

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Berbagai penelitian tentang motor induksi tiga fasa telah dilakukan, seperti pada penelitian tugas akhir **Rusdianto** yang berjudul perbandingan metode DTC dan PID pada pengaturan kecepatan motor induksi tiga fasa, mengatakan bahwa motor induksi tiga fasa memiliki ketelitian pengaturan kecepatan dan respon torsi yang lebih cepat dibandingkan motor DC sehingga penggunaan motor induksi tiga fasa mulai menggeser peran motor DC sebagai penggerak listrik, namun demikian peneliti juga mengatakan bahwa motor induksi tidak mampu mempertahankan kecepatannya ketika terjadi perubahan beban. Pada penelitian ini dapat ditarik kesimpulan bahwa perbandingan kedua pengendali tidak menunjukkan perbedaan yang signifikan dengan dengan nilai *rise time* 1.760 detik *settling time* (5τ) 2.989 detik *delay time* 0.4143 pada pengendali PID, *rise time* 1.728 detik *settling time* (5τ) 2.935 detik *delay time* 0.4068 detik pada pengendali DTC.

Pada penelitian lain yang dilakukan oleh **Ratna Ika. P** juga mengatakan bahwa motor induksi tiga fasa memiliki kelemahan yaitu tidak mampu mempertahankan kecepatannya dengan konstan bila diberi beban, oleh sebab itu dibutuhkan pengendali untuk mengatasi kelemahan motor induksi tiga fasa. Dalam penelitiannya ia menggunakan pengendali adaptif *fuzzy* dan dapat disimpulkan bahwa respon sistem dapat mencapai *set point* 1000 RPM dengan nilai parameter nilai *rise time* 2.295 detik *settling time* 3.78 detik *delay time* 2.16 detik.

Penelitian tentang pengendalian motor induksi dengan pengendali lain juga dilakukan oleh **Muhammad Ridho**, peneliti menggunakan *sliding mode control* untuk mengendalikan kecepatan motor induksi yang diberi beban, penelitian ini menyimpulkan bahwa pengendali *sliding mode* berhasil mempercepat waktu sistem menuju *set point* dengan *rise time* 7.45 detik dari waktu simulasi 70 detik, tetapi masih memiliki kekurangan yaitu adanya *overshoot* dan masih terjadinya *chattering* yang terjadi akibat proses pemeliharaan trayektori yang dipaksa mengikuti jalur *sliding surface*.

Selanjutnya penelitian yang dilakukan oleh **Karsino** yang meneliti tentang kecepatan motor induksi tiga fasa menggunakan pengendali PID. Dari penelitian ini didapatkan hasil yang cukup baik dengan nilai *rise time* 2,3 detik namun sama dengan pengendali dengan menggunakan metode *sliding mode* masih terdapat *overshoot* sebesar 12,75% walaupun tanpa adanya *chattering*.

penelitian **Linus A. Aloo, dkk** yang mendesain pengendali LQR dan pengendali PID untuk pengaturan kecepatan motor. Dari penelitian ini dapat disimpulkan bahwa penggunaan kendali PID memiliki *rise time* yang cepat yaitu 0.2 detik tetapi masih memiliki nilai *overshoot* yang besar yaitu 23.5% sedangkan dengan hanya menggunakan pengendali LQR sistem memiliki nilai *overshoot* yang lebih kecil yaitu hanya sebesar 4% tetapi dengan *rise time* lebih lama dibandingkan dengan menggunakan pengendali PID yaitu 0.9 detik. Penggabungan pengendali antara PID dan LQR menghasilkan *rise time* 0.2 detik dan *overshoot* 5%.

Penelitian yang dilakukan oleh **Endro Wahjono** yang berjudul pengaturan kecepatan motor induksi sebagai penggerak mobil listrik dengan pengendali *fuzzy logic* berbasis *direct torque control*. Hasil simulasi dengan *simulink* menunjukkan bahwa dengan kontroler memberikan respon yang baik dengan *setling time* 0.2 detik namun masih menunjukkan adanya *overshoot* sebesar 2,67% untuk kecepatan referensi motor 149,02 rad/detik dan torka beban 12,64 Nm.

Penelitian berjudul perbandingan sistem pengendalian motor induksi tiga fasa dengan metode *field oriented control* menggunakan *PI controller* dan *fuzzy logic controlleryang* dilakukan oleh **Gafur Nugroho, dkk** menyimpulkan bahwa *PI controller* menghasilkan *rise time*, *steady state* dan *over shoot* berturut – turut sebesar 0.0538s, 1.95s dan 13.448%.

Pada penelitian **Waluyo, dkk** berjudul Analisa Penalaan Kontrol PID pada Kecepatan Putaran Motor DC Berbeban Menggunakan Metode Heuristik dimana beliau mengatakan, selama ini kenyataannya kecepatan putar motor DC mengalami penurunan dari melonjaknya suatu kebutuhan yang mengakibatkan naiknya suatu beban sehingga menyebabkan putaran motor menjadi lambat dan kecepatannya menjadi tidak konstan. Untuk mengatasi hal itu maka diperlukan suatu perancangan sistem kendali kecepatan motor DC agar kecepatan motor DC tersebut berjalan stabil dan konstan sesuai dengan kecepatan yang diinginkan.

