

BAB II

TINJAUAN PUSTAKA

Bab ini berisi tentang penelitian terkait yang telah dilakukan sebelumnya dan terdapat penjelasan tentang dasar teori yang dipakai dalam mengerjakan Tugas Akhir ini. Dasar teori tersebut meliputi : dinamika sistem PCT - 100, permodelan matematis PCT-100, identifikasi sistem, pengendali *Linear Quadratic Regulator* (LQR), pengendali *Proportional Integral Derivatif* (PID), dan perangkat lunak MATLAB.

2.1 Penelitian Terkait

Dalam penelitian Tugas Akhir ini dilakukan studi literatur yang merupakan pencarian teori serta referensi yang relevan dengan kasus dan permasalahan yang akan diselesaikan, teori dan referensi didapat dari jurnal, paper, buku dan sumber lainnya.

Penelitian yang dilakukan oleh Raharjo dkk, yang berjudul Desain dan Implementasi Kontroler PID *Gain Scheduling* untuk Sistem Pengaturan Proses *Level* pada PCT - 100. Dari penelitian ini, dimana menggunakan waktu simulasi 2500 detik, menyebabkan respon kerja menjadi lambat saat terjadi perubahan *level*. Untuk mengatasi masalah tersebut, peneliti menambahkan pengendali LQR dan PID untuk membuat respon kerja yang terjadi pada perubahan *level* menjadi lebih cepat.

Raharjo dkk, dengan penelitian tentang pengendali PID *Gain Scheduling* untuk sistem pengaturan proses *level* pada PCT – 100, dimana pengendali akan mengatur nilai *gain* saat variabel mengalami perubahan sehingga hasil keluaran *set point* sesuai dengan yang diinginkan. Pengendali PID *Gain Scheduling* mengendalikan respon sistem proses *level* PCT-100 mengalami perubahann kondisi dengan rata – rata kesalahan sebesar 0,12% dimana waktu simulasi yang digunakan 2500 detik [1].

Pada penelitian yang dilakukan oleh Rachmad Ridwan, menggunakan pengendali *Hybrid Sliding Mode Control* dan PID *Gain Scheduling* pada sistem PCT – 100 menunjukkan bahwa respon kerja untuk menghasilkan *rise time* masih sangat lambat. Hal ini dapat dilihat dari nilai *settling time* (t_s), *rise time* (t_r), *delay time* (t_d) dan *error steady state* (e_{ss}). Pada kondisi ketinggian 25% diperoleh $t_s = 3953,97s$, $t_r = 2328,44s$, $t_d = 1965,05s$ dan $e_{ss} = 0,53$, untuk kondisi ketinggian 50% diperoleh $t_s = 2107,308s$, $t_r = 1240,97s$, $t_d = 1047,29s$ dan $e_{ss} = -0,01$, untuk kondisi ketinggian 75% diperoleh $t_s = 1343,199s$, $t_r = 790,99s$, $t_d = 667,54s$ dan $e_{ss} = -0,13$, dan untuk ketinggian 100% diperoleh $t_s = 1020,1835s$, $t_r = 600,77s$, $t_d = 507,01s$ dan $e_{ss} = -0,03$. Hasil simulasi menunjukkan

belum tercapainya respon yang optimal karena rata – rata waktu *rise time* yang cukup lambat, dan masih dapat ditingkatkan lagi. Untuk mendapatkan hasil yang lebih optimal dipilih pengendali optimal metode Linear Quadratic Regulator (LQR).

Penelitian pada sistem *steam drum* untuk mengendalikan *level* menggunakan pengendali optimal *Linear Quadratic Regulator* (LQR) menunjukkan bahwa pengendali optimal LQR didasari pada kemampuannya dalam mempercepat respon sistem, LQR mampu membuat sistem mencapai *set point* pada detik ke 1,5 dari total waktu simulasi selama 10 detik. Namun, masih terdapat error sebesar 0,02% [3].

Penelitian selanjutnya membahas tentang pengendalian rasio bahan bakar dan udara pada sistem *steam drum boiler* menggunakan pengendali LQR, yang difokuskan pada pengendalian tekanan dan level pada *steam drum boiler*. Untuk pengendalian level sangat cepat sehingga sistem stabil sesaat sudah *steam drum boiler* mulai bekerja. Artinya pengendali optimal LQR mampu mempercepat respon sistem [4]. Untuk itu, pengendali LQR cocok diterapkan untuk mengendalikan level pada modul training PCT - 100.

