

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penerapan PID *controller* pada sistem temperatur pada proses pembuatan kaca ini dijelaskan tentang pengendali sebuah mesin *lehr* pada *Annealing* untuk menjaga suhu temperatur dan sekaligus menurunkan temperatur kaca. Sebelum kaca masuk kedalam *lehr* kaca akan memasuki proses *melting* atau disebut juga peleburan pada suhu yang sangat tinggi yaitu 1600-1700°C, selanjutnya akan memasuki *glass forming* (pembentukan kaca) diproses dalam ruangan yang disebut dengan *metal bath*. Suhu kaca yang sangat panas akan memasuki *annealing room* (pendinginan) di *lehr* untuk proses pendinginan secara bertahap. Sistem kendali yang digunakan adalah PID dengan pendekatan menggunakan metode Ziegler-Nichols, data yang didapat dari metode tersebut adalah $T=330$ s, $L=12.2$ s untuk nilai PID adalah $K_1=32.45$, $T_i=24.4$ dan $T_d=6.1$, hasil respon yang diperoleh waktu naik 5 detik, waktu tunak pada detik ke 40 detik dan terdapat *overshoot* sebesar $\pm 12\%$. Sistem telah sampai pada keadaan tunak meskipun respon system lambat[3].

Penelitian ini menggunakan LQR sebagai controller pada sistem temperatur di *annealing lehr* untuk proses pembuatan lembaran kaca. Pada penelitian ini sistem yang dikendalikan hanya temperatur heater pada proses *annealing*. Perancangan pengendali optimal LQR mampu menghasilkan performansi yang optimal dimana pengendalian ini berkerja dengan cara pembobotan matrik Q dan R yang optimal yang dibuat kecil supaya dapat mengikuti setpoint yang diberikan dan hasil yang didapatkan dianalisa berdasarkan 2 hal yaitu indentifikasi respon sistem dan IAE, respon sistem sudah dianggap baik karna memiliki waktu yang cukup cepat dalam mencapai setpoint yaitu pada detik ke 6,4182 dan IAE juga sudah optimal dengan metode heuristik menghasilkan nilai yang terkecil yaitu sebesar 811.5. Respon pengendalian temperatur yang dihasilkan LQR mempunyai *settling time* sebesar 2,8s, *rise time* 6,4s, *delay time* 1,6s, dan *error* sebesar 0,5% [4].

Pada penelitian ini membahas mengenai *fuzzy PID based temperature control of furnance for glass tempering process* menyatakan pengujian sistem menggunakan PID dengan metode ZN *tuning*, Cohen-coon *tuning*, ITAE *tuning load poin*, ITEA *tuning set poin* dan matlab/simuling PID *tuning* memperoleh hasil yang berbeda-beda, setiap metode *tuning* menghasilkan respon sistem yang kurang baik dengan *rase time* yang lama,

memiliki osilasi dan *overshoot* yang besar, dari kelima metode ITAE menghasilkan respon yang baik ketimbang yang lain. Untuk mengatasi kekurangan yang di dapat pada *controller* PID penulis melakukan *hybrid* dengan *Fuzzy*, setelah *hybrid* dilakukan memperoleh respon waktu yang lebih cepat dan dapat meminimalisir *overshoot* dan *osilasi*. Dinyatakan dengan, *controller fuzzy* PID memperoleh *rising time* 45 detik, *setting time* 180 detik dan *overshoot* sebesar 2.129%, dan PID saja memperoleh waktu *rising time* 44 detik, *setting time* 290 detik dan *overshoot* sebesar 16.13% [5]

Penelitian ini menggunakan metode LQR-PID pada pengaturan kecepatan simulator Parallel Hybrid Electric Vehicle. Metode PID-*Linear Quadratic Regulator* (PID-LQR) digunakan untuk mengatur kecepatan motor listrik, sehingga dapat membantu ICE saat terjadi pembebanan lebih yang dapat menurunkan kecepatan pada kendaraan. Motor DC dengan kontroler PID-LQR dapat membantu kinerja ICE pada permasalahan regulator dengan nilai parameter $K_p = 3,163$, $K_i = 7,0711$, dan $K_d = 0,0173$. Hasil pengujian secara simulasi menggunakan kontroler PID-LQR, motor DC mampu membantu kinerja ICE pada saat terdapat beban sehingga kecepatan yang diinginkan dapat dipertahankan dengan menghilangkan kesalahan hingga 99,8 %. Pada hasil implementasi, motor DC juga mampu membantu ICE pada saat terjadi pembebanan sehingga kecepatan putar HEV mendekati *setpoint*, meskipun masih terdapat *error steady state* hingga 28,8 % [6].