Unis Badri, dkk dengan penelitian yang berjudul Kontrol Optimal Pada Motor DC Menggunakan Metode *Linear Quadratic Regulator* (LQR). Pada penelitian ini mengatakan, desain kontrol yang baik harus memenuhi persyaratan-persyaratan tertentu yang telah ditetapkan. Persyaratan yang harus dipenuhi sistem kendali itu disebut sebagai indeks unjuk kerja (*performance index*). Indeks ini berkaitan dengan ketelitian, kestabilan dan kecepatan tanggapan sistem. Sistem kendali optimal jika harga-harga parameter dipilih sedemikian rupa akan menghasilkan indeks unjuk kerja maksimum atau minimum. Sistem kendali yang dirancang berdasarkan optimasi indeks unjuk kerja ini disebut sistem kendali optimal. Analisa hasil respon dengan menggunakan respon transient orde satu yang meliputi 4 hal yaitu: *Rise Time*, *Setling Time*, *Error State* dan *Overshoot*. Penulis juga menjelaskan sebelum suatu *Plant* dikendalikan perlu diketahui nilai *Transfer Function* nya terlebih dahulu agar dapat dirubah kedalam bentuk *State Space* sehingga bisa di terapkan kedalam perhitungan metode LQR.

Widhayaka Aji Chandra, dkk Pengaturan Kecepatan pada Simulator *Paraller Hybrid Elektrik Vehicle* (PHEV) Menggunakan *Linear Quadratic Regulator* (LQR) Berdasarkan *Particle Swarm Optimazation* (PSO) menjelaskan, untuk mendapatkan indeks performansi dan nilai kesalahan yang minimum dapat di peroleh dengan menentukan matriks pembobot Q dan R yang optimal dimana Q ini adalah matriks $n \times n$ simetri, semi-definit positif dan R adalah matriks $m \times m$ simetri, definit positif yang apabila nilai matriks tersebut dirubah-ubah, maka akan mendapatkan hasil yang maksimal. Sebagaimana yang dijelaskan oleh penulis yaitu penentuan matriks pembobot Q dan R yang berpedoman pada :

1. Semakin besar harga matriks Q, maka akan memperbesar harga elemen matriks gain kendali dan mempercepat sistem mencapai *Steady State*.
2. Semakin besar harga matriks R, maka akan memperkecil harga elemen matriks gain kendali dan memperlambat sistem mencapai *Steady State*.

Dari beberapa penelitian terkait mengenai pengendalian kecepatan motor induksi tiga fasa, kendali optimal *Linear Quadratic Regulator* (LQR) dan kendali PID yang telah di jelaskan sebelumnya. Maka digunakanlah kendali optimal LQR dengan penambahan PID agar menghasilkan respon sistem yang cepat dan stabil dalam mengendalikan sistem. Untuk itu penelitian dititikberatkan pada desain kendali optimal LQR – PID dalam mengendalikan kecepatan motor induksi tiga fasa.

2.2 Dasar Teori

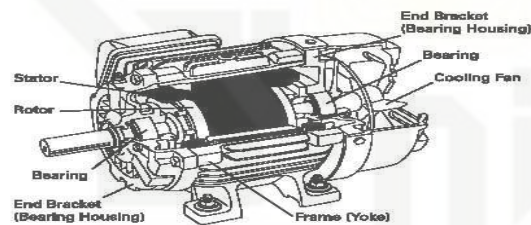
2.2.1 Motor Induksi Tiga Fasa

Motor induksi adalah suatu mesin listrik yang merubah energi listrik menjadi energi gerak dengan menggunakan gandingan medan listrik dan mempunyai slip antara medan stator dan medan rotor. Motor induksi merupakan motor yang paling banyak kita jumpai dalam industri (Fitzgerald, dkk, 1997).

Motor induksi tiga Fasa berputar pada kecepatan yang pada dasarnya adalah konstan, mulai dari tidak berbeban sampai mencapai keadaan beban penuh. Kecepatan putaran motor ini dipengaruhi oleh frekuensi, dengan demikian pengaturan kecepatan tidak dapat dengan mudah dilakukan terhadap motor ini. Walaupun demikian, motor induksi tiga fasa memiliki beberapa keuntungan, yaitu sederhana, konstruksinya kokoh, harganya relatif murah, mudah dalam melakukan perawatan, dan dapat diproduksi dengan karakteristik yang sesuai dengan kebutuhan industri (Fitzgerald, dkk, 1997).

2.2.2 Kontruksi Motor Induksi Tiga Fasa

Pada dasarnya motor induksi terdiri dari suatu bagian yang tidak berputar (stator) dan bagian yang bergerak memutar (rotor) seperti pada gambar 2.1.



Gambar 2.1 Bagian Motor Induksi Tiga Fasa
(Sumber: Fitzgerald, dkk, 1997).

2.2.3 Stator

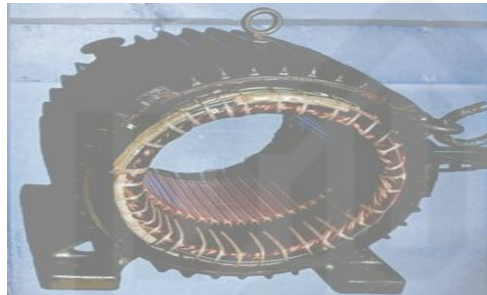
Stator merupakan bagian motor yang diam. Pada bagian stator terdapat beberapa slot yang merupakan tempat kawat (konduktor) dari tiga kumparan yang masing – masing berbeda fasa dan menerima arus dari tiap fasa tersebut yang disebut kumparan stator. Stator terdiri dari plat – plat besi yang disusun sama besar dengan rotor dan pada bagian dalam mempunyai banyak alur yang diberi kumparan kawat tembaga yang berisolasi. Jika kumparan stator mendapatkan suplai arus tiga fasa maka pada kumparan tersebut akan timbul fluks magnet putar. Karena adanya fluks magnet putar pada kumparan stator, mengakibatkan rotor berputar karena adanya induksi magnet dengan kecepatan putar

rotor sinkron dengan kecepatan putar stator (**Fitzgerald, dkk, 1997**). Kecepatan sinkron motor induksi tiga fasa diperoleh dari persamaan berikut :

$$N_s = \frac{120 f}{P} \quad (2.1)$$

Dimana :