Setelah pengujian dilakukan pada modul training PCT – 100 menggunakan pengendali LQR secara simulasi, diperoleh respon keluaran belum sesuai karakteristik LQR yang mampu mempercepat respon waktu. Respon pengendalian level yang dihasilkan LQR mempunyai *settling time* (t_s) sebesar 10,645s, *rise time* (t_r) sebesar 6,269s, *delay time* (t_d) sebesar 5,2906 dan *error steady state* (e_{ss}) sebesar -18009 dengan total waktu simulasi selama 50 detik. Besar kemungkinan hal ini disebabkan oleh karakteristik sistem yang berbeda. Penyelesaian permasalahan ini dilakukan dengan menambahkan pengendali PID, karena PID memiliki efek Proporsional yang mampu mempercepat respon, memiliki Integral yang mampu menghilangkan error dan memiliki derivatif yang mampu meredam osilasi.

Berdasarkan penelitian terkait dan pengujian sistem PCT – 100 menggunakan pengendali LQR, menunjukkan bahwa pengendali LQR memiliki kemampuan dalam mempercepat respon sistem, tetapi pada sistem PCT – 100 respon keluaran yang dihasilkan oleh pengendali LQR belum sesuai karakteristik LQR yang mampu mempercepat respon waktu. Dengan adanya permasalahan tersebut penulis akan mengkombinasikan pengendali optimal LQR dan pengendali PID agar kinerja LQR menjadi lebih baik dan optimal. Optimal difokuskan pada respon waktu yang cepat dengan error minimum. Sehingga penulis tertarik melakukan penelitian dengan judul “Analisis Pengendalian *Level*

menggunakan pengendali *Linear Quadratic Regulator (LQR)* dan *Propotional Integral Derivatif (PID)* pada Modul *Training Process Control Technology – 100 (PCT – 100)*”.

2.2 Dasar Teori

2.2.1. Proses Control Technology -100 (PCT-100)

PCT-100 merupakan modul *training* kendali proses yang dilengkapi dengan beberapa sensor dan aktuator untuk aplikasi dari suatu proses. Sensor yang terdapat pada PCT-100 yaitu sensor *level, temperature, flow* dan *pressure*, sedangkan aktuator berupa *pump, heating, valve* dan *cooling*. Semua sensor dan aktuator tersebut saling terintegrasi, tetapi tidak harus diaktifkan semua, dapat diaktifkan sesuai dengan kebutuhan. Sistem ini memiliki sistem yang non linier pada proses *level* yang mengalami perubahan nilai *set point* secara langsung mengakibatkan hasil keluaran menjadi tidak sesuai dengan yang diinginkan [1].

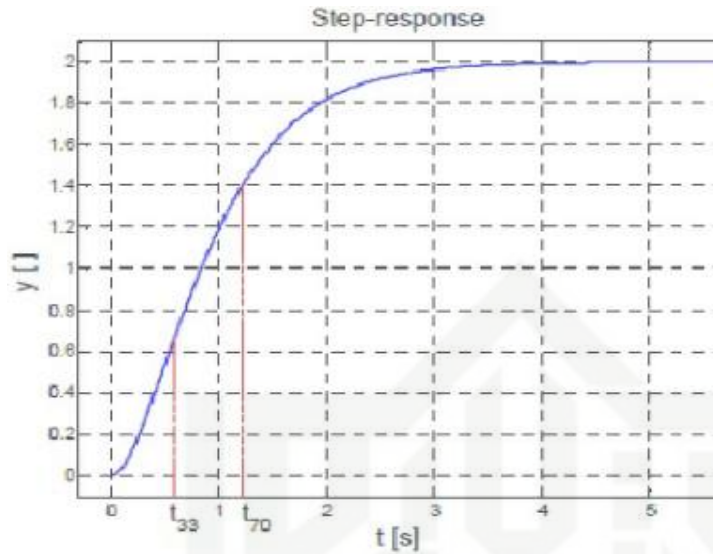


Gambar 2.1 Proses Control Techhnology -100 [1]

