

BAB II

TINJAUAN PUSTAKA

Bab ini berisi tentang penelitian terkait yang telah dilakukan sebelumnya dan terdapat penjelasan tentang dasar teori yang dipa kai dalam mengerjakan Tugas Akhir ini. Dasar teori tersebut meliputi : dinamika sistem *heat exchanger*, permodelan matematis *heat exchanger*, 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 Murie Dwiyani yang berjudul *Tuning Parameter PID dengan Metode Ciancone pada Plant Heat Exchanger*. Dari penelitian ini, dimana dilihat dari hasil respon yang belum optimal karena rata-rata waktu *rise time* yang cukup lambat. Untuk meningkatkan hasil respon yang lebih optimal, peneliti menambahkan pengendali LQR untuk membuat hasil respon yang terjadi pada perubahan temperatur lebih optimal.

Murie Dwiyani, dengan penelitian tentang pengendali PID untuk sistem pengaturan temperatur pada HE (*Heat Exchanger*), dimana pengendali akan mengatur parameter saat variabel mengalami perubahan sehingga hasil keluaran *set point* sesuai dengan yang diinginkan. Pengendali PID mengendalikan respon sistem proses HE (*Heat Exchanger*) mengalami perubahan kondisi dengan *overshoot* sebesar 2,27% dimana waktu simulasi yang digunakan 100 detik [4].

Pada penelitian yang dilakukan oleh Djulil Amri, menggunakan pengendali *feedforward* dan PID pada pengendalian temperatur HE (*Heat Exchanger*) menunjukkan respon kerja untuk menghasilkan *rise time* masih sangat lambat. Hal ini dapat dilihat dari nilai *settling time* yang terjadi sebesar 58 detik dan *overshoot* sebesar 5% [5].

Kemudian penelitian yang dilakukan oleh Eko Ariyanto dan Cekmas Cekdin, dengan penelitiannya tentang permodelan dan simulasi PID kontrol pada alat penukar panas menunjukkan kinerja pengendali PID *Feedback Controller* menghasilkan *overshoot* yang sangat tinggi dan pengendali PID IMC menghasilkan *overshoot* sebesar 6,2%, *settling time*

sebesar 88,8 detik dan *rise time* sebesar 59,8 detik. Dari beberapa penelitian tentang HE mempunyai kelemahan dari segi *time response* terutama *rise time* yang belum memuaskan. Kelemahan lainnya adalah terdapat pada nilai *overshoot*. Berdasarkan hal tersebut dipilih pengendali optimal dengan metode LQR (*Linear Quadratic Regulator*) [6].

Penelitian tentang pengendalian temperatur dengan metode *linear quadratic regulator* (LQR) yang dilakukan Ahmad Fahmi, menunjukkan bahwa pengendali optimal LQR didasari pada kemampuannya dalam mempercepat respon sistem, LQR mampu membuat sistem menjadi optimal dengan *setpoint* sebesar 1570°C dengan total waktu simulasi selama 10000 detik. Menghasilkan *settling time* sebesar 2000 detik dan lebih cepat 700 detik dari nilai *settling time* yang menggunakan pengendali PID, dan *error steady state* sebesar 0,18% [7].

Beberapa penelitian lainnya yang dapat membuktikan bahwa pengendali optimal LQR mampu membuat kinerja sistem lebih optimal yaitu: pertama penelitian yang dilakukan oleh Ilham, dengan judul pengendalian kecepatan motor induksi 3 fasa dengan pengendali optimal LQR. Informasi yang didapat dari penelitian ini adalah kecepatan putaran motor dengan pengendali LQR pada saat diberi torsi beban 1 Nm menghasilkan *overshoot* sebesar 0,0028% atau sebesar 1396 rpm, begitupun saat motor diberikan torsi beban 2 Nm menghasilkan *overshoot* sebesar 0,0028% atau 1396 rpm. dan ketika putaran torsi diturunkan 1 Nm akan menghasilkan *overshoot* sebesar 0,0028%. Dari informasi yang diperoleh tersebut dapat diketahui bahwa dengan menggunakan pengendali LQR dalam mengendalikan kecepatan motor induksi 3 fasa mampu memberikan respon dinamik yang cukup baik dan mampu memperbaiki kecepatan putaran motor induksi pada saat perubahan torsi beban [8].