- N_s : Kecepatan Sinkron (rpm)
- f : Besarnya Frekuensi (Hz)
- P : Jumlah Kutub



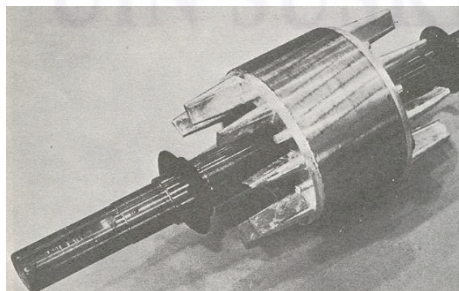
Gambar 2.2 Stator
(Sumber: Fitzgerald, dkk, 1997)

2.2.4 Rotor

Rotor merupakan bagian motor yang berputar. Berdasarkan jenis *rotor* nya, motor induksi tiga fasa dapat dibedakan menjadi dua jenis, yang juga akan menjadi penamaan untuk motor tersebut, yaitu *rotor* sangkar tupai (*squirrel cage rotor*) dan *rotor* belitan (*wound rotor*) (**Fitzgerald, dkk, 1997**).

1. Rotor Sangkar Tupai (*Squirrel Cage Rotor*)

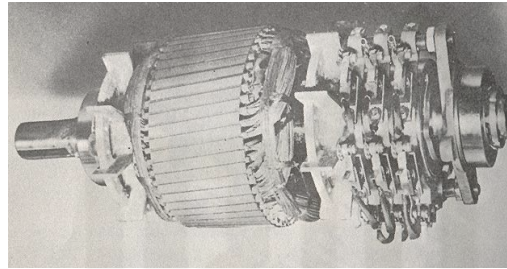
Rotor sangkar mempunyai kumparan yang terdiri atas beberapa batang konduktor yang disusun sedemikian rupa hingga menyerupai sangkar tupai. Rotor sangkar tupai konstruksinya sangat sederhana, yang mana rotor dari motor sangkar adalah konstruksi dari inti berlapis dengan konduktor dipasangkan paralel (**Bolton, 2006**).



Gambar 2.3 Rotor Sangkar Tupai
(Sumber: Fitzgerald, dkk, 1997)

2. Rotor Belitan (*Wound Rotor*)

Rotor belitan atau rotor cincin slip berbeda dengan rotor sangkar tupai dalam konstruksi rotornya. Bentuk rotor belitan dapat dilihat pada gambar 2.16. Seperti namanya rotor dililit dengan lilitan terisolasi serupa dengan lilitan stator. Lilitan fasa rotor dihubungkan secara Y dengan poros motor. Rotor belitan kurang banyak digunakan dibandingkan dengan rotor sangkar tupai karena harganya mahal dan biaya pemeliharaan lebih besar (Fitzgerald, dkk, 1997).



Gambar 2.4 Rotor Belitan
(Sumber: Fitzgerald, dkk, 1997)

2.2.5 Prinsip Kerja Motor Induksi Tiga Fasa

Pada dasarnya ada beberapa prinsip penting pada motor – motor induksi yaitu sebagai berikut :

1. Apabila sumber tegangan tiga fasa dipasang pada kumparan stator, timbullah medan putar dengan kecepatan.
2. Medan putar stator tersebut akan memotong batang konduktor pada rotor.
3. Akibatnya pada kumparan rotor timbul induksi (ggl). sebesar :

$$E = 4,44 fN\phi \quad (2.2)$$

Dimana :

- E = Tegangan Induksi ggl
- f = Frekuensi
- N = Banyak Lilitan
- ϕ = Fluks

4. Karena kumparan rotor merupakan rangkaian yang tertutup, ggl (E) akan menghasilkan arus (I).
5. Adanya arus didalam medan magnet menimbulkan gaya pada motor.

6. Bila kopel mula dihasilkan oleh gaya (F) pada rotor cukup besar untuk memikul kopel beban, rotor akan berputar searah dengan medan putar stator.
7. Tegangan induksi timbul karena terpotongnya batang konduktor (rotor) oleh medan putar stator. Artinya agar tegangan terinduksi diperlukan adanya perbedaan relatif antara kecepatan medan putar stator (n_s) dengan kecepatan berputar rotor (n_r).
8. Perbedaan kecepatan antara n_r dan n_s disebut slip (s) dinyatakan dengan :

$$S = \frac{n_s - n_r}{n_s} \times 100\% \quad (2.3)$$

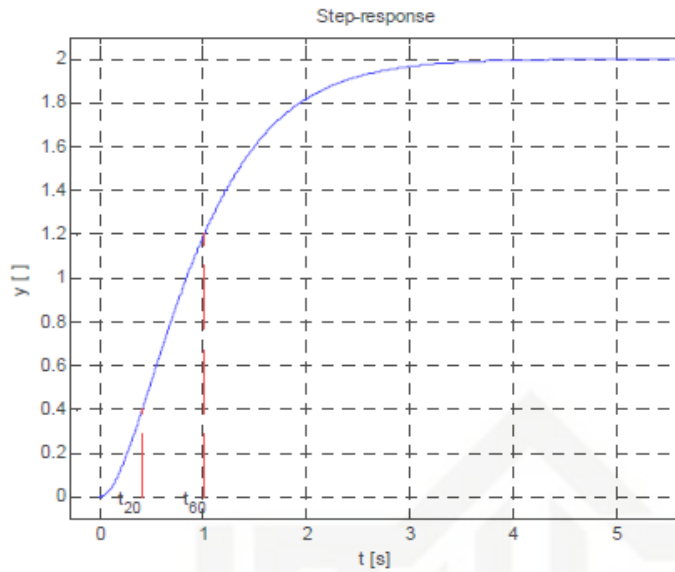
9. Bila $n_r = n_s$, tegangan tidak akan terinduksi dan arus tidak mengalir pada kumparan jangkar (rotor), dengan demikian tidak dihasilkan kopel. Kopel motor akan ditimbulkan apabila n_r lebih kecil dari n_s .
10. Dilihat dari cara kerjanya maka motor tak serempak disebut juga motor induksi atau motor asinkron.