Pada penelitian ini menggunakan LQR-PID untuk mengendalikan level pada modul training PCT-100. PCT-100 merupakan modul *training* kendali proses yang dilengkapi dengan sensor dan aktuator untuk aplikasi dari suatu proses. Parameter yang dapat diuji secara *loop* terbuka pada sistem PCT-100 antara lain *level*, *temperature*, *flow* dan *pressure*. Perancangan pengendali *hybrid Linear Quadratic Regulator* (LQR) – *Proportional Integral Derivatif* (PID) untuk pengendalian *Level* pada sistem PCT – 100 telah berhasil dirancang. Berdasarkan hasil dan analisa, maka didapatkan bahwa dari kombinasi pengendali LQR – PID mampu mencapai *setpoint* dengan respon waktu yang cepat pada PCT – 100. Hal ini dibuktikan berdasarkan hasil visual dan identifikasi sistem secara analisa diperoleh nilai *Rise Time* (tr) dari 1240.97 menjadi sebesar 2.0367 detik pada saat *setpoint* 50 [7].

Berdasarkan penelitian terkait dan pengujian sistem diatas dengan menghybrid LQR-PID ini dapat menghasilkan respon waktu yang cepat, menghilangkan error dan mempercepat kestabilan suatu sistem yang secara optimal

2.2 Dasar Teori

2.2.1 Annealing

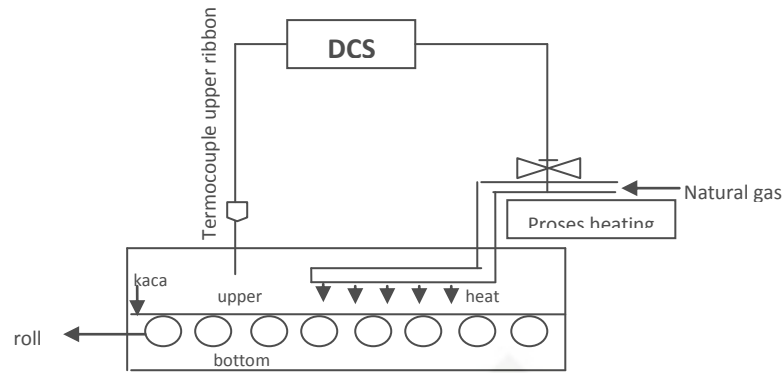
Annealing adalah suatu pendingin kaca secara lambat setelah kaca tersebut dipanaskan pada temperatur yang sangat tinggi. Proses pendingin kaca yang dipanaskan pada temperatur yang tinggi tersebut berlangsung secara perlahan-lahan. Ketika penurunan temperatur tersebut berhenti, kaca telah berada pada kondisi dengan energi yang sangat rendah[5]. *Annealing* adalah suatu metode optimasi yang berdasarkan pada proses pendinginan. Secara umum, pada saat suatu zat melewati proses pendinginan, pertamanya akan dipanaskan dulu sampai mencapai titik lebur untuk pencairannya, kemudian didinginkan secara perlahan-lahan dengan cara mengontrol proses pendinginannya hingga zat tersebut padat kembali. Sifat-sifat terakhir dari zat ini sangat bergantung pada jadwal pendinginan yang diterapkan, jika suhu pendinginan diturunkan secara cepat maka zat yang dihasilkan akan mudah rusak karena struktur yang tidak sempurna, sebaliknya jika suhu pendinginan diturunkan secara perlahan maka struktur yang dihasilkan tersusun dengan baik dan kuat.

Dalam proses *annealing* ada beberapa tahap yang terjadi dalam pendingin kaca.

1. Pada temperatur tinggi, atom-atom yang berada dalam kaca yang dipanaskan tersebut berorientasi acak dan mempunyai energi yang sangat tinggi.
2. Ketika temperatur diturunkan, atom-atom akan cenderung akan mengikuti arah atom tetangganya. Namun daerah yang berbeda akan memiliki arah yang berbeda pula.
3. Jika temperatur sudah sangat rendah, maka terjadi proses pembekuan sehingga atom-atom tersebut akan berkaitan erat, memiliki energi yang sangat rendah dan memiliki arah yang satu dengan yang lain.