2.2.2. Identifikasi Sistem Proses Level PCT-100

Tahap indentifikasi adalah tahap menggambarkan perilaku sistem ataupun mencari pemodelan matematika dari sistem. Pada penelitian Rachmad Dwi Raharjo Dkk, sistem terlebih dahulu di identifikasi secara *open loop* dengan respon yang digunakan adalah respon *set point 50%* [1]. Kemudian hasil dari respon *plant* yang terbaca. Setelah itu salah satu sampel data diterapkan pada 7 model metode berdasarkan *paper* penelitian, dari 7 metode untuk identifikasi sistem, metode Viteckova orde 1 yang digunakan dalam pendekatan model matematika sistem yang sesuai, adapun persamaan metode Viteckova orde 1 identifikasi sebagai berikut :

Pada metode Viteckova orde 1 persamaan :



Gambar 2.2 Parameter Analisa Metode Viteckova Orde 1 [5]

$$G_H(s) = \frac{K}{(\tau_{v1}s+1)} e^{-T_{dv1}s} \quad (2.1)$$

- a. Tentukan nilai t_{33} yaitu saat respon bernilai 33% dari Y_{ss}

$$t_{33} = 33\% \times Y_{ss} \quad (2.2)$$

- b. Tentukan nilai t_{70} yaitu saat respon bernilai 70% dari Y_{ss}

$$t_{70} = 70\% \times Y_{ss} \quad (2.3)$$

- c. Tentukan nilai T_{dv1} melalui persamaan

$$T_{dv1} = 1,498 t_{33} - 0,498 t_{70} \quad (2.4)$$

- d. Setelah itu hitung nilai τ_{v1} melalui persamaan

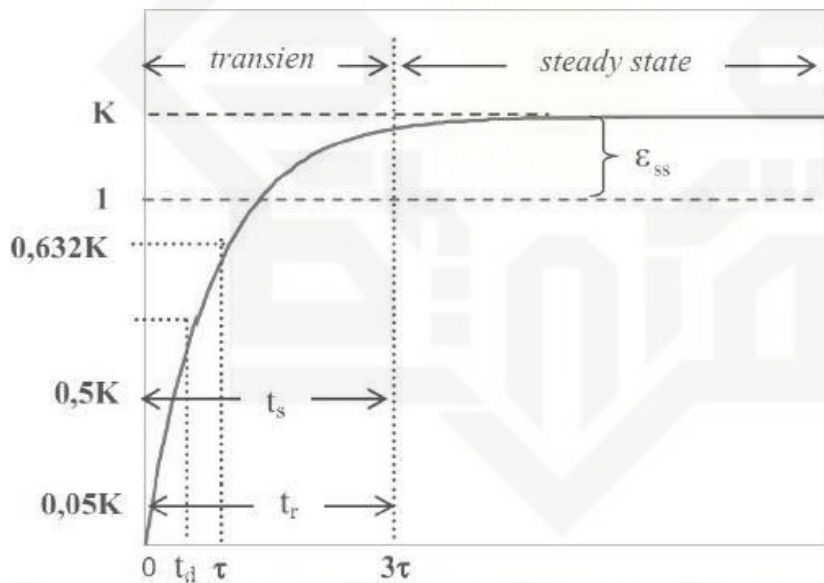
$$T_{v1} = 1,245 (t_{70} - t_{33}) \quad (2.5)$$

Hak Cipta Dilindungi Undang-Undang
 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.2.3. Identifikasi Sistem

Pengaturan proses *level* merupakan sistem berorde satu, oleh sebab itu metode identifikasi yang dilakukan adalah metode identifikasi statis. Metode identifikasi dilakukan dengan pendekatan grafis, di mana sinyal uji diberikan pada sistem untuk mengetahui respon *open loop* sistem. Dari respon sistem, dapat diketahui karakteristik-karakteristik penting dari system.

Salah satu metode identifikasi statis adalah metode pengamatan respon waktu. Identifikasi sistem dengan metode ini bekerja berdasarkan pengamatan grafis terhadap masukan *step*. Karakteristik respon waktu untuk sistem orde pertama diberikan berdasarkan respon sistem terhadap masukan sinyal *step*. Karakteristik respon waktu sistem orde pertama dibedakan menjadi karakteristik respon transien dan keadaan tunak (*steady state*). Grafik respon sistem orde pertama untuk $X_{ss} = 1$ dan $Y_{ss} = K$ ditunjukkan pada Gambar 2.3.