2.2.6 Pendekatan Model Matematika Sistem

Sistem motor induksi tiga fasa terlebih dahulu diidentifikasi secara *open loop* dengan memberikan sinyal uji *step* dan melihat respon berupa kecepatan putar, kemudian hasil dari respon sistem yang terbaca oleh *rotary encoder* ditampilkan pada komputer melalui proses akuisisi data dengan PLC Mitsubishi (Afif Alhamdi, 2014). Setelah itu salah satu sampel data berupa grafik sebagai respon asli sistem motor induksi tiga fasa diterapkan pada model metode Smith dengan pendekatan matematika sistem berdasarkan paper penelitian internasional, *Experimental Identification of Stable Nonoscillatory Systems from Step-Responses by Selected Method* (Jakaubek, 2009).

2.2.6.1 Metode Smith

Pada metode *Smith*, persamaan matematika sistem didapat dari respon grafik sistem yaitu pada saat respon mencapai 20% dan 60%.



Gambar 2.5 Contoh *Step Respon* dari Sistem dengan Waktu t_{20} dan t_{60}

(Sumber: Jakaubek, 2009)

Persamaan *plant* sistem metode smith adalah sebagai berikut :

$$GSM(s) = \frac{K}{(\tau_{SM1}s+1)(\tau_{SM2}s+1)} \quad (2.4)$$

Untuk mendapatkan nilai dari persamaan sistem diatas, terlebih dahulu menyelesaikan persamaan dibawah ini :

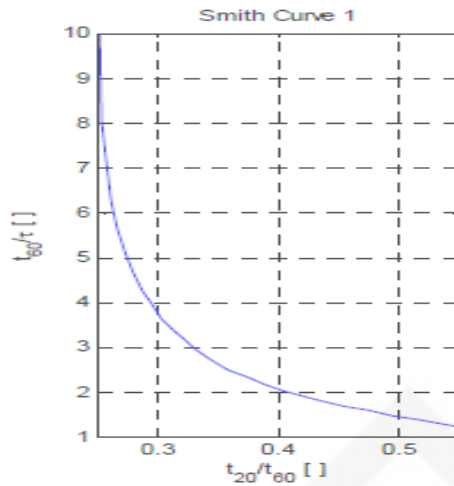
$$K = \frac{Y_{ss}}{X_{ss}} \quad (2.5)$$

Y_{ss} adalah keluaran respon sistem *plant* dalam keadaan *steady state*, sedangkan X_{ss} adalah *setpoint*.

$$\tau_{SM1} = \tau\zeta + \tau\sqrt{(\zeta^2 - 1)} \quad (2.6)$$

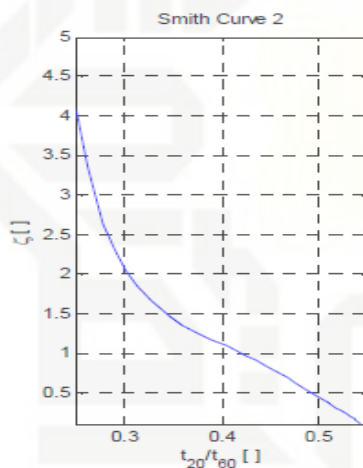
$$\tau_{SM2} = \tau\zeta - \tau\sqrt{(\zeta^2 - 1)} \quad (2.7)$$

Untuk menentukan nilai τ_{SM1} dan τ_{SM2} , terlebih dahulu tentukan nilai ζ yang didapat dari rasio t_{20} / t_{60} pada kurva *smith* dibawah ini :



Gambar 2.6 Kurva *Smith 1*
(Sumber: Jakaubek, 2009).

Dan rasio pada $t_{20}/60$ dapat digunakan untuk mencari nilai t_{60}/τ , sehingga didapat nilai τ pada kurva *smith* berikut :



Gambar 2.7 Kurva *Smith 2*
(Sumber: Jakaubek, 2009).

Setelah mendapatkan nilai ζ dan τ berdasarkan kurva *smith*. Maka dapat diselesaikan persamaan τ_{SM1} dan τ_{SM2} diatas. Dan akan menjadi nilai pada persamaan $GSM(s)$.

2.3 Pengendali Sistem

2.3.1 Sistem Kendali Optimal

Istilah optimal mengandung maksud hasil yang baik yang dapat dicapai dengan memperhatikan kondisi dan kendala dari suatu sistem (**Badri, 2008**). Dalam sistem kendali optimal, istilah optimal sering kali merujuk pada minimal, misalnya meminimalkan bahan bakar, waktu dan kesalahan. Pada dasarnya, solusi kendali optimal dititik beratkan pada pencarian nilai sinyal kendali $u(t)$ sehingga secara bersamaan nilai

indeks performansi dapat dioptimasi. Nilai indeks performansi yang terbaik tidak akan di ketahui sampai dengan proses pengendalian selesai. Hal ini tergantung pada kondisi plant yang dikendalikan dan spesifik indeks performansi yang dioptimalkan.

Secara umum 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) (**Ogata, 2002**). Melihat hal tersebut kendali optimal menggunakan metode *Linear Quadratic Regulator* (LQR) dapat diterapkan pada sistem motor DC yang berbeban untuk menyelesaikan permasalahan regulator pada kecepatan putaran motor yang tidak stabil dan mendapatkan nilai optimal indeks performansi yang terbaik.

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 disebut sebagai sistem kendali optimal (**Badri, 2008**).

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

$$J = \int_{t_0}^T L(x, u, t) dt \quad (2.8)$$

Dimana;

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 yang berbentuk *linear*. Suatu sistem kendali akan optimal pada indeks unjuk kerja yang diberikan, tetapi tidak optimal lagi pada indeks unjuk kerja yang lain (**Badri, 2008**).