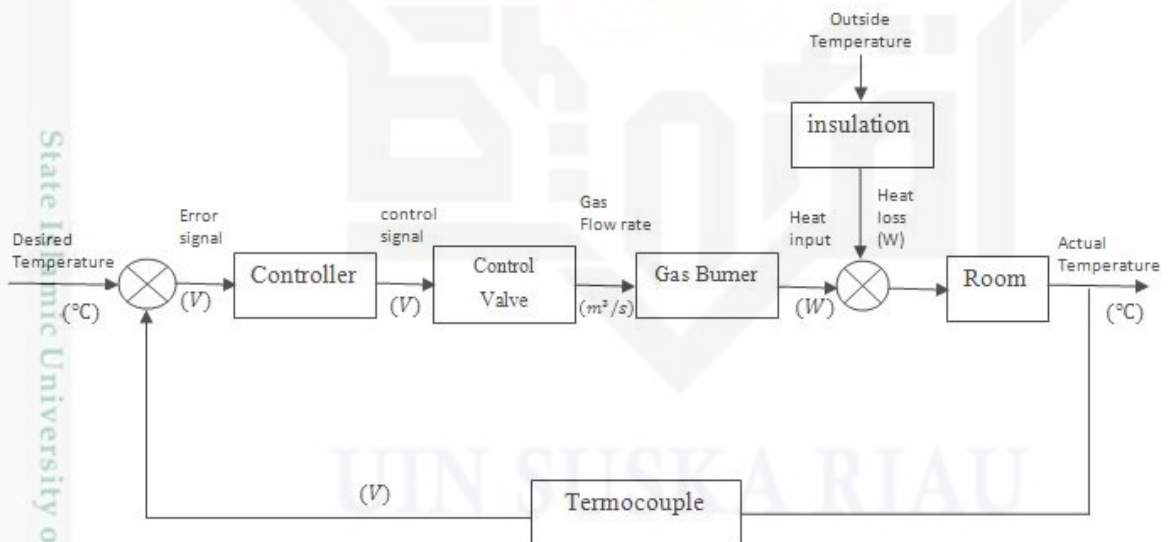
2.2.2 Pemodelan Annealing

Proses kerja pada annealing lehr dapat dilihat pada gambar berikut:



Gambar 2.1 Gambar sistem kerja Annealing Lehr[3].

Pada proses *Annealing* terdapat 2 macam pengendalian temperatur yaitu, proses *heating* dan *cooling*. *Annealing heater* adalah penurunan temperatur dengan suhu tertentu untuk memasuki pada tahap selanjutnya, sedangkan *annealing cooling* penurunan temperatur setelah pada tahapan sebelumnya hingga sama dengan suhu lingkungan[3], namun pada penelitian ini proses yang diteliti hanya pada proses *heating* saja. Dari sistem kerja yang terlihat pada gambar 2.1 menunjukkan bagaimana menjaga temperatur pada ruangan *lehr*, jika digambarkan dalam bentuk diagram blok adalah sebagai berikut:



Gambar 2.2 Diagram blok lehr pada proses heating [3].

Dimana variabel sistem blok tersebut adalah:

$$\Theta_d(t) = \text{Desired Temperature } (^\circ\text{C}).$$

$$\Theta_m(t) = \text{Maerused Temperature } (^\circ\text{C}).$$

$$\Theta_o(t) = \text{Actual Temperature } (^\circ\text{C}).$$

- $\Theta_s(t)$ = Outside Temperature ($^{\circ}\text{C}$).
- $u(t)$ = Control Signal (V).
- $v(t)$ = Gas Flow Rate (m^3/s).
- $Q_i(t)$ = Heat Flow into Room (W).
- $Q_0(t)$ = Heat Flow though wall (W).

Untuk menentukan dan mengetahui parameter serta kinerja pada *annealing lehr* perlu dilakukan pemodelan matematis sistem dari plant proses. Dari pemodelan tersebut akan di simulasi menggunakan MATLAB 2014. Dari data diagram blok diatas dapat dibuat matematis sistem dari *plant annealing lehr* tersebut. Dari gambar 2.2 setiap blok–blok diagram tersebut akan dimodelkan dalam persamaan matematis, pada *open loop* sistem kontroler diabaikan karna dianggap belum ada, blok pertama yang akan dimodelkan adalah *control valve*, *gas burner* dan *room lehr*.

Pemodelan *Control valve*, dalam pengendalian temperature dalam proses *annealing lehr* maka dibutuhkan *control valve* yang akan mengatur laju aliran gas menuju *room lehr*. Model matematis *control valve* adalah sebagai berikut:

$$\frac{KT}{1+T1s} \tag{2.1}$$

Dengan KT = kontrol gain *control valve*.
 $T1s$ = *Time constan control valve*.

Untuk mendapatkan gain *control valve*, dapat diperoleh dengan menggunakan persamaan:

$$K = \frac{\text{Laju aliran maksimum}}{\text{Perubahan tekanan masukan}}$$

Gain tranduser (I/P) diperoleh dengan:

$$GT = \frac{\text{Span output (Kg/m}^2\text{)}}{\text{Span input (mA)}}$$

Maka gain total dari *control valve* diperoleh dengan persamaan 2.2

$$KT = KCV \times GT \tag{2.2}$$

Konstanta waktu dari *control valve* diperoleh berdasarkan data pada penelitian sebelumnya dengan nilai $T1s = 12.24s$

Blok diagram Selanjutnya adalah *gas burner* dimana disimbolkan dengan K atau konstanta gas burner, gas burner merupakan elemen yang akan mengatur turun dan naiknya temperatur dalam ruangan lehr jika temperatur terlalu tinggi maka aliran gas akan dikurangi dan jika temperatur turun dari nilai yang diinginkan maka aliran gas burner akan

ditambah. Nilai konstanta gas burner adalah $K1 = 40000 \text{ kcal/hr}$ atau 11.1111 kcal/s , 46488 W .