Gambar 2.3 Respon sistem orde pertama [6]

Respon transien terdiri dari:

- a) Spesifikasi teoritis:

Konstanta waktu (τ), adalah waktu yang dibutuhkan respon mulai dari $t=0$ sampai dengan respon mencapai 63,2% dari respon *steady state*. Konstanta waktu menyatakan kecepatan respon sistem. Konstanta waktu yang lebih kecil akan mempercepat respon

b) Spesifikasi praktis:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

1 Waktu tunak atau *settling time* (t_s), adalah ukuran waktu yang menyatakan bahwa respon sistem telah masuk pada daerah stabil. Jika dihubungkan dengan konstanta waktu τ , maka t_s dapat diformulasikan sebagai berikut:

$$t_s(\pm 5\%) = 3\tau$$

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2 Waktu naik atau *rise time* (t_r), adalah ukuran yang menyatakan bahwa respon sistem telah naik dari nilai respon pada keadaan tunak (*steady state*). Jika dihubungkan dengan konstanta waktu τ , maka t_r dapat diformulasikan sebagai berikut:

$$t_r = (10\% - 90\%) \text{ set point}$$

3. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3 Waktu tunda atau *delay time* (t_d), adalah waktu yang dibutuhkan respon mulai $t=0$ sampai respon mencapai 50% dari nilainya pada keadaan tunak (*steady state*). Waktu tunda menyatakan besarnya faktor keterlambatan respon akibat proses *sampling*. Jika dihubungkan dengan konstanta waktu τ , maka t_d dapat diformulasikan sebagai berikut:

$$t_d = 0,5 \times \text{set point}$$

4. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

4 Karakteristik respon keadaan tunak (*steady state*) sistem orde pertama diukur berdasarkan kesalahan pada keadaan tunak atau *error steady state* (e_{ss}).

$$e_{ss} = R_{ss} - C_{ss}$$

dengan C_{ss} dan R_{ss} masing-masing adalah keluaran dan masukan sistem pada keadaan tunak.

2.3 Linear Quadratic Regulator (LQR)

Pokok bahasan dalam kendali optimal adalah menentukan sinyal kendali yang akan di proses untuk memenuhi batasan fisik sesuai dengan kriteria performansi yang diinginkan. *Linear Quadratic Regulator* (LQR) merupakan salah satu metode kendali optimal dengan kriteria kuadratik untuk menyelesaikan permasalahan regulator [8]. Disebut linier karena model dan bentuk kendalinya berupa sistem linear sedangkan disebut kuadratik karena memiliki *Cost Function* yaitu kuadrat dan karena referensi sistem bukan fungsi waktu maka disebut regulator [9]. Dari hasil linierisasi didapatkan suatu *plant* linear dalam bentuk :

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.6}$$

Dimana,

- A : matriks sistem
- B : matriks *input*
- C : matriks *output*
- y : *State output*
- x : *State sistem*
- u : *State input*

Pada perancangan pengendali optimal LQR, terlebih dahulu menentukan matriks Q dan R yang selanjutnya digunakan untuk menentukan indeks performansi sistem, harga matriks Q dan R ditentukan sesuai dengan kriteria yang diinginkan dengan menggunakan indeks performansi [10].

$$J(t_0) = \frac{1}{2} S(T)x^2(T) + \frac{1}{2} \int_{t_0}^T (Qx^2 + Ru^2) \tag{2.7}$$

Dengan syarat :

$$S(T) \geq 0, \quad Q \geq 0, \quad R > 0$$

dimana

- t_0 = waktu awal
- T = waktu akhir
- x = matriks *State* akhir
- Q = matriks semi definit positif
- R = matriks definit positif
- S = matriks semi definit positif

Sehingga diperoleh persamaan Hamilton

$$\begin{aligned} H(x, u, \lambda, t) &= L(x, u, t) + \lambda^T f(x, u, t) \\ H(x, u, \lambda, t) &= \frac{1}{2} (x^T Qx + u^T Ru) + \lambda^T (Ax + Bu) \end{aligned} \tag{2.8}$$

