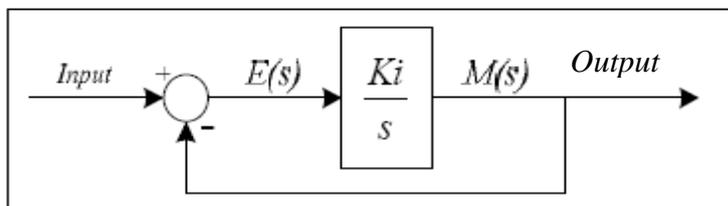
### 2.2.2 Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap pada kontroler proporsional sebelumnya. Pada kontroler dengan aksi integral, harga keluaran kontroler  $m(t)$  diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak  $e(t)$ .

Jadi,

$$\frac{dm(t)}{dt} = Ki \cdot e(t) \dots\dots\dots (2-9)$$

Jika harga  $e(t)$  diduakalikan, maka harga  $m(t)$  berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga  $m(t)$  tetap stasioner. Aksi kontrol integral seringkali disebut kontrol *reset* (Ogata K.,1997). Gambar 2.3 menunjukkan diagram blok kontroler integral.



Gambar 2 . 3 Diagram Blok Kontroler Integral

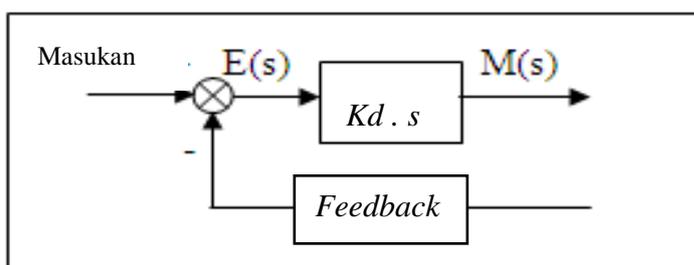
Sumber : Ogata K, 1997

### 2.2.3 Kontroler Diferensial

Kontroler ini digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem kontrol dengan cara memperbesar *phase lead* terhadap penguatan kontrol dan mengurangi *phase lag* penguatan tersebut (Ogata K.,1997). Kontroler diferensial tidak dapat mengeluarkan *output* bila tidak ada perubahan masukan, selain itu kontroler differensial tidak dapat digunakan untuk proses yang mengandung *noise*. Hubungan antara keluaran kontroler  $m(t)$  dan sinyal kesalahan penggerak  $e(t)$  adalah:

$$\frac{M(s)}{E(s)} = Kd \cdot s \dots\dots\dots(2-10)$$

Gambar 2.4 menunjukkan diagram blok kontroler diferensial.



Gambar 2 . 4 Diagram Blok Kontroler Diferensial

Sumber : Ogata K, 1997

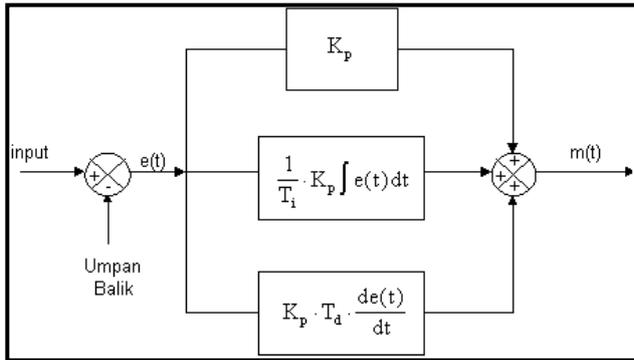
### 2.2.4 Kontroler Proporsional Integral Diferensial (PID)

Gabungan aksi kontrol proporsional, integral, dan diferensial mempunyai keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler. Elemen-elemen P, I, dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar. Persamaan kontroler PID ini dapat dinyatakan sebagai berikut:

$$m(t) = Kp \cdot e(t) + \frac{Kp}{Ti} \cdot e(t)dt + Kp \cdot Td \frac{de(t)}{dt} \dots\dots\dots(2-11)$$

Dalam transformasi Laplace dinyatakan sebagai berikut:

$$\frac{M(s)}{E(s)} = Kp \left( 1 + \frac{1}{Ti \cdot s} + Td \cdot s \right) \dots\dots\dots(2-12)$$



Gambar 2 . 5 Diagram Blok Kontroler PID

Sumber : Ogata K, 1997

### 2.3 Desain Kontroler menggunakan pidTuner

Ketika fungsi alih plant semakin kompleks, proses desain sistem kontrol secara manual akan menjadi semakin rumit. Matlab menyediakan aplikasi yang memiliki kemampuan melakukan penalaan kontroler PID berdasarkan fungsi alih plantnya secara otomatis. Aplikasi fungsi tersebut adalah *pidtune* yang digunakan bersama fungsi *pidTuner*. Fungsi *pidtune* untuk memprediksi jenis kontroler untuk fungsi alih plant dengan jenis kontroler kombinasi dari P,I dan D. jika variabel P mendefinisikan fungsi alih plant, maka untuk memprediksi jenis kontroler PID harus diberikan perintah:

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

Variabel C adalah jenis kontroler yang diprediksi berikut konstantanya yaitu  $K_p$  dan  $K_i$ . Jika diinginkan untuk memprediksi jenis kontroler yang lainnya misalnya PID, maka argumen harus dimasukkan ke parameter kedua. Fasilitas ini dapat juga digunakan untuk mendesain sistem kontrol secara manual. Untuk memprediksi parameter awal kontroler guna mendapatkan kestabilan sistem.

Selanjutnya desain dilanjutkan dengan penalaan kontroler dengan mengeksekusi fungsi *pidTuner* dengan perintah;

```
1 pidTuner(P, C)
```

Parameter pertama adalah fungsi alih plant dengan variabel P, parameter kedua adalah kontroler kontroler hasil prediksi. Dengan mengeksekusi fungsi tersebut. Penalaan dapat dilakukan dengan sangat mudah dan cepat untuk mendapatkan respon yang diinginkan.

## 2.4 Penalaan Kontroler menggunakan penala PID – pidTune

Untuk menentukan jenis kontroler dan konstanta penguatan pada fungsi alih  $C(s)$  dapat diperoleh secara cepat, mudah dan akurat, dengan menggunakan aplikasi-aplikasi yang tersedia di Matlab. (Siswojo,2017: 45).

**pidTune** – fungsi digunakan untuk menala kontroler dari fungsi alih untai maju dengan umpan balik satu (*unity feedback*).

Berdasarkan model plant berupa fungsi alih, pidtune secara otomatis dapat melakukan penalaan masing-masing penguatan kontroler PID untuk menyeimbangkan unjuk kerja (respon waktu) dan ketegaran (margin kestabilan). Dapat dipilih berbagai variasi dari konfigurasi kontroler PID dan dapat memberikan spesifikasi respon waktu dan margin fasa yang diinginkan. Perlu dicatat bahwa menaikkan unjuk kerja secara tipikal akan mengurangi ketegaran dan sebaliknya.

**c = pidtune(G, TYPE)** – menala kontroler PID untuk jenis SISO (*single masukan single output*) dari plant G. Dapat dispesifikasikan jenis sistem linier untuk G. Variabel c adalah fungsi alih controller hasil penalaan. Argumen string TYPE untuk memberikan spesifikasi jenis kontroler seperti yang di perlihatkan dalam tabel

**pidtune** – mengembalikan sebuah objek c PID/PID2 dengan waktu sampling yang sama sebagai G. Jika G adalah larik dari model LTI (*linier time-invariant*), **pidtune** mendesain sebuah kontroler untuk masing-masing model plant dan mengembalikannya dalam bentuk objek larik dari PID/PID2.

**C = pidtune(G, C0)** – memaksa C supaya cocok dengan struktur dari objek C0 dan PID, PID2, PIDSTD atau PIDSTD2. Hasil C akan berjenis sama, bentuk, dan formula integrator/derivatifnya C0. Sebagai contoh, untuk menala sebuah waktu diskrit kontrolerPI dalam bentuk standar dengan waktu sampling 0,1 dan formula trapezoidal.

**C0 = pidstd(1, 1, 'Ts', 0.1, IFormula', 'T')** dan **C = pidtune(C,C0,WC)** – menspesifikasikan sebuah nilai target WC (dalam *rad/TimeUnit* relatif terhadap *time unit* dari G) untuk frekuensi crossover penguatan 0dB dari respon untai terbuka  $L = L \cdot G$ . secara tipikal, WC kurang lebih men-set lebar pita kontrol untuk meningkatkan respon kecepatan dan mengurangi WC untuk memperbaiki stabilitas. Ketika dihilangkan, WC akan dipilih secara otomatis berdasarkan dinamika plant.

$C = \text{pidtune}(G, \dots, \text{OPTIONS})$  – menspesifikasikan pilihan penalaan tambahan seperti margin fasa yang ditargetkan. Penggunaan perintah `PIDTUNESOPTIONS` mengatur pilihan dari singkatan `OPTIONS`.

$[C, \text{INFO}] = \text{pidtune}(G, \dots)$  – juga mengembalikan sebuah stuktur `INFO` dengan informasi tentang stabilitas untai tertutup, pemilihan penguatan frekuensi crossover, dan margin fasa sebenarnya.