Kedua penelitian yang dilakukan oleh Dista Yoel Tadeus, dengan judul simulasi pengendalian daya reaktor nuklir dengan teknik kontrol optimal LQR. Dari penelitian tersebut diperoleh informasi bahwa pengendali LQR mampu membuat respon daya reaktor nuklir semakin cepat menuju titik stabil dengan harga q besar dan harga r kecil namun memiliki konsekuensi penggunaan energi yang besar [9].

Ketiga penelitian yang dilakukan oleh Modestus Oliver Asali, dkk, yang berjudul analisa pengendalian gerak model robot keseimbangan beroda dua menggunakan pengendali LQR. Dari penelitian ini, dapat diketahui untuk keluaran sudut kemiringan pendulum (α) memiliki keluaran *delay time* sebesar 0,23 detik, *rise time* sebesar 0,39 detik, *settling time* sebesar 2,23 detik dengan *overshoot* sebesar 0,0216% dan untuk keluaran

orientasi robot (θ) memiliki keluaran *delay time* sebesar 0,85 detik, *rise time* sebesar 1,42 detik, *settling time* sebesar 2,76 detik dengan *overshoot* sebesar 2,13%. Dari penelitian tersebut diperoleh informasi bahwa pengendali LQR dalam mengendalikan keseimbangan mampu memberikan respon waktu yang lebih cepat dan juga kesalahan keadaan tunak yang lebih kecil [10].

Berdasarkan penelitian terkait dan pengujian sistem HE menggunakan pengendali LQR, menunjukkan bahwa pengendali LQR mampu membuat sistem dalam mencapai keadaan *steady state* dan memperkecil *error steady state*, namun dengan pengendali LQR ini *rise time* yang dihasilkan belum sesuai dengan yang diinginkan. Dengan adanya permasalahan tersebut penulis akan mengkombinasikan pengendali optimal LQR agar kinerja LQR menjadi lebih baik dan optimal. Optimal difokuskan pada *rise time* yang cepat dengan *error* minimum. Sehingga penulis tertarik melakukan penelitian dengan judul “Analisa Pengendalian Temperatur menggunakan pengendali *Linear Quadratic Regulator* (LQR) dan *Proportional Integral Derivatif* (PID) pada sistem HE (*Heat Exchanger*) tipe *shell and tube*”.

2.2 Dasar Teori

2.2.1 HE (*Heat Exchanger*) Tipe *Shell and Tube*

2.2.1.1 HE (*Heat Exchanger*)

HE (*Heat Exchanger*) merupakan alat yang berfungsi memindahkan energi panas antara dua fluida atau lebih fluida, perpindahan ini dapat terjadi pada temperatur yang berbeda antar beberapa fluida yang mana satu fluida memiliki temperatur yang lebih tinggi dan fluida yang lain memiliki temperatur yang lebih rendah. Proses perpindahan panas tersebut dapat dilakukan secara langsung maupun tidak langsung [12].

Adapun penjelasan dari dua proses perpindahan tersebut yaitu :

1. Proses perpindahan secara langsung yaitu dimana fluida yang panas akan bercampur secara langsung dengan fluida dingin (tanpa adanya pemisah) dalam suatu bejana atau ruangan tertentu.
2. Proses perpindahan secara tidak langsung yaitu dimana fluida panas tidak berhubungan langsung dengan fluida dingin atau melalui media perantara seperti pipa, pelat atau peralatan jenis lainnya dalam melakukan proses perpindahannya.