Dari fungsi Hamilton tersebut dapat diperoleh syarat perlu dan syarat batas sebagai berikut:

1. Syarat perlu

a) Persamaan *State*

$$\dot{x} = \left(\frac{\partial x}{\partial x}\right)^T Qx + \left(\frac{\partial Qx}{\partial \lambda}\right)^T u = Ax + Bu \quad (2.9)$$

b) Persamaan *Costate*

$$-\dot{\lambda} = \frac{1}{2} \left[\left(\frac{\partial x}{\partial x}\right)^T Qx + \left(\frac{\partial Qx}{\partial x}\right)^T x + \left(\frac{\partial u}{\partial x}\right)^T Ru + \left(\frac{\partial Ru}{\partial x}\right)^T u \right] + \left(\frac{\partial \lambda}{\partial x}\right)^T (Ax + Bu) + \left(\frac{\partial (Ax + Bu)}{\partial x}\right)^T \lambda$$

$$-\dot{\lambda} = Qx + A^T \lambda \quad (2.10)$$

c) Kondisi stasioner

$$\frac{\partial H}{\partial u} = 0$$

$$\frac{1}{2} \left[\left(\frac{\partial x}{\partial x}\right)^T Qx + \left(\frac{\partial Qx}{\partial x}\right)^T x + \left(\frac{\partial u}{\partial x}\right)^T Ru + \left(\frac{\partial Ru}{\partial x}\right)^T u \right] + \left(\frac{\partial x}{\partial u}\right)^T (Ax + Bu) + \left(\frac{\partial (Ax + Bu)}{\partial u}\right)^T \lambda = 0$$

$$Ru + B^T \lambda = 0$$

$$u = -R^{-1} B^T \lambda \quad (2.11)$$

2. Syarat batas

a. Batas awal

$$t = 0 \quad x(0) = 0$$

b. Batas akhir

$$(\phi_x + \psi_x^T v - \lambda)^T dx|_T + (\phi_t + \psi_t^T v - H) dt|_{t=T} = 0 \quad (2.12)$$

Karena $\psi = 0$ dan

$$\phi(T) = \frac{1}{2} x^T(T) S(T) x(T)$$

$$\phi_x = \frac{\partial \phi}{\partial x} = S(T) x(T)$$

sehingga diperoleh persamaan

$$(S(T) x(T) + v \cdot 0 - \lambda)^T dt|_{t=T} = 0$$

$$S(t) x(t) = \lambda(t) \quad (2.13)$$

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Dari persamaan *Costate* dan (2.13), keduanya mengandung variabel λ . Bila keduanya disubstitusikan maka akan menghasilkan persamaan:

$$\begin{aligned}
 S(t)x(t) &= \lambda(t) \\
 \dot{\lambda}(t) &= \dot{S}(t)x(t) + S(t)\dot{x}(t) \\
 \dot{\lambda}(t) &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\dot{x}(t) &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\{Ax + Bu\} &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\{Ax + B(-R^{-1}B^T \lambda)\} &= -Qx - A^T \lambda \\
 -\dot{S}x &= (A^T S + SA - SBR^{-1}B^T S + Q)x \\
 -\dot{S} &= A^T S + SA - SBR^{-1}B^T S + Q
 \end{aligned} \tag{2.14}$$

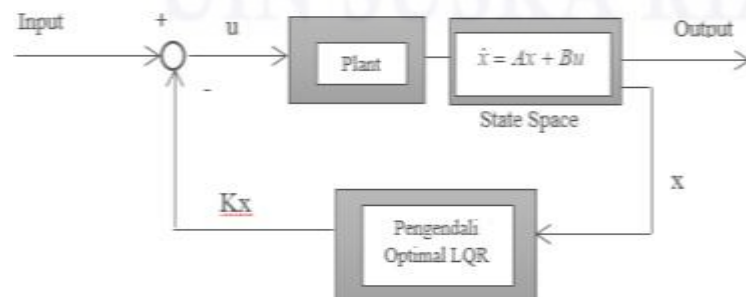
Persamaan (2.14) disebut persamaan differensial Riccati. Untuk $\dot{S} = 0$ persamaannya disebut *Algebraic Riccati Equation*(ARE) [10]. Dengan di dapatkannya matriks S dari persamaan ARE ini sehingga persamaan sinyal kendali pun dapat di hitung:

$$u = -R^{-1}B^T \lambda = -R^{-1}B^T Sx = -Kx \tag{2.15}$$

Maka konstanta umpan balik keadaan K dapat dicari sebelum menghasilkan sinyal kendali pada persamaan (2.16) [10].