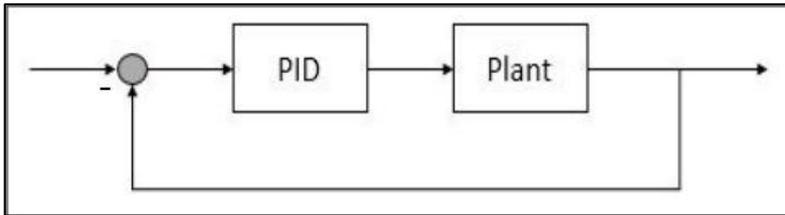
Tabel 2. 1 Tipe Kontroler

| TYPE    | Fungsi alih   |
|---------|---|
| 'P'     | Kontroler P saja  |
| 'I'     | Kontroler I saja  |
| 'PI'    | Kontroler PI  |
| 'PD'    | Kontroler PD  |
| 'PID'   | Kontroler PID   |
| 'PDF'   | Kontroler PD dengan filter orde pertama derivative                          |
| 'PIDF'  | Kontroler PID dengan filter orde pertama derivative                         |
| 'PI2'   | Kontroler 2-dof PI  |
| 'PD2'   | Kontroler 2-dof PD  |
| 'PDF2'  | Kontroler 2-dof PD dengan filter orde pertama derivative                    |
| 'PID2'  | Kontroler 2-dof PID   |
| 'PIDF2' | Kontroler 2-dof PID dengan filter orde pertama derivative                   |
| 'I-PD'  | Kontroler 2-DOF PID dengan $b = 0, c = 0$                                   |
| 'I-PDF' | Kontroler 2-DOF PID dengan filter orde pertama derivatif dan $b = 0, c = 0$ |
| 'ID-F'  | Kontroler 2-DOF PID dengan $b = 0, c = 1$                                   |
| 'IDF-P' | Kontroler 2-DOF PID dengan filter orde pertama derivatif dan $b = 0, c = 1$ |
| 'PI-D'  | Kontroler 2-DOF PID dengan $b = 1, c = 1$                                   |
| 'PI-DF' | Kontroler 2-DOF PID dengan filter orde pertama derivatif dan $b = 1, c = 1$ |

Matlab telah melengkapi sistem penalaan berbasis antarmuka pengguna secara grafis (*GUI, Graphical user interface*) sehingga desainer sistem kontrol PID dapat melakukan pendesainan secara cepat, mudah dan tepat. (Siswojo,2017: 45).

**pidTuner** – mengesekusi aplikasi interaktif berbasis GUI, untuk mendesain kontroler PID secara grafis.

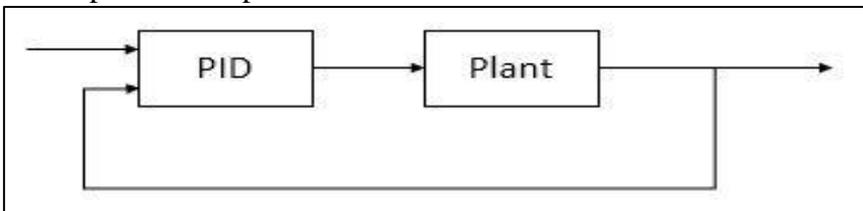
**pidTuner** – berfungsi untuk mendesain kontroler PID tipe 1-DOF atau 2-DOF. Penalaan kontroler PID tipe 1-DOF, konfigurasi sistem kontrol yang digunakan oleh penala PID diperlihatkan pada Gambar 2.6.



Gambar 2 . 6 Konfigurasi Kontroler PID 1-DOF secara diagram

Sumber : siswojo,2017: 45

Sedangkan kontroler 2-DOF, konfigurasi sistem kontrol yang digunakan oleh penala PID diperlihatkan pada Gambar 2.7.