2.3.2 *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 (**Chandra, 2014**). *Linear Quadratic Regulator* (LQR) merupakan salah satu metode kendali optimal pada sistem linear dengan kriteria kuadratic untuk menyelesaikan permasalahan regulator (**Chandra, 2014**). 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 (**Firmansyah, dkk, 2012**). Dari hasil linierisasi didapatkan suatu *Plant* linear dalam bentuk :

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.9}$$

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 (**Lewis, 1995**).

$$J(t_0) = \frac{1}{2} S(T)x^2(T) + \frac{1}{2} \int_{t_0}^T (Qx^2 + Ru^2) \tag{2.10}$$

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

$$\begin{aligned} H(x,u,\lambda,t) &= L(x,u,t) + \lambda^T f(x,u,t) \\ H(x,u,\lambda,t) &= \frac{1}{2} (Qx^2 + Ru^2) + \lambda^T (Ax + Bu) \end{aligned} \tag{2.11}$$

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

1. Syarat perlu

a) Persamaan *State*

$$\dot{x} = \left(\frac{\partial x}{\partial x} \right)^T Qx + \left(\frac{\partial Qx}{\partial \lambda} \right)^T u = Ax + Bu \quad (2.12)$$

b) Persamaan *Costate*

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

$$-\dot{\lambda} = Qx + A^T \lambda \quad (2.13)$$

c) Kondisi stasioner

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

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

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

$$u = -R^{-1} B^T \lambda \quad (2.14)$$

2. Syarat batas

a. Batas awal

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

b. Batas akhir

$$(\phi_x + \psi_x^T v - \lambda)^T dx \Big|_T + (\phi_t + \psi_t^T v - H) dt \Big|_{t=T} = 0 \quad (2.15)$$

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

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

$$S(t) x(t) = \lambda(t) \quad (2.16)$$

Dari persamaan *Costate* dan (2.16), keduanya mengandung variabel λ . Bila keduanya disubstitusikan maka akan menghasilkan persamaan:

$$\begin{aligned}
 S(t)x(t) &= \lambda(t) \\
 \dot{\lambda}(t) &= \dot{S}(t)x(t) + S(t)\dot{x}(t) \\
 \dot{\lambda}(t) &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\dot{x}(t) &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\{Ax + Bu\} &= -Qx - A^T \lambda \\
 \dot{S}(t)x(t) + S(t)\{Ax + B(-R^{-1}B^T \lambda)\} &= -Qx - A^T \lambda \\
 -\dot{S}x &= (A^T S + SA - SBR^{-1}B^T S + Q)x \\
 -\dot{S} &= A^T S + SA - SBR^{-1}B^T S + Q
 \end{aligned} \tag{2.17}$$

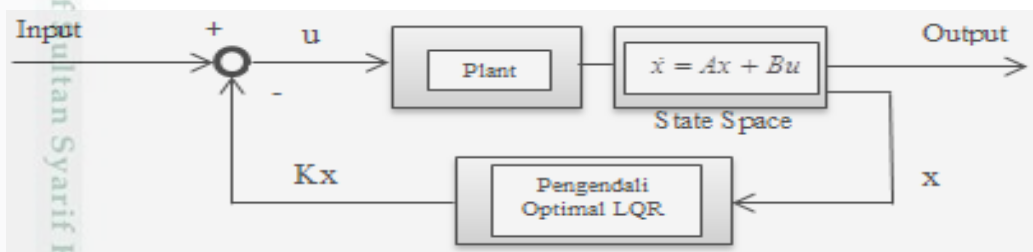
Persamaan (2.27) disebut persamaan differensial Riccati. Untuk $\dot{S} = 0$ persamaannya disebut *Algebraic Riccati Equation (ARE)* (Lewis, 1995). Dengan di dapatkannya matriks S dari persamaan ARE ini sehingga persamaan sinyal kendali pun dapat di hitung:

$$u = -R^{-1}B^T \lambda = -R^{-1}B^T Sx = -Kx \tag{2.18}$$

Maka konstanta umpan balik keadaan K dapat dicari sebelum menghasilkan sinyal kendali pada persamaan (2.19) (Lewis, 1995).

$$K = R^{-1}B^T S \tag{2.19}$$

Dimana nilai S adalah unik, solusi semidefinit positif untuk persamaan ARE harus memenuhi syarat persamaan (2.17). Selanjutnya dalam perancangan teknik kendali optimal LQR, Setelah nilai S diketahui maka, nilai S tersebut disubstitusikan kedalam persamaan (2.19) sehingga didapatkan nilai matriks optimal K, dengan K adalah *Gain State Feedback*, sehingga sistem kendali optimal LQR hasil desain dapat di perlihatkan pada gambar 2.8 berikut:



Gambar 2.8 Sistem Kendali dengan Skema Kendali Optimal LQR

(Sumber: Tadeus, 2012)