2.2.1.2 Klasifikasi HE (*Heat Exchanger*)

Dari beberapa jenis alat HE, dapat diklarifikasikan menjadi bermacam-macam pertimbangan yaitu:

1. Klasifikasi berdasarkan proses perpindahan panas
 - Tipe kontak langsung
 - Tipe kontak tidak langsung
2. Klasifikasi berdasarkan jumlah fluida yang mengalir
 - Dua jenis fluida
 - Tiga jenis fluida
 - N-jenis fluida (N lebih dari tiga)
3. Klasifikasi berdasarkan kompaknya permukaan
 - Tipe HE yang kompak
Density luas permukaannya $> 700 \text{ m}^2/\text{m}^3$
 - Tipe HE yang tidak kompak
Density luas permukaannya $< 700 \text{ m}^2/\text{m}^3$
4. Klasifikasi berdasarkan mekanisme perpindahan panas
 - Dengan cara konveksi, satu fase pada kedua sisi alirannya
 - Dengan cara konveksi pada satu sisi aliran dan pada sisi yang lainnya terdapat cara konveksi dua aliran
 - Dengan cara konveksi pada kedua sisi alirannya serta terdapat dua *pass* aliran masing-masing
 - Kombinasi cara konveksi dan radiasi
5. Klasifikasi berdasarkan konstruksi
 - Kontruksi tubular (*shell and tube*)
 - Kontruksi tipe pelat (*plate*)
 - Kontruksi dengan permukaan diperluas (*extended surface*)
Regenerative
6. Klasifikasi berdasarkan pengaturan aliran
 - Aliran dengan satu pass (*singel phase*)
 - Aliran multipass (*multiphase*)
Shell and tube aliran yang berlawanan
 - Multipass pelat

2.2.1.3 HE (*Heat Exchanger*) *Shell and Tube*

HE (*Heat Exchanger*) tipe ini adalah salah satu jenis HE yang menurut konstruksinya terdiri dari sebuah *Shell* silindris dibagian luar dan sejumlah *Tube* dibagian dalam. Alat ini memiliki temperatur berbeda pada dua bagiannya sehingga terjadi perpindahan panas antara aliran fluida di dalam *Tube* dan di luar *Tube*. Adapun yang berhubungan dengan bagian dalam *Tube* disebut *Tube Side* dan yang dibagian luar disebut *Shell Side* [12].

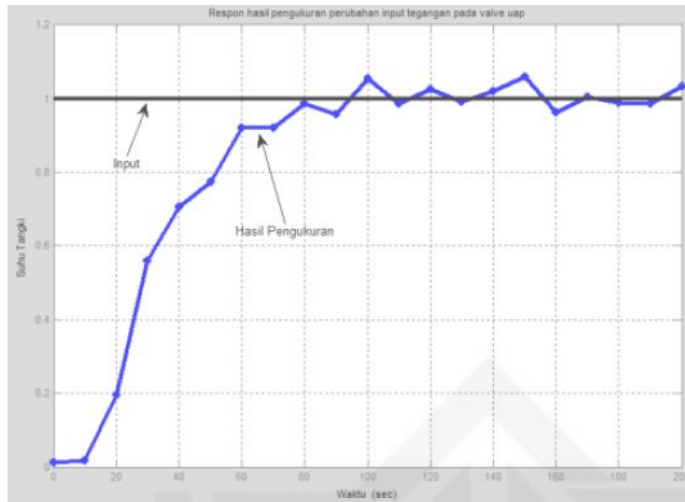
Pada HE tipe ini proses mengalirnya fluida terjadi secara terpisah, dimana satu fluida mengalir dalam pipa dan fluida yang lainnya dialirkan dalam *shell*. Ketika melakukan proses perpindahan fluida, pada bagian *tube* ditahan dengan sekat atau *baffle*. Pada bagian *shell* harus ada ruang bebas (*clearance*) antara *shell* dan sekat (*baffle*), yang mana ruang bebas tersebut dapat mempengaruhi aliran fluida di luar *tube*. Semakin besar ruang bebas akan semakin besar pula kebocoran antara *shell* dan sekat (*baffle*) [12].