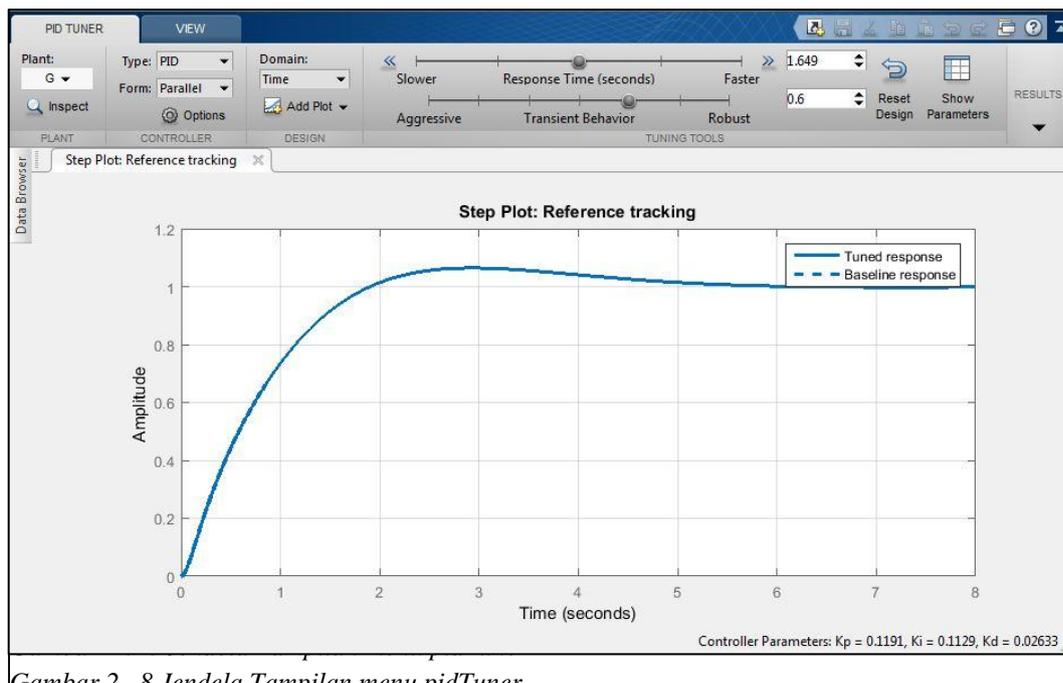
Gambar 2 . 7 Konfigurasi Kontroler PID 2-DOF secara diagram blok

Sumber : siswojo,2017: 45

**pidTuner(SYS, TYPE)** – mendesain sebuah kontroler PID untuk plant SYS. SYS adalah sistem SISO LTI seperti fungsi TF, ZPK, SS, FRD, atau sebuah model linier dihasilkan oleh aplikasi IDTF, IDFRD, IDGREY, IDPOLY, IDPROC dan IDSS. TYPE mendefinisikan jenis kontroler, dan dapat menggunakan salah satu dari yang diperlihatkan dalam tabel 2.1 Untuk sistem diskrit SYS, kontroler PID memiliki waktu sampling yang sama dengan SYS.

**pidTuner(SYS, C)** – mengambil sistem LTI sebagai kontroler panduan, selanjutnya digunakan sebagai pembanding unjuk kerja setelah dilakukan penalaan. Jika C adalah sebagai obyek PID, PIDSTD, PID2 atau PIDSTD2. Desain kontroler adalah jenis yang sama, bentuk dan metode diskritisasi sebagai C. C dapat juga sebagai sistem SS, TF, atau ZPK.

Ketika SYS adalah (1) sistem FRD atau (2) sistem SS yang memiliki waktu tunda (*delay*) secara internal dan tidak dapat diubah kedalam sistem ZPK, penala PID mengasumsikan bahwa plant tidak memiliki pole yang tidak stabil, jika terdapat pole yang tidak stabil, harus membuka dialog impor sistem linier sesudah penalaan PID diaktifkan dan mengimpor SYS dengan jumlah pole tidak stabil yang dispesifikasikan dalam dialognya. Jendela tampilan menu pidTuner dapat dilihat pada Gambar 2.8



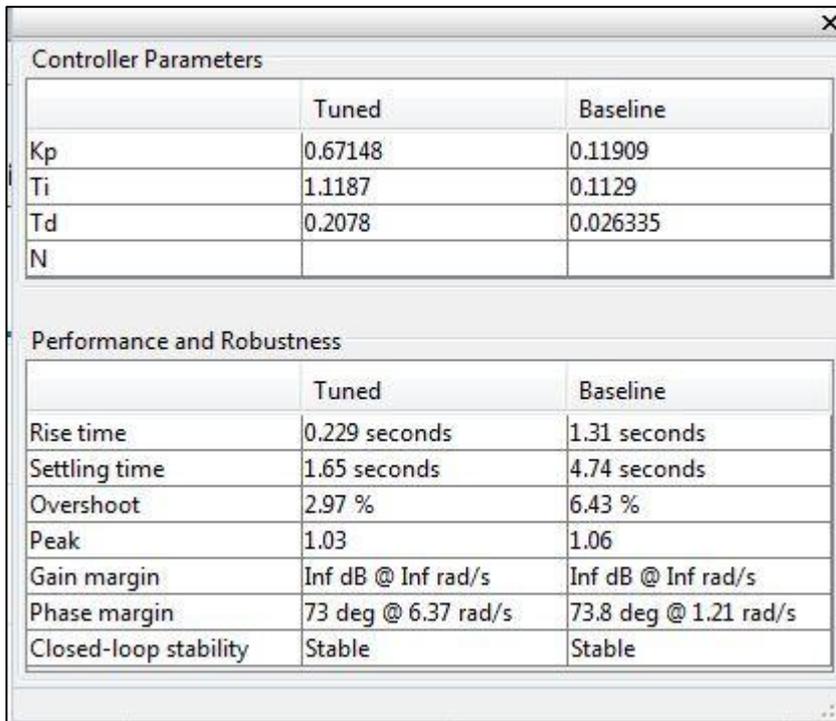
Gambar 2. 8 Jendela Tampilan menu pidTuner

**Plant** – menu daftar, dengan menu ini dapat memilih plant yaitu (1) vairabel yang telah teridefinisi, (2) membuat baru yaitu mengimpor atau mengidentifikasi plant baru.