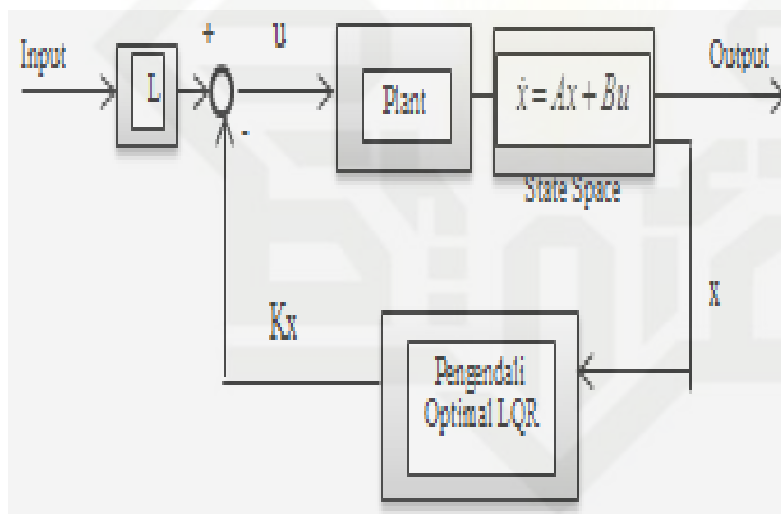
Permasalahan yang umum ditemui dalam bidang kendali yaitu tidak hanya menstabilkan sistem, tetapi juga menentukan bagaimana keluaran sistem mengikuti perubahan *Set Point* atau referensi yang ditentukan (**Badri, 2008**). Dalam hal tersebut jika diinginkan keluaran *Plant* (Y) sama dengan input referensi maka perlu dirancang sistemnya menggunakan *Nonzero Set Point/Tracking*. Bentuk umum sinyal kendali *Nonzero Set Point/Tracking* ini adalah :

$$u = -Kx + Lr \tag{2.20}$$

Dimana

$$L = [C(BK - A)^{-1} B^{-1}] \tag{2.21}$$

Dalam *Plant* nyata “ r ” merupakan nilai referensi atau jika dalam kendali kecepatan ini merupakan nilai referensi atau *Set Point* nilai kecepatan yang diinginkan dan “ x ” merupakan *Output* dari sensor ketepatan suatu kecepatan. Sehingga diagram sistem kendali optimal LQR setelah menggunakan matriks *Tracking* L dapat di lihat pada gambar 2.9 berikut ini:



Gambar 2.9 Skema Sistem Kendali Optimal Setelah menggunakan Matriks *Tracking*
 (Sumber: Mangkusasmito, 2013)

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 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

4. Hitung optimal *Gain Feedback K*
5. Tentukan nilai *Tracking L* dengan persamaan (2.21)

2.3.3 Kendali *Proportional-Integral-Derivatif* (PID)

Kendali ini dikenal sebagai pengendali tiga metode atau pengendali PID. Penggabungan semua dari ketiga mode kendali (proporsional, integral, dan derivatif) memungkinkan untuk mendapatkan sebuah pengendali yang tidak mempunyai *error* keadaan tunak serta dapat mereduksi kecenderungan terjadinya osilasi (Waluyo, 2013). Persamaan yang menggambarkan aksi pengendali ini adalah :

$$u(t) = K_p e(t) + K_i \int e dt + K_d \frac{de}{dt} \quad (2.22)$$

2.3.3.1 Kendali Proporsional

Kendali proporsional adalah besarnya keluaran pengendali terhadap besarnya *error* yang terjadi. Jadi didapat relasi : keluaran pengendali α masukan pengendali (Waluyo, 2013). Relasi ini dapat dituliskan secara matematis sebagai :

$$\text{Keluaran pengendali} = K_p \times \text{masukan pengendali} \quad (2.23)$$

Dimana K_p adalah konstanta yang disebut *gain* (peroleh). Ini berarti bahwa elemen koreksi sistem kendali akan memiliki masukan berupa sebuah sinyal yang berbanding lurus terhadap besarnya koreksi yang diperlukan (Waluyo, 2013).

2.3.3.2 Kendali Integral

Kendali integral merupakan mode kendali dimana keluaran pengendali berbanding lurus dengan integral *error* terhadap waktu. Jadi, keluaran pengendali α integral *error* terhadap waktu (Waluyo, 2013). Sehingga dapat ditulis secara matematis sebagai :

$$\text{Keluaran pengendali I} = K_i \times \text{integral } error \text{ terhadap waktu} \quad (2.24)$$

2.3.3.3 Kendali Derivatif

Kendali derivatif adalah perubahan keluaran pengendali dari titik pengaturan akan berbanding lurus dengan laju perubahan terhadap waktu dari sinyal *error* (Waluyo, 2013). Jadi keluaran pengendali α laju perubahan *error* dapat dirumuskan sebagai :

$$\text{Keluaran pengendali D} = K_d \times \text{laju perubahan } error \quad (2.25)$$

2.3.4 Analisa *Time Respons*

Analisa *time respons* sistem kendali terdiri dari respon *transient* dan *steady state*. Respon *transient* adalah respon sistem yang berlangsung dari keadaan awal sampai keadaan akhir, sedangkan respon *steady state* adalah kondisi keluaran sesudah habis respon *transient*. Parameter – parameter yang dapat dianalisa adalah :

1. Konstanta waktu

Ketika waktu $t = \tau$, maka $y = k(1 - e^{-1}) = 0,63k$. Jadi, konstanta waktu τ untuk sistem orde pertama ketika diberikan sinyal masukan berbentuk tangga adalah waktu yang dibutuhkan oleh keluaran sistem untuk mencapai 0,63 dari nilai keadaan tunaknya (Waluyo, 2013).

2. Waktu Tunda (*Delay Time*)

Waktu tunda (t_d) adalah waktu yang dibutuhkan oleh keluaran sistem untuk mencapai nilai 50% dari nilai keadaan tunaknya (Waluyo, 2013).

3. Waktu Naik (*Rise Time*)

Waktu naik (t_r) adalah waktu yang dibutuhkan oleh keluaran sistem untuk naik dari 10% nilai keadaan tunaknya hingga mencapai nilai 90% nilai keadaan tunaknya (Waluyo, 2013).

4. Waktu Pemantapan (*Settling Time*)

Waktu pemantapan (t_s) merupakan ukuran dari waktu yang diperlukan oleh osilasi respon sistem untuk menghilang. Lebih spesifiknya, waktu pemantapan adalah waktu yang diperlukan oleh respon sistem untuk jatuh menuju suatu nilai tertentu dan kemudian tetap berada pada nilai ini, yang biasanya dinyatakan sebagai persentase terhadap nilai keadaan tunaknya (Waluyo, 2013).