Blok diagram yang terakhir adalah *room lehr* diasumsikan bahwa *room lehr* diisolasi untuk mencegah kehilangan panas ke udara sekitar. Diasumsikan pula tidak terdapat panas masuk dan kaca pada *room lehr* pada temperatur yang seragam.

Anggap bahwa temperatur kaca pada aliran masuk dijaga tetap dan laju aliran panas masuk sistem (panas dari gas burner) tiba-tiba berubah dari H menjadi $H + h_i$. dimana h_i menyatakan perubahan kecil laju panas masuk. Laju aliran panas keluar kemudain diubah dari H menjadi $H + h_o$. Temperatur kaca pada aliran keluar juga akan berubah dari Θ menjadi $\Theta + \theta$. Dalam hal ini besarnya perubahan kecil laju panas keluar h_o , kapasitas termal C dan tahanan thermal R berturut-turut adalah:

$$h_o = Gc\theta$$

$$C = Mc$$

$$R = \frac{\theta}{h_o} = \frac{1}{Gc}$$

Persamaan differensial untuk sistem ini adalah :

$$C \frac{d\theta}{dt} = h_i - h_o$$

dengan mensubsitusikan persamaan h_o , C , R ke dalam Persamaan differensial sistem diperoleh sebagai berikut :

$$RC \frac{d\theta}{dt} + \theta = Rh_i$$

dengan menggunakan transformasi laplace didapatkan sebagai berikut :

$$RCs\Theta(s) + \Theta(s) = RH_i(s)$$

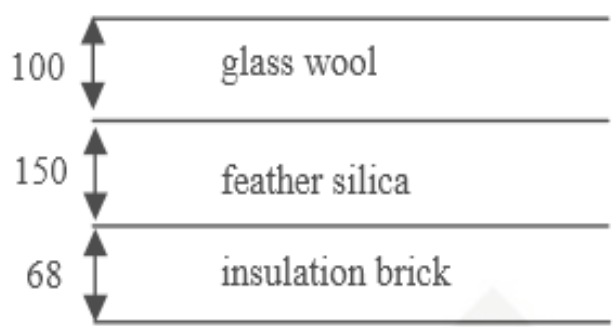
Sehingga fungsi alih yang menghubungkan $\Theta(s)$ dan $H_i(s)$ adalah :

$$\frac{\theta(s)}{H(s)} = \frac{R}{RCs + 1} \tag{2.3}$$

Untuk nilai parameter CT yang berasal dari *element room*. CT merupakan kapasitas termal udara dalam ruangan. resistansi dinding pada Ruangan lehr disimbolkan dengan RT yang akan di peroleh dari dinamika termal ruangan. Dimana dinding tersebut terbuat dari *Glass Wool*, *Feather Silica*, *Insulation brick* seperti ditunjukkan pada gambar 2.3. Resistansi dinding (RT) diperoleh dari hasil bagi antara ketebal dinding (d) dan konduktivitas termal (k).

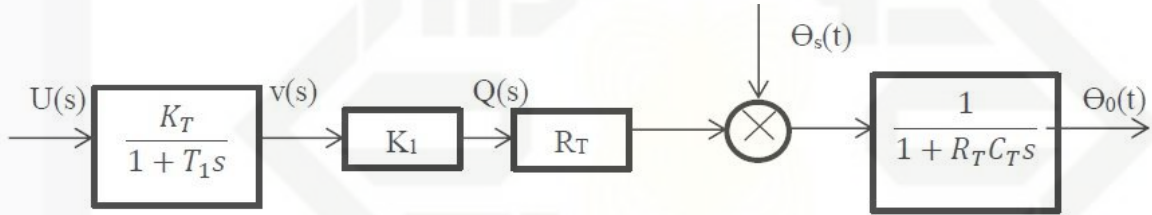
$$RT = \frac{d_1}{k_1} + \frac{d_2}{k_2} + \frac{d_3}{k_3} \tag{2.4}$$

Untuk ketebalan dinding pada *lehr* dapat dilihat pada gambar dibawah berikut :



Gambar 2.3 insulation pada dinding *Lehr*[8].

Setelah mendapatkan model matematis setiap variabel blok dan dibuat blok diagram setelah dimodelkan akan memperoleh fungsi alih *open loop* sistem dengan asumsi temperatur dari lingkungan $\Theta_s(t) = 0$ (tidak berpengaruh/insulation baik) sebagai berikut:



Gambar 2.4 model matematis blok diagram lehr.