1. **Type** – menu daftar untuk pemilihan jenis kontroler yaitu konfigurasi kontroler seperti yang terlihat pada table.
2. **Form** – menu daftar untuk pemilihan model kontroler standar atau parallel
3. **Domain** – menu daftar untuk pemilihan domain waktu dan frekuensi.
4. **Response Time** – menu geser untuk menala waktu respon sistem kontrol, ketika ditala grafik respon akan berubah secara realtime, dapat juga secara langsung memasukkan nilainya.
5. **Transient Behavior** – menu geser, untuk menala perilaku transiennya, ketika ditala grafik respon akan berubah secara realtime, dapat juga secara langsung memasukkan nilainya.
6. **Reset Design** – menu klik untuk mengembalikan penalaan seperti semula.

7. **Show Parameter** – menu klik yang menampilkan parameter kontroler, unjuk kerja dan ketegaran (*robustness*). Dapat dilihat pada gambar 2.9.

**Export** – menu yang menyediakan untuk (1) mengekspor kontroler atau plant ke ruang kerja (*workspace*) Matlab. Selanjutnya fungsi alih dapat digunakan di konsol Matlab, (2) menyimpan sebagai acuan penalaan.



| Controller Parameters |         |          |
|-----------------------|---------|----------|
|                       | Tuned   | Baseline |
| Kp                    | 0.67148 | 0.11909  |
| Ti                    | 1.1187  | 0.1129   |
| Td                    | 0.2078  | 0.026335 |
| N                     |         |          |

| Performance and Robustness |                     |                       |
|----------------------------|---------------------|-----------------------|
|                            | Tuned               | Baseline              |
| Rise time                  | 0.229 seconds       | 1.31 seconds          |
| Settling time              | 1.65 seconds        | 4.74 seconds          |
| Overshoot                  | 2.97 %              | 6.43 %                |
| Peak                       | 1.03                | 1.06                  |
| Gain margin                | Inf dB @ Inf rad/s  | Inf dB @ Inf rad/s    |
| Phase margin               | 73 deg @ 6.37 rad/s | 73.8 deg @ 1.21 rad/s |
| Closed-loop stability      | Stable              | Stable                |

Gambar 2. 9 Jendela tampilan Show Parameter

## 2.5 Fungsi Alih

Dalam teori kontrol, fungsi yang disebut “fungsi alih” seringkali digunakan untuk mencirikan hubungan masukan-keluaran dari sistem linier parameter konstan. Konsep fungsi alih hanya digunakan pada sistem linier parameter konstan, walaupun dapat diperluas untuk suatu sistem kontrol nonlinier.

Fungsi alih sistem linier parameter konstan didefinisikan sebagai bandingan dari transformasi Laplace keluaran (fungsi respon) dan transformasi Laplace masukan (fungsi penggerak), dengan anggapan bahwa semua syarat awal adalah nol.

Tinjau sistem linier parameter konstan yang didefinisikan persamaan differensial berikut:

$$a_0y^{(n)} + a_1y^{(n-1)} + \dots + a_{n-1}\dot{y} + a_ny = b_0x^{(m)} + b_1x^{(m-1)} + \dots + b_{m-1}\dot{x} + b_mx \quad (n \geq m) \dots\dots\dots(2-13)$$

di mana  $y$  adalah keluaran sistem dan  $x$  adalah masukan. Fungsi alih dari sistem ini diperoleh dengan mencari transformasi Laplace dari kedua ruas persamaan (2-13), dengan menganggap bahwa semua syarat awal adalah nol

$$\text{Fungsi alih} = G(s) = \frac{Y(s)}{X(s)} = \frac{b_0s^m + b_1s^{m-1} + \dots + b_{m-1}s + b_m}{a_0s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n} \dots\dots\dots (2-14)$$

Fungsi alih adalah suatu ekspresi yang merealisasikan keluaran dan masukan suatu sistem linier parameter konstan dalam bentuk parameter sistem dan merupakan sifat dari sistem itu sendiri, tidak bergantung pada fungsi masukan dengan keluaran, meskipun demikian, fungsi alih tidak memberikan informasi mengenai struktur fisik dari sistem.

