

## BAB IV

### PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas mengenai perancangan dan pembuatan alat meliputi pembahasan pada setiap blok rangkaian, cara kerja masing-masing blok rangkaian, perhitungan dan juga fungsi masing-masing blok rangkaian tersebut.

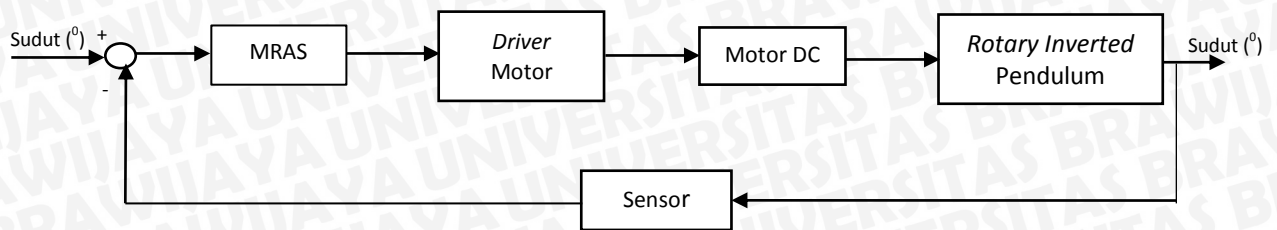
#### 4.1 Spesifikasi Alat

Spesifikasi alat ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang dirancang yaitu :

1. Sistem penggerak menggunakan motor DC.
2. Menggunakan catu daya 5V untuk catu driver motor EMS 5A H-Bridge.
3. Menggunakan catu daya 12V untuk motor DC.
4. Sensor posisi menggunakan rotary encoder.
5. Menggunakan Arduino Mega 2560 sebagai pengendali utama.
6. Berat tongkat pendulum (rod) 200 g dan panjang 30 cm dengan bahan alumunium.
7. Panjang lengan pendulum/chart 15 cm.

#### 4.2 Diagram Blok Sistem

Secara garis besar, diagram blok perancangan *hardware* sistem secara keseluruhan ditunjukkan dalam Gambar 4.1. *Setpoint* merupakan masukan berupa sudut. Sensor yang digunakan adalah *rotary encoder* berfungsi untuk mengetahui posisi sudut pendulum. Pengontrolan menggunakan *Model Reference Adaptive Systems* (MRAS). Arduino Mega 2560 sebagai pusat pengendalian sistem yang memberikan keluaran berupa *duty cycle* PWM kepada driver motor. *Driver* motor EMS 5A H-Bridge dengan IC MC33887VW memberikan perintah ke aktuator (motor DC) yang akan menggerakkan *rod* pada *Rotary Inverted* Pendulum sesuai arah kemiringannya dengan kecepatan tertentu. *Output* yaitu gerakan dari *rod* berupa sudut.



Gambar 4.1 Diagram Blok Perancangan *Hardware*

### 4.3 Prinsip Kerja Alat

Pendulum terbalik dikatakan tegak (setimbang) apabila rotary encoder memberikan tegangan yang menunjukkan bahwa kondisi pendulum terbalik berada tegak lurus terhadap permukaan bumi (sesuai hasil pengujian rotary encoder), apabila pendulum terbalik tidak tegak lurus terhadap garis normal atau miring, maka kontroler akan memberikan perintah pada *driver* motor untuk mengendalikan motor sehingga motor berputar sesuai arah kemiringan pendulum terbalik dengan kecepatan tertentu hingga pendulum terbalik mencapai keadaan setimbang kembali.

Masukan kontroler berupa kemiringan, sementara keluaran kontroler adalah berupa *duty cycle* PWM yang kemudian diteruskan ke-*driver* motor dan menghasilkan tegangan untuk memutar motor DC dengan kecepatan tertentu.