5. Kesalahan Tunak (*Error Steady State = ess*)

Yaitu kesalahan yang merupakan selisih antara keluaran yang sebenarnya dengan keluaran yang diinginkan.

2.3.5 Penalaan Parameter PID

Tuning atau penalaan parameter P, I dan D merupakan hal krusial dalam desain pengendali PID. Untuk itu perlu dilakukan penalaan terhadap parameter tersebut. Metode yang digunakan dalam penalaan parameter pengendali adalah metode Heuristik yaitu dengan mengkombinasikan antara parameter pengendali hingga didapatkan respon yang memuaskan.

Hubungan dari ketiga aksi pengendali tersebut ditunjukkan pada tabel berikut

Tabel 2.1 Hubungan antara kendali P, I dan D.

Respon lup tertutup	Waktu naik	<i>overshoot</i>	Waktu turun	Kesalahan keadaan tunak
Kp	Menurun	Meningkat	Perubahan kecil	Menurun
Ki	Menurun	Meningkat	Meningkat	Hilang
Kd	Perubahan kecil	Menurun	Menurun	Perubahan kecil

(Sumber : Ogata, 2002)

2.3.6 Pengendali PID Metode Heuristik

Metode Heuristik merupakan sebuah metode pemecahan masalah menggunakan eksplorasi dan cara coba-coba Heuristik adalah suatu aturan atau metode untuk bisa menyelesaikan solusi secara penalaan. Rancangan metode Heuristik ini diperoleh dengan cara perubahan parameter yang disesuaikan dengan kinerja plant yang akan dikendalikan

Untuk perancangan sistem pengendalian PID dilakukan pencarian nilai besarnya Kp, Ki, dan Td. Maka pengujian dilakukan dalam beberapa tahap, dengan penalaan (*Heuristic Method*):

1. Penalaan parameter pengendali dimulai dengan hanya menggunakan pengendali P,
2. kemudian baru ditambahkan pengendali I.
3. dan terakhir ditambahkan dengan pengendali D.

Pemberian nilai parameter disesuaikan dengan karakteristik respon sistem yang diperoleh (Waluyo, 2013).

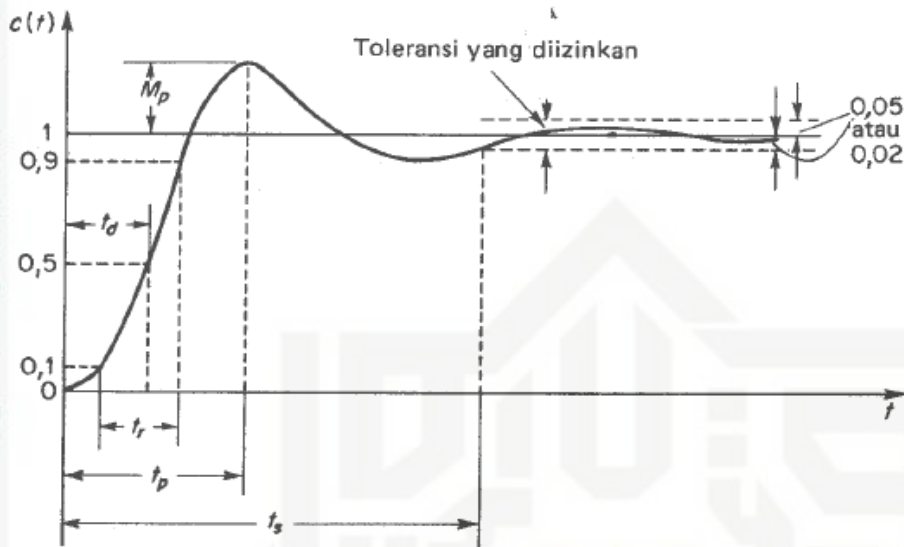
2.4 Identifikasi Sistem

Identifikasi sistem pada penelitian ini menggunakan metode pengamatan respon waktu dan kriteria integral IAE (*Integral of Absolut Error*) untuk mengidentifikasi tanggapan transien lingkaran tertutup yang menghasilkan error sekecil mungkin.

2.4.1 Identifikasi Respon Waktu

Metode ini mengidentifikasi sistem berdasarkan pengamatan grafis terhadap masukan step. Dimana sinyal uji diberikan kepada sistem untuk mengetahui respon sistem *open*

loop. Dari respon sistem, dapat diketahui karakteristik-karakteristik penting dari sistem. Sistem yang digunakan pada pengaturan kecepatan motor Induksi tiga fasa merupakan sistem orde 2 yang ditunjukkan pada Gambar (2.6) berikut:



Gambar 2.10 Respon Orde Kedua
(Sumber: Ogata, 2002)

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 sistem

b. Spesifikasi praktis:

1. Waktu penetapan (*Settling Time*) adalah waktu yang diperlukan kurva respon untuk mencapai dan menetap dalam daerah disekitar harga akhir yang ukurannya ditentukan dengan persentase mutlak dari harga akhir (5% atau 2%). Jika dihubungkan dengan nilai konstanta waktu τ , maka dapat diformulasikan sebagai berikut:

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

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

$$t_s (\pm 0.5\%) \approx 5\tau$$

2. Waktu naik atau *Rise Time* (t_r), adalah ukuran waktu 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 nilai konstanta waktu τ , maka 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 nilai konstanta waktu τ , maka dapat diformulasikan sebagai berikut:

$$t_d(\pm 3\%) \approx \tau \ln 2$$

4. Waktu puncak, t_p adalah waktu yang diperlukan respon untuk mencapai puncak lewatan yang pertama kali

5. Maksimum *Overshoot* (M_p): nilai yang menyatakan perbandingan antara nilai maksimum respon (*Overshoot*) yang melampaui nilai *Steady State* dibanding dengan nilai *Steady State*. Besarnya persen *Overshoot* dinyatakan sebagai berikut:

$$\%M_p = \frac{c_{\max} - c_{\text{akhir}}}{c_{\text{akhir}}} \times 100\% \quad (2.26)$$

Overshoot ini hanya terjadi pada sistem yang kurang teredam ($\zeta < 1$).