Dengan menggunakan konsep ini, kita dapat menyatakan dinamika sistem dengan beberapa persamaan aljabar dalam  $s$ . pangkat tertinggi  $s$  penyebut fungsi alih sama dengan orde suku turunan tertinggi dari keluaran. Jika pangkat tertinggi dari  $s$  tersebut adalah  $n$ , maka sistem tersebut disebut sistem orde ke  $n$  (Ogata, 1997).

## 2.6 Analisis Respon Transien

Langkah pertama dalam menganalisis sistem kontrol adalah menurunkan model matematik sistem. Setelah model diperoleh, kita dapat menggunakan berbagai metoda untuk analisis performansi sistem(system performance).

Dalam praktek, sinyal masukan sistem kontrol tidak dapat diketahui sebelumnya, tetapi mempunyai sifat acak, sehingga masukan sesaat tidak dapat dinyatakan secara analitis. Hanya pada beberapa kasus khusus sinyal masukan dapat diketahui terlebih dahulu sehingga dapat dinyatakan secara analitis atau dengan kurva, seperti kasus kontrol otomatis pada perkakas potong.

Dalam menganalisis dan mendisain sistem kontrol, kita harus mempunyai suatu dasar perbandingan performansi berbagai sistem kontrol. Dasar ini dapat disusun dengan menetapkan sinyal-sinyal uji tertentu dan membandingkan respon berbagai sistem terhadap sinyal-sinyal masukan ini.

Beberapa kriteria disain didasarkan pada sinyal-sinyal semacam itu atau pada respon sistem terhadap perubahan syarat awal (tanpa suatu sinyal uji). Penggunaan sinyal uji dapat dibenarkan karena ada suatu korelasi antara karakteristik sistem terhadap sinyal masukan uji tertentu dan kemampuan sistem untuk mengikuti sinyal masukan yang sebenarnya.

**Sinyal uji yang khas**, sinyal masukan sinyal masukan uji (test input signals) yang biasa digunakan adalah fungsi tangga, fungsi “ramp”, fungsi percepatan, fungsi impuls, fungsi

sinusoida dan sebagainya. Dengan sinyal uji ini dapat dilakukan analisis matematik dan eksperimental sistem kontrol secara mudah karena sinyal-sinyal ini merupakan fungsi waktu yang sangat sederhana.

Jenis sinyal masukan yang akan digunakan untuk menganalisis karakteristik sistem diantara sinyal-sinyal masukan khas ini, dapat ditentukan dari bentuk masukan yang paling sering akan diberikan ke sistem pada operasi normal. Jika masukan sistem kontrol merupakan fungsi waktu yang berangsur-angsur berubah, maka fungsi waktu “ramp” mungkin merupakan sinyal uji yang baik. Demikian pula, jika sistem dikenai gangguan secara tiba-tiba, maka fungsi waktu tangga (step) mungkin merupakan sinyal uji yang baik mungkin adalah sinyal impuls. Setelah sistem kontrol didisain berdasarkan sinyal uji, performansi sistem dalam memberikan respon terhadap masukan yang sebenarnya biasanya memuaskan. Penggunaan sinyal uji memungkinkan kita untuk membandingkan performansi semua sistem dengan basis yang sama.

**Respon transien dan respon steady state**, Respon waktu sistem kontrol terdiri dari dua bagian: respon transien dan respon keadaan mantap(steady state). Yang dimaksud dengan respon transien adalah respon sistem yang berlangsung dari keadaan awal sampai keadaan akhir. Untuk respon keadaan mantap, kita maksudkan sebagai perilaku keluaran sistem jika  $\tau$  mendekati tak terhingga.(ogata,1985:222).

### **Karakteristik Respon Sistem Orde Satu**

Fungsi alih sistem orde satu dinyatakan sebagai berikut:

$$\frac{C(s)}{R(s)} = \frac{K}{\tau s + 1} \dots \dots \dots (2-15)$$

Dimana :

K = Gain Overall

$\tau$  = Konstanta Waktu

Untuk masukan sinyal *unit step*, , transformasi Laplace dari sinyal masukan . Maka, respon keluaran sistem orde satu dengan memasukkan sinyal *step* dalam kawasan *s* adalah:

$$C(s) = \frac{K/\tau}{s(s+1/\tau)} = \frac{A}{s} + \frac{B}{(s+1/\tau)} \dots \dots \dots (2-16)$$