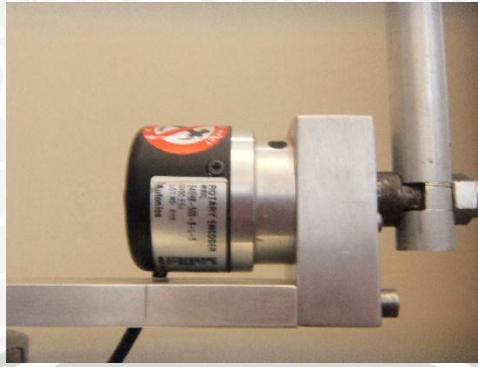
### 4.4 Perancangan Perangkat Keras (*Hardware*)

Berdasarkan diagram blok perancangan alat yang telah disusun, perancangan perangkat keras meliputi pembuatan perancangan mekanik, *rotary encoder* dan *driver* motor.

#### 4.4.1 *Rotary Encoder*

*Rotary encoder* merupakan perangkat elektromekanik yang dapat memonitor gerakan dan posisi, dengan cara meletakkan ujung bawah tiang pendulum terbalik pada poros *rotary encoder* maka setiap pergeseran sudut dari pendulum terbalik akan ikut mempengaruhi perubahan resistansi pada keluaran *rotary encoder*. Di bawah ini Gambar 4.2 merupakan *rotary encoder* yang terpasang pada *rod*.



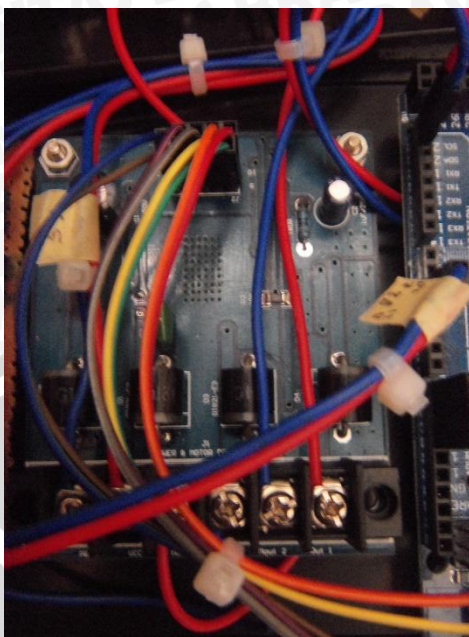


Gambar 4.2 Rotary Encoder pada Rod

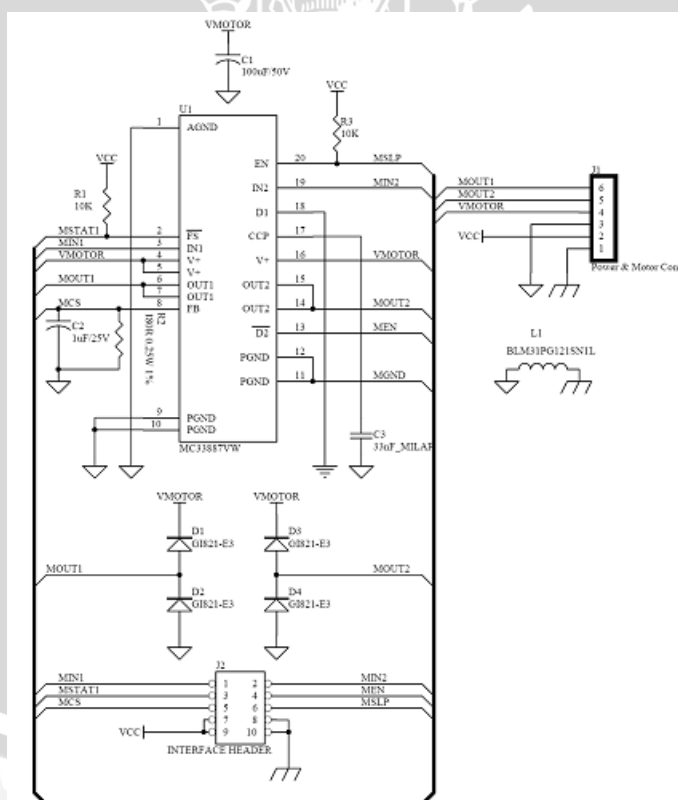
#### 4.4.2 Perancangan Driver Motor

Pada perancangan ini *driver* motor DC digunakan *Embedded Module Series* (EMS) 5A H-Bridge yang menggunakan IC MC33887VW. Modul ini merupakan driver H-Bridge yang dirancang untuk menghasilkan *drive* 2 arah dengan arus kontinyu sampai dengan 5 A pada tegangan 5 Volt sampai 40 Volt. Modul ini mampu men-*drive* beban-beban induktif seperti misalnya relay, solenoid, motor DC, motor stepper dan berbagai macam beban lainnya. Modul EMS 5A digambarkan dalam Gambar 4.3.