Sehingga menghasilkann fungsi *Transfer Function* sebagai berikut:

$$\frac{\theta_0(t)}{U(s)} = \frac{K_T K_1 R_T}{(1 + T_1 s)(1 + R_T C_T s)} \tag{2.5}$$

Untuk memperoleh nilai K_t dan R_t dilihat pada persamaan 2.2 dan 2.4 yang nilai parameternya diperoleh di tabel 2.1 parameter annealing kemudian semua nilai tersebut dimasukan kedalam persamaan 2.5 hingga memperoleh fungsi alih *Annealing Lehr* sebagai berikut:

$$K_T = K_{cv}GT = (3.3333 \times 10^{-9} \text{ m}^5/\text{kgs}) (6250000 \text{ kg}/(\text{m}^2\text{A})) = 0.028 \text{ m}^3\text{kg}/\text{sA}.$$

$$R_T = \frac{d_1}{K_1} + \frac{d_2}{K_2} + \frac{d_3}{K_3} = \frac{100 \text{ m}}{0.11619 \text{ W}/\text{m}^\circ\text{C}} + \frac{150}{0.08135 \text{ W}/\text{m}^\circ\text{C}} + \frac{68}{0.16270 \text{ W}/\text{m}^\circ\text{C}}$$

$$= 3.1224 \text{ m}/\text{W}^\circ\text{C}$$

$$\frac{\theta(s)}{U(s)} = \frac{0.028 \text{ m}^3\text{kg}/\text{sA} \times 46488.8 \text{ W} \times 3.1224 \text{ m}/\text{W}^\circ\text{C}}{(1 + 12.245 \text{ s})(1 + 3.1224 \text{ mW}^\circ\text{C} \times 80 \text{ J}/\text{K})} \tag{2.6}$$

$$\frac{\theta(s)}{U(s)} = \frac{3019.257}{3057.245 \text{ s}^2 + 262.032 \text{ s} + 1} \tag{2.7}$$

Tabel 2.1 Parameter pada *annealing lehr* pada proses *heating* [3].

No	Parameter <i>Annealing lehr</i> pada proses <i>heating</i>	Nilai
1	Gain kontrol valve (K_{CV})	$3.3333 \times 10^{-9} m^5 / (kgs)$
2	Gain Transuder (G_T)	$6250000 \text{ kg} / (m^2 A)$
3	Konstanta waktu kontrol valve (T_1)	12.24 s
4	Ketebalan dinding <i>Glass wool</i> (d_1)	100 m
5	Ketebalan dinding <i>Feather silica</i> (d_2)	150 m
6	Ketebalan dinding <i>Insulation brick</i> (d_3)	68 m
7	Konduktifitas termal <i>Glass wool</i> (k_1)	$0.11619 \text{ W} / m^{\circ}C$
8	Konduktifitas termal <i>Feather silica</i> (k_2)	$0.08135 \text{ W} / m^{\circ}C$
9	Konduktifitas termal <i>Insulation brick</i> (k_3)	$0.16270 \text{ W} / m^{\circ}C$
10	Konstanta gas burner (K_1)	46488 W
11	Kapasitansi termal udara (C_T)	80s J/K

2.3 Identifikasi Sistem

Identifikasi sistem pada penelitian ini menggunakan metode pengamatan respon waktu.

2.3.1 Identifikasi Respon Waktu

Dalam beberapa kasus praktek, karakteristik kerja yang diinginkan dari suatu sistem kendali dicirikan oleh suku besaran daerah asal waktu. Sistem dengan tenaga tidak dapat memberikan respon seketika dan akan menunjukkan tanggapan transien walaupun diberi masukan atau gangguan. Biasanya, karakteristik kinerja suatu sistem kendali dicirikan oleh suku respon transien terhadap masukan tangga satuan karena hal itu mudah ditimbulkan dan cukup drastis. Jika respon terhadap masukan tangga diketahui, secara matematis dapat dihitung respon untuk sembarang masukan [9].

Respon transien suatu sistem terhadap masukan tangga satuan tergantung transien terhadap variasi sistem, terdapat cara praktis yang biasa digunakan. Yaitu dengan syarat

awal standar bahwa sistem mula-mula diam dengan keluaran dan semua turunannya nol. Dengan demikian, karakteristik tanggapannya dapat dengan mudah dibandingkan [9].

Respon transien suatu sistem kendali secara praktek selalu menunjukkan osilasi teredam sebelum mencapai keadaan tunaknya. Dalam menggolongkan karakteristik tanggapan transien suatu sistem kendali terhadap masukan tangga satuan secara umum dikelompokkan sebagai berikut [9] :

1. Kontanta waktu (τ)

Waktu yang dibutuhkan respon mulai dari $t = 0$ sampai dengan respon mencapai 63.2 % dari respon *steady state*.