Gambar 2.1 *Heat Exchanger* (HE) [13]

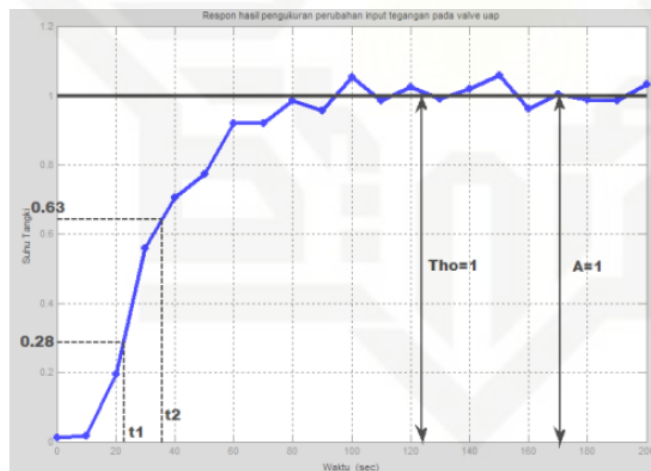
2.2.2 Identifikasi Sistem Proses Temperatur HE

Tahap identifikasi adalah tahap menggambarkan perilaku sistem ataupun mencari pemodelan matematika dari sistem. Pada penelitian yang dilakukan oleh Murie Dwiyanti, sistem terlebih dahulu diidentifikasi secara *open loop*, kemudian *input step* berupa tegangan (v) diberikan ke *valve* lalu keluaran sistem berupa temperatur pada tangki direkam, hasil pengidentifikasi sistem dapat dilihat pada Gambar 2.2 berikut:



Gambar 2.2 Data Pengukuran Sistem [4]

Dari data yang terlihat pada Gambar 2.2 diatas, dibuatlah pemodelan matematika sistem dalam bentuk fungsi alih dengan pendekatan orde satu. Fungsi alih berguna untuk melakukan analisa dinamik dan perancangan sistem pengendali, adapun persamaan fungsi alih dari sistem sebagai berikut:



Gambar 2.3 Parameter Sistem [4]

Dari Gambar 2.3 diketahui langkah-langkah dalam menentukan parameter:

- Melakukan pendekatan orde satu terhadap data (Gambaran 2.3) dengan pemodelan grafik ciancone dengan menghitung penguatan *proportional* (K_p) yang merupakan nilai keluaran (Δ) dibagi dengan nilai masukan (δ)

$$A = \Delta = 1 \tag{2.1}$$

$$Tho = \delta = 1 \tag{2.2}$$

$$K_p = \frac{\Delta}{\delta} = \frac{1}{1} = 1 \tag{2.3}$$

2. Menentukan konstanta waktu (τ) dengan mencari waktu yang diperlukan untuk mencapai 28% dari keadaan mapan ($t_{28\%}$) dan waktu yang diperlukan untuk mencapai 63% keadaan mapan ($t_{63\%}$)

$$\Delta 28\% = 1 \times 28\% = 0,28 \tag{2.4}$$

$$\Delta 63\% = 1 \times 63\% = 0,63 \tag{2.5}$$

$$t_{28\%} = t_1 = 21,8 \text{ detik} \tag{2.6}$$

$$t_{63\%} = t_2 = 36 \text{ detik} \tag{2.7}$$

$$\tau = 1,5(t_{63\%} - t_{28\%}) = 1,5(36 - 21,8) = 21,3 \tag{2.8}$$

3. Mencari waktu tunda (θ) dengan persamaan:

$$\theta = t_{63\%} - \tau = 36 - 21,3 = 14,70 \tag{2.9}$$

Dari identifikasi yang telah dilakukan oleh Murie Dwiyanti, dalam menentukan fungsi alih sistem Murie Dwiyanti merujuk pada penelitian yang dilakukan oleh Hsiao-Ping Huang, Ming-Wei Lee dan Cheng-Liang Chen dengan judul penelitian “*A System of Procedures for Identification of Simple Models Using Transient Step Response*” didapatkan fungsi alih sistem sebagai berikut:

$$\frac{Y(s)}{X(s)} = \frac{K_{pe} e^{-\theta s}}{\tau s + 1} \tag{2.10}$$

2.2.3 Identifikasi Sistem

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

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 sinyal *step*. Karakteristik respon waktu sistem orde pertama dibedakan menjadi karakteristik respon transien dan keadaan tunak (*steady state*). Grafik respon sistem orde satu untuk $X_{ss} = 1$ dan $Y_{ss} = K$ seperti yang terlihat pada Gambar 2.3.