Modul H-Bridge memiliki 1 set *interface header* dan 1 set terminal konektor. *Interface Header* berfungsi sebagai *input* untuk antarmuka dengan *input-output* digital serta *output* analog dari modul H-Bridge. Terminal konektor berfungsi sebagai konektor untuk catu daya dan beban. Rangkaian *driver* motor berdasarkan *datasheet* ditunjukkan dalam Gambar 4.4.



Gambar 4.3 Driver Motor EMS 5A H-Bridge



Gambar 4.4 Rangkaian Elektrik Driver

Dengan konfigurasi *driver* dalam Gambar 4.4, maka akan didapatkan keadaan-keadaan seperti ditunjukkan dalam Tabel 4.1. Sebuah modul EMS 5A hanya dapat



dihubungkan pada satu buah motor. Pin MOUT1 dan MOUT2 dihubungkan pada motor. Pin MEN merupakan pin *enable* yang digunakan sebagai pengaktif sinyal masukan pada *driver*. Pin MIN1 dan MIN2 merupakan pin yang digunakan untuk menentukan arah putaran motor. Apabila MIN1 berlogika *high* dan MIN2 berlogika *low* maka motor akan berputar searah dengan arah jarum jam. Apabila MIN1 berlogika *low* dan MIN2 berlogika *high* maka motor akan berputar berlawanan arah dengan arah jarum jam. Pin MSTAT1 merupakan pin *output digital* yang melaporkan kondisi *fault* pada modul. Pin MSTAT1 akan berlogika *low* jika ada *fault* pada modul atau *output*. Pin MSLP merupakan pin *input* yang mengatur kerja dari modul. Modul akan bekerja pada mode *full-operation* pada saat pin ini diberi logika *high* dan akan bekerja pada mode *sleep* pada saat pin ini diberi logika *low*.

Tabel 4.1 Tabel Kebenaran *Driver* Motor EMS 5A H-Bridge

Status kerja modul H-Bridge	Input				Status Flag	Output	
	MSLP	MEN	MIN1	MIN2		MOUT 1	MOUT 2
<i>Forward</i>	H	H	H	L	H	V MOT	MGND
<i>Reverse</i>	H	H	L	H	H	MGND	V MOT
<i>Freewheeling Low</i>	H	H	L	L	H	MGND	MGND
<i>Freewheeling High</i>	H	H	H	H	H	V MOT	V MOT
<i>Free Running Stop</i>	H	L	X	X	L	Z	Z
MIN1 tidak terhubung	H	H	Z	X	H	V MOT	X
MIN2 tidak terhubung	H	H	X	Z	H	X	V MOT
MEN tidak terhubung	H	Z	X	X	L	Z	Z
<i>Undervoltage</i> <sup>1</sup>	H	X	X	X	L	Z	Z
<i>Overtemperature</i> <sup>1</sup>	H	X	X	X	L	Z	Z
<i>Short Circuit</i> <sup>1</sup>	H	X	X	X	L	Z	Z
<i>Mode Sleep</i>	L	X	X	X	H	Z	Z

Keterangan :

H = High

L = Low

X = don't care

Z = High Impedance (tri-state)

#### 4.4.3 Perancangan Arduino Mega 2560

Pada pendulum terbalik ini digunakan Arduino Mega2560 sebagai pusat dari pengolah data. Agar sebuah Arduino Mega2560 dapat bekerja sebagai pengontrol, kaki-kaki *pin* Arduino Mega2560 dihubungkan dalam rangkaian-rangkaian eksternal. Dalam perancangan ini, *pin* yang digunakan adalah sebagai berikut :