2. Waktu tunda (t_d)

Waktu tunda adalah waktu yang diperlukan oleh tanggapan untuk mencapai setengah nilai akhir untuk waktu yang pertama.

$$t_d(3\%) \approx \tau \ln 12 \tag{2.8}$$

3. Waktu naik (t_r)

Waktu naik adalah waktu yang diperlukan oleh tanggapan untuk naik dari 10% menjadi 90%, 5% menjadi 95%, atau 0 menjadi 100% dari nilai akhir yang biasa digunakan.

$$t_r(5\% - 95\%) \approx \tau \ln 19 \tag{2.9}$$

$$t_r(10\% - 90\%) \approx \tau \ln 9 \tag{2.10}$$

4. Waktu puncak (t_p)

Waktu puncak adalah waktu yang diperlukan tanggapan untuk mencapai puncak pertama *overshoot*.

5. *Overshoot* maksimum (M_p)

Overshoot maksimum adalah nilai puncak kurva tanggapan diukur dari satuan. Apabila nilai akhir keadaan tunak tanggapannya jauh dari satu, maka biasanya digunakan persen *overshoot* maksimum dan didefinisikan oleh.

$$M_p = \frac{C(t_p) - C(\infty)}{C(\infty)} 100\% \tag{2.11}$$

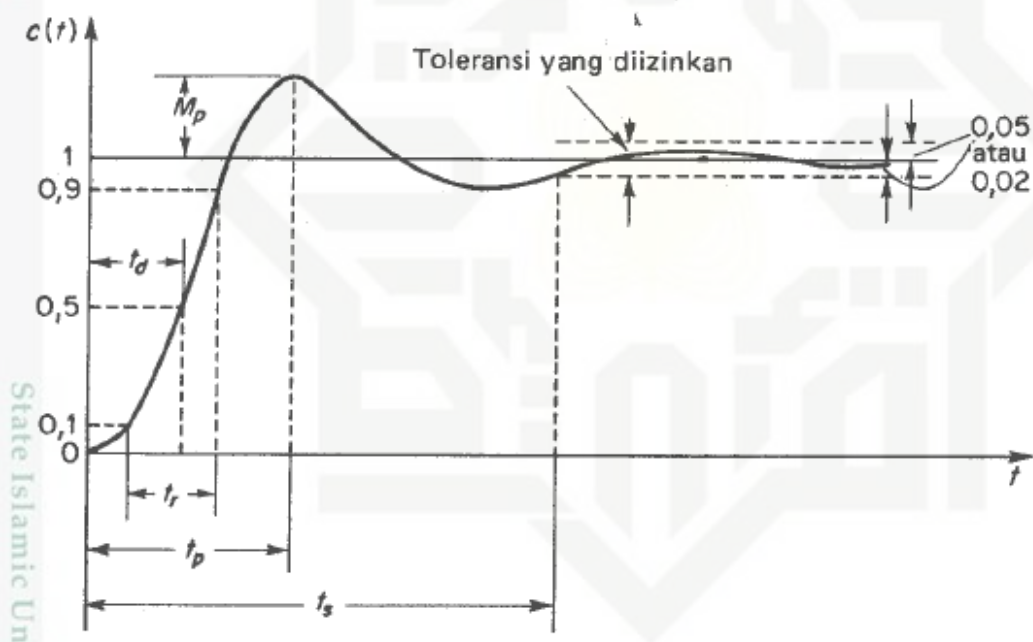
6. Waktu tunak (t_s)

Waktu tunak adalah waktu yang diperlukan untuk menanggapi kurva agar dapat mencapai dan tetap berada dalam gugus nilai akhir ukuran yang disederhanakan dengan presentase mutlak harga akhirnya (biasanya 2% atau 5%) waktu tunak tadi dihubungkan dengan sebesar sistem kendali. Apabila kita menemukan kriteria kesalahan presentase kesalahan presentase untuk sistem, kita boleh menetapkannya dari tujuan desain sistem dalam pertanyaan.

$$t_s(\pm 3\%) \approx 5\tau \tag{2.12}$$

$$t_s(\pm 4\%) \approx 2\tau \tag{2.13}$$

$$t_s(\pm 5\%) \approx 0.5\tau \tag{2.14}$$



Gambar 2.5 Respon waktu orde 2 [9].

2.3.2 Kriteria Integral Menggunakan IAE (*Integral of Absolute Error*)

Kriteria integrasi membutuhkan data tanggapan mulai dari hingga mencapai keadaan tunak. Dengan demikian kriteria ini didasarkan pada seluruh tanggapan dari proses yang bersangkutan. Kriteria integral yang digunakan pada penelitian ini adalah kriteria IAE (*Integral of Absolute Error*) yang bertujuan untuk mendapatkan nilai IAE sekecil mungkin dengan pemilihan kriteria tergantung pada karakteristik sistem proses dan beberapa syarat tambahan yang diperoleh dari tanggapan loop tertutup. Kriteria IAE lebih dipopuler

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.

dikalangan praktisi industri sebab mudah dalam pemakaiannya sertasangat cocok untuk menekan error yang kecil.

$$IAE = \int_0^{\infty} |e|dt \tag{2.15}$$

2.3.3 Sistem Kontrol Optimal

Dalam sistem kendali optimal, pada dasarnya, solusi kendali optimal dititik beratkan pada pencarian nilai sinyal kendali sehingga indeks performansi dapat dioptimasi. Nilai indeks performansi yang terbaik tidak akan diketahui sampai dengan proses pengendalian selesai. Hal ini tergantung pada kondisi plant yang dikendalikan dan spesifik indeks performansi yang dioptimalkan.