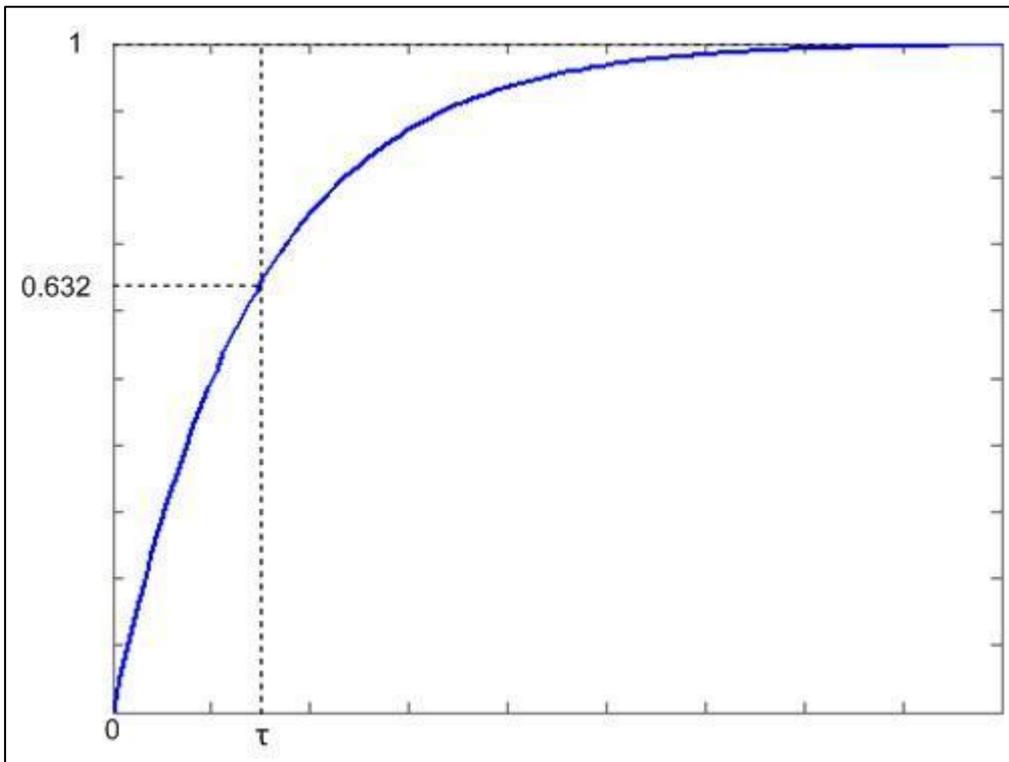
$$A = \frac{K/\tau}{s(s+1/\tau)} \Big|_{s=0} = K \dots \dots \dots (2-17)$$

$$A = \frac{K/\tau}{s} \Big|_{s=1/\tau} = -K \dots \dots \dots (2-18)$$

Dengan menggunakan inversi transformasi Laplace diperoleh respon dalam kawasan waktu yang dinyatakan dalam persamaan berikut:

$$c(t) = K \left( 1 - e^{-\frac{t}{\tau}} \right), t \geq 0 \dots\dots\dots(2-19)$$

Kurva respon orde satu untuk masukan sinyal unit *step* ditunjukkan pada Gambar 2.10.



Gambar 2. 10 Respon Orde Satu Terhadap Masukan Unit Step

Sumber : Ogata :1997

Ketika diberi masukan *unit step*, keluaran sistem  $c(t)$  mula-mula adalah nol dan terus naik hingga mencapai nilai K. salah satu karakteristik sistem orde satu adalah ketika nilai  $t = \tau$ , yaitu ketika nilai keluaran mencapai 63,2% dari nilai akhirnya.

**Karakteristik Respon Sistem Orde Dua**

Persamaan umum sistem orde dua dinyatakan oleh persamaan berikut

$$\frac{C(s)}{R(s)} = \frac{K\omega_n^2}{s^2+2\zeta\omega_n+\omega_n^2} \dots\dots\dots(2-20)$$