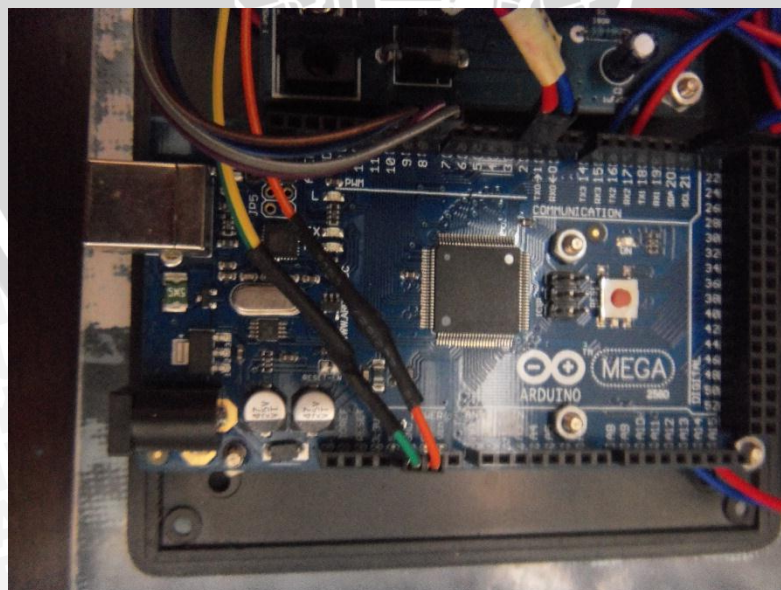
1. *Pin 2 (Interrupt 0) dan pin 3 (Interrupt 1)*

*Pin* ini terhubung dengan sensor rotary encoder bawah. *Pin* ini merupakan *input* dari kontroler.

2. *Pin 8, Pin 9, Pin 10 dan Pin 10*

*Pin* ini merupakan *pin output* dari Arduino Mega 2560. Bentuk sinyal keluaran dari *pin* ini adalah sinyal *Pulse Width Modulation (PWM)*. Pada perancangan ini, sinyal pin 11 dan pin 10 digunakan sebagai penentu arah putaran motor. Jika pin 11 mempunyai logika 1 dan pin 10 mempunyai besar mempunyai logika 0 maka motor akan berputar searah dengan arah jarum jam. Jika pin 11 mempunyai besar mempunyai logika 0 dan pin 10 mempunyai besar mempunyai logika 1 maka motor akan berputar berlawanan dengan arah jarum jam. Pin 9 digunakan sebagai *enable*. Pin 8 digunakan untuk mengatur kerja modul H-Bridge. Diberi logika High untuk Full Operation, diberi logika Low untuk Mode Sleep

Selanjutnya, Arduino Mega2560 dihubungkan dengan komputer dengan komunikasi serial. Dihubungkan disini bertujuan agar dapat dilakukan pemrograman di komputer, selanjutnya program tersebut dapat di tanamkan pada Arduino Mega2560. Selain itu dapat digunakan sebagai catu untuk Arduino Mega2560. Gambar dari Arduino Mega2560 ditunjukkan pada Gambar 4.5