Perancangan sistem kendali optimal mempunyai 2 tujuan (fungsi), yaitu: sebagai regulator (menstabilkan sistem dengan variabel/*output* agar tetap kecil) dan sebagai *Tracker/Servomechanism* (mengontrol sistem agar mengikuti trayektori dan keadaan selalu dalam batas-batas tertentu) [10].

Sistem kendali yang baik adalah sistem kendali yang memiliki daya tanggap yang cepat dan stabil, tetapi tidak memerlukan energi yang berlebihan. Sistem yang demikian dapat dicapai melalui indeks performansi yang tepat. Berdasarkan optimasi indeks performansi inilah yang disebut sebagai sistem kendali optimal [11].

Pada suatu sistem indeks performansi dipilih sesuai dengan bagian yang akan dioptimalkan. Bentuk umum dari indeks adalah sebagai berikut :

$$J = \pi \int_{t_0}^T L(x, u, t)dt \tag{2.16}$$

J = indeks performansi

L(x,u,t) = fungsi dari x, u dan t

T = waktu

Salah satu metode yang biasa digunakan untuk meminimalkan indeks performansi yaitu persamaan *Aljabar Riccati*, yang digunakan untuk mengoptimalkan sistem proses berbentuk *linear*. Suatu sistem kendali akan optimal pada indeks unjuk kerja yang diberikan, tetapi tidak optimal lagi pada indeks unjuk kerja yang lain [11].

2.4 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 pada sistem linear dengan kriteria kuadratik untuk menyelesaikan permasalahan regulator. Disebut linier karena model dan bentuk kontrolernya berupa sistem linear sedangkan disebut kuadratik karena memiliki *Cost Function* yaitu kuadrat dan karena referensi sistem bukan fungsi waktu maka disebut regulator[12].

Diberikan suatu *plant linear* dalam persamaan *state space* sebagai berikut :

$$\dot{x} = A_x + B_u \quad t \geq t_0 \quad (2.17)$$

Dimana :

A : $n \times n$ matriks *State*

B : $n \times r$ matriks kontrol

x(t) : n th vektor *State*

u(t) : r th vector kontrol

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 [13].

$$J(t_0) = \frac{1}{2} x^2(T)S(T)x T + \frac{1}{2} \int_{t_0}^T (x^2 Qx + u^2 Ru) \quad (2.18)$$

Dengan syarat sebagai berikut :

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

Dimana :

t_0 = waktu awal t

τ = waktu akhir

x(τ) = matriks *State* akhir

Q = matriks semi definit positif

R = matriks definit positif

S = matriks semi definit positif

Sehingga diperoleh persamaan Hamilton

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

$$H(x, u, \lambda, t) = \frac{1}{2} (x^T Qx + u^T Ru) + \lambda^T (Ax + Bu) \quad (2.19)$$

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

1. Syarat perlu

a) Persamaan *State*

$$\dot{x} = \left(\frac{\partial x}{\partial x}\right)^T Qx + \left(\frac{\partial Qx}{\partial \lambda}\right)^T u = Ax + Bu \quad (2.20)$$

b) Persamaan *costate*

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

$$-\dot{\lambda} = Qx + A^T \lambda \quad (2.21)$$

c) Kondisi stasioner

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

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

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

$$u = -R^{-1} B^T \lambda \quad (2.22)$$

2. Syarat batas

a. Batas awal

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

b. Batas akhir

$$\left(\phi_x + \psi_x^T v - \lambda\right)^T dx|_{t=\tau} + (\phi_t + \psi_t^T v - H) dt|_{t=\tau} = 0 \quad (2.23)$$

Karena $\psi = 0$ dan

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

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

sehingga diperoleh persamaan

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

$$S(\tau) x(\tau) = \lambda(\tau) \quad (2.24)$$

Dari persamaan *costrate* dan (2.23), keduanya mengandung variable λ . Bila keduanya disubstitusikan maka akan menghasilkan persamaan:

$$\begin{aligned}
 S(t)x(t) &= \lambda(t) \\
 \dot{\lambda}(t) &= \dot{S}(t)x(t) + S(t)\dot{x}(t) \\
 \dot{\lambda}(t) &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\dot{x}(t) &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\{Ax + Bu\} &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\{Ax + B(-R^{-1}B^T \lambda)\} &= -Qx - A^T \lambda \\
 -\dot{S}x &= (A^T S + SA - SBR^{-1}B^T S + Q)x \\
 -\dot{S} &= A^T S + SA - SBR^{-1}B^T S + Q
 \end{aligned}
 \tag{2.25}$$