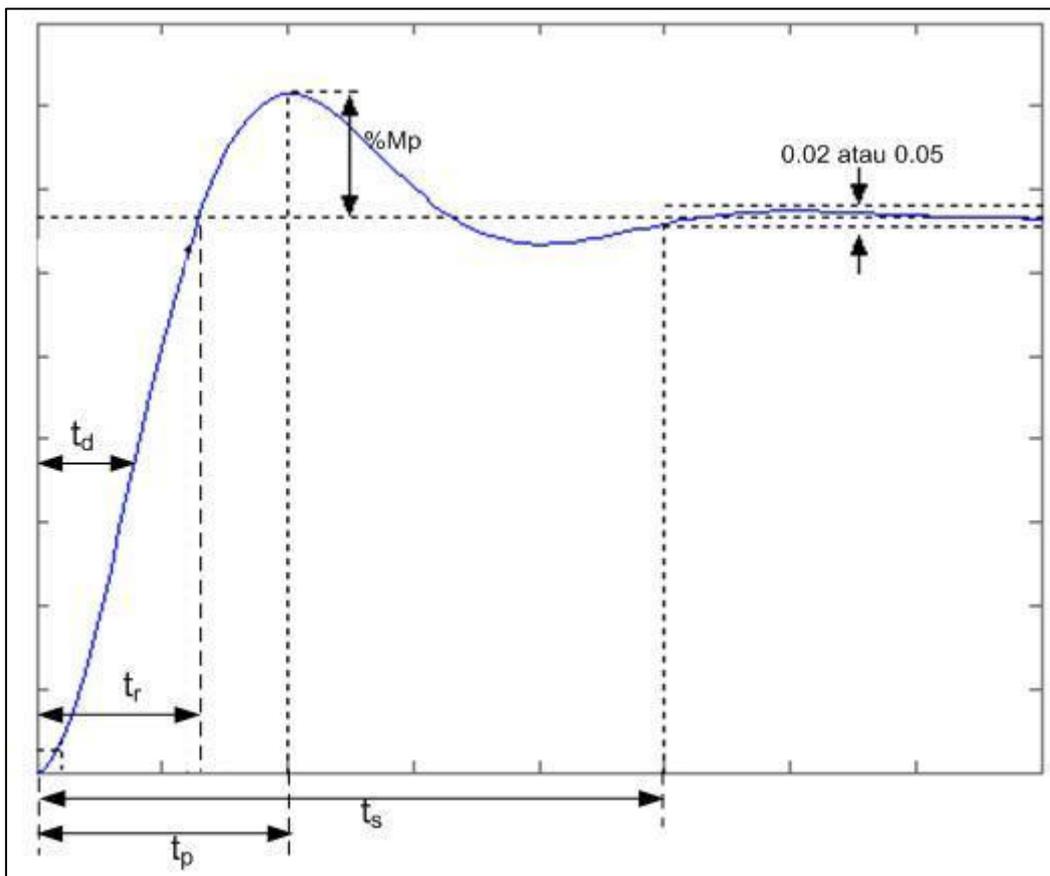
Dimana:

K = Gain

$\zeta$  = Ratio Redaman

$\omega_n$  = Frekuensi Alami

Bentuk umum kurva respon orde dua untuk masukan sinyal *unit step* ditunjukkan pada Gambar 2.11



Gambar 2 . 11 Respon Orde Dua Terhadap Masukan Unit Step

Sumber : Ogata : 1997

Dari grafik di atas diketahui karakteristik keluaran sistem orde dua terhadap masukan *unit step*, yaitu:

1. Waktu tunda (*delay time*),  $\tau_d$

Ukuran waktu yang menyatakan faktor keterlambatan respon output terhadap input, diukur mulai  $t = 0$  s/d respon mencapai 50% dari respon *steady state*. Persamaan berikut menyatakan besarnya waktu tunda dari respon orde dua.

$$\tau_d = \frac{0,742}{\zeta\omega_n} \dots\dots\dots(2-21)$$

2. Waktu naik (*rise time*),  $\tau_r$

Waktu naik adalah ukuran waktu yang di ukur mulai dari respon  $t= 0$  sampai dengan respon memotong sumbu *steady state* yang pertama. Besarnya nilai waktu naik dinyatakan pada persamaan berikut:

$$\tau_r = \frac{1}{\omega_n \sqrt{1-\zeta^n}} \left( \pi \tan^{-1} \frac{\sqrt{1-\zeta^n}}{\zeta} \right) \dots\dots\dots (2-22)$$

3. Waktu puncak (*peak time*),  $\tau_p$

Waktu puncak adalah waktu yang diperlukan respon mulai dari  $t=0$  hingga mencapai puncak pertama *overshoot*. Waktu puncak dinyatakan pada persamaan berikut:

$$\tau_p = \frac{1}{\omega_n \sqrt{1-\zeta^2}} \dots\dots\dots (2-23)$$

4. *Overshoot* maksimum,  $M_p$

Nilai relatif yang menyatakan perbandingan antara nilai maksimum respon (*overshoot*) yang melampaui nilai *steady state* dibanding dengan nilai *steady state*. Besarnya persen overshoot dinyatakan dalam persamaan berikut:

5. Waktu tunak (*settling time*),  $\tau_s$

Waktu tunak adalah ukuran waktu yang menyatakan respon telah masuk  $\pm 5\%$ , atau  $\pm 2\%$ , atau  $\pm 0.5\%$  dari keadaan *steady state*, dinyatakan dalam persamaan berikut:

$$\tau_s(\pm 5\%) = \frac{3}{\zeta \omega_n} \quad \tau_s(\pm 2\%) = \frac{4}{\zeta \omega_n} \quad \tau_s(\pm 0,5\%) = \frac{3}{\zeta \omega_n} \quad \dots\dots\dots (2-24)$$