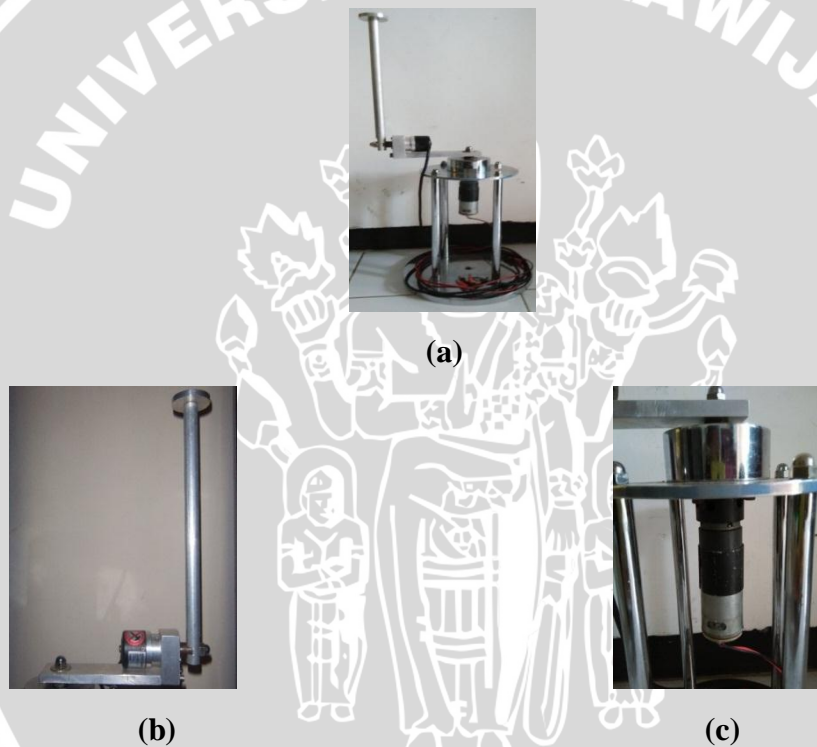
Gambar 4.5 Arduino Mega2560



#### 4.4.4 Perancangan Mekanik Rotary Inverted Pendulum

Mekanik dari *Rotary Inverted Pendulum* mempunyai tiga bagian penting yaitu *rod* atau stik yang akan dipertahankan posisinya, motor DC, dan lengan pemutar. Ujung bawah *rod* diletakkan pada poros *rotary encoder*, sehingga setiap pergeseran sudut pada *rod* akan terjadi pergeseran sudut pada poros *rotary encoder* tersebut. *Rotary encoder* diletakkan pada sebuah lengan yang terhubung dengan poros motor DC, sehingga pada saat terjadi kemiringan pada pendulum terbalik maka motor DC akan memutar lengan tersebut dengan arah sesuai dengan kemiringan dari pendulum dengan kecepatan tertentu.

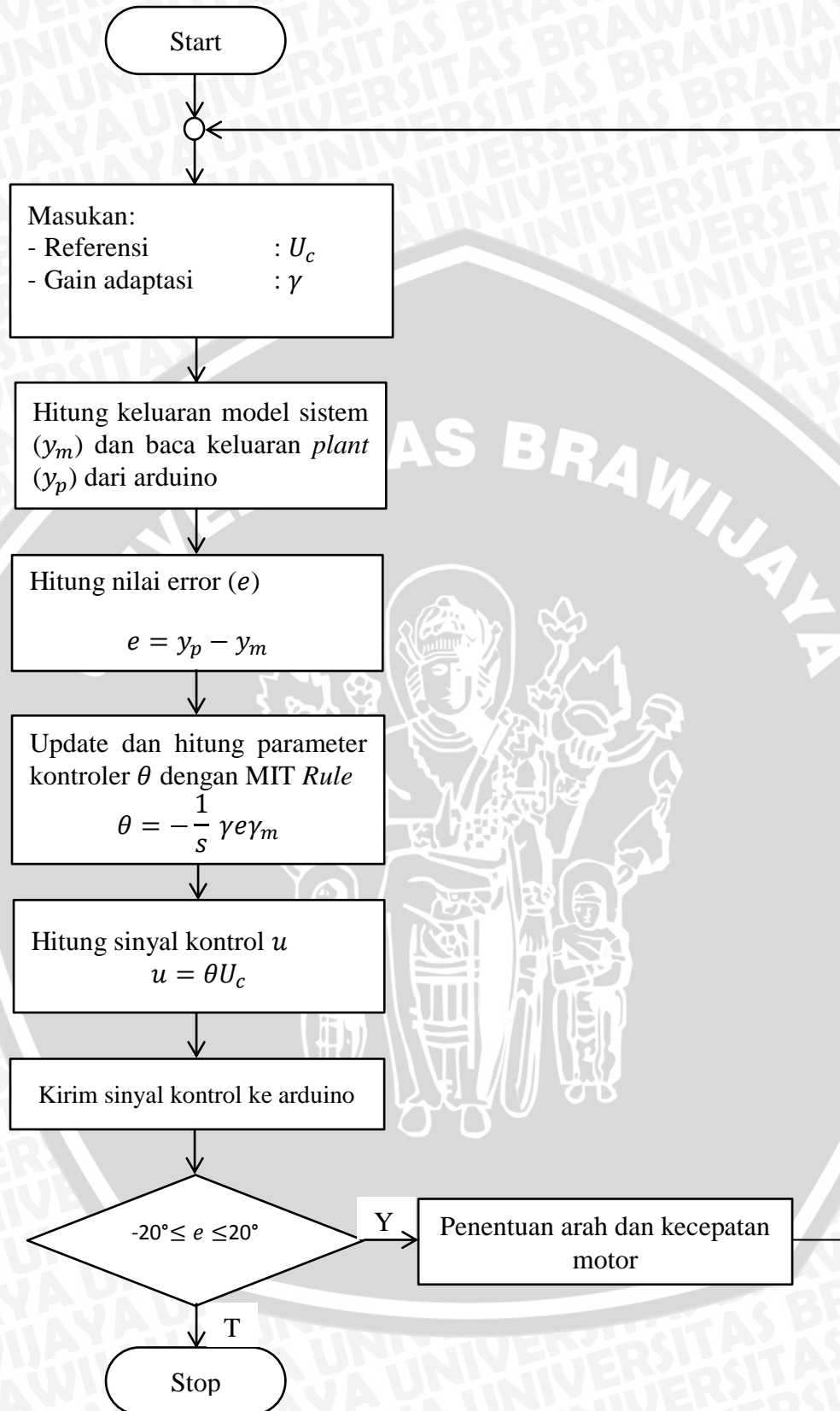
Gambar hasil perancangan mekanik ditunjukkan dalam Gambar 4.6 :