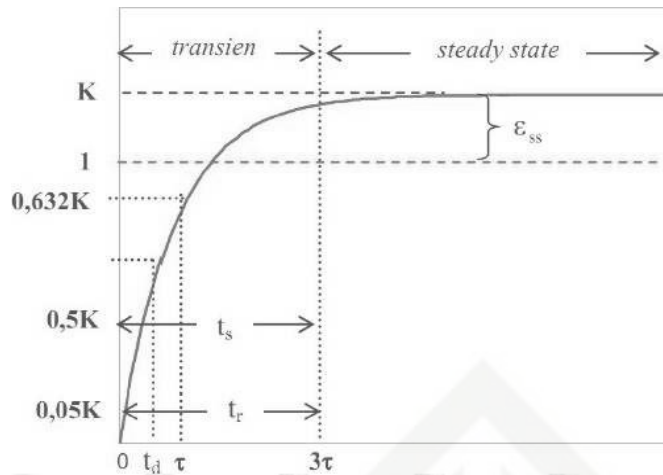
Hak Cipta Dilindungi Undang-Undang

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 2.4 Respon sistem orde satu [14]

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. 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:

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

$$t_s(\pm 4\%) \approx 2\tau$$

$$t_s(\pm 5\%) \approx 0,5\tau$$

2. Waktu naik atau *rise time* (t_r), adalah ukuran yang menyatakan bahwa respon sistem telah naik dari 5% ke 95% atau 10% ke 90% dari nilai respon pada keadaan tunak (*steady state*). Jika dihubungkan dengan konstanta waktu τ , maka t_r dapat diformulasikan sebagai berikut:

$$t_r(5\% - 95\%) \approx \tau \ln 19$$

$$t_r(10\% - 90\%) \approx \tau \ln 9$$

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:

$$t_d(\pm 3\%) \approx \tau \ln 12$$

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 kuadratic untuk menyelesaikan permasalahan regulator [15]. Disebut linier karena model dan bentuk kendalinya berupa sistem linear sedangkan disebut kuadrat karena memiliki *Cost Function* yaitu kuadrat dan karena referensi sistem bukan fungsi waktu maka disebut regulator [16]. Dari hasil linierisasi didapatkan suatu *plant* linear dalam bentuk :

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.11}$$

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 [16].

$$J(t_0) = \frac{1}{2} S(T)x^2(T) + \frac{1}{2} \int_{t_0}^T (Qx^2 + Ru^2) \quad (2.12)$$

Dengan syarat:

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

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

$$H(x, u, \lambda, t) = L(x, u, t) + \lambda^T f(x, u, t)$$

$$H(x, u, \lambda, t) = \frac{1}{2} (x^T Qx + u^T Ru) + \lambda^T (Ax + Bu) \quad (2.13)$$

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

1. Syarat perlu

a) Persamaan *State*

$$\dot{x} = \left(\frac{\partial x}{\partial x} \right)^T Qx + \left(\frac{\partial Qx}{\partial \lambda} \right)^T u = Ax + Bu \quad (2.14)$$

b) Persamaan *Costate*

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

$$-\dot{\lambda} = Qx + A^T \lambda \quad (2.15)$$

c) Kondisi stasioner

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

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

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

$$u = -R^{-1} B^T \lambda \quad (2.16)$$

Syarat batas

a. Batas awal

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

b. Batas akhir

$$(\phi_x + \psi_x^T v - \lambda)^T dx|_T + (\phi_t + \psi_t^T v - H) dt|_{t=T} = 0 \quad (2.17)$$

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

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

$$S(t) x(t) = \lambda(t) \quad (2.18)$$

Dari persamaan *Costate* dan (2.18), keduanya mengandung variabel λ . Bila keduanya disubstitusikan maka akan menghasilkan persamaan:

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

$$\dot{\lambda}(t) = \dot{S}(t) x(t) + S(t) \dot{x}(t)$$

$$\dot{\lambda}(t) = -Qx - A^T \lambda$$

$$S(t) x(t) + S(t) \dot{x}(t) = -Qx - A^T \lambda$$

$$\dot{S}(t) x(t) + S(t) \{Ax + Bu\} = -Qx - A^T \lambda$$

$$\dot{S}(t) x(t) + S(t) \{Ax + B(-R^{-1} B^T \lambda)\} = -Qx - A^T \lambda$$

$$-\dot{S}x = (A^T S + SA - SBR^{-1} B^T S + Q)x$$

$$-\dot{S} = A^T S + SA - SBR^{-1} B^T S + Q \quad (2.19)$$

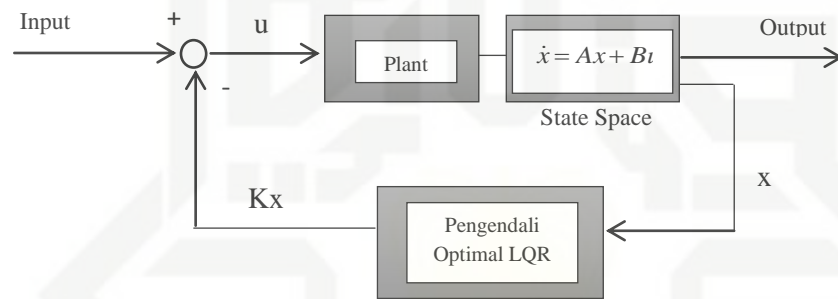
Persamaan (2.19) disebut persamaan differensial Riccati. Untuk $\dot{S} = 0$ persamaannya disebut *Algebraic Riccati Equation* (ARE). Dengan di dapatkannya matriks S dari persamaan ARE ini sehingga persamaan sinyal kendali pun dapat di hitung:

$$u = -R^{-1} B^T \lambda = -R^{-1} B^T Sx = -Kx \quad (2.20)$$

Maka konstanta umpan balik keadaan K dapat dicari sebelum menghasilkan sinyal kendali pada persamaan (2.21) [17].

$$K = R^{-1} B^T S \quad (2.21)$$

Dimana nilai S adalah unik, solusi semi definit positif untuk persamaan ARE harus memenuhi syarat persamaan (2.19). Selanjutnya dalam perancangan teknik kendali optimal LQR, Setelah nilai S diketahui maka, nilai S tersebut disubstitusikan kedalam persamaan (2.20) sehingga didapatkan nilai matriks optimal K , dengan K adalah *Gain State Feedback*, sehingga sistem kendali optimal LQR hasil desain dapat di perhatikan pada gambar 2.7 berikut:



Gambar 2.5 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 [18] [19]. 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 :

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de}{dt} \quad (2.22)$$

Persamaan 2.0- diubah ke dalam bentuk Laplace menjadi

$$\frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s \quad (2.23)$$

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.

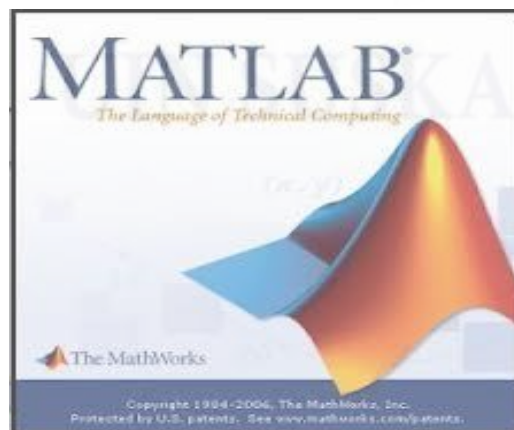
Hubungan dari ketiga aksi pengendali tersebut ditunjukkan pada Tabel berikut :

Tabel 2.1 Hubungan antara kendali P, I, dan D [11]

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



Gambar 2.6 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. 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.

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*. Inimemungkinkan bagi kita untuk melakukan kedua hal baik “pemrograman dalam lingkupsederhana ” untuk mendapatkan hasil yang cepat, dan “pemrograman dalam lingkup yang lebihbesar” untuk memperoleh hasil-hasil dan aplikasi yang kompleks.

4. Graphics

MATLAB memiliki fasilitas untuk menampilkan vector dan matrices sebagai suatugrafik. 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 diriuntuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan *graphicaluser 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 untukpemanggilan *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.