$$K = R^{-1}B^T S \tag{2.16}$$

Dimana nilai S adalah unik, solusi semi definit positif untuk persamaan ARE harus memenuhi syarat persamaan (2.14). Selanjutnya dalam perancangan teknik kendali optimal LQR, Setelah nilai S diketahui maka, nilai S tersebut disubstitusikan kedalam persamaan (2.16) sehingga didapatkan nilai matriks optimal K, dengan K adalah *Gain State Feedback*, sehingga sistem kendali optimal LQR hasil desain dapat di perlihatkan pada gambar 2.7 berikut:



Gambar 2.4 Sistem Kendali dengan Skema Kendali Optimal LQR

Dari penjabaran di atas dapat disimpulkan bahwa langkah-langkah untuk merancang kendali LQR adalah sebagai berikut :

1. Jika persamaan matematis masih dalam bentuk *transfer function*, ubah bentuk *transfer function plant* nonlinier menjadi bentuk *state space* dan dapatkan matriks A, B, C dan D dari *plant* tersebut.
2. Tentukan matriks pembobot Q dan R.
3. Selesaikan persamaan aljabar riccati hingga menghasilkan suatu matriks S yang definit positif

Hitung optimal *gain feedback* K.

2.4 Proportional Integral Derivative (PID)

PID adalah salah satu pengendali otomatis yang terdiri dari perpaduan tiga aksi dasar kendali yaitu Proporsional, Integral, dan Derivatif. pada masing-masing memiliki aksi yang khas pada tiap-tiap kekurangan tiap pengendali P, I dan D dapat saling menutupi dengan menggabungkan ke tiganya dan merangkai secara paralel menghasilkan pengendali PID. Pada masing- masing aksi keseluruhan memiliki tujuan untuk mempercepat reaksi sebuah sistem yang mampu menghilangkan offset dan menghasilkan perubahan yang besar [12]. Oleh karena itu, pengendali PID merupakan pengendali berumpan balik dengan baik pada beberapa aplikasi proses petroleum, proses polimer, dan lain sebagainya.

Adapun bentuk umum dari aksi kendali PID sebagai berikut :

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de}{dt} \tag{2.17}$$

Persamaan 2.0- diubah ke dalam bentuk Laplace menjadi

$$\frac{Y(s)}{U(s)} = k_p + \frac{k_i}{s} + k_d s \tag{2.18}$$

Dimana :

k_p = Proporsional Gain

k_i = Integral Gain

k_d = Derivatif Gain

Pengendali proporsional (K_p) akan memberikan efek mengurangi waktu naik, tetapi tidak menghapus *error steady state*. Pengendali integral (K_i) akan memberikan efek menghapus *error steady state* tetapi berakibat buruk pada saat menanggapi respon. Pengendali derivatif (K_d) berdampak pada stabilitas sistem, mengurangi kesalahan keadaan

tunak dan menaikkan respon transien. Hubungan dari ketiga aksi ditunjukkan pada Tabel 2.1.

2.4.1 Tuning (Penalaan) Parameter PID

Istilah penalaan (*tuning*) untuk menggambarkan metode-metode yang digunakan dalam memilih pengontrolan agar mendapat unjuk kerja yang diinginkan. Pada penelitian ini, ini dilakukan penalaan pada pengendali P,I dan D. Metode penalaan parameter pengendali yang digunakan adalah metode uji coba atau Metode Heuristik yaitu dengan mengkominasikan antara parameter antara pengendali hingga didapatkan respon yang diinginkan.