Gambar 4.6 (a) Hasil Perancangan Tampak Keseluruhan, (b) Sambungan antara Sensor dan Rod, (c) Motor DC yang Menggerakkan Lengan Pemutar

#### 4.5 Perancangan Perangkat Lunak (*Software*)

Perangkat lunak pada skripsi ini dikerjakan pada program matlab/simulink yang telah kompatibel dengan arduino. *Flowchart* perancangan perangkat lunak di tunjukkan dalam Gambar 4.7.



Gambar 4.7 Flowchart Program



## 4.6 Perancangan *Model Reference Adaptive Systems* (MRAS)

### 4.6.1 Model Matematis Sistem

Sistem stabilisasi gerak tongkat pendulum meliputi beberapa komponen sistem yang bekerja di dalamnya, yaitu: kontroler, driver motor, motor dc, rotary inverted pendulum, sensor rotary encoder. Masukan dalam sistem ini adalah sudut tongkat pendulum yang diinginkan yaitu  $0^\circ$ , sedangkan keluarannya adalah sudut tongkat pendulum yang sebenarnya.

Adapun data parameter-parameter utama yang akan digunakan pada sistem ini adalah sebagai berikut:

**Tabel 4.2 Data Parameter-parameter Utama Sistem**

Parameter	Nilai	Parameter	Nilai
$m_1$ (kg)	0,7	$I_1$ (kg m <sup>-2</sup> )	0,0525
$m_2$ (kg)	0,2	$I_2$ (kg m <sup>-2</sup> )	0,03
$L_1$ (m)	0,15	$g$ (m s <sup>-2</sup> )	9,81
$L_2$ (m)	0,3	$R_a$ ( $\Omega$ )	22,6
$C_1$ (Nms)	0	$K_v$	0,014
$C_2$ (Nms)	0	$K_t$ (N m A <sup>-1</sup> )	0,014
$l_2$ (m)	0,2	$V_a$ (V)	12

Guna mendapatkan model matematis dari sistem *rotary inverted pendulum*, dapat diperoleh dengan memasukkan nilai parameter ke dalam persamaan 2.16 :

$$\dot{x}(t) = Ax(t) + BV_a$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -d_1(h_6 + C_1) & -d_2h_4 & -d_2C_2 \\ 0 & 0 & 0 & 1 \\ 0 & -d_3(h_6 + C_1) & -d_4h_4 & -d_2C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ d_1h_5 \\ 0 \\ d_3h_5 \end{bmatrix} V_a$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.00015472 & 1.1054 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.00002443 & -10.5008 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.0111 \\ 0 \\ -0.0017 \end{bmatrix} 12 \quad (4.1)$$