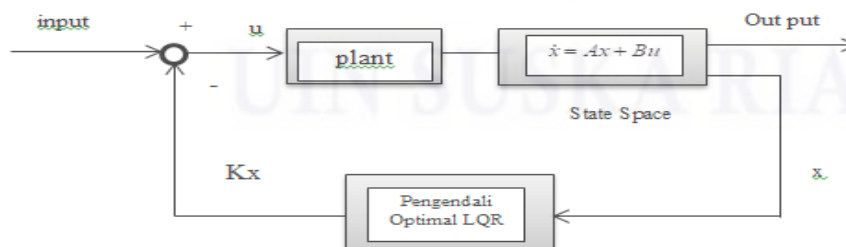
Persamaan (2.24) disebut persamaan differensial *Riccati*. Untuk $\dot{S} = 0$ persamaannya disebut *Algebraic Riccati Equation (ARE)* [12]. Dengan didapatkannya matriks S dari persamaan ARE ini sehingga persamaan sinyal kendali pun dapat di hitung:

$$u = -R^{-1}B^T \lambda = -R^{-1}B^T Sx = -Kx \tag{2.26}$$

Maka konstanta umpan balik keadaan K dapat dicari sebelum menghasilkan sinyal kendali pada persamaan (2.21) [19].

$$K = R^{-1}B^T S \tag{2.27}$$

Dimana nilai * S adalah unik, solusi semidefinit positif untuk persamaan ARE harus memenuhi syarat persamaan (2.24). Selanjutnya dalam perancangan teknik kendali optimal LQR, Setelah nilai S diketahui maka, nilai S tersebut disubstitusikan kedalam persamaan (2.26) sehingga didapatkan nilai matriks optimal K, dengan K adalah *Gain State Feedback*, sehingga sistem kendali optimal LQR hasil desain dapat di perlihatkan pada gambar 2.6.



Gambar 2.6 diagram alir kendali optimal LQR[5].

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

1. Ubah bentuk *Transfer Function Plant* nonlinier menjadi bentuk *State Spaced* dan dapatkan matriks A, B, C dan D dari *Plant* tersebut.
2. Tentukan matriks pembobot Q dan R.
3. Selesaikan persamaan *Ljabarriccati* hingga menghasilkan suatu matriks S yang definit positif.
4. Hitung optimal *Gain Feedback* K.

2.5 PID (*Proportional Integral Derivatif*)

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 [9]. Oleh karena itu, pengendali PID merupakan pengendali berumpan balik dengan baik pada beberapa aplikasi proses petrolem, proses polimer, dan lain sebagainya.

Adapun bentuk umum dari aksi kendali PID sebagai berikut :

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de}{dt} \quad (2.28)$$

Persamaan diubah ke dalam bentuk Laplace menjadi

$$\frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s \quad (2.29)$$

Dimana :

k_p = proportional 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 2.2.

2.5.1 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 mengkombinasikan antara parameter antara pengendali hingga didapatkan respon yang diinginkan.

Hubungan dari ketiga aksi pengendali tersebut ditunjukkan pada Tabel berikut :

Tabel 2.2 Hubungan antara kendali P, I, dan D [9].

Respon lup tertutup	Waktu naik	Overshoot	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

2.6 MATRIX LABORATORY (MATLAB)

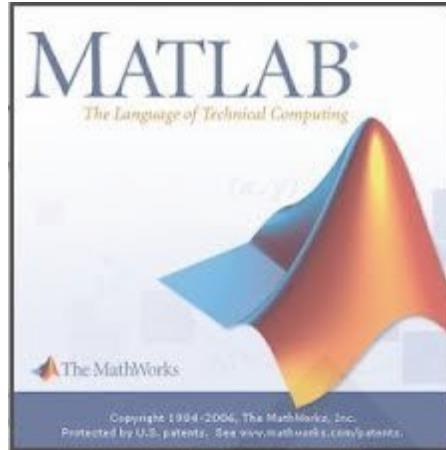
MATLAB adalah singkatan dari MATrix LABoratory. Pertama kali dibuat untuk mempermudah penggunaan dua koleksi subrutin pada pustaka FORTRAN yaitu: LINPACK dan EISPACK, dalam menangani komputasi matriks. Sejak itu, MATLAB berkembang menjadi sebuah sistem yang interaktif sekaligus sebagai bahasa pemrograman 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 2.7 Tampilan Matlab[4].

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 antara lain sebagai berikut:

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*. 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 dalambahasa C dan Fortran mampu berinterakasi 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 program pembanding 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.