Hubungan dari ketiga aksi pengendali tersebut ditunjukkan pada Tabel berikut :

Tabel 2.1 Hubungan antara kendali P, I, dan D [12]

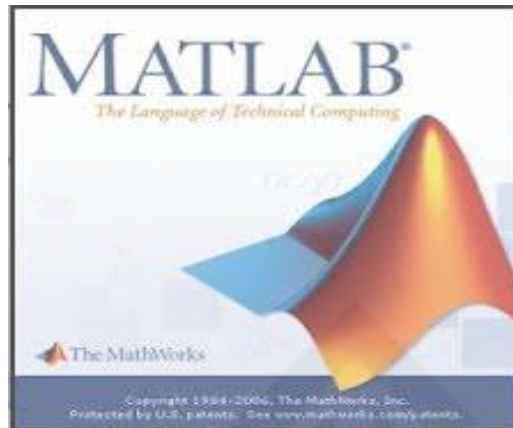
| Respon lup tertutup | Waktu naik | <i>Overshoot</i> | Waktu turun | Kesalahan keadaan tunak |
|----------------------|-----------------|------------------|-----------------|-------------------------|
| <i>K_p</i> | Menurun | Meningkat | Perubahan Kecil | Menurun |
| <i>K_i</i> | Menurun | Meningkat | Meningkat | Hilang |
| <i>K_d</i> | Perubahan Kecil | Menurun | Menurun | Perubahan Kecil |

2.5 MATRIX LABORATORY (MATLAB)

MATLAB adalah singkatan dari MATrix LABoratory. Pertama kali dibuat untuk mempermudah penggunaan dua koleksi subrutin pada pustaka FORTRAN yaitu: LINPACK dan EISPACK, dalam menangani komputasi matriks. Sejak itu, MATLAB berkembang menjadi sebuah sistem yang interaktif sekaligus sebagai bahasa pemrograman untuk keperluan-keperluan ilmiah, komputasi teknis, dan visualisasi. Elemen data dasar MATLAB adalah matriks. Perintah-perintah diekspresikan dalam bentuk yang sangat mirip dengan bentuk yang digunakan dalam matematika dan bidang teknik. Contoh persamaan $b=Ax$, dengan A, b, dan x matriks, ditulis: $b=A*x$. Untuk mendapat solusi x dari A dan b, tulis: $x=A\b$. Tidak diperlukan penulisan program khusus untuk operasi-operasi matriks seperti perkalian matriks atau invers matriks. Oleh karena itu bahasa MATLAB menyelesaikan masalah tersebut memerlukan waktu lebih cepat dibanding waktu yang dibutuhkan bahasa pemrograman tingkat tinggi lain.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 2.5 Tampilan Matlab

Pada pertengahan tahun 1970, Cleve Moler dan beberapa rekan tergabung dalam suatu team pengembangan *software* yang dibiayai oleh *The National Science Foundation* untuk tujuan membuat subrutin-subrutin dalam pustaka FORTRAN yang dinamai LINPACK dan EISPACK. LINPACK berisi koleksi subrutin untuk penyelesaian persamaan linear, sementara EISPACK adalah koleksi subrutin untuk penyelesaian masalah nilai pribadi (*eigenvalue*). Baik LINPACK maupun EISPACK pada prinsipnya merupakan program untuk komputasi matriks.

Dipenghujung tahun 1970, Cleve ingin dapat mengajarkan kepada mahasiswa materi aljabar linear di Universitas New Mexico menggunakan LINPACK dan EISPACK tanpa harus menulis rutin-rutin program dalam bahasa FORTRAN. Berdasarkan keinginan tersebut, Cleve mulai menulis program untuk memberikan kemudahan akses interaktif pada LINPACK dan EISPACK. Cleve menamakan programnya dengan MATLAB yang merupakan singkatan dari MATrix LABoratory. Beberapa tahun kemudian, ketika Cleve berkunjung ke universitas lain untuk berbicara, atau sebagai *Visiting Professor*, Cleve meninggalkan duplikasi MATLABnya pada komputer di universitas tersebut. Hanya dalam satu atau dua tahun, MATLAB versi pertama ini telah menjadi buah bibir pembicaraan orang, terutama yang berada dalam komunitas matematika terapan.

Dari hasil kunjungan Cleve di Universitas Stanford, sekitar awal tahun 1983, John Little, seorang *engineer*, menampilkan MATLAB dengan memperkenalkan penerapan MATLAB yang potensial dalam bidang-bidang keteknikan. Karena itu, dalam tahun 1983, Little, Moler, dan Steve Bangert membentuk team untuk mengembangkan MATLAB generasi kedua. MATLAB versi ini dibuat menggunakan bahasa C dan terintegrasi dengan

grafik. *The MathWorks, Inc.* didirikan tahun 1984 untuk memasarkan dan melanjutkan pengembangan MATLAB.