Sehingga akan didapatkan fungsi alih dari *state space* sistem yaitu:

$$T(s) = \frac{0.01105s^2 + 0.1141}{s^4 + 0.0001547s^3 + 10.5s^2 + 0.001598s} \quad (4.2)$$

#### 4.6.2 Pendekatan Reduksi Orde Model

Fungsi alih model pada persamaan (4.3) merupakan model berorde 4, persamaan tersebut sulit digunakan untuk mendesaian model referensi pada MRAS. Oleh karena itu diperlukan desain dari orde tinggi melalui reduksi orde model yang cocok. Hal ini dapat diselesaikan melalui pendekatan model orde 2 dan sedikit kesalahan indeks kumulatif ( $J$ ). Prosedur untuk menentukan reduksi orde model secara singkat dapat dijelaskan di bawah ini.

Orde- $n$  original sistem pada persamaan (4.2) disamakan terhadap orde- $k$  pada reduksi orde model dengan parameter yang tidak diketahui, dimana persamaannya adalah:

$$G_n(s) = R_k(s)$$

$$\frac{0.01105s^2 + 0.1141}{s^4 + 0.0001547s^3 + 10.5s^2 + 0.001598s} = \frac{d_0 + d_1s}{e_0 + e_1s + e_2s^2}$$

$$(0.01105s^2 + 0.1141)(e_0 + e_1s + e_2s^2) = (s^4 + 0.0001547s^3 + 10.5s^2 + 0.001598s)(d_0 + d_1s)$$

$$0.01105s^2e_0 + 0.00115s^3e_1 + 0.01105s^4e_2 + 0.1141e_0 + 0.1141se_1 + 0.1141s^2e_2 = s^4d_0 + s^5d_1 + 0.0001547s^3d_0 + 0.0001547s^4d_1 + 10.5s^2d_0 + 10.5s^3d_1 + 0.001598sd_0 + 0.001598s^2d_1 \quad (4.3)$$

Penyamaan koefisien dari kemiripan persamaan (4.3) diperoleh hubungan sebagai berikut:

$$0.1141e_1 = 0.001598d_0 \quad (4.4)$$

$$0.01105e_0 + 0.1141e_2 = 10.5d_0 + 0.001598d_1 \quad (4.5)$$

$$0.00115e_1 = 0.0001547d_0 + 10.5d_1 \quad (4.6)$$

$$0.01105e_2 = d_0 + 0.0001547d_1 \quad (4.7)$$

$$0 = d_1 \quad (4.8)$$

Dengan memisalkan  $d_0=1$  dan menggunakan persamaan (4.4)-(4.8) parameter yang tidak diketahui dapat diketahui nilainya sehingga reduksi model orde 2 yaitu:

$$\frac{d_0 + d_1s}{e_0 + e_1s + e_2s^2} = \frac{1}{15,7654 + 0,14s + 90,4477s^2}$$



#### 4.6.3 Penentuan Model Referensi

Secara teoritis model referensi ditentukan dengan mempertimbangkan derajat relatif proses. Derajat relatif adalah selisih antara derajat polinomial *pole* dengan derajat polinomial *zero*. Derajat relatif model referensi lebih besar atau sama dengan derajat relatif proses.

Namun perlu dicatat bahwa pada praktiknya ketika kebutuhan teoritis tidak dapat terpenuhi maka model referensi harus ditentukan dengan hati-hati dengan syarat keluaran proses sebenarnya dapat mengikuti keluaran model referensi. Jika model referensi yang dipilih memiliki respon yang terlalu cepat, sebagai contoh, maka sinyal kontrol yang dibutuhkan harus sangat besar, sehingga menyebabkan efek saturasi pada masukan atau dinamika tak termodelkan yang merusak sistem adaptif (Butler, 1992).