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

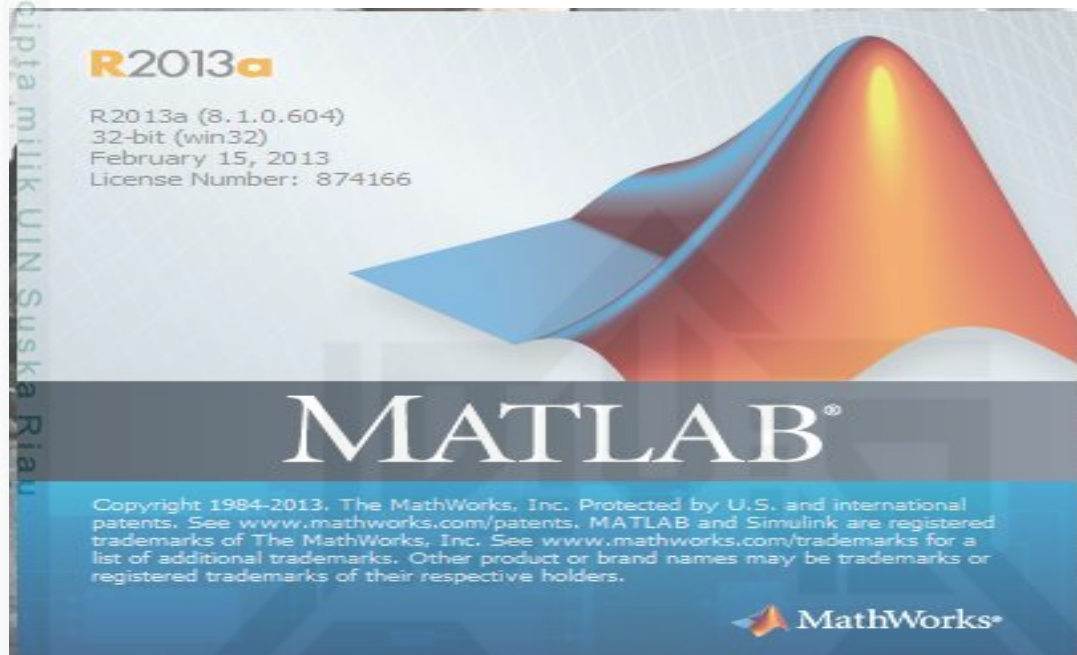
Kriteria integrasi membutuhkan data tanggapan mulai dari $t=0$ hingga mencapai keadaan tunak. Dengan demikian kriteria ini didasarkan pada seluruh tanggapan dari proses yang bersangkutan. Kriteria integral yang digunakan pada penelitian ini adalah kriteria IAE (*Integral of Absolute Error*) yang bertujuan untuk mendapatkan nilai IAE sekecil mungkin dengan pemilihan kriteria tergantung pada karakteristik sistem proses dan beberapa syarat tambahan yang diperoleh dari tanggapan lup tertutup. Kriteria IAE lebih dipopuler dikalangan praktisi industri sebab mudah dalam pemakaiannya serta sangat cocok untuk menekan error yang kecil.

$$IAE = \int_0^{\infty} |e| dt \quad (2.27)$$

2.5 Program Matlab

Matlab merupakan singkatan dari *Matrix Laboratory* yang berarti bahasa pemrograman level tinggi (semakin tinggi level bahasa pemrograman maka semakin mudah cara menggunakannya) dengan kinerja tinggi untuk komputasi masalah teknik.

Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan tunggal. Matlab memberikan sistem interaktif yang menggunakan konsep *array/matrix* sebagai variabel elemennya tanpa membutuhkan pendeklarasian *array*.



Gambar 2.11 MATLAB R2013

(Sumber: Mathwork, 2013)

Matlab dikembangkan oleh Mathwork pada tahun 1970. Aplikasi Matlab itu sendiri banyak digunakan dalam bidang yang membutuhkan perhitungan matematika yang rumit, dimana seluruh operasi perhitungan dalam Matlab berupa operasi matrik. Matlab dapat menampilkan hasil perhitungan dalam bentuk *plot* grafik dan dapat juga dirancang menggunakan GUI (*Graphical User Interface*) yang kita rancang. Pada *software* Matlab terdapat beberapa bagian penting yang digunakan dalam menjalankan program yaitu:

1. *Command window* digunakan untuk mengetik fungsi yang diinginkan.
2. *Command history* berfungsi agar fungsi yang telah digunakan sebelumnya dapat digunakan kembali.
3. *Workspace* digunakan untuk membuat variabel yang ada dalam Matlab.

2.5.1 Simulink-Matlab

Simulink adalah sebuah kumpulan aplikasi dalam Matlab untuk melakukan modeling, simulasi, dan untuk melakukan analisis dinamik pada suatu sistem. Program *simulink* memudahkan user untuk membuat suatu simulasi lebih interaktif. Tiruan sistem diharapkan mempunyai perilaku yang sangat mirip dengan sistem fisik. Jika digunakan

dengan benar, simulasi akan membantu proses analisis dan desain sistem. *Simulink* dalam matlab juga dapat menunjukkan performansi sistem dalam bentuk grafik dua dimensi ataupun tiga dimensi.

Dalam perancangannya *user* dibantu oleh blok-blok diagram yang dapat dengan mudah diatur sedemikian rupa, sesuai dengan model matematis dari sistem atau plant yang dikendalikan.