Penggunaan MATLAB:

1. Adanya *toolbox* yang disebut MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer
2. Komputasi matematik dan komputasi teknik
3. Analisis data.
4. Simulasi dan pemodelan
5. Grafik-grafik perhitungan.
6. Manipulasi matriks
7. Pem-plot-an fungsi dan data
8. Implementasi algoritma
9. Pembuatan antarmuka pengguna / *interface* (visualisasi)
10. Peng-antarmuka-an dengan program dalam bahasa lainnya

Fungsi-fungsi yang digunakan MATLAB:

1. Fungsi matematika
2. Fungsi fisika
3. Fungsi statistik
4. Fungsi visualisasi.

MATLAB merupakan *software* yang digunakan untuk:

1. Para pendidikan (untuk melakukan riset dalam bentuk komputasi, pemodelan, simulasi dan demonstrasi / visualisasi)
2. Para ilmuwan
3. Matematikawan
4. Pelajar dan mahasiswa terutama para pendidik matematika
5. Praktisi
6. Insinyur.
7. dan untuk siapa saja tanpa ada batasannya

Sebagai sebuah system, MATLAB tersusun dari 5 bagian utama:

1. *Development Environment*.

Merupakan sekumpulan perangkat dan fasilitas yang membantu anda untuk menggunakan fungsi-fungsi dan file-file MATLAB. Beberapa perangkat ini merupakan sebuah *graphical user interfaces* (GUI). Termasuk didalamnya adalah MATLAB desktop dan *Command Window*, *command history*, sebuah *editor* dan *debugger*, dan *browsers* untuk melihat *help*, *workspace*, *files*, dan *search path*.

2. *MATLAB Mathematical Function Library*

Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti: *sum*, *sin*, *cos*, dan *complex arithmetic*, sampai dengan fungsi-fungsi yang lebih kompleks seperti *matrix inverse*, *matrix eigenvalues*, *Bessel functions*, dan *fast Fourier transforms*.

3. *MATLAB Language*

Merupakan suatu *high-level matrix/array language* dengan *control flow statements*, *functions*, *data structures*, *input/output*, dan fitur-fitur *object-oriented programming*. Ini memungkinkan bagi kita untuk melakukan kedua hal baik “pemrograman dalam lingkup sederhana ” untuk mendapatkan hasil yang cepat, dan “pemrograman dalam lingkup yang lebih besar” untuk memperoleh hasil-hasil dan aplikasi yang kompleks.

4. *Graphics*

MATLAB memiliki fasilitas untuk menampilkan vector dan matrices sebagai suatu grafik. Didalamnya melibatkan *high-level functions* (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, *image processing*, *animation*, dan *presentation graphics*. Ini juga melibatkan fungsi level rendah yang memungkinkan bagi anda untuk membiasakan diri untuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan *graphical user interfaces* pada aplikasi MATLAB anda.

5. *MATLAB Application Program Interface (API)*

Merupakan suatu *library* yang memungkinkan program yang telah anda tulis dalam bahasa C dan Fortran mampu berinteraksi dengan MATLAB. Ini melibatkan fasilitas untuk pemanggilan *routines* dari MATLAB (*dynamic linking*), pemanggilan MATLAB sebagai sebuah *computational engine*, dan untuk membaca dan menuliskan *MAT-files*. Kehadiran Matlab memberikan jawaban sekaligus tantangan. Disini kita

bisa memilih apa aja yg ingin kita pelajari, mempelajari metoda visualisasi saja, pemrograman saja, atau keduanya di sini intinya Matlab memang dihadirkan bagi mereka yang tidak ingin disibukkan dengan rumitnya sintak dan alur logika pemrograman sementara pada saat yang sama membutuhkan hasil komputasi dan visualisasi yang maksimal untuk mendukung pekerjaannya. Selain itu Matlab juga memberikan keuntungan bagi *programmer* dan *developer* program yaitu menjadi programpembanding yang sangat handal, hal tersebut karena kekayaannya akan fungsi matematika, fisika, statistik dan visualisasi.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