Penentuan fungsi alih model referensi ditentukan dengan kriteria sebagai berikut:

- Settling time* model referensi sebesar 0,25 detik.
- Prosentase *overshoot* tidak lebih dari 10% dari *output* respon.
- Derajat relatif model referensi adalah dua. Besar derajat relatif ini diperoleh dengan melihat fungsi alih proses yang akan dikendalikan dimana yang digunakan adalah fungsi alih proses hasil dari pendekatan reduksi orde model yaitu:

$$\frac{1}{15,7654 + 0,14s + 90,4477s^2}$$

Penentuan model referensi dilakukan dengan hati-hati sesuai kriteria di atas, yaitu:

$$PO = 100e^{-\left(\frac{\xi\pi}{\sqrt{1-\xi^2}}\right)}$$

$$10\% = 100e^{-\left(\frac{\xi\pi}{\sqrt{1-\xi^2}}\right)}$$

$$\xi = 0,9108$$

$$t_s = \frac{4}{\xi\omega_n}$$

$$0,25 = \frac{4}{0,9108\omega_n}$$

$$\omega_n = \frac{4}{0,9108(0,25)}$$

$$\omega_n = 17,5669 \text{ rad/s}$$

sehingga diperoleh model referensi yang ditunjukkan pada persamaan berikut:

$$\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} = \frac{308,59}{s^2 + 31,99s + 308,59}$$

#### 4.6.4 Penentuan Hukum Adaptasi

Proses adaptasi berlangsung dengan adanya perubahan nilai parameter pengendali. Proses ini berlangsung selama terjadi *error* antara keluaran *plant* dengan keluaran model referensi. Proses adaptasi diatur melalui cara yang disebut dengan hukum adaptasi. Nilai parameter kontroler akan diubah sedemikian rupa sehingga *error* antara  $y_p$  dan  $y_m$  sama dengan nol. Penyesuaian parameter pengendali akan menghasilkan sinyal kontrol sebagai masukan bagi *plant*.

Pada pengendalian adaptif dengan MRAS, *error* antara keluaran *plant* dengan keluaran model referensi digunakan secara langsung untuk mengubah nilai parameter kontroler. Proses adaptasi pada MRAS disebut dengan adaptasi langsung.

Hukum adaptasi yang digunakan adalah MIT *Rule*. Untuk menyetengahkan aturan MIT, dimisalkan sebuah sistem *loop* tertutup dimana kontrolernya mempunyai sebuah parameter  $\theta$  yang dapat diatur. Respon *loop* tertutup yang diinginkan ditentukan dengan suatu model yang memiliki *output*  $y_m$ . Misalnya  $e$  sebagai *error* antara *output* sistem *loop* tertutup  $y$  dan *output* model  $y_m$ . Satu kemungkinan untuk mengatur parameter dengan meminimalisasi *loss function* (fungsi kerugian),  $J$ :

$$J(\theta) = \frac{1}{2} e^2 \quad (4.9)$$

Untuk mendapatkan nilai  $J$  kecil, merupakan hal yang beralasan untuk mengubah parameter dalam arah negatif gradient dari  $J$ , yaitu:

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (4.10)$$

$$e = y_p - y_m = kG(s)\theta u_c - k_o G(s)\theta u_c \quad (4.11)$$

$$\frac{\partial e}{\partial \theta} = kG(s)\theta u_c = \frac{k}{k_o} y_m \quad (4.12)$$

MIT *Rule* memberikan hukum adaptasi berikut:



$$\frac{d\theta}{dt} = -\gamma e \frac{k}{k_o} y_m \quad (4.13)$$

Dari persamaan diatas dapat dirubah menjadi domain-s untuk mendapatkan  $\theta$ ,

$$s\theta = -\gamma e \frac{k}{k_o} y_m \quad (4.14)$$

$$\theta = -\frac{1}{s} \gamma e \frac{k}{k_o} y_m = -\frac{1}{s} \gamma e y_m \quad (4.15)$$

Turunan parsial  $\partial e/\partial \theta$  yang disebut dengan turunan sensitivitas sistem, menunjukkan bagaimana *error* dipengaruhi oleh parameter yang dapat diatur. Jika diasumsikan bahwa parameter berubah lebih lambat daripada variabel yang lain di dalam sistem, sehingga turunan  $\partial e/\partial \theta$  dapat dievaluasi di bawah asumsi bahwa  $\theta$  adalah konstan.

